

## Computational Symmetry in Computer Vision and Computer Graphics

By Yanxi Liu, Hagit Hel-Or,  
Craig S. Kaplan and Luc Van Gool

### Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What is symmetry?	6
1.2	Why is symmetry relevant to computational science?	7
1.3	Why is computational symmetry challenging?	9
1.4	Historical Perspective	11
1.5	Organization of this Survey	12
<b>2</b>	<b>Symmetry and Symmetry Groups</b>	<b>15</b>
2.1	Symmetry in Euclidean Space	15
2.2	Symmetry in Affine and Projective Spaces	31
2.3	Symmetry in non-Euclidean Geometry	56
<b>3</b>	<b>Symmetry Detection</b>	<b>60</b>
3.1	Symmetry Detection by Methods	61
3.2	Symmetry Detection by Symmetry Types	77
3.3	Quantitative Comparison of Symmetry Detection Algorithms	98

<b>4 Near Regular Texture (NRT)</b>	<b>105</b>
4.1 NRT Formalization	106
4.2 NRT Quantification	109
4.3 NRT Applications	110
4.4 Discovery of NRTs in the Real World: Deformed-Lattice Extraction	117
4.5 Dynamic NRTs	132
<b>5 Continuous Symmetry</b>	<b>140</b>
5.1 Measuring Symmetry as a Continuous Feature	141
5.2 Facial Asymmetry as a Biometric	152
5.3 Statistical Brain Asymmetry from Volumetric Images	156
<b>6 Symmetry in Graphics</b>	<b>158</b>
6.1 Kaleidoscopes	158
6.2 Tilings	159
6.3 Ornamental Design	161
6.4 Non-Euclidean Geometry	166
6.5 Polyhedra and Sculpture	167
6.6 Chaos and Fractals	170
<b>7 Summary</b>	<b>173</b>
<b>Acknowledgements</b>	<b>176</b>
<b>References</b>	<b>177</b>

Foundations and Trends® in  
Computer Graphics and Vision  
Vol. 5, Nos. 1–2 (2009) 1–195  
© 2010 Y. Liu, H. Hel-Or, C. S. Kaplan  
and L. Van Gool  
DOI: 10.1561/0600000008



## Computational Symmetry in Computer Vision and Computer Graphics

Yanxi Liu<sup>1</sup>, Hagit Hel-Or<sup>2</sup>,  
Craig S. Kaplan<sup>3</sup>, and Luc Van Gool<sup>4</sup>

<sup>1</sup> Pennsylvania State University, USA, yanxi@cse.psu.edu

<sup>2</sup> University of Haifa, Israel, hagit@cs.haifa.ac.il

<sup>3</sup> University of Waterloo, Canada, csk@uwaterloo.ca

<sup>4</sup> KU Leuven, Belgium and ETH Zurich, Switzerland,  
Luc.Vangool@esat.kuleuven.ac.be

### Abstract

In the arts and sciences, as well as in our daily lives, symmetry has made a profound and lasting impact. Likewise, a computational treatment of symmetry and group theory (the ultimate mathematical formalization of symmetry) has the potential to play an important role in computational sciences. Though the term *computational symmetry* was formally defined a decade ago by the first author, referring to algorithmic treatment of symmetries, seeking symmetry from digital data has been attempted for over four decades. *Computational symmetry* on real world data turns out to be challenging enough that, after decades of effort, a fully automated symmetry–savvy system remains elusive for real world applications. The recent resurging interests in computational symmetry for computer vision and computer graphics applications have shown promising results. Recognizing the fundamental relevance and

potential power that computational symmetry affords, we offer this survey to the computer vision and computer graphics communities. This survey provides a succinct summary of the relevant mathematical theory, a historic perspective of some important symmetry-related ideas, a partial yet timely report on the state of the arts symmetry detection algorithms along with its first quantitative benchmark, a diverse set of real world applications, suggestions for future directions and a comprehensive reference list.

# 1

---

## Introduction

---

Symmetry is a pervasive phenomenon presenting itself in all forms and scales in natural and man-made environments, from galaxies to biological structures (Figure 1.1), as well as in the arts (Figure 1.2 from the classic book by Jones [106]). Much of our understanding of the world is based on the perception and recognition of repeated patterns that are generalized by the mathematical concept of symmetries [47, 277]. Humans and animals have an innate ability to perceive and take advantage of symmetry in everyday life, but harnessing this powerful insight for machine intelligence remains an elusive goal for computer science.

In the basic sciences, the understanding of symmetry played a profound role in several important discoveries, including: the theory of relativity (the discovery of the isometries of Minkowski spacetime under the Poincaré group, the full symmetry group) [195]; the double helix structure of DNA (with two-fold rotation symmetry) [275]; the discovery of quasi-crystals (the first observation of an unusual fivefold symmetry indicated by diffraction pictures of samples from an alloy of aluminium and manganese) [247] and their mathematical counterpart Penrose tiles [209]. It is not a coincidence that all of these symmetry-related discoveries led, directly or indirectly, to Nobel prizes — an indication of the fundamental relevance of symmetry in science.

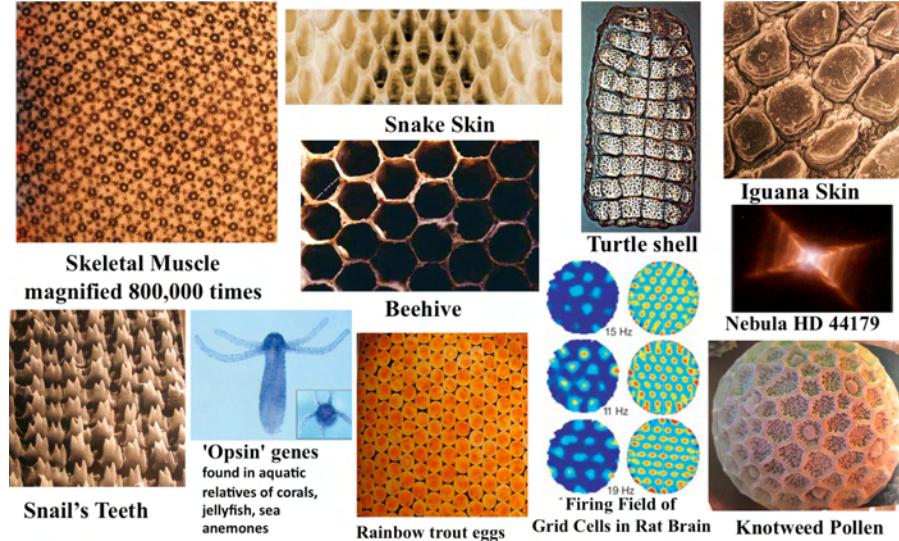


Fig. 1.1 The ubiquitous appearance of real world symmetries in nature: from Nebula to the firing field of grid cells in rats' brains. The photos from four corners, and bottom-middle, are courtesy of [182].

Given the evidence of the powerful role of symmetry in the history of the natural sciences, we hypothesize that *computational symmetry*, defined by the first author as *using computers to model, analyze, synthesize and manipulate symmetries in digital forms, imagery or otherwise* [152], will likewise play a crucial role in the advancement of our understanding in artificial/machine intelligence.

In human perception, symmetry is considered a pre-attentive feature that enhances object recognition [40, 144, 268]. Although humans, primates, dolphins, birds, and insects have an innate ability to recognize and use real world symmetries that have been quantitatively documented [74, 226, 268], the symmetry cue is hardly used in today's object recognition, categorization or scene understanding systems due, largely, to a lack of computational models and available robust algorithms.

In computational science, the development of symmetry detection algorithms has had a long history. The earliest attempt at an algorithmic treatment of bilateral reflection symmetry detection predates computer vision itself [18]. In spite of years of effort, we are still short of

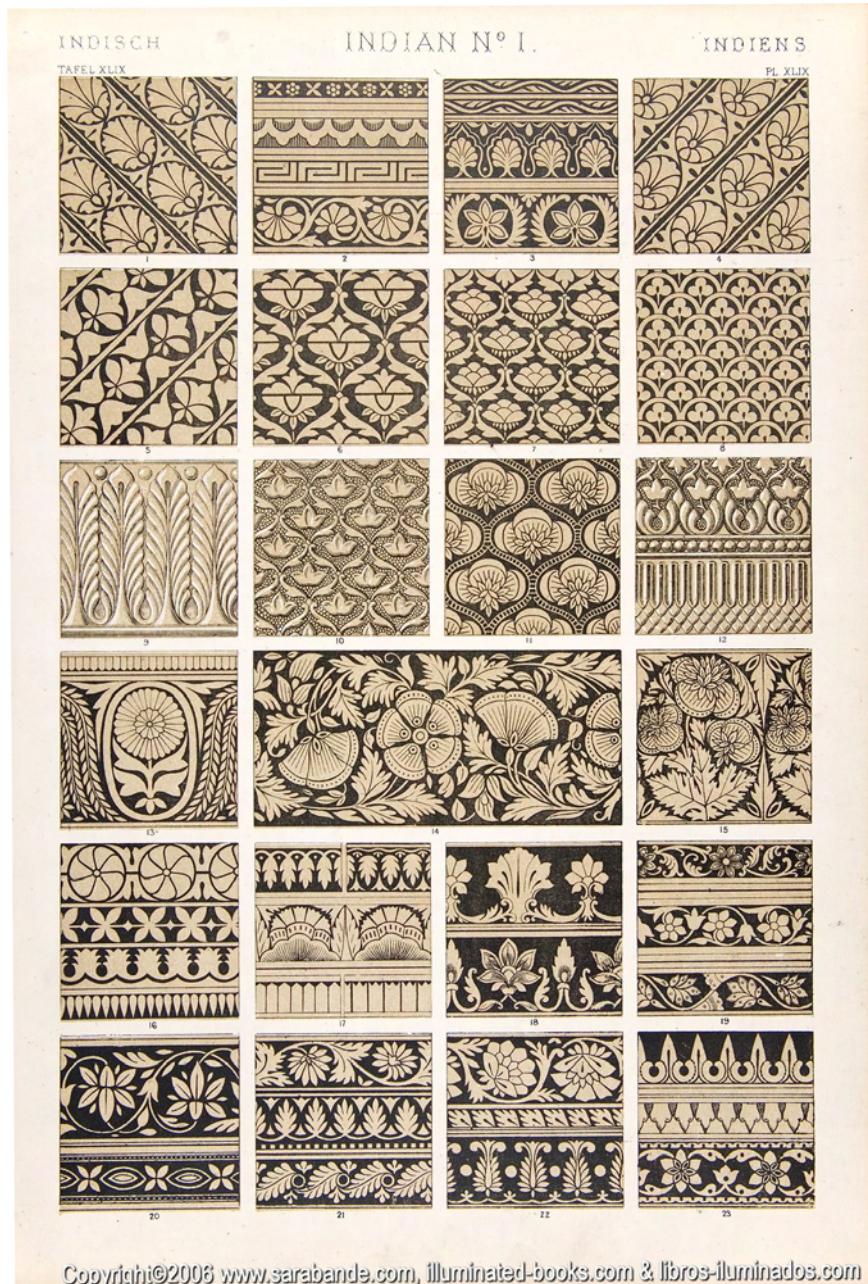


Fig. 1.2 Symmetries in art.

## 6 Introduction

a robust, widely applicable “symmetry detector” for real images. Furthermore, though an initial effort has been made [33, 203], we have yet to see a large-scale, systematic, quantitative evaluation and a publicly available test-image database to gauge the progress in this important, widely applicable research direction.

Without attempting to cover all published related-work in this survey, we primarily focus on the theory and techniques that make explicit use of symmetry groups. Due to a lack of coverage in the literature on *discrete* finite and infinite symmetry groups and their relevance to and impact on computer vision and computer graphics problems, we pay special attention to these types of algorithms without excluding affine, perspective, and non-Euclidean geometries. We hope to provide a clear conceptual roadmap of group theory (Section 2), and its multi-facet applications in computer vision and computer graphics. In particular, we hope to achieve these goals:

- (1) to de-mystify group theory, discrete and finitely generated (infinite) groups in particular, using concrete examples from 2D and 3D Euclidean, projective and hyperbolic geometries;
- (2) to illustrate the ubiquitous and persistent appearances of symmetry structures, particularly those associated with discrete symmetry groups, in real world data;
- (3) to appreciate the substantial computational challenges as well as promises in current and future computational symmetry research.

### 1.1 What is symmetry?

From the spirit of the Felix Klein’s Erlangen program [79]: geometry is the study of a space that is invariant under a given transformation group, to the Gestalt principles of perception [7]: among others, *Laws of Symmetry*, symmetries and group theory play an important role in describing the geometry and the appearance of an object. Informally, we may think of symmetry as expressing the notion that a figure or object is made from multiple copies of the same smaller unit that are interchangeable somehow. Mathematically, we formalize this notion by examining the effect of transformations on the object in a certain space

such that its sub-parts coincide (map to each other). The following quote from Weyl [277] captures the essence of symmetry eloquently:

“Starting from the somewhat vague notion of symmetry = harmony of proportions, . . . rise to the general idea . . . of invariance of a configuration of elements under a group of automorphic transformations.”

More formally, in a metric space  $M$ , a *symmetry*  $g \in G$  of a set  $S \subseteq M$  is an isometry (a distance preserving transformation) that maps  $S$  to itself (an automorphism),  $g(S) = S$ . The transformation  $g$  keeps  $S$  invariant as a whole while permuting its parts. Symmetries  $G$  of  $S$  form a mathematical *group*  $\{G, *\}$ , closed under transformation composition  $*$ , called the *symmetry group* of  $S$  [44].

Group theory provides a level of abstraction that leads to simplicity and completeness in practical algorithm design and execution. In two-dimensional (2D) Euclidean space, for example, there are four distinct *atomic* transformations as *primitive* symmetries [41, 44, 277]: translation, rotation, reflection and glide-reflection<sup>1</sup> (Figure 1.3). One somewhat surprising mathematical discovery of a century ago is the answer to the first part of Hilbert’s 18th problem: there is only a finite number of symmetry groups for all periodic patterns in  $R^n$  of any  $n$ . These groups are referred to as *crystallographic groups* [44, 84, 277]. While the number of distinct 1D (frieze) and 2D (wallpaper) crystallographic groups is finite and relatively small, there is an infinite number of potential instantiations; some samples can be observed in Figures 1.1, 1.2 and 1.3.

## 1.2 Why is symmetry relevant to computational science?

A computational model for symmetry is especially pertinent to computer vision and computer graphics, or machine intelligence in general,

---

<sup>1</sup> Glide-reflection is defined as a symmetry composed of a translation along and a reflection about the same axis.

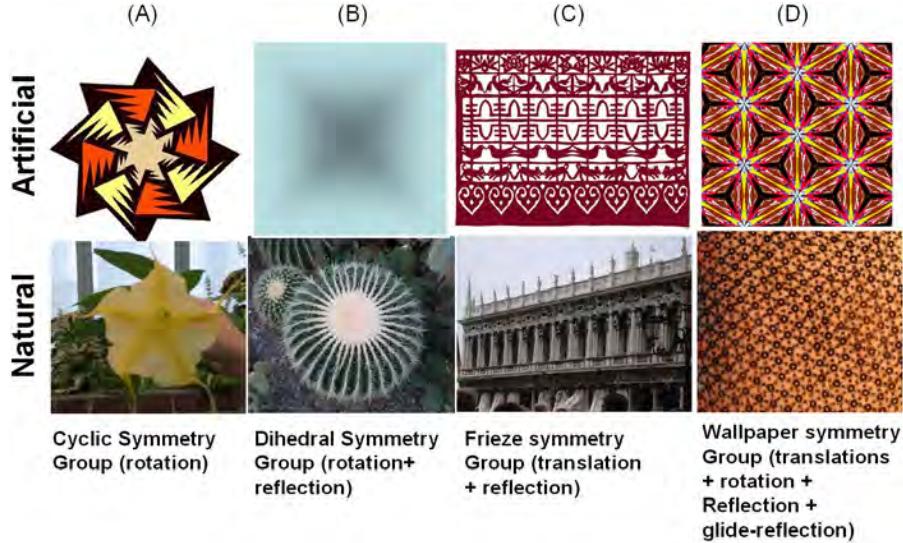


Fig. 1.3 Sample images of symmetry categorized by their respective ideal symmetry groups: column (A) cyclic group containing rotation symmetries only, (B) dihedral group (reflection and rotation), (C) frieze group (translation plus reflection), and (D) wallpaper group (translation, rotation, reflection and glide-reflection). The top row contains synthetic patterns while the bottom row shows photos of real world scenes (bottom-right is an image of a transverse slice of skeletal muscle magnified with a transmitter microscope 800,000 times).

because of its

- *ubiquitousness*: both the physical and the digital worlds are filled with various forms of symmetry, near-symmetry and distorted symmetry patterns (Figures 1.1, 1.2 and 1.3). The applicability of such a computational model can only be limited by one's imagination;
- *essentiality*: intelligent beings perceive and interact with the chaotic real world in the most efficient and effective manner by capturing its essential structures and sub-structures — the generators of symmetry, near-symmetry, distorted symmetry and/or repeated patterns;
- *compactness*: the recognition of symmetries is the first step towards minimizing redundancy, often leading to drastic reductions in computation; and

- *aestho-physiology*: from a butterfly to an elephant, from a tea cup to a building, symmetry or deviation from it, has been a time-honored principle for design (by nature or by human) that can guide machine perception, detection, recognition and synthesis of the real world.

Since the earliest bilateral reflection symmetry detection algorithm [18], attempts in computational treatment of symmetry and regularity have been made continuously. Figure 1.4 shows the statistics of published papers<sup>2</sup> in several major computer vision/graphics conferences/journals during the period of 1974–2009 (36 years). An increasing level of interests can be observed in both computer vision and computer graphics.

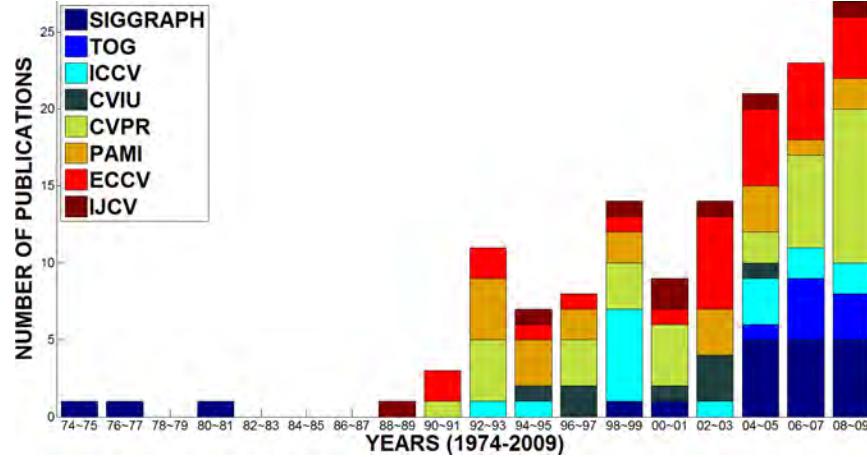
### 1.3 Why is computational symmetry challenging?

Humans are experts in symmetry detection and appreciation [144, 268]. Our ability to recognize and tolerate departures from perfect symmetries reflects a level of sophistication in human perception. From an engineering point of view, however, it remains unclear how to capture and simulate this perceptual capability of humans and animals for machine/artifitial intelligence. From a theoretical point of view, even though group theory itself (especially Euclidean group and its subgroups) is a mature field, little formal theory exists connecting the elegant group theory to the noisy, incomplete and often inconsistent real world. From an educational point of view, group theory is usually introduced in classrooms as an abstract theory instead of a theoretical basis for algorithmic treatment of real world problems, such like those in computer vision and computer graphics.

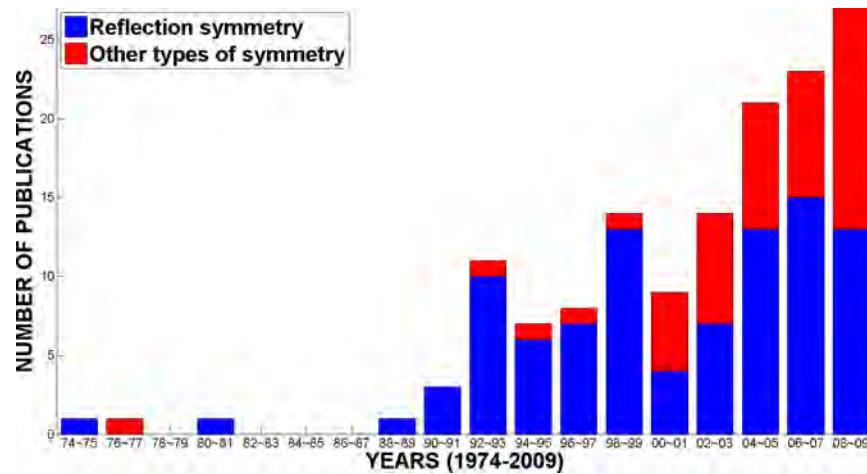
Regardless of how powerful computers have become, one fundamental limitation of computers is their finite representation power. One simple floating point round-up error destroys any perfect symmetry in the data. In addition, and perhaps more importantly, the non-coherent

---

<sup>2</sup>A list of the papers counted for can be found here: <http://www.cse.psu.edu/~yanxi/symmetryCitation.htm>.



(A) Publications on symmetry detection and applications in major computer vision and computer graphics conferences/journals.



(B) Dividing papers into on reflection symmetry alone versus other types of symmetries (rotation, translation and glide-reflection).

Fig. 1.4 From the publication statistics, it is obvious that research on reflection symmetry has been dominating the field in the past, with a growing awareness of the whole symmetry spectrum. A similar reflection-symmetry-dominating trend has also been observed in the psychology literature for human perception of symmetries [268].

(discrete versus continuous, and finite versus infinite) topological nature of symmetry groups poses serious problems for their representation and computation on computers under a uniform framework [151].

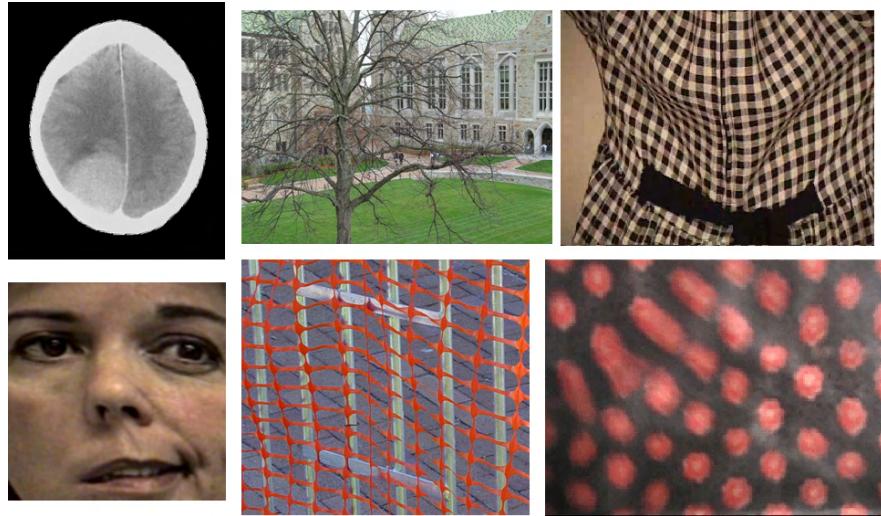


Fig. 1.5 Sample images of distorted, disguised and layered real world symmetries. Top and bottom-middle photos are courtesy of David R. Martin.

In summary, computational symmetry is challenged by at least two acute discrepancies:

- the clean formal concepts of group theory versus imperfect, noisy, ambiguous, distorted and often hidden symmetry patterns in digitized real world data (Figure 1.5).
- the complete, concise and uniform mathematical theory of symmetry, group theory, versus limitations of representational power of computers (hardware) and a lack of computational models for real world symmetry (software);

Without explicitly and effectively addressing each of these challenges, it is impossible for computational symmetry to release its potential power and play a substantial role in computer vision and computer graphics research even though the importance and relevance of symmetry in these fields become increasingly obvious (Figure 1.4).

## 1.4 Historical Perspective

To gain some historic perspective and insight, interested readers can find several influential symmetry-related papers, for example, the

## 12 *Introduction*

wonderful exposition on the role of symmetry in “Biological Shape and Visual Science” by Blum in 1973 [20]; in 1977, the “Description and Recognition of Curved Objects” reported by Nevatia and Binford, where bilateral symmetry of the object about different axes is examined [197]<sup>3</sup>; the method of detecting angle/side regularities of closed curves and plateaus in one-dimensional patterns by Davis et al. [22, 48]; the introduction of the term *skewed symmetry* by Takeo Kanade in 1981 [109]; the exposition on “Smoothed Local Symmetries and Their Implementation” by Brady and Asada (1984) [24]; the theory of *recognition-by-components* (RBC) proposed by Biederman in 1985 [14]<sup>4</sup>; “Perceptual Grouping and the Natural Representation of Natural Form” using superquadrics as restricted *generalized cylinders* (GC) by Pentland in 1986 [210]; “Perceptual Organization and Visual Recognition” by Lowe [175], where the non-coincidental appearance of symmetry in the real world was noted; and the “Symmetry-seeking Models for 3D Object Reconstruction” (1987) illustrated by Terzopoulos, Witkin and Kass [256].

### 1.5 Organization of this Survey

In order to make this survey concise, self-contained and easily searchable, we organize this survey into eight relatively independent sections. They are:

- (1) **Introduction**, where we motivate the necessity of this survey;
- (2) **Symmetry and symmetry groups**, where we provide a set of standard definitions and proofs with intuitive explanations of these concepts. The theoretical basis for symmetry is

---

<sup>3</sup>The authors stated: because of the simple descriptors used for pieces, the symmetry calculation is correspondingly crude. However, we feel that with improved descriptions, symmetry can be very useful.” (p. 90) [197].

<sup>4</sup>The fundamental assumption of the proposed theory of RBC “is that a modest set of components [ $N$  probably  $5 \leq 36$ ] can be derived from contrasts of five readily detectable properties of edges in a two-dimensional image: curvature, collinearity, symmetry, parallelism, and coterminal. The detection of these properties is generally invariant over viewing position and image quality and consequently, allows robust object perception when the image is projected from a novel viewpoint or degraded.”

established under Euclidean, affine and perspectively-skewed and non-Euclidean geometries;

- Symmetry in Euclidean geometry
- Symmetry-based invariants in perspective transformation (affine as a special case)
- Symmetry in non-Euclidean geometry

(3) **Symmetry and Symmetry group detection**, where we describe and demonstrate the methods and the output of several representative state of the art symmetry detection algorithms followed by the results from the first quantified benchmarking in this area;

(4) **Near regular textures**, where we show their mathematical roots to crystallographic groups, and their increasingly wide applicability in both computer vision and computer graphics research;

(5) **Continuous and quantified symmetry**, where we demonstrate a variety of real world applications using quantified symmetry (or deviations from it) as a continuous measure;

(6) **Symmetry in graphics**, where a range of visually appealing symmetry-based graphics applications is illustrated;

(7) **Summary**, where we summarize this effort, provide pointers to existing resources and lay out tangible future directions in this research area; and

(8) **References**, a comprehensive reference list in computational symmetry completes this survey.

Given limitations in space and time, some important aspects of symmetry related topics are not covered in this survey, they include:

- *Medial axis* and its wide range of applications: for a recent survey on this topic, refer to [249]; a classic shock graph reference [250], and some recent developments [140, 253].
- Human and animal symmetry perceptions: these are well studied and well documented. Interested readers can start with [268, 144]; some excellent exploration of symmetry and culture research can be found in these books [236, 274].

## 14 *Introduction*

- Alternative representations of symmetry groups: in this survey, we follow the classic group theory for symmetry. But we recommend the book by Conway et al. [41] for great intuition and a modern representation of symmetry groups.

# 2

---

## Symmetry and Symmetry Groups

---

We start by providing some formal definitions of symmetry, and symmetry groups as subgroups of the Euclidean group, including cyclic, dihedral, frieze and wallpaper groups, and their subgroup inter-relations. We then move on to affine, projective and non-Euclidean spaces (hyperbolic spaces) to demonstrate a wide range of symmetries.

### 2.1 Symmetry in Euclidean Space

#### 2.1.1 Basic Concepts

---

**Definition 1.** Let  $S$  be a subset of  $R^n$ . Then an isometry<sup>1</sup>  $g$  is a symmetry of  $S$  if and only if  $g(S) = S$ .

---

---

**Definition 2.** A symmetry  $g$  for a set  $S \in R^n$  is a primitive symmetry if and only if for any non-trivial decompositions of  $g = g_1 g_2$ , neither  $g_1$  nor  $g_2$  is a symmetry of  $S$ .

---

<sup>1</sup> An isometry is a distance preserving mapping.

## 16 Symmetry and Symmetry Groups

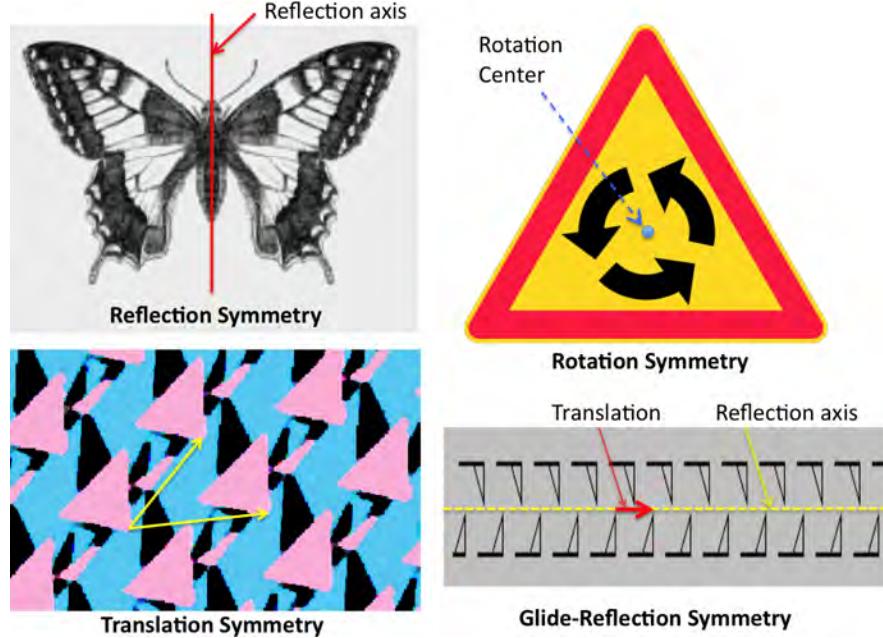


Fig. 2.1 An illustration of patterns with the four different primitive symmetries (PS) respectively, in the 2D Euclidean space. Reflection symmetry has the reflection axis as its pointwise invariance. Rotation symmetry has the center of rotation as its invariant point. An  $n$ -fold rotational symmetry with respect to a particular point (in 2D) or axis (in 3D) means that, set-wise, the rotation by an angle of  $360/n$  does not change the object. Glide reflection is composed of a translation that is  $1/2$  of the smallest translation symmetry  $t$  and a reflection  $r$  with respect to a reflection axis along the direction of the translation. There are no invariant points under translation and glide-reflection symmetries.

For example, in 2D Euclidean space  $R^2$ , there are four types [41, 44, 277] of *primitive symmetries*  $g(S) = S$ . They are, without loss of generality, for the four images  $f(x, y)$  shown in Figure 2.1:

- (1) *Reflection*:  $f(x, y) = f(-x, y)$ , its reflection axis (plane) remains invariant under the reflection.
- (2) *Rotation*:  $f(x, y) = f(r \cos(2\pi/n), r \sin(2\pi/n))$ ,  $r = \sqrt{x^2 + y^2}$ ,  $n$  is an integer ( $n = 3$  in Figure 2.1), its rotation center point (axis) remains invariant under the rotation.
- (3) *Translation*:  $f(x, y) = f(x + \Delta x, y + \Delta y)$ , for some  $\Delta x, \Delta y \in R$ , no invariant points exist.

(4) *Glide reflection*:  $f(x, y) = f(x + \Delta x, -y)$ , for some  $\Delta x \in R$ , no invariant points exist. A glide-reflection  $g$  can be expressed as  $g = tr$ , where  $t$  is a translation and  $r$  is a reflection whose axis of reflection is along the direction of the translation. Note: neither  $t$  or  $r$  alone is a symmetry of  $S$  thus  $g$  is a primitive symmetry of  $S$ .

Euclidean space of three-dimension  $R^3$  has two additional primitive symmetries:

- (1) *Rotoreflection symmetry*: invariant to the center of rotation;
- (2) *Helical symmetry* (screw operation or rotation–translation in 3D): invariant to its axis of rotation.

Given an arbitrary subset  $S \subset R^n$ ,  $S$  has at least one symmetry — the *identity mapping* which maps each point in  $S$  to itself. Thus symmetry groups can conceptually capture both regular shaped objects as well as irregular shaped objects as having a trivial symmetry group (with cardinality one).

### Symmetries beyond isometries

$S$  can be a purely point set or something beyond, such as surface orientations in 3D space [166], for example, or color and texture of an image (e.g., [202]).

- Color symmetry and *counterchange symmetry* are isometries plus permutations of the color space.
- We can also expand the notion of symmetry to include dilations. In that case, a shape like a logarithmic spiral in the plane has a “dilate and rotate” symmetry that would not be possible with ordinary isometries.
- Symmetry groups in the hyperbolic plane (Section 2.3), when viewed as perspective transformations of the 2D Euclidean plane under the Klein projection, form perfectly well-behaved groups of non-isometries.

## 18 Symmetry and Symmetry Groups

Now let us review the mathematical notion of an algebraic *group*  $G$ :

---

**Definition 3.** Let  $G$  be a non-empty set with a well-defined binary operation  $*$  such that for each ordered pair  $g_1, g_2 \in G$ ,  $g_1 * g_2$  is also in  $G$ .  $(G, *)$  is a *group* if and only if:

---

- (1) there exists an *identity* element  $e \in G$  such that  $e * g = g = g * e$  for all  $g \in G$ ;
- (2) any element  $g$  in  $G$  has an *inverse*  $g^{-1} \in G$  such that  $g * g^{-1} = g^{-1} * g = e$ .
- (3) the binary operation  $*$  is associative:  $a * (b * c) = (a * b) * c$  for all  $a, b, c \in G$ .

Using the composition of mappings in  $R^n$  as the binary operation  $*$ , one can prove that *symmetries of a subset  $S \subset R^n$  form a group*, which is called the **symmetry group** of  $S$ .<sup>2</sup>

---

**Proposition 2.1.** Symmetries of a subset  $S \subset R^n$  form a symmetry group  $G_S$  of  $S$ .

---

*Proof.* Assume  $G_S$  contains all the symmetries of  $S$  and  $*$  is the transformation composition defined on pairs of symmetries. Obviously, the identity mapping  $e$  is a symmetry of  $S$  since  $e(S) = S \rightarrow e \in G_S$  (condition #1 of Definition 3). Since all symmetries  $g$  of  $S$  are rigid transformation in Euclidean space, each of them has a natural inverse  $g^{-1}$  to reverse the transformation of  $g$ . If  $g \in G_S$  then  $g(S) = S$ , multiplying by  $g^{-1}$  on both sides, we have  $S = g^{-1}g(S) = g^{-1}(S)$  therefore  $g^{-1} \in G_S$  (condition #2). Finally, if  $g_1, g_2, g_3 \in G_S$  then  $(g_1g_2)g_3(S) = g_1(g_2(S)) = g_1(S) = S = (g_1(g_2 * g_3))(S)$  thus  $*$  is associative (condition #3). By the definition of a group (Definition 3)  $(G_S, *)$  is a group.  $\square$

---

**Definition 4.** All the symmetries of  $R^n$  form the Euclidean Group  $\mathcal{E}$ .

---

<sup>2</sup>Note, the difference between *symmetry group* and *symmetric group*, the latter is the group consisting of all permutations of a finite set with function composition as the group operation [69, 181].

---

**Definition 5.** All the handedness-preserving isometries, i.e., excluding reflections in  $\mathcal{E}$ , form the Proper Euclidean Group  $\mathcal{E}^+$ .  $\mathcal{E}^+$  is a subgroup of  $\mathcal{E}$ . The symmetry group of any  $S \subset R^n$  is a subgroup of  $\mathcal{E}$ , or of  $\mathcal{E}^+$  if reflections are excluded.

---

Table 2.1 lists some typical subgroups of  $\mathcal{E}^+$  summarized in [151].

---

**Definition 6.**  $G_1$  and  $G_2$  are subgroups of group  $\mathcal{E}^+$ ,  $G_1$  is conjugate to  $G_2$  iff there exists  $g \in \mathcal{E}^+$  such that  $G_1 = gG_2g^{-1}$ .

---



---

**Proposition 2.2.** If  $G$  is the symmetry group of  $S \subset R^n$  then for any rigid transformation  $g$  in  $\mathcal{E}^+$ ,  $gGg^{-1}$  is the symmetry group of  $g(S)$ .

---

Table 2.1. Some typical canonical subgroups of  $\mathcal{E}^+$  in  $R^3$  [151].

Canonical groups	Representative group element
Identity group	
$G_{\text{id}}$	$\{1\}$
Rotation subgroups	
$SO(3)$	$\{\text{rot}(\mathbf{i}, \theta)\text{rot}(\mathbf{j}, \sigma)\text{rot}(\mathbf{k}, \phi)   \theta, \sigma, \phi \in R\}$
$O(2)$	$\{\text{rot}(\mathbf{k}, \theta)\text{rot}(\mathbf{i}, n\pi)   \theta \in R, n \in \mathcal{N}\}$
$SO(2)$	$\{\text{rot}(\mathbf{k}, \theta)   \theta \in R\}$
$D_n$	$\{\text{rot}(\mathbf{k}, 2\pi/n)\text{rot}(\mathbf{i}, m\pi)   m \in \mathcal{N}\}, n \in \mathcal{N}$
$C_n$	$\{\text{rot}(\mathbf{k}, 2\pi/n)\}, n \in \mathcal{N}$
Translation subgroups	
$\mathcal{T}^1$	$\{\text{trans}(0, 0, z)   z \in R\}$
$\mathcal{T}_{\text{dis}}^1(t_0)$	$\{\text{trans}(0, 0, t_0)\}, t_0 \in R$
$\mathcal{T}^2$	$\{\text{trans}(x, y, 0)   x, y \in R\}$
$\mathcal{T}^3$	$\{\text{trans}(x, y, z)   x, y, z \in R\}$
Mixed subgroups	
$G_{\text{cyl}}$	$\{\text{trans}(0, 0, z)\text{rot}(\mathbf{k}, \theta)\text{rot}(\mathbf{i}, n\pi)   n \in \mathcal{N}, \theta, z \in R\}$
$G_{\text{dir\_cyl}}$	$\{\text{trans}(0, 0, z)\text{rot}(\mathbf{k}, \theta)   z, \theta \in R\}$
$G_{\text{plane}}$	$\{\text{trans}(x, y, 0)\text{rot}(\mathbf{k}, \theta)\text{rot}(\mathbf{i}, n\pi)   x, y, \theta \in R, n \in \mathcal{N}\}$
$G_{\text{dir\_plane}}$	$\{\text{trans}(x, y, 0)\text{rot}(\mathbf{k}, \theta)   x, y, \theta \in R\}$
$G_{\text{screw}}(p)$	$\{\text{trans}(0, 0, z)\text{rot}(\mathbf{k}, 2z\pi/p)   z \in R\}, p \in R$
$G_{T_1 C_2}$	$\{\text{trans}(0, 0, z)\text{rot}(\mathbf{i}, n\pi)   n \in \mathcal{N}, z \in R\}$
$\mathcal{E}^+$	$\{\text{trans}(x, y, z)\text{rot}(\mathbf{i}, \theta)\text{rot}(\mathbf{j}, \sigma)\text{rot}(\mathbf{k}, \phi)   x, y, z, \theta, \sigma, \phi \in \mathfrak{R}\}$

The major orthogonal axes are represented by  $\mathbf{i}, \mathbf{j}, \mathbf{k}$ .  $\text{rot}(\mathbf{k}, \theta)$  is a rotation about  $\mathbf{k}$  by  $\theta$  radius.  $\text{trans}(x, y, z)$  is a translation by  $x, y, z$  units in the  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  directions respectively.  $R$  is real numbers and  $\mathcal{N}$  is natural numbers.

## 20 Symmetry and Symmetry Groups

*Proof.* If  $a \in G$ ,  $gag^{-1}(g(S)) = ga(S) = g(S)$  thus  $gag^{-1}$  is a symmetry of  $g(S)$ . Obviously,  $1 \in gGg^{-1}$  and  $(gag^{-1})^{-1}(g(S)) = ga^{-1}g^{-1}(g(S)) = ga^{-1}(S) = g(S)$ , therefore  $(gag^{-1})^{-1}$  is also a symmetry of  $g(S)$  and belongs to  $gGg^{-1}$ . Finally, for  $a, b \in G$ ,  $gag^{-1}gbg^{-1}(g(S)) = gag^{-1}gb(S) = gag^{-1}(g(S)) = ga(S) = g(S)$ . So  $gGg^{-1}$  is the symmetry group of  $g(S)$ .  $\square$

### 2.1.2 Symmetry Group Categorization

Groups can be divided into different categories. For example,  $G$  is a *finite* group if there is a finite number of elements in  $G$ , an infinite group otherwise.

One way to describe a symmetry group is to use the concept of *orbit*.

---

**Definition 7.** An orbit of a point  $x \in R^n$  under group  $G$ , or the  $G$ -orbit of  $x$ , is  $G(x) = \{g(x) | g \in G\}$ .

---

A *discrete* symmetry group can be defined in terms of its orbits:

---

**Definition 8.** A discrete group  $G$  is a subgroup of  $\mathcal{E}$  such that for any  $x \in R^n$  and any sphere  $B_r = \{y | y \in R^n, ||y|| \leq r\}$  there is only a finite number of points in the  $G$ -orbit of  $x$  that are contained in  $B_r$ .

---

According to this definition, finite groups are always discrete but discrete groups are not necessarily finite. For example, the cyclic group  $C_n$  is a *discrete* and *finite* group while the symmetry group of an infinite line of dots with equal intervals is not *finite* though indeed *discrete*. A continuous group, such as the symmetry group of a sphere  $SO(3)$ , is neither finite nor discrete.

Another way to categorize different types of groups is by observing what points in  $R^n$  are left invariant by a part or all the group actions. These points are often called *invariant points*.

---

**Definition 9.** For every  $x$  in  $X$ , we define the *stabilizer subgroup* of  $x$  (also called the *isotropy group* or *little group*) as the set of all elements in  $G$  that fix  $x$ :  $G_x = \{g \in G | g(x) = x\}$ .

---

---

**Definition 10.** A *point group*  $G$  is a symmetry group that leaves a single point  $x$  fixed. In another word,  $G = G_x$ , here  $G_x$  is the stabilizer subgroup on  $x$ .

---



---

**Definition 11.** In crystallography, a space group is a group  $G$  of operations which leave the infinitely extended, regularly repeating pattern of a crystal unchanged. In  $R^3$  there are 230 such groups. In another word, such a group  $G$  as a whole leaves no point invariant, i.e., for all possible  $x \in R^n$ ,  $G_x$  is either a proper subgroup of  $G$ ,  $G_x \subset G$  or empty. Here  $G_x$  is the stabilizer subgroup on  $x$ .

---

A list of finite symmetry groups in  $R^3$  is shown in Table 2.2, including symmetry groups of the five platonic solids (Figure 2.2). In  $R^2$ , there are only four types of discrete (finite and infinite) symmetry groups as

Table 2.2. Finite point groups in 3D Euclidean space.

Group name	Size $ G $	Comments
$G_{\text{id}}$	1	The identity group
$C_n = G_{\text{cyclic}}$	$n$	Generated by the $2\pi/n$ rotation
$D_n = G_{\text{dihedral}}$	$2n$	Generated by the $2\pi/n$ rotation + a $180^\circ$ rotation for a regular $n$ -gon, $n + 1$ rotation axes
$G_{\text{tetrahedral}}$	24	7 Rotation axes, 12 rotation symmetries
$G_{\text{octahedral}}$	48	24 Rotation symmetries, 13 axes, same as cube
$G_{\text{icosahedral}}$	120	31 Rotation axes, 60 rotations, same as dodecahedron

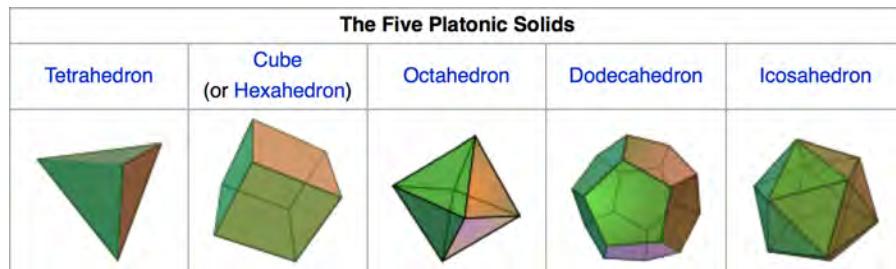


Fig. 2.2 The five platonic solids. The cube and octahedron (the dodecahedron and icosahedron) are dual solids with the same symmetry group. The symmetry groups of the five platonic solids are all finite groups as shown in Table 2.2. Figure source: [http://en.wikipedia.org/wiki/Platonic\\_solid](http://en.wikipedia.org/wiki/Platonic_solid).

## 22 Symmetry and Symmetry Groups

Table 2.3. Discrete symmetry groups in  $R^2$ .

Name	Group type	Symbol	Order	Atomic symmetry	Example
Cyclic	Point	$C_n$	$n$	Rotation	Figure 1.3(A)
Dihedral	Point	$D_n$	$2n$	Rotation and reflection	Figure 1.3(B)
Frieze	Space	$G_{\text{frieze}}$	$\infty$	All 4 primitives	Figure 1.3(C)
Wallpaper	Space	$G_{\text{wallpaper}}$	$\infty$	All 4 primitives	Figure 1.3(D)



Fig. 2.3 Examples of the only three types of rotation symmetry groups in the 2D Euclidean space: Left:  $C_5$  is the cyclic group of order 5 with rotation symmetries only; middle:  $D_5$  is the dihedral group of order  $2 \times 5 = 10$  with both rotation and reflection symmetries; and right:  $O(2)$  is the infinite, continuous orthogonal group in 2D containing infinitesimal rotation symmetries and reflection symmetries.

shown in Table 2.3. There are two special cases for cyclic group  $C_n$  and dihedral group  $D_n$  respectively: when  $n = 1$ ,  $C_n$  becomes the identity group of size 1 and  $D_n$  of size 2 contains only a bilateral reflection symmetry and the identity mapping; when  $n \rightarrow \infty$ ,  $C_n \rightarrow \text{SO}(2)$  (special orthogonal group), the symmetry group of an oriented disk (without any reflection symmetries) and  $D_n \rightarrow O(2)$  (orthogonal group), the symmetry group of an un-oriented disk (all rotations and reflections that keep the center of the disk invariant). Figure 2.3 demonstrates the three sample shapes associated with its respective, distinct, non-trivial 2D point groups.

### 2.1.3 Crystallographic Groups

A mature mathematical theory for periodic patterns<sup>3</sup> has been known for over a century [13, 61, 62, 63], namely, the *crystallographic groups*. These are groups composed of symmetries of periodic patterns in

<sup>3</sup> An  $n$ -dimensional periodic pattern is formed by repeating a pattern-unit in equal intervals along  $n$  or less than  $n$  directions.

$n$  dimensional Euclidean space. An essential mathematical theory of periodic patterns is the answer to the first part of Hilbert's 18th problem [189]: regardless of dimension  $n$  and despite an infinite number of possible instantiations of periodic patterns, the number of distinct symmetry groups for periodic patterns in any Euclidean space  $R^n$  is always *finite*! These groups are often referred to as *crystallographic groups* [42, 84, 277]. For 2D monochrome patterns, there are 7 *frieze-symmetry groups* translating along one direction (strip-patterns) [42] (Figure 1.3(C)). and 17 *wallpaper-groups* covering the whole plane [61, 84] (Figure 1.3(D)). In 3D, there are 230 different *space groups* [98] generated by three linearly independent translations (regular crystal patterns).

### Frieze Symmetry Group

A frieze pattern is a 2D strip in the plane that is periodic along one dimension. Any frieze pattern  $P$  is associated with one of the seven unique symmetry groups (Figure 2.4). These seven symmetry groups, denoted by crystallographers as  $l1, lg, ml, l2, mg, lm, mm$  [42], are called the *frieze groups*. These frieze groups are infinite yet discrete. Without loss of generality, assume the direction of translation symmetry of a frieze pattern is horizontal, the frieze pattern can exhibit five different types of symmetries (Figure 2.4(A)):

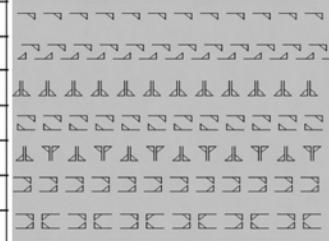
- (1) horizontal translation;
- (2) two-fold rotation;
- (3) horizontal reflection (reflection axis is placed horizontally);
- (4) vertical reflection; and
- (5) horizontal glide-reflection composed of a half-unit translation followed by a horizontal reflection.

The primitive symmetries in each group (the inner-structure of a frieze group) and the relationship among the seven frieze groups (inter-structure of frieze groups) are depicted in Figure 2.4(A) and (B) respectively. Each frieze pattern is associated with one of the seven possible frieze groups, depending on the unique and legal combination of these five primitive symmetries presented in the pattern (Figure 2.4(A)). Not all possible combinations of symmetries form legit-

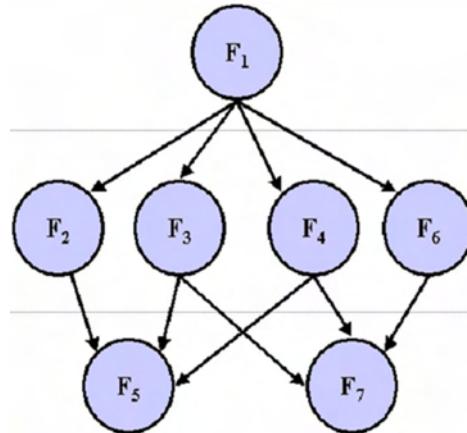
## 24 Symmetry and Symmetry Groups

Symmetry Group	translation	2-fold rotation	Horizontal reflection	Vertical reflection	Glide reflection
F1 11	yes	no	no	no	no
F2 1g	yes	no	no	no	yes
F3 ml	yes	no	no	yes	no
F4 12	yes	yes	no	no	no
F5 mg	yes	yes	no	yes	yes
F6 lm	yes	no	yes	no	no
F7 mm	yes	yes	yes	yes	no

Seven Frieze Patterns



(A) Primitive symmetries (PS) in frieze groups.



(B) Inter-structure of frieze groups.

Fig. 2.4 The inner (A) and inter-structures (B) of the seven frieze groups are shown. The subgroup structures of frieze groups provide a sound regularity scale: the larger the symmetry group the higher the regularity in general.

imate symmetry groups, for example, a horizontal frieze pattern as shown in Figure 2.4(A) cannot exhibit both horizontal reflection and glide reflection symmetries simultaneously.

### Wallpaper Symmetry Group

A wallpaper pattern is a 2D periodic pattern extending along two linearly independent directions [42, 235] (Figure 2.5). Any wallpaper pattern is associated with one of the 17 wallpaper groups. Wallpaper group theory [84] states that all translationally symmetric patterns  $P_r$  can be generated by a pair of linearly independent, shortest (among all

## Examples of 17 Wallpaper Patterns

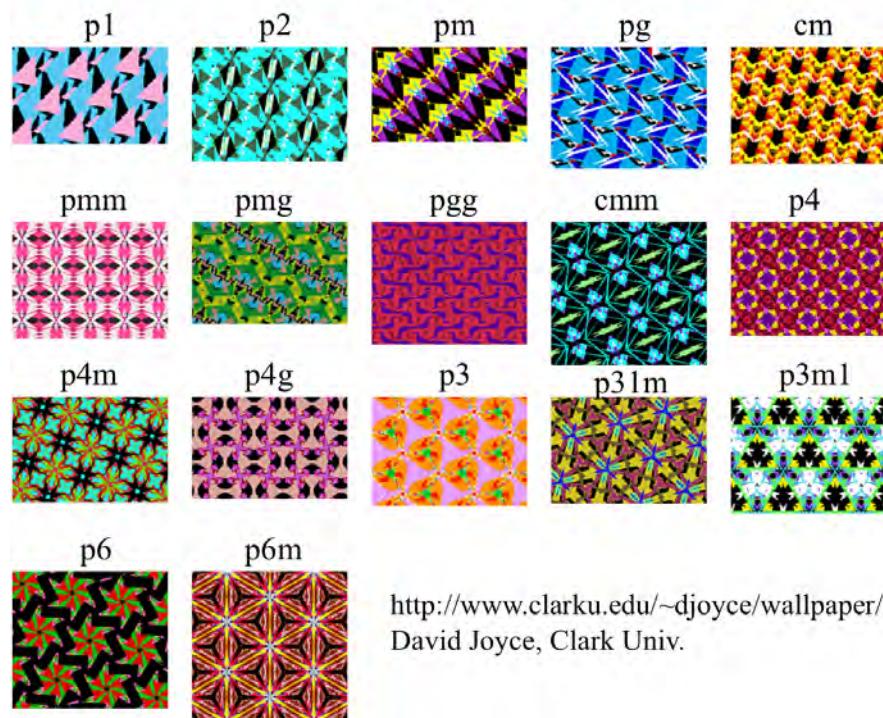


Fig. 2.5 Sample wallpaper patterns associated with the 17 distinct wallpaper groups.

possible) vectors  $t_1, t_2$  applied to a minimum sized tile. The *orbits* of this pair of translation symmetry generator vectors form a 2D *quadrilateral lattice*, which simultaneously defines all 2D tiles (partitions the space into its smallest generating regions) and a topological lattice structure relating all tiles (Figure 2.6). The unit lattice of each of the 17 wallpaper groups and their relations are shown in Figure 2.7.

### Motifs of Wallpaper Patterns

When translational symmetry subgroup of a periodic pattern is determined, it fixes the size, shape and orientation of the unit lattice, but leaves open the question of where the unit lattice should be located in the pattern. Any parallelogram of the same size and shape carves out an equally good tile that can be used to tile the plane. However,

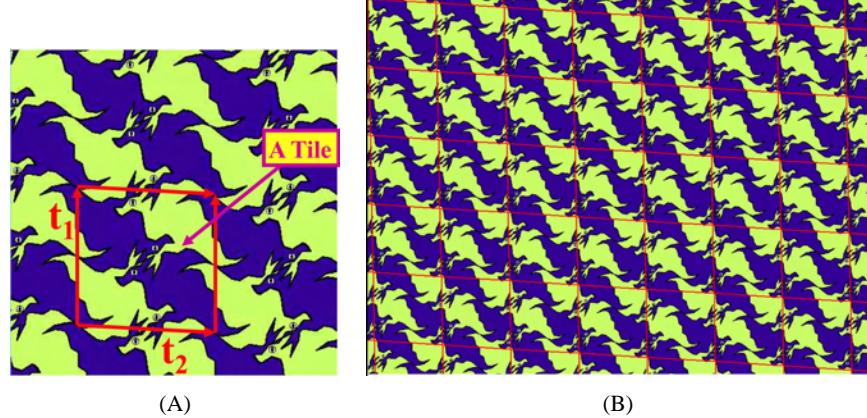


Fig. 2.6 (A) The generators  $t_1$  and  $t_2$  simultaneously determine the *lattice unit* (shape, size, orientation) and a *tile* (a parallelogram-shaped piece of the wallpaper) that can tile the entire 2D plane. (B) The 2D lattice captures the translation subgroup of the wallpaper pattern.

from a perception point of view, some parallelograms produce tiles that are better descriptors of the underlying symmetry of the overall pattern than others. For example, if the whole pattern has some rotation symmetries, a tile located on the rotation centers can reflect the global symmetry property of the wallpaper pattern instantly (Figure 2.6(A)). Such *motifs*, as representative tiles of a periodic pattern, can be defined mathematically:

---

**Definition 12.** A motif of a wallpaper pattern is a tile that is cut out by a lattice whose lattice unit is centered on the fixed point of the largest stabilizer group.

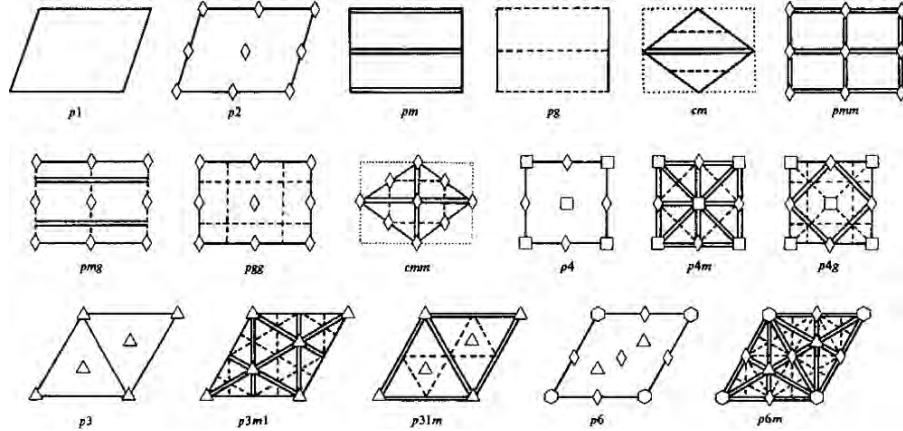
---

Candidate motifs can then be determined systematically by enumerating each *distinct* centers of the highest-order rotation in the wallpaper pattern ([28]).

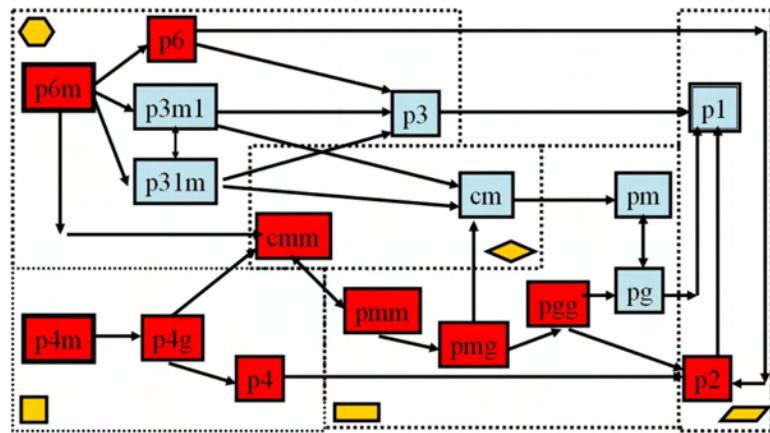
---

**Definition 13.** Two rotation centers of the rotation subgroups with the same order are distinct if they lie in different orbits of the symmetry group, that is, if one cannot be mapped into the other by applying any translation, rotation, reflection or glide-reflection symmetries in the symmetry group.

---



(A) Unit lattice and PSs of the 17 wallpaper groups.



(B) Inter-structure of wallpaper groups.

Fig. 2.7 The inner (A) and inter-structures (B) of the 17 wallpaper groups are shown. The symbols  $p_1, p_2, \dots, p_4m, p_6m$  are crystallographers' representations for wallpaper group [235]. The diamond, triangle, square and hexagon shapes correspond to 2, 3, 4 and 6-fold rotation centers. Solid single line, dotted single line and double parallel lines denote unit translation, glide-reflection and reflection symmetries. (B), first appeared in [156], is a pictorial interpretation of wallpaper group relations from [44].

Algorithmically, one can characterize a wallpaper pattern by determining the distinct orbits of a given pattern and shifting the found unit lattice to center on each of the highest order of rotation symmetries (Figures 2.8, 2.9 and 2.10).

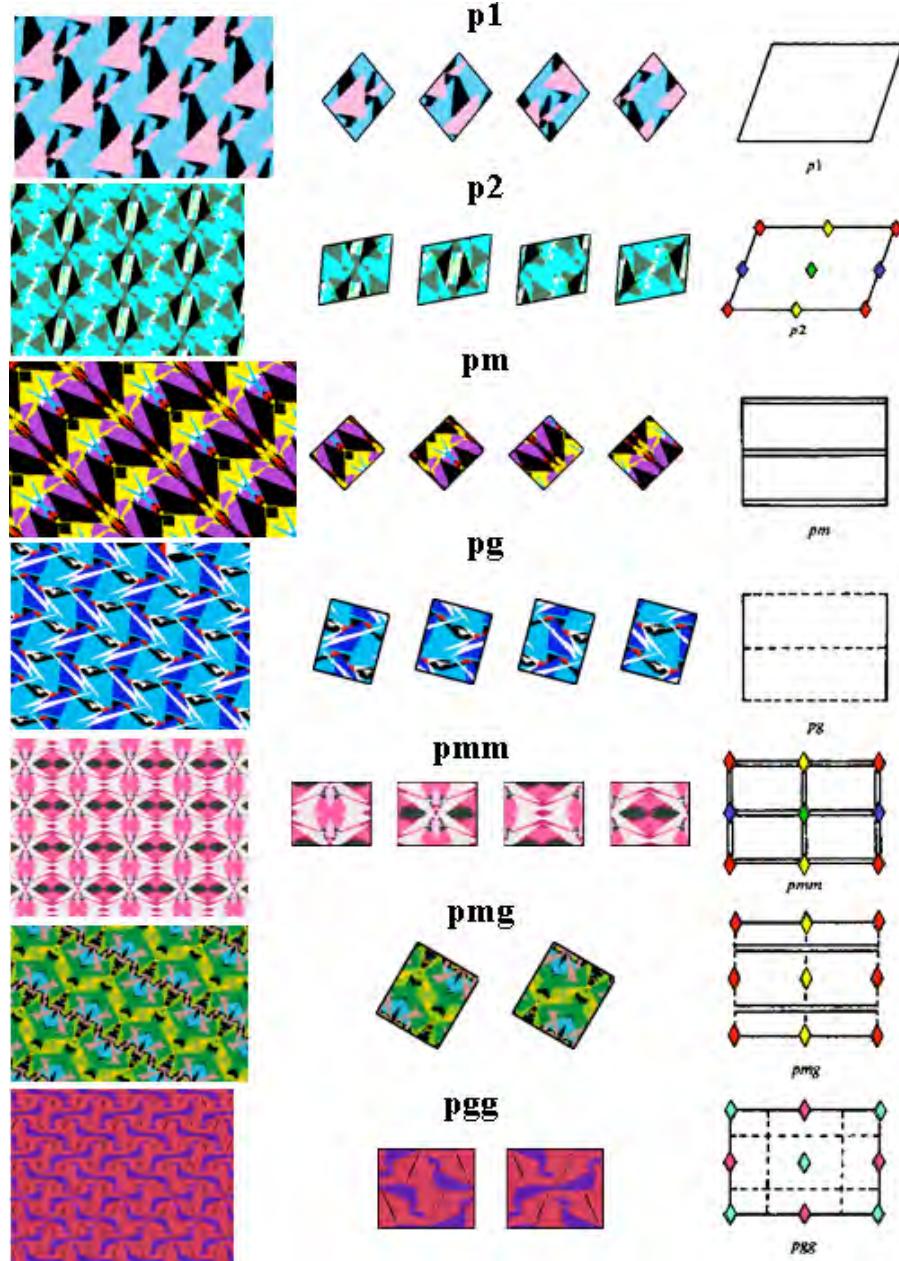


Fig. 2.8 Motifs centered on the highest order of rotation symmetry centers. Different colors indicate distinct orbits. For symmetry groups without rotation centers ( $p1$ ,  $pm$ ,  $pg$ ,  $cm$ ), approximate rotation symmetry centers are used (courtesy of [159]).

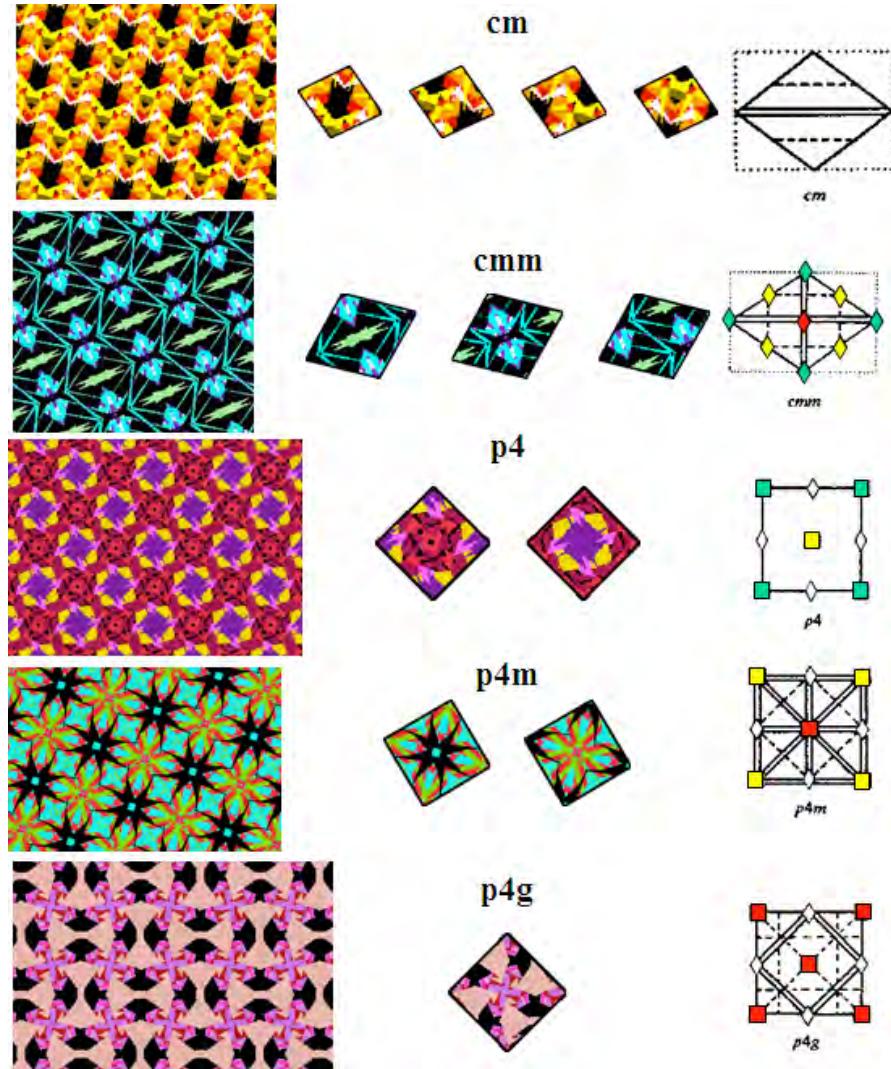


Fig. 2.9 Motifs centered on the highest order of rotation symmetry centers. Different colors indicate distinct orbits. For symmetry groups without rotation centers ( $p1$ ,  $pm$ ,  $pg$ ,  $cm$ ), approximate rotation symmetry centers are used (courtesy of [159]).

### Infinite versus Finite Periodic Patterns

Mathematically, frieze and wallpaper groups are defined only for infinite periodic patterns  $P$ . In practice, we use the term “symmetry group  $G$  of  $P$ ” equivalently for the infinite periodic pattern  $P$  or a finite

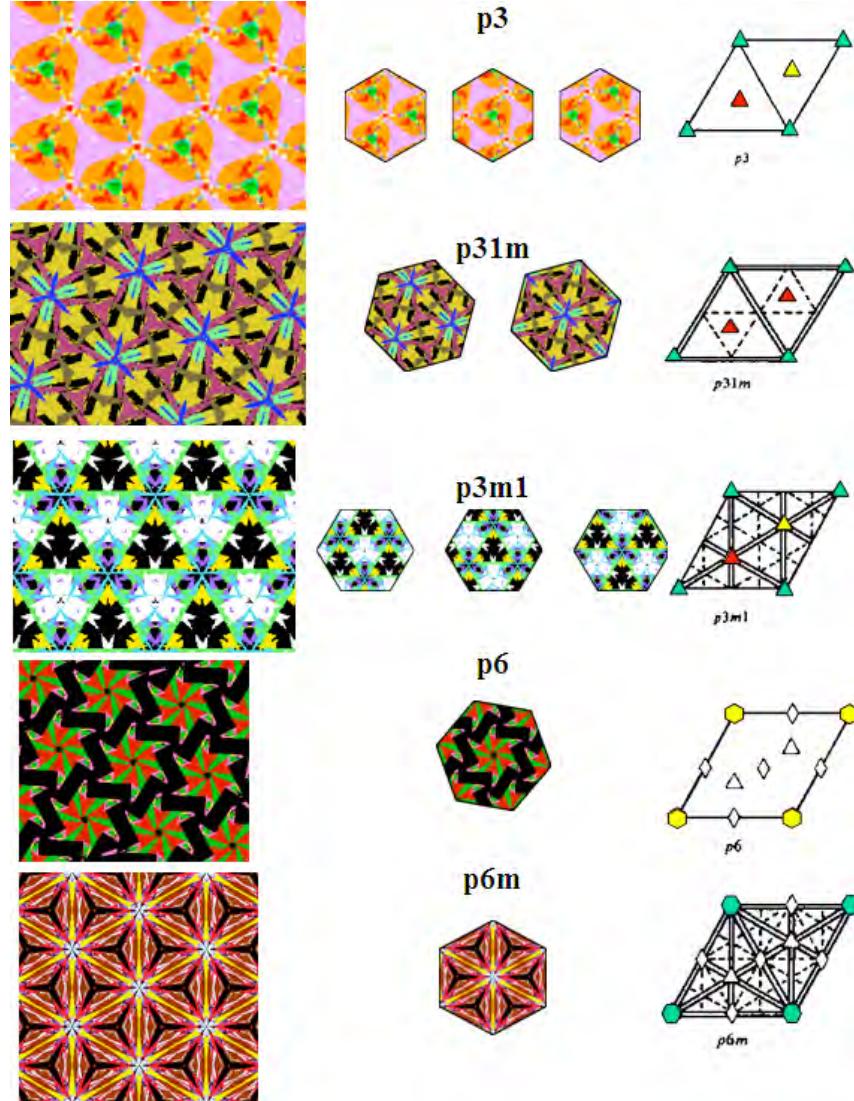


Fig. 2.10 Motifs centered on the highest order of rotation symmetry centers. Different colors indicate distinct orbits. For symmetry groups without rotation centers ( $p1$ ,  $pm$ ,  $pg$ ,  $cm$ ), *approximate rotation symmetry centers* are used (courtesy of [159]).

segment of  $P$ . An intuitive “computer vision” interpretation is to view any real world finite periodic pattern segment as an *occluded* or *cropped* version of an infinite periodic pattern.

$E(3)$ : Euclidean group in  $\mathbb{R}^3$   
 $E^+$ : the proper Euclidean group  
 $O$   $\rightarrow$  Orthogonal group  
 $SO$   $\rightarrow$  Special Orthogonal group  
 $T$   $\rightarrow$  Translation group  
 $T_{dis}$   $\rightarrow$  Discrete Translation group  
 $D$   $\rightarrow$  Dihedral group  
 $C$   $\rightarrow$  Cyclic group  
 $G_{crystal}(3)$   $\rightarrow$  Crystallographic groups in  $\mathbb{R}^3$   
 $G_{plato}$   $\rightarrow$  Symmetry groups for Platonic solids  
 $G_{id}$   $\rightarrow$  The identity group  
 **$A \rightarrow B$  means:  $B$  is a subgroup of  $A$**   
 **$A \dashrightarrow B$  means:  $B$  is a subgroup of  $A$  for some  $n$**

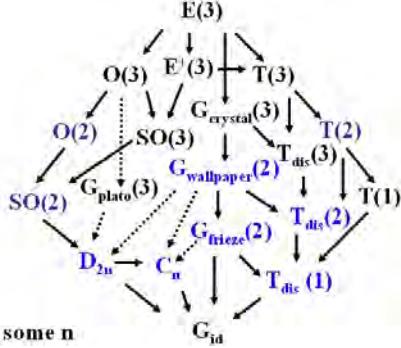


Fig. 2.11 The Euclidean group and its major subgroups, providing an overview of the types of symmetry groups in 3D Euclidean space. In 2D, the four light-blue colored symmetry groups (cyclic, dihedral, frieze and wallpaper) are the most representative regularities, with  $T_{dis}(1), T_{dis}(2)$  as their subgroups and  $O(2), SO(2)$  and  $T(2)$  as their extreme (continuous) cases.

### 2.1.4 Euclidean Group and Its Subgroup Hierarchy

Figure 2.11 depicts the hierarchical structure of the 3D Euclidean group and its subgroups (defined in Tables 2.1 and 2.2).

A full understanding of symmetry group types and their hierarchical structures (Table 2.3, Figures 2.4, 2.6, 2.11) provides a theoretically sound set of qualitative standards for regularity. Computationally, this understanding also helps us to focus on a comprehensive yet computationally tractable set of symmetry groups. In 2D space for example, symmetry discovery becomes no more and *no less* than searching for cyclic, dihedral, frieze and wallpaper symmetry groups (Table 2.3).

An understanding of symmetry groups also helps us to develop more robust computer algorithms. Instead of looking for isolated symmetries in a noisy real world image, we can search for specific, predictable, structured symmetries: the symmetry groups.

## 2.2 Symmetry in Affine and Projective Spaces

### 2.2.1 Affine-Skewed Symmetry Groups [156]

Symmetry groups are composed of rigid Euclidean transformations (Section 2.1). We define *skewed symmetry groups* as affinely transformed Euclidean symmetry groups. The term *skewed symmetry groups*

## 32 Symmetry and Symmetry Groups

first appeared in [156] where a computational treatment for 2D wallpaper groups under affine transformations is presented. This seems to be the first time that wallpaper groups are studied under full affine deformations (conjugated by affine transformations) for computer vision applications [43].

The study of skewed symmetry groups [156] is motivated by the following observations: (1) even though the appearance of a periodic pattern can change infinitely and sometimes drastically under all possible affine transformations, its symmetry group stays in a small finite set (much smaller than 17) and relatively stable, and thus provides a good index for regular textures viewed at arbitrary angles [156]; (2) according to a well-defined symmetry group hierarchy (Figure 2.6) [44], different wallpaper patterns have their respective highest order of symmetry group under all possible affine deformations. For example, all three patterns in Figure 2.12 initially (leftmost patterns) have the 2-fold ( $180^\circ$ ) rotation as their highest order of symmetry. Under certain affine deformations the symmetry group of each pattern follows a different path to reach its own ‘highest’ order: (1) has 4-fold rotational symmetry, (2) has 6-fold and (3) still only has 2-fold rotation as its highest order of rotation symmetry.

In robotics manipulation related research, where only rigid transformations are considered, we can show (Proposition 3.3.1 [166]):

**Proposition:** If  $G$  is a symmetry group of  $P$  and  $A$  is a rigid transformation, then  $AGA^{-1}$  is the symmetry group of  $A(P)$ .

Here,  $AGA^{-1}$  is a conjugation of group  $G$  via  $A$  [69]. When  $A$  is a rigid transformation or a uniform scaling, there exists a bijection between the original symmetry group and the conjugated symmetry group, and thus  $G$  and  $AGA^{-1}$  are considered equivalent (isomorphic).

When a periodic pattern undergoes non-rigid transformations (other than uniform scaling), the above proposition no longer holds. In  $AGA^{-1}$  the original symmetry group  $G$  is being skewed by  $A$ , thus the term *skewed symmetry groups* [156]. It is useful to note that certain symmetries do survive some constrained or even general affine transformations. For example, 2-fold rotation symmetry remains under all non-degenerative affine transformations. The question is whether

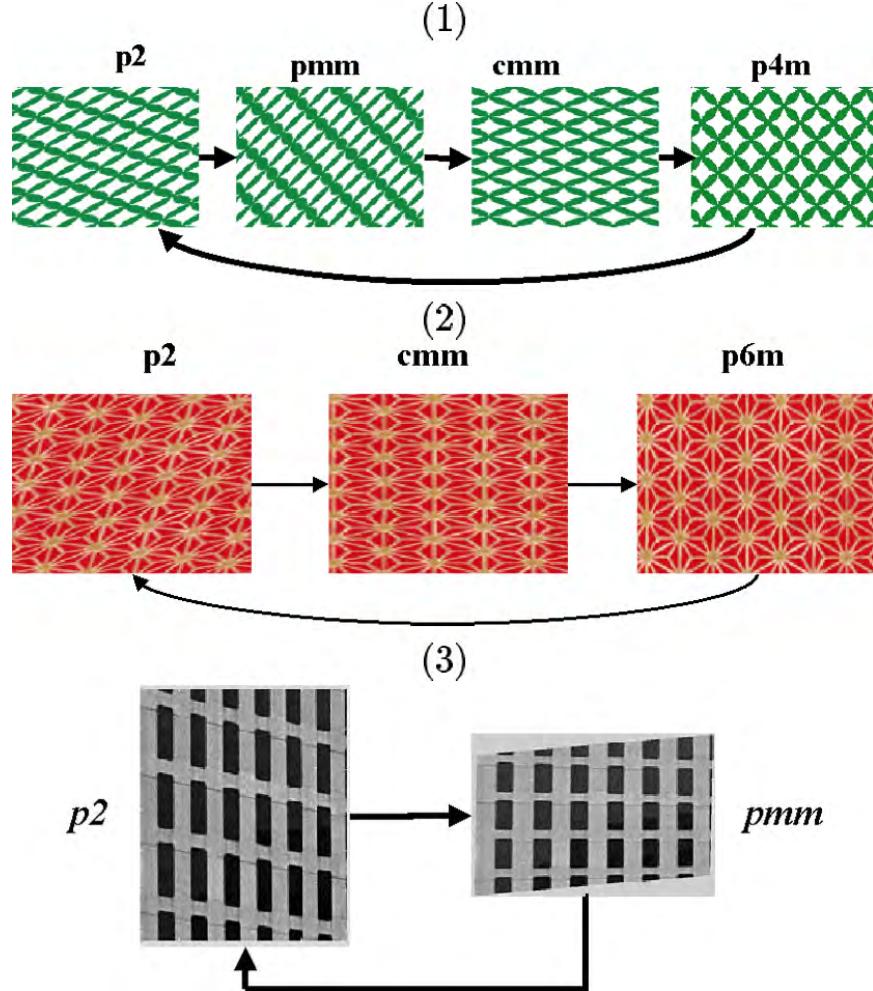


Fig. 2.12 When a pattern is deformed by affine transformations, its symmetry group migrates to different groups within its *special orbit*: (1)  $p2 \rightarrow pmm \rightarrow cmm \rightarrow p4m$ , (2)  $p2 \rightarrow cmm \rightarrow p6m$ , (3)  $p2 \rightarrow pmm$ . Note: the labels  $p1, p2, p3, p6, \dots$  are classic notations for crystallographic groups. For details see [44, 235].

$AGA^{-1}$  retains its original group structure, and if not, which symmetry group can it migrate to?

Let us examine under what conditions a symmetry of a pattern  $P$  remains when an affine transformation  $A$  is applied to  $P$ . If  $g$  is a symmetry of pattern  $P$ ,  $g(P) = P$  (definition of symmetry). For each

$x \in P$ , there exists a unique  $y \in P$  such that  $g(x) = y$ . Let  $A = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$ .

Applying  $A$  to  $P$ , and assuming  $g$  remains a symmetry of  $A(P)$  so  $g(A(P)) = A(P)$  (modulo translations and rotations in  $A$  acting on  $g$  since we know they do preserve symmetries), we have

$$g(A(x)) = A(y) \Rightarrow g(A(x)) = A(g(x)) \Rightarrow gA = Ag.$$

From this relation, we derive constraints that affine transformation  $A$  must satisfy to maintain symmetry  $g$

**1) When  $g$  is a 2-fold rotation** (rotation by  $180^\circ$ ),

$$g = \begin{vmatrix} -1 & 0 \\ 0 & -1 \end{vmatrix},$$

$$gA = \begin{vmatrix} -a & -b \\ -c & -d \end{vmatrix}, \quad Ag = \begin{vmatrix} -a & -b \\ -c & -d \end{vmatrix}$$

There are thus no constraints on the values of  $a, b, c$ , and  $d$ . This means that a 2-fold rotational symmetry is invariant to any non-singular affine transformation.

**2) W.l.g. when  $g$  is a reflection** about the  $Y$  axis

$$g = \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix},$$

$$gA = \begin{vmatrix} -a & -b \\ c & d \end{vmatrix}, \quad Ag = \begin{vmatrix} -a & b \\ -c & d \end{vmatrix}$$

The derived constraints are:  $b = 0$  and  $c = 0$ . This means that a reflection is invariant **only** to non-uniform scaling parallel and perpendicular to the axis of reflection.

**3) When  $g$  is an  $n$ -fold rotation where  $n \neq 2$**  (in this case, rotations by  $120^\circ$ ,  $90^\circ$  and  $60^\circ$ )

$$g = \begin{vmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{vmatrix}$$

$$gA = \begin{vmatrix} a \cos \theta - c \sin \theta & b \cos \theta - d \sin \theta \\ a \sin \theta + c \cos \theta & b \sin \theta + d \cos \theta \end{vmatrix}$$

$$Ag = \begin{vmatrix} a \cos \theta + b \sin \theta & -a \sin \theta + b \cos \theta \\ c \cos \theta + d \sin \theta & -c \sin \theta + d \cos \theta \end{vmatrix}$$

Equating the two sides, we have the constraints:

$$\begin{aligned} a \cos \theta - c \sin \theta &= a \cos \theta + b \sin \theta \Rightarrow b = -c \\ b \cos \theta - d \sin \theta &= -a \sin \theta + b \cos \theta \Rightarrow a = d \\ a \sin \theta + c \cos \theta &= c \cos \theta + d \sin \theta \Rightarrow a = d \\ b \sin \theta + d \cos \theta &= -c \sin \theta + d \cos \theta \Rightarrow b = -c \end{aligned}$$

Therefore

$$A = \begin{vmatrix} a & b \\ -b & a \end{vmatrix} = \underbrace{\sqrt{a^2 + b^2}}_S \underbrace{\begin{vmatrix} \frac{a}{\sqrt{a^2 + b^2}} & \frac{-b}{\sqrt{a^2 + b^2}} \\ \frac{b}{\sqrt{a^2 + b^2}} & \frac{a}{\sqrt{a^2 + b^2}} \end{vmatrix}}_R$$

where  $S$  is a uniform scaling and  $R$  is a rotation with  $\theta = \text{atan2}(b, a)$ . Therefore, 3-fold, 4-fold and 6-fold rotational symmetries are only invariant to similarity transformations.

Based on the above results, we can derive a set of conditions that specify when two *different* symmetry groups,  $G_1$  and  $G_2$ , *cannot* be transformed into each other:

- (1)  $G_1$  has a 2-fold rotation symmetry but  $G_2$  does not (2-fold rotation survives any nonsingular affine distortion);
- (2)  $G_1$  and  $G_2$  have the same lattice type;
- (3) after deforming the lattice type of group  $G_1$  into the lattice type of group  $G_2$ , at least one remaining symmetry in the deformed  $G_1$  differs from all symmetries in  $G_2$ ; and
- (4)  $G_1$  and  $G_2$  do not have a subgroup relationship (Figure 2.6).

We can now construct a  $17 \times 17$  “migration map” that lists the complete set of groups that any one of the 17 wallpaper groups can be transformed into under affine transformations (Table 2.4).

For the first time, a given periodic pattern can be viewed not simply under its Euclidean symmetry group but its highest *potential symmetry group* under all possible affine deformations. The importance of this result is a constructive, well-defined *canonical* representation of any periodic patterns (its most symmetrical form) that can be used in practice for diverse applications, ranging from an indexing and retrieval scheme for regular patterns, to a maximal-symmetry-based shape and orientation estimator from texture under unknown views [156].

Table 2.4. Wallpaper group migration map.

	p1	p2	pm	pg	cm	pmm	pmg	pgg	cmm	p4	p4m	p4g	p3	p3m1	p6	p6m
p1	<b>A</b>	1	<b>P</b>	<b>P</b>	1	1	1	1	1	<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>	1	1	1
p2	1	<b>A</b>	1	<b>P</b>	1	<b>P</b>	<b>P</b>	<b>P</b>	<b>P</b>	1	1	1	<b>P</b>	1	<b>P</b>	<b>P</b>
pm	<b>A</b>	1	<b>N</b>	2	3	1	1	1	1	1	1	1	4	3	3	1
pg	<b>A</b>	1	2	<b>N</b>	3	1	1	1	1	1	1	1	4	3	3	1
cm	<b>A</b>	1	3	3	<b>N</b>	1	1	1	1	1	1	1	4	<b>P</b>	<b>P</b>	1
pmm	1	<b>A</b>	1	1	1	<b>N</b>	2	2	<b>P</b>	3	3	1	1	1	1	4
pmg	1	<b>A</b>	1	1	1	2	<b>N</b>	2	3	3	3	1	1	1	1	4
pgg	1	<b>A</b>	1	1	1	2	2	<b>N</b>	<b>P</b>	3	3	<b>P</b>	1	1	1	4
cmm	1	<b>A</b>	1	1	1	<b>P</b>	3	<b>P</b>	<b>N</b>	3	<b>P</b>	<b>P</b>	1	1	1	<b>P</b>
p4	1	<b>A</b>	1	1	1	3	3	3	<b>S</b>	2	2	1	1	1	1	4
p4m	1	<b>A</b>	1	1	1	<b>N</b>	3	3	<b>N</b>	2	<b>S</b>	2	1	1	1	3
p4g	1	<b>A</b>	1	1	1	3	3	<b>N</b>	<b>N</b>	2	<b>S</b>	1	1	1	1	3
p3	<b>A</b>	1	4	4	4	1	1	1	1	1	<b>S</b>	2	2	1	1	4
p3m1	<b>A</b>	1	3	3	<b>N</b>	1	1	1	1	1	1	2	<b>S</b>	2	1	1
p3m	<b>A</b>	1	3	3	<b>N</b>	1	1	1	1	1	1	1	2	2	1	1
p6	1	<b>A</b>	1	1	1	4	4	4	3	4	3	3	1	1	1	<b>S</b>
p6m	1	<b>A</b>	1	1	1	3	3	3	<b>N</b>	3	3	4	1	1	1	2

This is a  $17 \times 17$  table where each entry indicates whether the row-group to the left of the table can be affinely transformed into a column-group on the top of the table by: **S**, a similarity transformation, **N**, a non-uniform scaling  $\perp$  or  $\parallel$  to all reflection axes in the group to the left, **A**, a general affine transformation other than **S** or **N**. An entry of **P** means that row-group may or may not be transformed into the column-group, depending on the particular pattern (see Figure 2.12 for some examples). Entries with numbers state that there are no affine deformations that can possibly transform the row-group to the column-group, and the number listed indicates which of the four conditions listed above hold (only one number is listed when multiple justifications hold) [156].

### 2.2.2 Symmetry-specific Invariants under Perspective Transformation

A large part of this survey deals with symmetries of planar shapes. But when these are observed from oblique viewing directions, this leads to deformations in the image. These deformations tend to destroy some of the characteristics that make symmetric patterns special in the first place, but fortunately not all such characteristics go lost. Depending on the viewing conditions, i.e., whether they can be modeled by pseudo-orthographic projection or the full perspective model needs to be used, more or fewer such characteristics may survive. Such characteristics are called *invariant features* or *invariants* for short. In this section, we will first have a closer look at characteristics that remain invariant under perspective projection, then we will comment on the pseudo-orthographic case, which is far more benign. In particular, we will derive invariants that are specific for skewed symmetries as they appear for the ornamental groups (cyclic and dihedral symmetry groups).

#### The groups governing skewed symmetries

Much of our analysis is based on the classifications of subgroups of the plane projectivities. Nevertheless, the planar shapes that are involved can be part of non-planar configurations. Moreover, sometimes the same geometrical analysis applies to the grouping of curved surfaces. Surfaces of revolution are a good case in point, as their outlines share their geometrical constraints with skewed planar symmetric shapes [211].

First, we recapitulate some issues of general, projective invariants, to then press on with their specialisation towards the symmetry-specific subgroups. This also stands to reason from a mathematical point of view. As the symmetry-specific transformations will form subgroups of the projectivities, any projective invariant will also be an invariant for the symmetry-related groups.

##### 2.2.2.1 Projective invariants

**Introduction and notations.** When observing a planar pattern from two different viewpoints, corresponding points in the two image

projections are related by a projective transformation (or projectivity). If  $(x', y')$  are the image coordinates in the second image of the point which corresponds to the point  $(x, y)$  in the first image, then there are nine coefficients  $p_{ij}$  such that,

$$\begin{aligned} x' &= \frac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + p_{33}} \\ y' &= \frac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + p_{33}} \end{aligned} \quad (2.1)$$

For other pairs of corresponding points the same coefficients  $p_{ij}$  and thus the same projectivity apply. Although there are nine coefficients, a projectivity only has eight degrees of freedom, as multiplying each coefficient with the same non-zero scalar keeps the transformation unchanged.

The symmetric patterns that we discuss here are co-planar, although parts of this discussion also pertain to relations between planar but not necessarily co-planar patterns. When co-planar, symmetric subpatterns are viewed from different viewpoints, the transformation between their images will still be projective. Due to the special nature of the symmetric pattern, special subgroups of the projectivities will apply, however. This issue will be discussed in the following subsections. There we will derive features that remain unchanged between the different views, i.e., invariants. But before we do that, we first have a look at the features that remain invariant under just any projectivity.

We will assume that we have available points for which we can also identify the corresponding points in other views. The points are characterised by their image coordinates  $\mathbf{x} = (x, y)^T$ , where “T” stands for the transpose. In order to distinguish between multiple points, we will use subscripts:  $\mathbf{x}_i = (x_i, y_i)^T$  are the coordinates of point  $i$ . Moreover, sometimes points lie on curves. The coordinates of  $\mathbf{x}(t) = (x(t), y(t))^T$  of such points are ordered along the curve, with some parameter  $t$  increasing monotonously when moving along the curve. For points along a curve with sufficient continuity, one can also extract their derivatives with respect to the parameter  $t$ . These will be written as:

$$\mathbf{x}^{(j)} = \begin{pmatrix} x^{(j)} \\ y^{(j)} \end{pmatrix} = \begin{pmatrix} \frac{d^j x}{dt^j} \\ \frac{d^j y}{dt^j} \end{pmatrix}.$$

Please note that these derivatives will change their values upon changing the parameterisation of the curve, i.e., when replacing the parameter  $t$  by another parameter  $t'$ . More about this is to follow. The notation  $\mathbf{x}_i^{(j)}$  then indicates the  $j$ th derivative of the coordinates at  $\mathbf{x}_i$ . The notation  $\mathbf{x}^{(j:t)} = (x^{(j:t)}, y^{(j:t)})^T$  is used to denote derivatives with respect to a specific parameter  $t$ . The notation  $s$  will be reserved for Euclidean arclength, i.e., the actual length traversed when moving along the curve, measured in the image from some starting point for which  $s = 0$ .

Continuing our introduction of notations for this section, single bars ( | ) will indicate determinants, and double bars ( || ) norms. As an example

$$|\mathbf{x}_1 - \mathbf{x}_2 \mathbf{x}_1 - \mathbf{x}_3| = (x_1 - x_2)(y_1 - y_3) - (y_1 - y_2)(x_1 - x_3)$$

and

$$\|\mathbf{x}^{(1)}\| = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2},$$

with  $t$  an arbitrary parameterisation of the curve. With these notations, one also has

$$s(t) = \int_0^t \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt$$

**Building blocks for invariants:** In our quest for projective invariants, we will take a shortcut when compared to the strict derivation through so-called Lie prolongations. This shortcut will make use of some simple expressions, which will serve as building blocks of the invariants. As we will see, extracting invariants becomes an issue of solving simple systems of linear equations, rather than the systems of differential equations coming from the Lie prolongations.

First, recalling the expression for planar projective transformations, we have

$$\begin{aligned} x' &= \frac{p_{11}x + p_{12}y + p_{13}}{p_{31}x + p_{32}y + p_{33}}, \\ y' &= \frac{p_{21}x + p_{22}y + p_{23}}{p_{31}x + p_{32}y + p_{33}}. \end{aligned}$$

In the sequel,  $N$  will serve as a shorthand notation for the denominator of these expressions:

$$N = p_{31}x + p_{32}y + p_{33}$$

and similarly  $N_i = p_{31}x_i + p_{32}y_i + p_{33}$ . Note that the projective transformation can then be rewritten as

$$N \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}.$$

Hence, for three transformed points the action of the projectivity can be summarized as

$$\begin{pmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1/N_1 & 0 & 0 \\ 0 & 1/N_2 & 0 \\ 0 & 0 & 1/N_3 \end{pmatrix}.$$

Taking the determinant of the left and right hand sides immediately yields

$$|\mathbf{x}'_1 - \mathbf{x}'_2 \ \mathbf{x}'_1 - \mathbf{x}'_3| = \frac{|P|}{N_1 N_2 N_3} |\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1 - \mathbf{x}_3| \quad (2.2)$$

where  $P$  denotes the matrix of the transformation parameters  $p_{ij}$ . The determinant expressions on both hand sides, i.e., before and after the transformation, are related by a factor. This factor in turn is composed of a number of factors. In order to find invariants one can combine several such determinants in order to eliminate all such factors. One should be careful not to generate trivial invariants (i.e., constants) though.

This type of determinant, which combines the coordinates of three different points, is one type of building block for our projective invariants. There are two other types, which change with similar factors under the application of a projectivity. These are, together with

their factors:

Building block	Factor
$ \mathbf{x}_1 - \mathbf{x}_2 \mathbf{x}_1 - \mathbf{x}_3 $	$\rightarrow \frac{ P }{N_1 N_2 N_3}$
$ \mathbf{x}_1 - \mathbf{x}_2 \mathbf{x}_1^{(1)} $	$\rightarrow \frac{ P }{N_1^2 N_2}$
$ \mathbf{x}_1^{(1)} \mathbf{x}_1^{(2)} $	$\rightarrow \frac{ P }{N_1^3}$

Each of these building blocks has an especially simple interpretation. The first building block gives the area of the parallelogram defined by the three points needed to calculate it. Also the other two building blocks take on a simple meaning *if* the derivatives are calculated with respect to Euclidean arclength  $s$ ,

$$s(t_x) = \int_0^{t_x} \|\mathbf{x}^{(1:t)}\| dt$$

with  $t$  an arbitrary parameter and  $t_x$  its particular value at the point  $\mathbf{x}$  which moves along the curve when  $t$  changes. It corresponds to length measured along the contour. Using  $s$  for the parameter, the second block measures the distance of point  $\mathbf{x}_2$  to the tangent line to the contour at point  $\mathbf{x}_1$ . This is the case because the tangent vector  $\mathbf{x}_1^{(1)}$  is a unit vector (choice of Euclidean arclength as parameter!). The third block expresses (Euclidean) curvature at the point  $\mathbf{x}_1$ . Again, this interpretation hinges on the choice of Euclidean arclength for the parameter, since in general curvature is given by

$$\frac{|\mathbf{x}^{(1)} \mathbf{x}^{(2)}|}{\|\mathbf{x}^{(1)}\|^3}.$$

These interpretations are illustrated in Figure 2.13.

Finally, an example is given to illustrate how the combination of building blocks can yield invariants. Consider

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \mathbf{x}_1 - \mathbf{x}_5| |\mathbf{x}_3 - \mathbf{x}_4 \mathbf{x}_3 - \mathbf{x}_5|}{|\mathbf{x}_1 - \mathbf{x}_3 \mathbf{x}_1 - \mathbf{x}_5| |\mathbf{x}_2 - \mathbf{x}_4 \mathbf{x}_2 - \mathbf{x}_5|} . \quad (2.3)$$

It is an easy exercise to check that all the factors produced under a projective transformation cancel. Notice that this invariant uses five

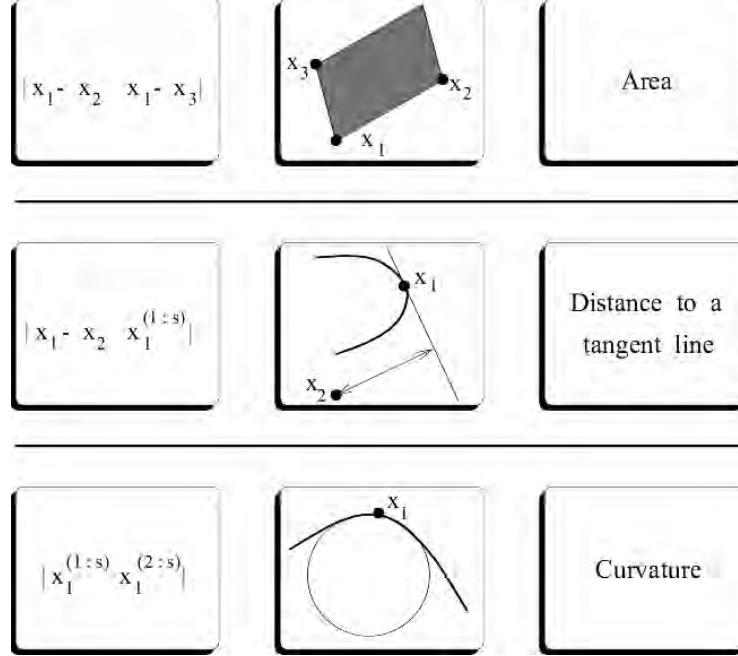


Fig. 2.13 Euclidean interpretations for the different projective building blocks. Left: formula for the building block; middle: graphical interpretation; right: interpretation in wording.

points. The above expression has the same value, whether one uses the coordinates of the five points before or those of the corresponding five points after the projective transformation. A second, independent invariant on the basis of the same five points can be constructed:

$$\frac{|\mathbf{x}_3 - \mathbf{x}_2 \ \mathbf{x}_3 - \mathbf{x}_5| \ |\mathbf{x}_1 - \mathbf{x}_4 \ \mathbf{x}_1 - \mathbf{x}_5|}{|\mathbf{x}_1 - \mathbf{x}_3 \ \mathbf{x}_1 - \mathbf{x}_5| \ |\mathbf{x}_2 - \mathbf{x}_4 \ \mathbf{x}_2 - \mathbf{x}_5|}.$$

Additional invariants based on the coordinates of the five points are dependent on the two foregoing invariants.

**Reparameterisation.** As we have seen, two out of the three projective building blocks contain derivatives. These can only be extracted in as far as the points where they are evaluated only on contours. Such derivatives are taken with respect to some contour parameter, say  $t$ . When the parameterization is changed, the derivatives will take on different values, however. Usually, when given two contours and wanting

to know whether they are possibly the projection of the same contour, we would also not know which point on one contour could then correspond to which point on the other. We would have to start our investigations with parameterizations of the contours in the different images that are still completely unrelated. Thus, if invariants should help us with the comparison, they should be invariant under such changes of parameter — i.e., under reparameterisation — as well.

Consider a general curve parametrised first with some parameter  $t$  as  $(x(t), y(t))$  and then with a parameter  $t^*$  as  $(x(t^*), y(t^*))$ . Then,

$$\frac{dx}{dt} = \frac{dt^*}{dt} \frac{dx}{dt^*}.$$

The same holds for  $y$ . We see that the first derivative changes with a factor  $dt^*/dt$  when we change the parameter. This factor corresponds to the relative ‘speed’ of the two parameterisations at the point where we consider the derivative. This factor will in general differ from point to point, depending on the variations in the mapping  $t = \phi(t^*)$  between the parameterisations. Denoting differentiation with respect to  $t$  by a dot, and differentiation with respect to  $t^*$  by a prime, the following transformation matrix for the derivatives is obtained:

$$\begin{pmatrix} x' \\ x'' \\ x''' \\ \vdots \end{pmatrix} = \begin{pmatrix} \phi' & 0 & 0 & \dots \\ \phi'' & \phi'^2 & 0 & \dots \\ \phi''' & 3\phi'\phi'' & \phi'^3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dddot{x} \\ \vdots \end{pmatrix} \quad (2.4)$$

Similar relations hold for  $y$ . Taking into account that the function  $\phi$  is an arbitrary monotone function, we see that the derivatives transform as follows under reparameterization:

$$\begin{pmatrix} x \\ x' \\ x'' \\ x''' \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & a & 0 & 0 & \dots \\ 0 & b & a^2 & 0 & \dots \\ 0 & c & 3ab & a^3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \\ \vdots \end{pmatrix} \quad (2.5)$$

where  $a = \phi' > 0$  (or  $a \neq 0$ ) for orientation-preserving (resp. arbitrary) reparametrizations and where  $b = \phi'', c = \phi''', \dots$  can take on arbitrary values.

## 44 Symmetry and Symmetry Groups

It is easy to check that the set of the above type of matrices is closed under matrix multiplication. In fact, they form a group, which we will refer to as the *reparametrization group*. Note that for both  $x$  and  $y$  coordinates of a single point the same reparameterization matrix applies. But at different points a different matrix will have to be considered and each such matrix will add to the list of parameters to remain invariant under.

In summary, in order to extract useful features for the recognition of plane curves, both the projective group and the reparameterization group will have to be considered and the features should be invariant under both. Hence, the question arises how the reparameterisation group affects the building blocks we have just derived. Fortunately, under reparameterisation they again change by simple factors:

Building block	Factor
$ \mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1 - \mathbf{x}_3 $	$\rightarrow 1$
$ \mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1^{(1)} $	$\rightarrow a_1$
$ \mathbf{x}_1^{(1)} \ \mathbf{x}_1^{(2)} $	$\rightarrow (a_1)^3$

The subscripts indicate values for the different points used in the expression. We want to emphasise again that, in contrast to the projective transformations, the parameters of the reparameterisation will differ for each point.

An example of an invariant under reparameterisation is

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1^{(1)}|}{\|\mathbf{x}_1^{(1)}\|},$$

which is not invariant under projectivities though. An example invariant under both projectivities and reparameterisation is

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_2^{(1)}|^3 \ |\mathbf{x}_1^{(1)} \ \mathbf{x}_1^{(2)}|}{|\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1^{(1)}|^3 \ |\mathbf{x}_2^{(1)} \ \mathbf{x}_2^{(2)}|}$$

This expression requires finding two corresponding points before and after the transformation. We suppose these points are lying on some

curves, so that we can calculate the first and second derivatives of their coordinates with respect to the curve's parameterisation. Needing two rather than five corresponding points as when no derivatives would have been used, avoids a serious combinatorial bottleneck. On the other hand, this expression is still far simpler than the simplest purely differential invariant, i.e., when one would work solely with derivatives in a single point.

It is interesting to note that, as soon as an expression is invariant under reparameterisation, its building blocks can be given the meaning of Figure 2.13 again, even when calculated on the basis of a parameterisation other than Euclidean arclength. Indeed, once such invariance has been achieved, the choice of parameter no longer matters. As a result, projective invariants can often be expressed as relations between more intuitive, Euclidean concepts. Taking the last expression as an example, it consists of the ratio of the two curvatures in the two points, times the cube of a ratio of two distances, each measured from one of the points to the tangent line in the other.

### 2.2.2.2 Fixed structures and subgroups

Suppose we are not observing just any planar pattern, but one which has some symmetry. From what we saw in the previous sections, we know that the repeated subpatterns will all have undergone the same projective transformation. Thus, if we want to discover that these subpatterns are repeated, we could try to match them on the basis of their projective invariants. Even if they look different in the image, these invariants should still be the same.

Yet, using general projective invariants is not necessarily the optimal approach. The reason is that fragments that match under projectivities, are not necessarily the result of a symmetry. Any other fragment with the same shape, would be seen as projectively equivalent. These fragments do not even have to be coplanar. More symmetry-specific invariants can be derived, however. And they may be simpler than the general projective invariants.

The existence of such symmetry-specific invariants hinges on the existence of projective subgroups to which the skewed symmetries

## 46 Symmetry and Symmetry Groups

belong. So-called fixed structures yield a direct route to finding such subgroups. As already seen earlier in this chapter, all projective transformations that keep certain structures fixed (e.g., a point) form a subgroup of the projectivities. Thus, we are interested in the fixed structures of the symmetries, as these implicitly define the corresponding subgroups. A more complete discussion on the different types of projective subgroups defined by fixed points and fixed lines is given by Van Gool [269].

Consider two planar shapes in 3D space. It is typical that a symmetry keeps certain structures fixed. For instance, a mirror symmetry maps all points of its symmetry axis onto themselves. Moreover, lines connecting symmetric points are also mapped onto themselves, in the sense that a point on such line is always mapped onto a point on the same line. For our further analysis, it is crucial that a point of the symmetric shape maps onto a point in the image, and a line to a line. Thus, if a point on the original, symmetric shape is mapped onto itself under the symmetry, so is the point under the projective transformation in the image which corresponds to this symmetry. The same holds for a line mapped onto itself under this symmetry. Fixed structures under the symmetry map to fixed structures under the corresponding projectivity.

If one wants to systematically determine the fixed structures under a transformation like a projectivity, then one needs to analyse its eigenvalue structure [245]. The fact that the same fixed structures are found under image projection, follows from the conjugation relation between the original symmetry and its image. Suppose the symmetry transformation matrix is given by  $S$ , and the projection from the original pattern plane to the image is given by  $P$ , then the transformation between the projection of the symmetric subpatterns is given by  $P S P^{-1}$ . This is a so-called conjugation, as we have already seen earlier in this chapter. It is known from algebra, that conjugation does not alter the eigenvalue structure. Eigenvalues being real, imaginary, zero, or identical, are properties that are preserved. As a consequence, the fixed structures are of the same type as those of the original symmetries, i.e., fixed points under  $s$  yield fixed points under the skewed symmetry, the same for lines, etc.

**The fixed structures of symmetries.** Fixed structures define subgroups of the projectivities. For instance, all projectivities that keep the same point fixed form such a subgroup. This can be easily verified by checking out that all properties that together define a group structure are still valid, like e.g., the fact that the composition of two projectivities that keep that point fixed will also keep the point fixed. Whereas the full group of projectivities has 8 degrees of freedom (DOF), this subgroup has only 6 DOF. In [269], Van Gool discusses a taxonomy of fixed structures and the DOF of the corresponding projective subgroups. From that discussion, it is useful to highlight two special cases. One is the case of a fixed line, for which also all the individual points are kept fixed. Such line will in the sequel be referred to as a line of fixed points. The other is a pencil of fixed lines. This entity contains all lines going through a single point and all these lines are fixed lines. Both cases are special in the sense that they only require two parameters to specify them, but each lifts 5 DOF. Subgroups that share a line of fixed points have 3 DOF, and so subgroups that share a pencil of fixed lines (under the assumption that no other fixed structures are specified).

The fact that the specification of a mere two parameters reduces the number of DOF from 8 to 3, causes the invariants for such subgroups to be quite simple when compared to those for the complete group of projectivities. These are also the very fixed structures that we see appear with several of the symmetries we are interested in. In particular, mirror symmetry has both. The symmetry axis is a line of fixed points and symmetric points are connected by fixed lines which all intersect in the same point (at infinity when the symmetry is not skewed). A point symmetry has the same fixed structures. The pencil has its center at the point about which the shape is symmetric. The line of fixed points corresponds to the line at infinity of the shape's plane. As a matter of fact, projectively a mirror and a point symmetry are equivalent.

For the sake of brevity, we will refer to a line of fixed points as *axis* and to a pencil of fixed lines as *pencil*. Transformations with both an axis and pencil as fixed structures are called planar homologies or, in case the pencil has its center on the axis, elations. All projective transformations sharing the same axis and pencil form a subgroup with

1 DOF. The remaining DOF corresponds to the cross ratio for four special points along the fixed lines of the pencil: a pair of symmetric points, the intersection of the line with the axis, and the center of the pencil. All such quadruples of points yield the same cross ratio. A mirror or point symmetry correspond to a cross ratio with value 2. This case is sometimes referred at as a harmonic homology. A transformation which yields the identity when repeated twice is sometimes also referred to as an involution. Mirror and point symmetries are also involutions.

Other than mirror and point symmetries, rotational symmetries also get special attention in this tutorial. Also when skewed, such transformations have a set of fixed structures. First, there is the center point about which the pattern is rotationally symmetric. This is a fixed point. Moreover, the line at infinity for the shape's plane is a fixed line. Having a fixed point and a fixed line as such are not special properties, however. Every projective transformation has a fixed point and a fixed line. Yet, our skewed rotational symmetry also has fixed sets of points. Indeed, each time we apply the symmetry, a point goes to a new point, until after sufficient such applications of the symmetry, we are back at the start. This entire set of points is mapped onto itself under the symmetry. We have an infinite number of such point sets, where each point belongs to one and only one such set.

**Building blocks and symmetries.** We now analyse the behaviour of the building blocks as introduced in Section 2.2.2.1, in the presence of several types of fixed structures. We also give some example invariants for the corresponding symmetry groups. However, we want to emphasize that the sets of invariants are not complete and that the discussion is not exhaustive.

**A pencil.** If there is a pencil of fixed lines, then every point is known to stay on the line of the pencil on which it lies. Denoting the pencil vertex with  $\mathbf{x}_v = (x_v, y_v)^T$ , one therefore knows that there exists a factor  $k_i$  such that for a point  $(x_i, y_i)^T$  and its image  $(x'_i, y'_i)^T$

$$\begin{aligned}(x'_i - x_v) &= k_i(x_i - x_v), \\ (y'_i - y_v) &= k_i(y_i - y_v).\end{aligned}$$

Such a factor  $k_i$  exists for every point  $\mathbf{x}_i$ . It immediately follows that

$$\frac{(y_i - y_v)}{(x_i - x_v)}$$

is an invariant, requiring only two points, one of which is the vertex.

In order to derive additional invariants (combinations with the different building blocks of Section 2.2), it is important to know more about the factor  $k_i$ . Consider

$$|\mathbf{x}'_1 - \mathbf{x}_v \ \mathbf{x}'_2 - \mathbf{x}_v| = \frac{|P|}{N_1 N_2 N_v} |\mathbf{x}_1 - \mathbf{x}_v \ \mathbf{x}_2 - \mathbf{x}_v|.$$

This can also be written as

$$|\mathbf{x}'_1 - \mathbf{x}_v \ \mathbf{x}'_2 - \mathbf{x}_v| = k_1 k_2 |\mathbf{x}_1 - \mathbf{x}_v \ \mathbf{x}_2 - \mathbf{x}_v|$$

and therefore

$$k_1 k_2 = \frac{|P|}{N_1 N_2 N_v}.$$

From the fact that this latter equality holds for any choice of the points  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , it follows that

$$k_i = \pm \sqrt{\text{abs}\left(\frac{|P|}{N_v}\right)} \frac{1}{N_i}. \quad (2.6)$$

We conclude that  $(x_i - x_v)$  and  $(y_i - y_v)$  come as additional building blocks with the pencil of fixed lines, easing the construction of invariants. An example invariant parameter is

$$\int \text{abs}\left(\frac{|\mathbf{x} - \mathbf{x}_v \ \mathbf{x}^{(1)}|}{(x - x_v)^2}\right) dt.$$

Given that this parameterisation of a curve on which the point  $\mathbf{x}$  lies is invariant under projective skewing, it is remarkably simple. Knowledge of the center of the pencil is required to exploit the parameterisation though.

**An axis.** If there is a line of fixed points — in the sequel referred to as *the axis* — then any point  $\mathbf{x}_{ai}$  on it is fixed. Hence,

$$\frac{|P|}{N N_{a1} N_{a3}} = \frac{|\mathbf{x}' - \mathbf{x}_{a1} \ \mathbf{x}' - \mathbf{x}_{a3}|}{|\mathbf{x} - \mathbf{x}_{a1} \ \mathbf{x} - \mathbf{x}_{a3}|} = \frac{|\mathbf{x}' - \mathbf{x}_{a1} \ \mathbf{x}' - \mathbf{x}_{a2}|}{|\mathbf{x} - \mathbf{x}_{a1} \ \mathbf{x} - \mathbf{x}_{a2}|} = \frac{|P|}{N N_{a1} N_{a2}}$$

and thus  $N_{a1} = N_{a2} = N_a$  where  $N_a$  is one and the same value for all the points on the axis.

It follows that, e.g.,

$$\frac{|\mathbf{x}_{a1} - \mathbf{x}_1 \ \mathbf{x}_{a1} - \mathbf{x}_2|}{|\mathbf{x}_{a2} - \mathbf{x}_1 \ \mathbf{x}_{a2} - \mathbf{x}_2|}$$

is an invariant, which requires knowledge about the axis and only two additional points, hence a total of six parameters (the two points on the axis can be chosen arbitrarily). A geometrical interpretation of this invariant is that the lines  $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$  and  $\langle \mathbf{x}'_1, \mathbf{x}'_2 \rangle$  intersect the axis in the same point. Again, we get access to an invariant which is simpler than a similar invariant under any projectivity, which would require the combination of a minimum of five points in a double ratio instead of four in a single ratio (see e.g., Equation (2.3)). The complexity of this invariant is comparable to that of simpler affine invariants. We cannot exploit it without the knowledge of the axis, however.

**A pencil and an axis.** If both a pencil of fixed lines and a line of fixed points exist — i.e., if we are dealing with a planar homology — then the previous results can be combined. If one considers  $(x_a - x_v)$  where both the point on the axis  $\mathbf{x}_a$  and the pencil vertex  $\mathbf{x}_v$  are fixed points now, this expression is a trivial invariant, i.e.,

$$k_a = \pm \sqrt{\text{abs}\left(\frac{|P|}{N_v}\right)} \frac{1}{N_a} = 1 \quad (2.7)$$

and therefore  $N_a = \pm \sqrt{\text{abs}(|P|/N_v)}$ , or, equivalently,  $N_v = |P|/N_a^2$ . For a single point, we can then find the invariant

$$\frac{(x - x_v)}{|\mathbf{x} - \mathbf{x}_v \ \mathbf{x} - \mathbf{x}_a|}.$$

As usual, this invariant can only be used if one has complete knowledge about the pencil and the axis.

**Fixed sets of points.** As already briefly discussed for the case of rotational symmetry, a set of points may, rather than being fixed individually, map onto each other. The set is fixed, not its points. Such cases are important, because they correspond to discrete symmetries. Mirror

symmetry is an example where every point belongs to a fixed pair of symmetric points. Ornamental symmetries include all cyclic and dihedral symmetry groups of different orders. Cyclic symmetry of order  $n$  is synonymous to  $n$ -fold rotational symmetry. Dihedral symmetry groups add mirror symmetries. There is an isomorphism between the skewed symmetries as observed in the image and the ‘ornamental symmetry group’ of the shape. Vice versa, the existence of fixed sets of points typically are a strong indication for the presence of skewed ornamental symmetries, and in some cases it even gives a guarantee (e.g., if there is a fixed triple [245]).

As in the case of a line of fixed points or a pencil of fixed lines, the presence of fixed discrete sets of points yields specialized invariants. And again, these are based on further constraints on the factors of the building blocks in Section 2.2.2.1.

Consider a fixed  $n$ -tuple of points,  $\mathbf{x}, \mathbf{x}', \mathbf{x}'', \dots, \mathbf{x}^{[n-1]}$ . Consider what happens to  $|\mathbf{x} - \mathbf{x}' \mathbf{x} - \mathbf{x}''|$ . Applying the transformation  $n$  times brings all the points back to their original positions. Hence, following the factor brought about by such building blocks according to Equation (2.2)

$$\frac{|P|^n}{(NN'N'', \dots, N^{[n-1]})^3} = 1$$

and therefore

$$NN'N'', \dots, N^{[n-1]} = |P|^{n/3}. \quad (2.8)$$

A degenerate case of a point cycle is the  $n$ -fold repetition of the rotation center. It follows from Equation 2.8 that for this point —  $\mathbf{x}_c$  — say —  $N_c^n = |P|^{n/3}$ , i.e.,  $N_c = \pm|P|^{1/3}$ . This holds irrespective of the angle of rotation.

As an example, if one is looking obliquely at a 3-fold rotational symmetry,

$$\frac{|\mathbf{x}_1 - \mathbf{x}_2 \mathbf{x}'_1 - \mathbf{x}_2| |\mathbf{x}_1 - \mathbf{x}_3 \mathbf{x}''_1 - \mathbf{x}_3|}{|\mathbf{x}_1 - \mathbf{x}_2 \mathbf{x}_1 - \mathbf{x}_3|}.$$

is an invariant under the transformation that corresponds to the  $120^\circ$  rotation as seen in the image. The symmetrically positioned counterparts of  $\mathbf{x}_2$  and  $\mathbf{x}_3$  yield the same values, while also cycling through

## 52 Symmetry and Symmetry Groups

$\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}''_1$  in the appropriate way, e.g.:

$$\begin{aligned} & \frac{|\mathbf{x}'_1 - \mathbf{x}'_2 \ \mathbf{x}''_1 - \mathbf{x}'_2| \ |\mathbf{x}'_1 - \mathbf{x}'_3 \ \mathbf{x}_1 - \mathbf{x}'_3|}{|\mathbf{x}'_1 - \mathbf{x}'_2 \ \mathbf{x}'_1 - \mathbf{x}'_3|} \\ &= \frac{|\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}'_1 - \mathbf{x}_2| \ |\mathbf{x}_1 - \mathbf{x}_3 \ \mathbf{x}''_1 - \mathbf{x}_3|}{|\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1 - \mathbf{x}_3|}. \end{aligned}$$

Although this invariant uses a total of five points just like a general point-based projective invariant would, it is both simpler and more selective. This expression is not invariant under general projectivities. Note that — as usual — this symmetry-specific invariant contains information on the fixed structures of the symmetry, i.e., the fixed triple  $\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}''_1$ .

The following expression is invariant under all skewed rotations about the point  $\mathbf{x}_c$ :

$$\frac{|\mathbf{x}_1 - \mathbf{x}_c \ \mathbf{x}_1^{(1)}| \ |\mathbf{x}_2 - \mathbf{x}_c \ \mathbf{x}_2^{(1)}| \ |\mathbf{x}_3 - \mathbf{x}_c \ \mathbf{x}_3^{(1)}|}{|\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1^{(1)}| \ |\mathbf{x}_2 - \mathbf{x}_3 \ \mathbf{x}_2^{(1)}| \ |\mathbf{x}_3 - \mathbf{x}_1 \ \mathbf{x}_3^{(1)}|} |\mathbf{x}_1 - \mathbf{x}_2 \ \mathbf{x}_1 - \mathbf{x}_3|.$$

In order to use this expression, one has to be able to extract a direction vector at each point, e.g., the tangent vector if the points  $\mathbf{x}_1, \mathbf{x}_2$ , and  $\mathbf{x}_3$  lie on curves.

**Skewed mirror symmetries.** Skewed mirror and point symmetries deserve some special attention, as they have an axis, pencil and fixed pairs of points. They thus contain all the types of fixed structures that we have considered so far.

Consider a pair of symmetric points  $\mathbf{x}$  and  $\mathbf{x}'$ . As we have already seen, the case of projectively skewed mirror or point symmetry are equivalent. Hence, for such pair of points we have

$$NN' = |P|^{2/3}. \quad (2.9)$$

since a point symmetry is nothing but a rotational symmetry of order 2 and therefore  $n = 2$  in Equation (2.8). It also follows that for fixed points, which are their own symmetric point,  $N = N'$  and therefore

$$N = |P|^{1/3}. \quad (2.10)$$

In particular, this is the case for all points on the axis as well as for the vertex of the pencil, i.e.,  $N_v = N_a = |P|^{1/3}$ . It then also follows from

Equation (2.6) that

$$(x' - x_v) = \frac{N_v}{N}(x - x_v) \quad \text{and} \quad (y' - y_v) = \frac{N_v}{N}(y - y_v),$$

with  $(x - x_v)$  and  $(y - y_v)$  building blocks at our disposal.

In the case of *affinely* skewed symmetry, Ponce [216] has shown that in a pair of symmetric points

$$\frac{\kappa}{\kappa'} = \frac{\sin^3 \theta}{\sin^3 \theta'}$$

where  $\kappa$  and  $\kappa'$  are the curvatures of the symmetric contours in these points and the meaning of  $\theta$  and  $\theta'$  is explained in Figure 2.14.

A similar relation has been used by Cham and Cipolla [31] for the detection of affinely skewed symmetries. Based on our prior discussion, it can be shown that this relation also holds under *projective* skewing. In order to prove this, we can rewrite Ponce's relation in the equivalent form

$$\frac{|\mathbf{x}^{(1)} \mathbf{x}^{(2)}|}{|\mathbf{x} - \mathbf{x}' \mathbf{x}^{(1)}|^3} = \frac{|\mathbf{x}'^{(1)} \mathbf{x}'^{(2)}|}{|\mathbf{x}' - \mathbf{x} \mathbf{x}'^{(1)}|^3}$$

This equality follows from the behaviour of the two types of building blocks in these expressions, as specified in Section 2.2.2.1, when combined with the mirror/point symmetry (involution) specific rule (2.9). The expression on the left-hand side is an invariant under the symmetry, since it has the same value upon replacement of  $\mathbf{x}$  by the symmetric

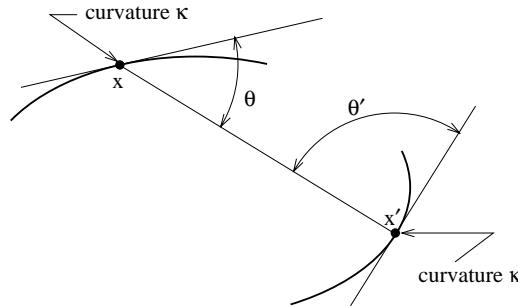


Fig. 2.14 Variables used in Ponce's symmetry invariant.

## 54 Symmetry and Symmetry Groups

point  $\mathbf{x}'$  and vice versa. This relation holds for a single pair of symmetric points. To check the symmetry of two contour segments, one can start from a single (hypothesised) correspondence  $\mathbf{x}_1, \mathbf{x}'_1$ , and find additional pairs of symmetric points by using the symmetry-specific invariant parameter

$$\int \text{abs} \left( \frac{|\mathbf{x}^{(1)} \mathbf{x}^{(2)}|^{1/3}}{|\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1 - \mathbf{x}|} \right) dt,$$

which is the affine arclength with a local correction factor  $|\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1 - \mathbf{x}|^{-1}$ . Corresponding points on symmetric contours will have the same value if this parameterisation is used on each, starting from the point  $\mathbf{x}_1$  and  $\mathbf{x}'_1$ , respectively, and upon swapping their roles in the expression. For each new point correspondence, the Ponce relation can then be checked. A further example of an invariant for this case is

$$\frac{|\mathbf{x} - \mathbf{x}_1 \mathbf{x}_1^{(1)}|^2 |\mathbf{x} - \mathbf{x}'_1 \mathbf{x}^{(1)}|}{|\mathbf{x} - \mathbf{x}_1 \mathbf{x} - \mathbf{x}'_1|^2 |\mathbf{x} - \mathbf{x}_1 \mathbf{x}^{(1)}| |\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1^{(1)}|^2}.$$

Observe that the limiting behaviour of this invariant when  $\mathbf{x}$  approaches the reference point  $\mathbf{x}_1$  is

$$\frac{|\mathbf{x}_1^{(1)} \mathbf{x}_1^{(2)}|}{|\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1^{(1)}|^3}.$$

In case the contour under scrutiny contains corner points (i.e., points at which the contour is not differentiable), then it is natural to take such a corner point as a reference point, because it can more easily be identified. For such points, however, the above invariants cannot be used. But a corner point being an isolated point at which the contour is not differentiable, one can compute its left and right derivatives (i.e., replacing the two-sided limit in the definition of the tangent vector by a left and a right limit, respectively). The following expression is a (full) projective invariant:

$$\frac{|\mathbf{x} - \mathbf{x}'_1 \mathbf{x}_1'^{(1:\ell)}| |\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1'^{(1:r)}|}{|\mathbf{x} - \mathbf{x}'_1 \mathbf{x}_1'^{(1:r)}| |\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1'^{(1:\ell)}|}. \quad (2.11)$$

A second, symmetry-specific invariant involving left and right derivatives in the point  $\mathbf{x}_1$  is given by

$$\frac{|\mathbf{x} - \mathbf{x}_1 \mathbf{x} - \mathbf{x}'_1|^2 |\mathbf{x} - \mathbf{x}_1 \mathbf{x}^{(1:ell)}| |\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1^{(1:\ell)}| |\mathbf{x}_1 - \mathbf{x}'_1 \mathbf{x}_1^{(1:r)}|}{|\mathbf{x} - \mathbf{x}'_1 \mathbf{x}^{(1:r)}| |\mathbf{x} - \mathbf{x}_1 \mathbf{x}_1^{(1:\ell)}|^2}. \quad (2.12)$$

Additional examples and a more elaborate discussion can be found in [270].

**A note on the affine case.** The case where affine rather than projective transformations suffice to describe the effects of image projection, the situation gets a lot simpler. This corresponds to the case where orthographic projection combined with a scaling model the situation, i.e., when perspective effects like parallel lines converging after projection do not happen. The same and additional building blocks can be used, but they change with factors that are less complicated than for the projective case. An affine transformation takes the general form

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

The three projection building blocks now all change with the simple factor  $|A|$ , which is the determinant of the linear part of the transformation. Indeed, for an affine transformation, which is a special projectivity, we have  $|P| = |A|$  and  $N = 1$  for all points. From Equation `refeq-pencil+axis` and Equation (2.8) it follows that for affinely skewed mirror and rotational symmetries one has  $(\text{abs})(|A|) = 1$ . This means that those symmetries correspond to so-called *equi-affine* transformations, i.e., they preserve areas also after image projection. Thus, all areas are invariant under these skewed symmetries.

In order to test for affinely skewed mirror symmetry, one has sometimes tested for two conditions that must hold between pairs of symmetric points in the image. On the one hand, the lines through such pairs are all parallel (the pencil has a vertex at infinity in this case). On the other hand, the midpoints between corresponding points are all collinear. It can be shown [271] that these pair of conditions - the parallelism and collinearity constraints - which are both sufficient and

necessary, can be replaced by the equivalent pair that the transformation between the symmetric parts is both equiaffine and an involution. In practice, the latter pair is sometimes easier and more efficient to check, based on invariant representations of the symmetric parts, e.g., when there are contour segments that one can describe in such way.

### 2.3 Symmetry in non-Euclidean Geometry

In the nineteenth century, several mathematicians realized independently that the fifth postulate of Euclidean geometry — the “parallel postulate” — was independent of the other four. Indeed, the negation of the parallel postulate could be combined with the other axioms of geometry to yield a self-consistent non-Euclidean (hyperbolic) geometry in which a line may have multiple distinct parallels through a point. This realization marked one of the greatest revolutions in the history of science [79].

It is impossible to draw a hyperbolic plane without distortion, but there are several Euclidean models of hyperbolic geometry that we can choose for visualization purposes. These models may be thought of as projections that distort some aspects of hyperbolic geometry and preserve others. Each has advantages and disadvantages for computer graphics applications [94].

The *Beltrami–Klein* model maps the entire hyperbolic plane into the interior of a circular disc. It is *projective* — that is, straight lines in the hyperbolic plane are represented by straight chords of the disc, including diameters. It also has the remarkable property that isometries of the hyperbolic plane become perspectivities of the Euclidean plane under projection [86, 87], making the rendering of images in the Beltrami–Klein model compatible with a standard graphics hardware pipeline. The disadvantage of this model is that it is not very well suited to making drawings of hyperbolic patterns, because it tends to compress most of the information in a drawing towards the boundary of the disc.

The *Poincaré disc* model also fits the plane into a disc, but *conformally* — it preserves angles. Straight hyperbolic lines are represented by arcs of circles that meet the bounding disc at right angles.

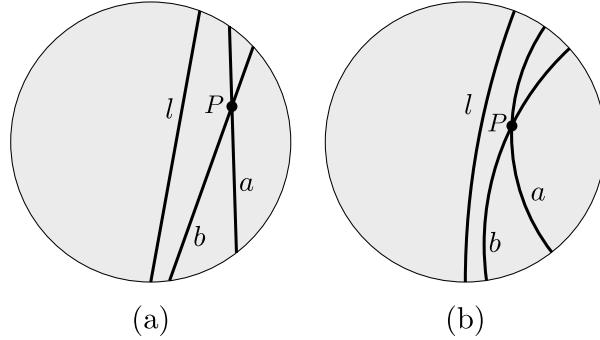


Fig. 2.15 Visualizations of identical lines in the Beltrami–Klein (a) and Poincaré disc (b) models of the hyperbolic plane. The diagrams also demonstrate the impossibility of satisfying the Euclidean parallel postulate in the hyperbolic plane: the line  $l$  is parallel to lines  $a$  and  $b$ , both of which pass through the point  $P$ .

Drawings in this model are more legible than in the Beltrami–Klein model. For example, Escher’s “Circle Limit” prints are all made using the Poincaré disc (Escher developed an intuitive understanding of hyperbolic geometry after seeing diagrams in a book by Coxeter [238]).

Figure 2.15 illustrates a simple configuration of lines in the Beltrami–Klein and Poincaré disc models of the hyperbolic plane.

In the *hyperboloid model* (sometimes also known as the *Minkowski model*), the points of the hyperbolic plane are taken from one sheet of the hyperboloid given by the equation  $x^2 + y^2 - z^2 = -1$ . Lines are the intersections of the hyperboloid with planes that pass through the origin of 3D space. The hyperboloid model is a useful internal model for computation on the hyperbolic plane. Isometries can be represented by matrices, and therefore composed via matrix multiplication. Constructions in the hyperboloid model can then be projected down to either of the two disc models.

The Euclidean wallpaper groups do not map directly into hyperbolic geometry; most obviously, the hyperbolic plane does not have an affine structure, meaning that even a simple Euclidean group like  $p1$  has no hyperbolic equivalent. But there are simple infinite families of discrete hyperbolic symmetry groups that are useful for creating appealing imagery. Dunham et al. observe that for every pair of positive integers  $p$  and  $q$  satisfying  $(p - 2)(q - 2) > 4$ , there exists a regular tiling  $\{p, q\}$

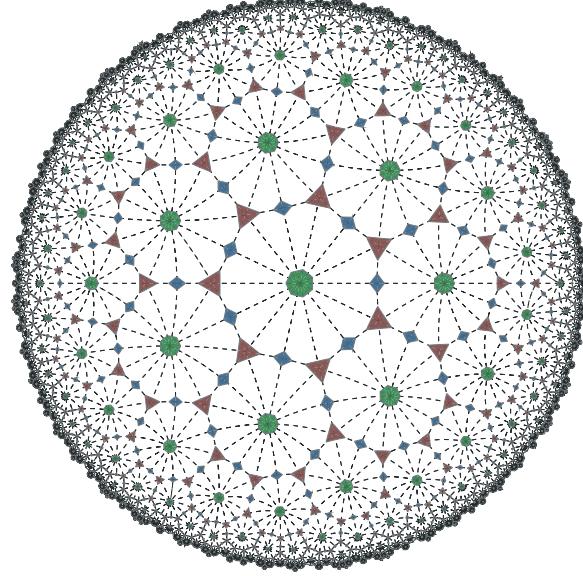


Fig. 2.16 A visualization of the hyperbolic symmetry group  $[7,3]$ , the group of symmetries of the tiling of the hyperbolic plane by regular heptagons meeting three at a vertex. The green, red, and blue shapes denote centres of sevenfold, threefold, and twofold rotations respectively. The dashed arcs (which are straight lines in the hyperbolic plane) denote lines of reflection.

of the hyperbolic plane made from regular  $p$ -gons meeting  $q$  around every vertex [56]. This tiling has a symmetry group  $[p,q]$ , generated by reflections in the sides of a right angle triangle with interior angles  $\pi/p$  and  $\pi/q$ . For example, the group  $[7,3]$  is visualized in Figure 2.16. These regular tilings can be seen as the generalization of the three tilings of the plane by regular squares, triangles and hexagons, as well as the five tilings of the sphere that correspond to the Platonic solids. The groups of the form  $[p,q]$ , together with their orientation-preserving subgroups, form a practical set for experimenting with hyperbolic imagery in computer graphics. A more general set of discrete groups can be found by taking those groups generated by reflections in a triangle with interior angles  $\frac{\pi}{k}$ ,  $\frac{\pi}{m}$ , and  $\frac{\pi}{n}$ , for any  $k$ ,  $m$ , and  $n$  with  $\frac{1}{k} + \frac{1}{m} + \frac{1}{n} < 1$  [273].

It is tempting also to include the sphere as another non-Euclidean plane, one in which parallels do not exist. This step would be too hasty, however: given a line and a point not on it, the first four axioms of

Euclid guarantee the existence of at least one parallel through the point. Fortunately, the foundations of geometry can be reformulated in such a way that we can consider the Euclidean plane, the hyperbolic plane, and the sphere side-by-side. One approach is to use Birkhoff's ruler-and-protractor postulates [121].

In practice, the spherical case is straightforward compared to the hyperbolic. The isometries of the sphere are precisely the elements of the Orthogonal Group  $O(3)$ . Its discrete symmetry groups are those of the platonic solids, prisms and antiprisms (Table 2.2).

We will revisit both hyperbolic and spherical symmetry, and their uses in computer graphics, in Section 6.4. For an excellent general introduction to the history and mathematical foundations of non-Euclidean geometry, see the book by Greenberg [79]. Dunham has written extensively on the algorithmic generation of symmetric drawings in the hyperbolic plane; several of his papers include the necessary matrices, formulas, and pseudocode [57, 55]. Levy [142] offers a more sophisticated, table-driven replication algorithm based on the theory of automatic groups.

# 3

---

## Symmetry Detection

---

Symmetry is an important cue for humans, animals, insects as well as machine perception [144, 268, 74, 226] of the world. Automatic symmetry detection from digital images/patterns has been a standing topic in computer vision and, more recently, computer graphics. The development of symmetry detection algorithms has a long history: the earliest attempt at detection of reflection symmetry (1932) even predates computer vision itself [18]. In spite of years of effort, computationally we are still short of a robust, widely applicable general “symmetry detector” that can parallel other types of computer vision/image processing tools for the more primitive structural features, such as an “edge” or “corner” detector. As shown in Figure 1.4, reflection symmetry detection used to dominate the field of symmetry detection in computer vision. Recently, we observe a surge of new symmetry detection algorithms from unsegmented real images that go beyond bilateral reflection symmetry detection. Here, we attempt to put forward a summary of the state of the art work on symmetry detection, organized by:

- (1) the types of methods applied: from correlation-based, local feature-based, transformation-driven, graphic models

and energy-based methods, ... to, more recently, spectral methods, MRF-based and other statistical learning-based approaches.

- (2) the types of primitive symmetries and symmetry groups detected: from primitive symmetries (reflection, rotation, translation and glide-reflection symmetries) to symmetry groups (cyclic, dihedral, crystallographic groups, ...).

In particular, we recap the results from a recent (and the first) systematic assessment of a set of state of the arts rotation, reflection and translation symmetry detection algorithms published within the past couple of years. The symmetry detection results are quantitatively evaluated using a set of carefully chosen synthetic and real images (publicly available), with labeled groundtruth, that contain both single and multiple symmetries within one image. This publicly available test-image database is the first standard data set for gauging the progress in this important and widely applicable research direction. These quantified results indicate that even after several decades of effort, symmetry detection from digital data (2D images or otherwise) remains a challenging and largely unsolved problem.

### 3.1 Symmetry Detection by Methods

Given its relative simplicity, the detection of bilateral reflection symmetry (mirror-symmetry) and its affinely and perspectively skewed version from processed images has been the dominant focus in computer vision for about 40 years. Some of the representative work includes [24, 30, 48, 71, 81, 109, 135, 194, 197, 216, 224, 270, 303]. The detection of rotation and translation symmetries and their skewed versions have also been explored in the literature, while the *glide reflection* symmetry seems to be completely ignored until recently [133, 159].

The basic question of symmetry detection is discussed in early papers of geometry and theoretical mathematics. These early studies deal with evaluating deviation from symmetry of sets in Euclidean space and are based on cord length, surface area (contour length) and internal volume (internal area). Examples include Minkowski, Winternitz and Kovner–Besicovitch measures of symmetry. These studies

approach symmetry evaluation from the theoretical point of view and do not suggest methods to efficiently evaluate these measures. A review of geometrical evaluation of symmetry of convex sets is given in [82].

A different approach to symmetry definition can be found in a collection of papers in which a computational approach was taken [5, 8, 58, 99, 279]. These studies present algorithms for detecting symmetry in collections of geometrical objects such as points, line segments, circles, etc. The basic idea in these algorithms is to reformulate the problem as a 1D pattern matching problem which can be solved efficiently (e.g., using known techniques such as KMP [126]). The complexity of these algorithms are shown to be  $O(n \log n)$  where  $n$  is the number of geometric objects. These algorithms are simple and efficient, however they are highly sensitive to noise. In fact, slight perturbations of the elements location, or slight computer precision errors, will cause the algorithm to fail in finding symmetry. Extension to higher dimensions [5], namely points in  $R^d$ , is implemented using a simplified Extended Gaussian Images. This method is also efficient ( $O(n \log n)$ ), however, as in the 2D case, it is not robust to noise, perturbation of points, nor to numerical imprecision. The matter of imprecision and approximate symmetry in these cases is described in [5] in terms of time complexity.

### 3.1.1 Symmetry in 2D images — Direct Approach

Detection of 2D symmetry in digital images has been widely studied and numerous approaches suggested. The most basic method, often referred to as the *Direct Approach*, for determining if a given image is mirror or rotationally symmetric is to apply the symmetry transformation (i.e., reflection or rotation) to the image and then compare with the original image. Such an approach is assumed in [272] where an optical-mechanical system is described to optically determine 2D symmetry in images. In [130], comparison of an image and its reflection is used for detection of vehicles. A similar approach is taken also in [29, 129]. In [29] this approach is combined with a recursive strategy using a multiresolution representation of an image — specifically, an image pyramid. This method recursively tests for perfect symmetry,

initiating the process at low resolution where time complexity is low, and continuing to higher resolution images. In [129], the overlap method is used to determine rotational symmetry by performing autocorrelation of the image in polar coordinates. These Direct Approach methods assume that the evaluated object is either perfectly symmetric or it is not at all, consequently, they are highly sensitive to noise and occlusion and cannot serve as satisfactory measures of symmetry.

Although in [272] one may suggest that the area of overlap between an object and its reflection can be used as a measure of symmetry, this would not always support our intuition of the symmetry measure value. For example, the images in Figure 3.1 are intuitively evaluated as highly symmetric (almost symmetric), however, the suggested overlap symmetry measure would give very low values.

### 3.1.2 Voting Schemes

A different approach to symmetry detection uses a voting scheme. The voting scheme is based on the fact that the symmetry axis is uniquely determined by two points in the object and, similarly, the center of rotational symmetry is determined by two pairs of points. In the voting scheme pairs of points are tested and a vote for their preferred symmetry axis is recorded. The oriented line with highest vote is selected as the symmetry axis of the object. In [206], a digitized polygon is represented by its chain code imposing a discrete number of possible mirror-symmetry axes which are voted on. In [141, 198] the Hough transform is used for the voting scheme. For every pair of points in the

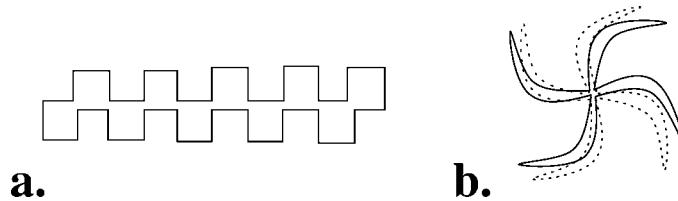


Fig. 3.1 Two images which give low symmetry content values using overlap techniques, though they are intuitively “almost” symmetric. (a) Folding across the mid horizontal axis, the region of overlap is minimal. (b) Rotating about the center by  $\pi/2$  (obtaining the shape with the dotted outline), the overlap is minimal.

image, the midpoint and the direction perpendicular to the segment connecting the pair are determined and voted for in the Hough space. The line with maximal votes is selected as the mirror symmetry axis. A variant of this method is used in [291] for detecting rotational symmetry. Another variant of the Hough transform voting scheme is the projection scheme [197, 217, 218]. The voting scheme can also be found in [307] where a feed-forward network is used to detect and enhance edges that are symmetric in terms of edge orientation.

The voting schemes are robust, to a certain degree, under noise and occlusion in the input image, however, they have high complexity. Several methods have been suggested to reduce complexity by grouping points into regions or into curve-segments, thus reducing the number of possible pairs involved in the voting [75, 229].

The voting schemes usually assume the existence of symmetry axes as well as the knowledge of the number of such axes (although thresholding heuristics could provide these variables, the process is image and noise dependent and generally unstable). These studies generally approach symmetry as a binary feature, where thresholding is performed to overcome noise in the input. Although a measure of certainty can be associated with a voted symmetry axis, the discrete nature of voting in bins provides only approximate symmetry detection and an approximate measure of symmetry, not to mention imprecision in symmetry axis location and orientation.

### 3.1.3 Global vs. Local Symmetry

Symmetry can be discussed as a global feature where all object points contribute to determining the symmetry, or as a local feature where every symmetry element is supported locally by some subset of the object (Figure 3.2).

The global symmetry methods are much more efficient in run time, usually having a linear time complexity, however they are generally sensitive to noise and occlusion. The local symmetry methods are more robust to noise and occlusion, and they are easily parallelized, however they have high time complexity.

In the case of global symmetry, the image or shape is assumed to be symmetric on a global scale, with symmetry axis supported by all object

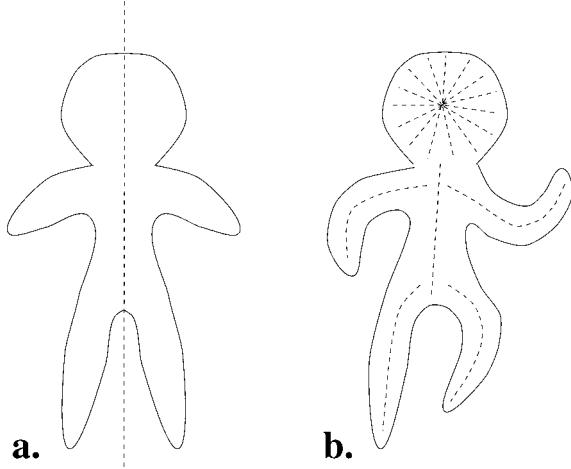


Fig. 3.2 Global vs. local symmetry. (a) Global symmetry — the mirror-symmetry axis is supported by all points of the shape. (b) Local symmetry — the symmetry axes (curves) are each supported locally by a subset of the shape.

points. The voting schemes described above fall into this category. Two other prominent global approaches are methods based on basis functions and methods based on moments.

**Basis Function Methods.** In [287], the Walsh functions are used as basis functions for evaluating mirror-symmetry (horizontal, vertical or both) and rotational-symmetry (of order 2). The 2D basis functions (Figure 3.3a) denoted  $W_{n,m}$  (with integer  $n, m$  values) are equally divided into four sets according to the type of symmetry they represent: vertical mirror ( $m$ -even,  $n$ -odd), horizontal mirror ( $m$ -odd,  $n$ -even), doubly mirror ( $m$ -even,  $n$ -even) and rotational-symmetry ( $m$ -odd,  $n$ -odd). Summing the Walsh coefficients according to this classification, a vector of four values is obtained representing the symmetries of an image. An overall evaluation of symmetry is obtained for the image by taking the entropy of these four values (accordingly, this single value is termed “symmetropy”).

In [16, 17, 15], Radial Basis Functions are used to detect rotationally symmetric images. These functions are spiral-like with varying number of “arms” and variable curvature (Figure 3.3b). Rotational symmetry of an image is evaluated by computing the weighted averages of its radial basis transform. Similar to the Fourier Basis functions, computations

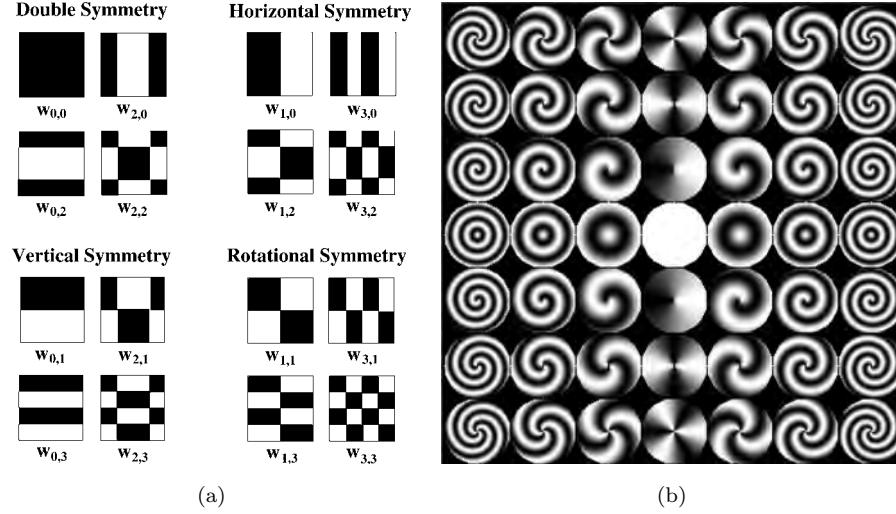


Fig. 3.3 Basis functions for global symmetry detection. (a) The Walsh basis functions, divided into four classes according to symmetry, are used to detect horizontal, vertical and double mirror-symmetry and to detect rotational symmetry (of order 2). (b) The radial basis functions are used to detect rotational and circular symmetry in images.

in the radial frequency domain can be exploited, reducing time complexity to  $O(n \log n)$ . Similarly, the power spectrum of the transform is invariant to rotation. In addition to the symmetry evaluation, an associated uncertainty value is computed. Low values of uncertainty reflect high symmetry content.

**Moments Based Methods.** Another global symmetry approach is that based on moments. In [103], moment invariants are developed for pattern recognition. As a specific case, an invariant was developed to distinguish between “mirror-images”. Additionally, constraints on possible invariants for circular and rotational symmetries are discussed. These are based on the non-uniqueness of the principle axes.

In [183], a method is presented for finding the  $n$ -mirror axes of a 2D shape which is rotationally symmetric of order  $n$  and having  $n$  mirror-axes. The method is based on the fact that all mirror-axes of a shape pass through the centroid of the shape. In [35, 261] a method is suggested for overcoming the problem of non-uniqueness of the principle axes for rotationally symmetric shapes. In [261] a method was suggested

for generalizing the principle axes to rotationally symmetric shapes. The rotationally symmetric shape is transformed into a shape which generally has no rotational symmetry and thus the principle axes can be computed. The computed principle axes are then transformed back as generalized principle axes of the original rotational symmetric shape.

The general use of moments either as invariants for shape description and recognition or as determinants for shape localization and orientation, are very sensitive to noise and occlusion, thus they are usually impractical for use in digital images. The method in [183] as presented in its theoretical formulation, must assume perfect  $n$ -fold symmetry. However, the method is extended to deal efficiently with imperfectly symmetric images, thus becoming more robust to noise and occlusion. Additionally, this method finds the number of reflection axes of the image. The method in [261] is a priori defined for all shapes including imperfectly symmetric shapes. Whereas the method in [183] might be defined as a measure of symmetry, the study in [261] deals with finding the principle axes and does not extend as a measure. Additionally, the methods in [35, 261] assume a priori knowledge of the order  $n$  of the rotational symmetry.

### 3.1.4 Local Symmetry

In the case of local symmetry, only part of a shape or a subset of its points is symmetric with respect to any given symmetry (rotational symmetry or mirror-symmetry). The subset supporting a given symmetry is typically forms a continuous section of the shape's contour or a continuous neighborhood around an image point.

It should be noted that global symmetry approaches such as those described above, can be implemented for detecting local symmetry by segmenting the shape or image into parts or regions and applying the global symmetry methods to each region independently. However, in contrast with the global symmetry approaches, the methods described here as local symmetry methods, are inherently local by definition.

#### Symmetry Axis Transform — SAT.

The *Symmetry Axis Transform* — SAT introduced in [21] (also known as the Medial Axis Transform — MAT), is defined as the loci

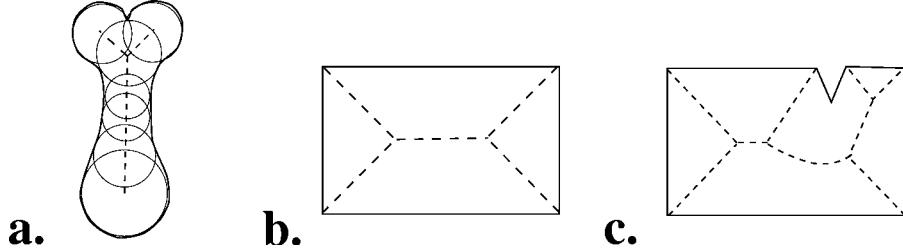


Fig. 3.4 Symmetry Axis Transform (SAT). (a-c) The SAT is the loci of all maximal disks enclosed in the shape (dashed lines). The SAT description of a shape is very sensitive to noise (c).

of the centers of all maximal disks contained within the shape (a 2D example is shown in Figure 3.4a). The SAT is shown to be piecewise smooth [23] and forms a graph-like structure. Associating every point of the SAT with the radius of the corresponding maximal disk, allows perfect reconstruction of the shape from its SAT. A weakness of the SAT description is that it is very sensitive to noise. Thus, small perturbations of the bounding curve of the shape, induce extreme effects on the SAT (Figure 3.4c). The SAT reflects various properties of the shape; parts of the SAT structure reflects the general shape and orientation of the shape and can be viewed as the major axis or spine of the shape (for example the SAT's horizontal axis in Figure 3.4b). Other parts of the SAT reflect local boundary formations (e.g., the oblique segments of the SAT in Figure 3.4b are associated with the corners of the shape).

In [214], the sensitivity to noise and the variance in SAT saliency are dealt with by extending the SAT to a hierarchical description. Smoothing a shape simplifies its SAT graph-structure and eliminates non-salient limbs. Thus an hierarchy of saliency of the SAT limbs can be obtained. A further extension in [70] builds a hierarchical SAT description of gray-scale images by thresholding the image at successive levels and applying standard 2D SATs to each of the obtaining binary images, independently. These 2D SATs are combined to create a 3D SAT of the intensity image. In [196] the SAT is extended to proper 3D where the SAT description is a 3D graph structure representing the loci of all maximal balls bounded in the 3D object. In digital images, the SAT has been extended to deal with graylevel images in [139, 208].

Additional variations of the SAT extend the *front propagation* interpretation of the SAT so as to regularize the original definition and allow robustness to noise. In [241] propagation is combined with diffusion to find the medial axis/. In [248] *shock grammar* is introduced in which a wave propagation scheme continuously smooths the shape boundary from which the medial axis can be easily determined. A similar approach based on image edge smoothing was introduced in [255].

**Generalized Ribbons — GR.** In [26], a class of shapes called *Generalized Ribbons* — GR is described. These shapes are generated from a 2D curve serving as a spine and from a generator segment which follows the spine trajectory at a constant angle (with possible length change). The area swept out by the segment is a generalized ribbon (Figure 3.5). The spine of the GR can be taken as a description of the ribbon shape. This definition of an axial shape description is more flexible than the SAT description [21], but is not uniquely defined (see Figure 3.5a, b). Note that the spine of a GR need not be straight and may be curved. Thus, the spine represents local symmetry of the shape (Figure 3.5c). The GR are 2D versions of the generalized cylinders (cones) [197] which are generated by moving a planar surface along a 3D curve (spine) at a constant angle to the spine and allowing changes of size in the planar shape (see also [230]).

**Smoothed Local Symmetries — SLS** An extension to the GR are the *Smoothed Local Symmetries* — SLS [24]. The GR of 2D shapes involves two steps: the determination of local symmetry and the formation of maximal smooth loci of these local symmetries. A local symmetry exists between points A and B on the contour of a shape if the

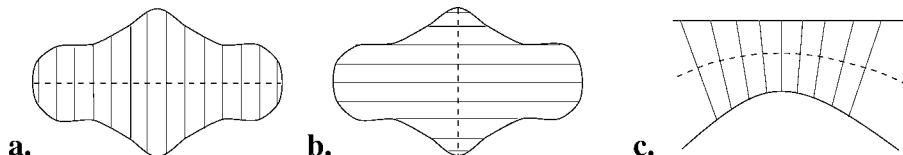


Fig. 3.5 Generalized Ribbons (GR). The GR is defined as the shape swept out by a segment moving along a 2D curve or spine. The 2D curve is the GR representation of a shape. (a, b) The GR is not uniquely defined. (c) The GR may be curved thus representing local symmetry of the shape.

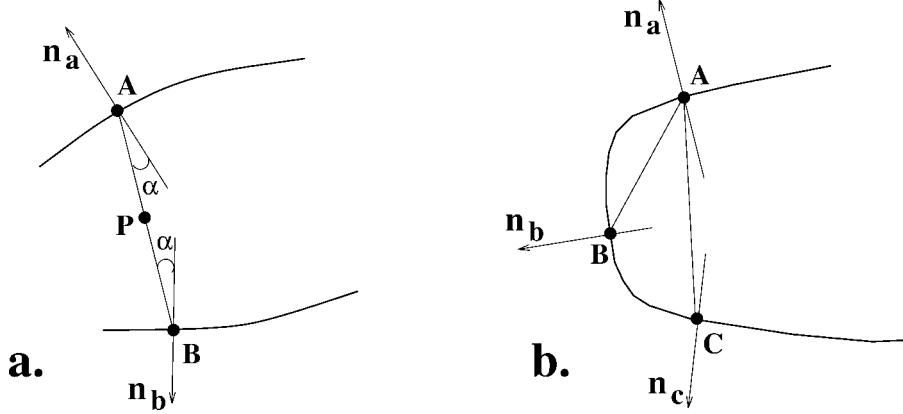


Fig. 3.6 Smoothed Local Symmetries (SLS). (a) The SLS is defined between two points A and B on a bounding contour of a shape if the two angles between the segment  $\bar{AB}$  and the two normals to the curves  $n_A, n_B$  at points A and B, are equal. The midpoint P of a segment connecting a pair of locally symmetric points is a symmetry locus. (b) In general a point (e.g., A) may be locally symmetric with several other points.

two angles between the segment  $\bar{AB}$  and the two normals to the curves at points A and B are equal (see Figure 3.6a). In general, a point may be locally symmetric with several other points (Figure 3.6b). The midpoint of a segment connecting a pair of locally symmetric points is a symmetry locus. These loci are connected into smooth curves (spines) creating the SLS representation of the shape (Figure 3.7a, b). This definition of SLS gives rise to multiple limbs of the SLS in shapes that have deep concavities or protrusions (Figure 3.7c).

In [34], an extension to SLS is presented, namely *Hierarchical Local Symmetry* — HLS, which eliminate these redundancies in the SLS. In [67] the SLS is extended to circular symmetries and local rotation symmetries (LRS) were presented.

**Additional Axial Representations.** Additional discussion on axial representations can be found in [217] where skew symmetry is shown to be a special case of GR with a straight spine. Also, in [143], Process Inferring Symmetry Analysis (PISA) was introduced. PISA points have a 1-to-1 correspondence with the SAT points but are differently located so that they are more adequate for inferring the process of shape formation.

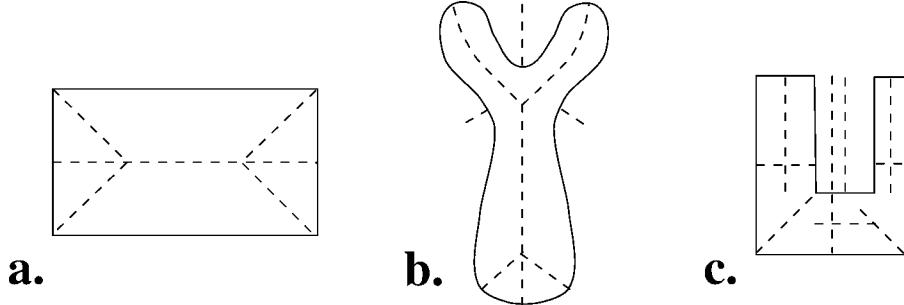


Fig. 3.7 Smoothed Local Symmetries (SLS). (a, b) The midpoint of segments connecting pairs of locally symmetric points are connected into smooth curves (spines) creating the SLS representation of a shape (dashed lines). (c) The SLS gives rise to multiple limbs of the SLS in shapes that have deep concavities or protrusions.

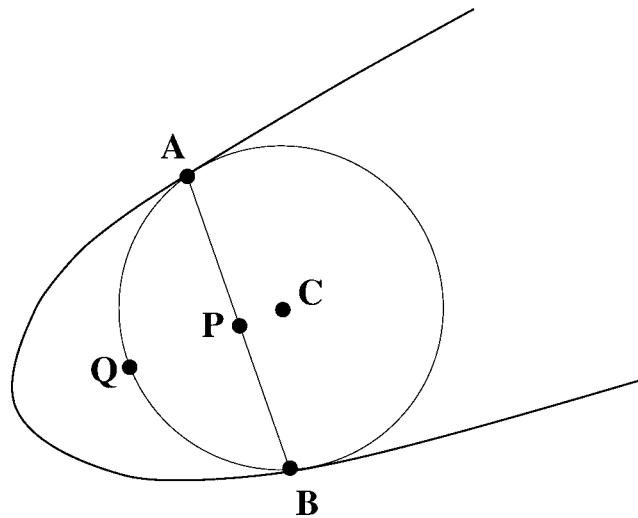


Fig. 3.8 The geometry of different local symmetry points. C, SAT; P, SLS, HLS, GR; Q, PISA.

In [227], the above described axial descriptions of 2D shapes are discussed and compared. It is shown that for the special case where the spines are straight and constant angles of the GR are set to  $90^\circ$ , the following relationship holds:  $SAT \subset GR \subset SLS$ . However, in [217], it is shown that in the general case the above relationship does not hold. In [34] the geometry of different symmetry points is discussed (see Figure 3.8).

**Local Symmetry in Images.** The local symmetries described above apply to shapes defined by bounding contours and the locality is in terms of subparts of the contour. For grey level images, locality is in terms of local regions or neighborhoods.

In [222], the local symmetry approach is followed in order to find points of local mirror and circular symmetry. These points are defined as points of interest in the digital image. The local symmetries are found using a symmetry operator which is applied to every point in the image and with respect to a local predefined neighborhood. For every point  $P_k$  in the image, a gradient vector  $V_k = (r_k, \theta_k)$  is defined where  $r_k$  denotes the gradient intensity and  $\theta_k$  denotes the gradient direction. For every two points  $P_i$  and  $P_j$ ,  $l$  denotes the line passing through them and  $\alpha_{ij}$  denotes the angle between  $l$  and the  $x$ -axis (see Figure 3.9a). For a given direction  $\psi$ , a symmetry measure  $S(P, \psi)$  is defined as follows:

$$S(P, \psi) = \sum_{P_i, P_j \in \Gamma(P)} \frac{r_i r_j (1 - \cos(\theta_i + \theta_j - 2\alpha_{ij})) (1 - \cos(\theta_i - \theta_j))}{\|P_i - P_j\|}$$

where  $\Gamma(P)$  is a neighborhood of  $P$ . Points  $P_i$  and  $P_j$  contribute maximally when there is good correlation between the two gradients ( $r_i, r_j$ ) and when  $(\theta_i - \alpha_{ij}) + (\theta_j - \alpha_{ij})$  is close to  $\pi$  (excluding the case where  $\theta_i - \alpha_{ij} = \theta_j - \alpha_{ij} = \pi/2$  which occurs when both  $P_i$  and  $P_j$  are on the same straight edge). Accumulating the values  $S(P, \psi)$  in bins according to the direction  $\psi$ , allows detection of various symmetries. Thus, considering the two bins in Figure 3.9b, horizontal symmetry is detected when  $S(P, \psi)$  is summed over  $\psi$  values falling in bin 1 and vertical

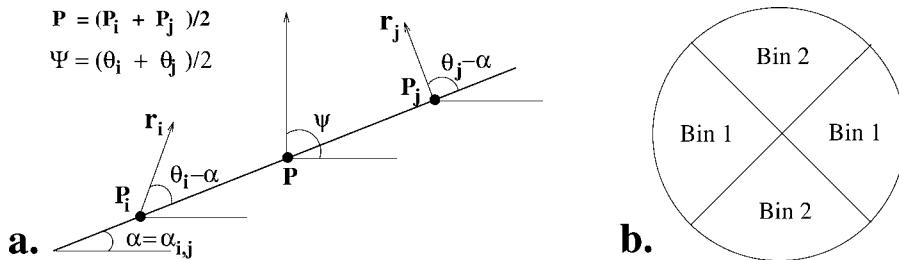


Fig. 3.9 Local symmetries in digital images. (a) A symmetry operator is applied to every point in the image with respect to a local neighborhood of the point (see text). (b) Dividing the angular orientations into bins, different symmetries can be detected.

symmetry in bin 2. Circular symmetry is detected by summing over all  $\psi$  values.

This method is easily parallelized and is applied directly on gradients of the original image with no need of edge detection or segmentation. This method has been applied to detection of facial features for normalization and recognition [223].

In [242], local mirror symmetry is found in images by autocorrelating a local region in the image with its reflection across a mirror-axis at angle  $\theta$ . A 1D function of  $\theta$  is obtained describing the symmetry content of the region with respect to mirror symmetry at angle  $\theta$ . Local mirror symmetry is determined to be at the angle with highest correlation value. The correlation is simplified by first averaging the grey values along radial rays so that the angular correlations are then performed on a 1D grey scale function of the averages.

Computationally, finding Axial Shape Description and finding symmetric loci in images is computationally intense. The basic methods use voting schemes (described earlier) where pairs of points vote for a local symmetry axis or point. According to the highest vote, local symmetry axes are determined. In order to reduce complexity, contours are either approximated by parameterized curves (circular arcs [24], splines [229], etc.) or a projection method is assumed as described above [197, 34, 217]. In images, a smoothing process or hierarchical representation is used to regularize the method.

Local symmetries are typically used for shape and object representation, or for shape formation characterization. The notion of a Measure of Symmetry (as discussed in the following) is difficult to associate with local symmetries as the output of these representations are either structures (SAT, MAT, Skeletons) or collections of symmetry focal points. Additionally, without using smoothing or regularization the axial representations are highly sensitive to variations in the shape and they do not necessarily change smoothly with these variations.

### 3.1.5 Grouping using Coupled Diffusion Maps

Proesmans, Van Gool and A. Oosterlinck [220, 221] implemented a set of grouping principles, based on coupled diffusion equations. Each equation yields a so-called map, like an image but containing not the

raw intensities but one or the other evolving feature. One map may, e.g., contain pixel intensities as they get enhanced, while a coupled map contains intensity discontinuities that are being detected in parallel, i.e., edges. Such Coupled Diffusion Maps or CODIMS have also been proposed for the detection of different symmetries, including imperfect 2D repeated patterns and mirror symmetries. Thus, in the CODIM context, these symmetries are considered as examples among several possible grouping principles.

Grouping tasks are divided into two classes — local and bilocal — and for each a prototypical set of equations was presented. The local grouping processes grow regions that are reasonably homogeneous in one or the other basic feature (intensity, gradient orientation, etc.) by considering in parallel all individual pixels and their immediate neighborhoods. The regions get connected and grow through the diffusion nature of the equations. Examples are the search for regions of the same color or orientation of their texture. Simultaneously, the corresponding edge maps develop, as a coupled process. Bilocal processes follow similar processes, but consider two distinct places in an image or multiple images, comparing basic cues at these pixel pairs and their respective neighborhoods. Examples are the extraction of contiguous motion regions in single images or depth regions on the basis of stereo disparity in pairs of images.

In this CODIM framework, both the search of imperfect mirror symmetries and distorted 2D periodicities are examples of bilocal grouping cases. Both processes work by initializing several optical flow type procedures by Horn and Schunck [221, 220], but implemented bi-directionally and enriched with discontinuity preserving diffusion operators and additional discontinuity maps. Such processes are initialized with constant motion vector maps, each map having a different orientation and length for its vectors. Maps with vectors similar to periodicities or a joint connecting symmetric points, soon lock into such order. It suffices that this locking happens in small regions. From there onwards, the diffusion can spread the solution in a flexible manner, adapting the vectors along the way. This procedure allows for local deviations from perfect symmetry. For mirror symmetries, the process was adapted as to encourage pixels on both sides of a symmetry axis to

grow the symmetry in opposite directions, whereas in all other processes like optical flow, stereo, or periodicity search, pixel pairs try to expand the regularities by both evolving in the same direction. In order to keep the computational cost under control, only vectors up to some maximal length are considered. The price one pays is that the symmetry in a region close to the axis needs to be visible to kick-start the locking in process. Interestingly, also human vision does not pick up symmetry if a limited region around a symmetry axis is made asymmetric [27]. For the case of 2D periodicities, several processes initialized with different directions and lengths lock into different periodicities, where the same periodicity may be found by multiple, initial vectors. Together, the different periodicities allow such process to recover the tessellation of the periodic image regions.

Of course, one could initialize the matching processes in a more parsimonious fashion, by searching for matching interest points, for instance. It may well be however, that the brain throws in its powerful parallel processing capabilities, as would be beneficial for the CODIM algorithms.

Figure 3.10 shows examples of imperfect mirror symmetries, detected by a system of coupled diffusion equations. The figure shows the symmetry axes as they can be extracted by connecting the midpoints of corresponding points as found by the algorithm.



Fig. 3.10 Two examples of detected mirror symmetries with the CODIM framework.

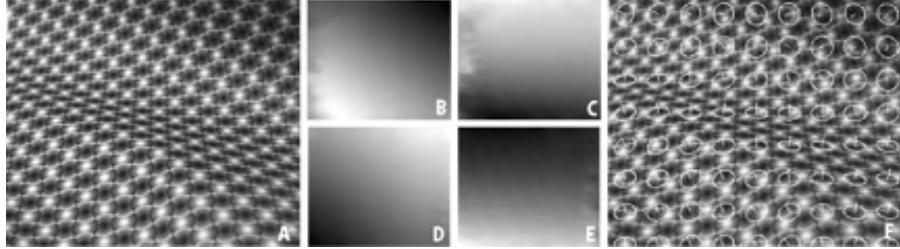


Fig. 3.11 (A): Original image, (B, C):  $x$  and  $y$  components for the first periodicity underlying the near regular tessellation; (D, E):  $x$  and  $y$  components of the second periodicity underlying the tessellation — together these two periodicities define the local tiles; (F): shape-from-texture (surface normals) based on the observed deformations of the tessellation (attributing the source of the deformation to the effect of projection).

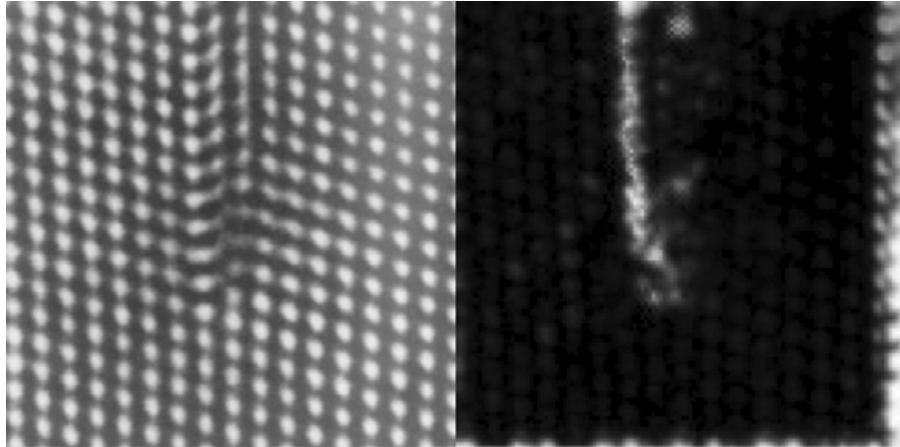


Fig. 3.12 Left image shows an atomic structure with a dislocation, i.e., an imperfection in the regularity. On the right, one of the CODIM discontinuity maps highlighting this deviation.

Figure 3.11 shows a curling sheet of wallpaper. Different periodicities were detected using CODIMs. The two main periodicities are shown in the middle of the figure. The horizontal and vertical components of each of those periodicities are shown in a first and second row, respectively. These maps exhibit the smooth variations in the periodicity vectors.

Figure 3.12 shows another example of an imperfect periodicity. This is a dislocation as found in atomic layouts. CODIMs come with maps

which explicitly indicate discontinuities. Here such discontinuity in the periodicity is found at the location of the dislocation (bright region in the image on the right).

### 3.2 Symmetry Detection by Symmetry Types

*Primitive symmetries* are the ‘atomic’ set-invariant transformations of an object, individually, each of them cannot be further divided and remains to satisfy the definition of symmetry (Section 2). In 2D Euclidean space there are only four different primitive symmetries: rotation, translation, reflection and glide-reflection. We focus on algorithms that detect these types of primitive symmetries and their combinations. Given its relative simplicity, the detection of bilateral reflection symmetry (mirror-symmetry) and its skewed version from images has been the dominant focus in computer vision (Figure 1.4).

As summarized in Table 2.3, there are only four types of *discrete symmetry groups* in 2D Euclidean space:

- (1) cyclic group  $C_n$  (rotation),
- (2) dihedral group  $D_n$  (rotation and reflection),
- (3) the seven frieze groups, and
- (4) the 17 wallpaper groups.

with the following special/limit cases:

- the identity group (asymmetrical figures with only one trivial symmetry),
- $SO(2)$  (Figure 2.3, special orthogonal group containing continuous rotation symmetry, a limiting case of  $C_n$ ),
- $O(2)$  (Figure 2.3, continuous rotation plus infinite number of reflections, a limiting case of  $D_n$ ),
- $T_1(T_2)$  symmetry group (a limiting case of the frieze/wallpaper groups, with infinitesimal translation symmetries, Table 2.1), and
- the symmetry group the 2D plane (Table 2.1).

Though the world is filled with various repeated patterns that can be characterized by discrete symmetry groups (Figure 1.1), the automatic

detection of discrete symmetry groups from real, non-segmented, digital imagery is relatively new. In texture analysis (Section 4), for example, many algorithms for finding the repeated image patches (texels) exist but few is aiming at an indepth analysis of the underlying symmetry group structures.

### 3.2.1 Reflection and Rotation

#### 3.2.1.1 From Reflection Symmetry to Dihedral $D_n$ and Frieze symmetry groups [162]

Recognizing that reflection symmetry plays an important visual as well as functional role in the folk art of paper-cut, this work [162] is perhaps the first algorithmic treatment (analysis and synthesis) of digitized paper-cut patterns (Figure 3.13) using automatically detected reflection symmetries. Making a connection between a reflection symmetry of the pattern and a folding action of the artist, the algorithm tries to discover all the potential single ‘folding lines’ (reflection symmetry axes) and the patterns formed by them, leading to more sophisticated symmetry structures such as patterns with a dihedral and a frieze symmetry group (Figure 3.13). This is done by searching through the entire 2D polar coordinates space  $(\theta, r)$ , for reflection axes that are supported by a sufficient set of *edge feature* pairs (Figure 3.14). This work proposes a hierarchical symmetry detection algorithm starting by finding individual reflection symmetries and moving on to find dihedral and frieze group structures: axes of a dihedral group forms a pencil shape and those of the frieze group form a set of equal distance parallel lines. The algorithm is voting based and uses edge features as the main matching cue. Figure 3.14 depicts the main steps in this algorithm [162].

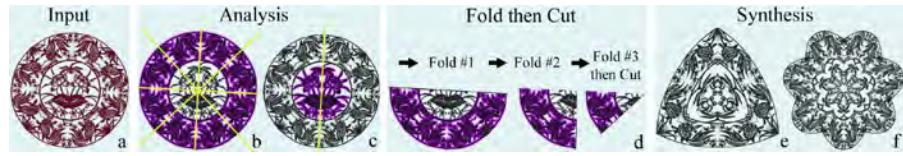


Fig. 3.13 Detect dihedral and frieze symmetry groups from reflection symmetries [162].

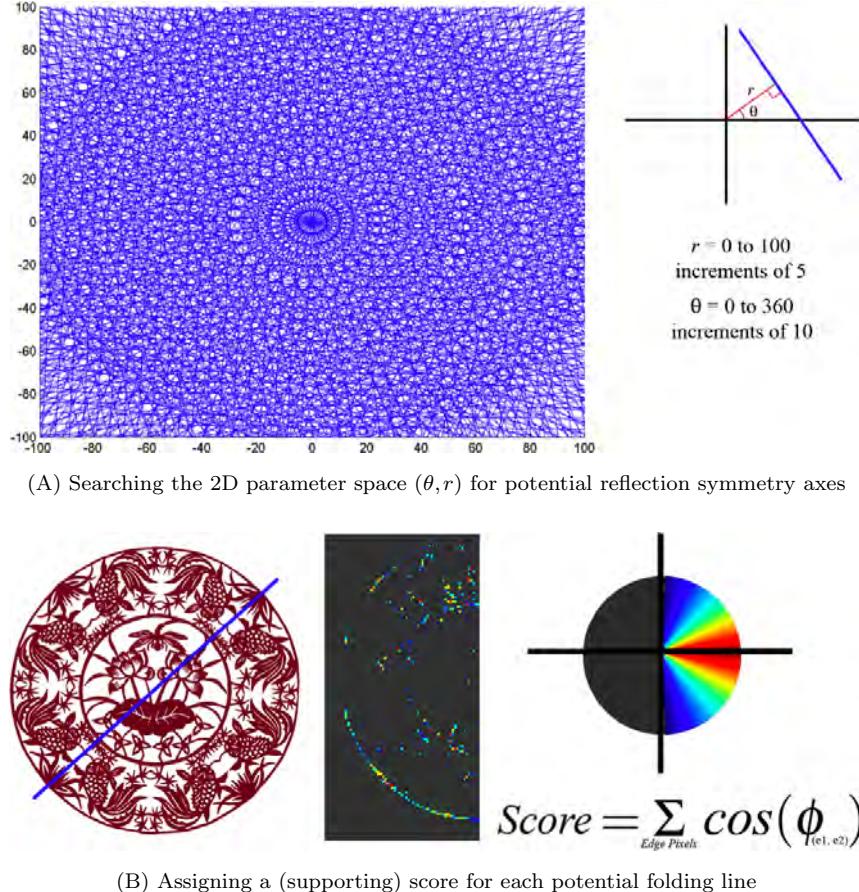


Fig. 3.14 The algorithm [162] discovers and organizes the potential reflection axes in the image. (A) Searching the  $\theta, r$  parameter space of the reflection axes (a voting method); (B) treating each reflection axis as a folding line, trying to match the edge points and assigning a score to the potential axis.

Given a paper-cut-pattern  $P$ , for each potential reflection symmetry axis  $l(r, \theta)$  on  $P$ , the algorithm first verifies whether  $l$  is a potential reflection symmetry axis of  $P$  by: (1) reflecting  $P$  about line  $l$  to obtain  $P_{ref(l)}$ ; (2) computing a symmetry score  $S_l = \sum \cos(\phi)$  where  $\phi$  is the angle between the orientations of corresponding edges of  $P(x, y)$  and  $P_{ref(l)}(x, y)$  at pixel  $(x, y)$ . The algorithm searches for peaks in  $M(r, \theta) = S_{l(r, \theta)}$  to locate those axes  $l$  with the highest scores. Finally, the

collections of  $l(r, \theta)$  are examined to see whether any of their structures indicate a desired symmetry group: (1) dihedral group: all reflection axes intersect in one point; or (2) frieze group: all reflection axes are parallel with each other and with equal intervals. Near-symmetry is also treated locally (Figure 3.14). The algorithm is implemented in MATLAB (available upon request at <http://vision.cse.psu.edu/data.html>).

Since the paper-cut patterns are human designed and hand cut, the images present various degrees of local and global symmetry or deviations from symmetry. The computational challenge to this algorithm is to distinguish symmetry from asymmetry, partial from global, and primitive reflection symmetries from higher level symmetry groups. This is one of the symmetry detection algorithms that are quantitatively evaluated and compared with in [203] (Section 3.3).

### 3.2.1.2 Reflection and rotation symmetry detection using local features [177]

This is a feature-based reflection and rotation symmetry detection algorithm proposed by Loy and Eklundh in 2006 [177], which takes advantage of local oriented features expressed as SIFT keys [176]. The basic idea is to collect pairwise feature matches to vote for candidate symmetry foci, either a straight reflection axis from multiple pairs of reflection symmetries (Figure 3.15) or a rotation symmetry center (or centers) from pairs of cyclic  $C_n$  symmetries (Figure 3.17), in a Hough transform fashion. The algorithm detects the rotation symmetry center and

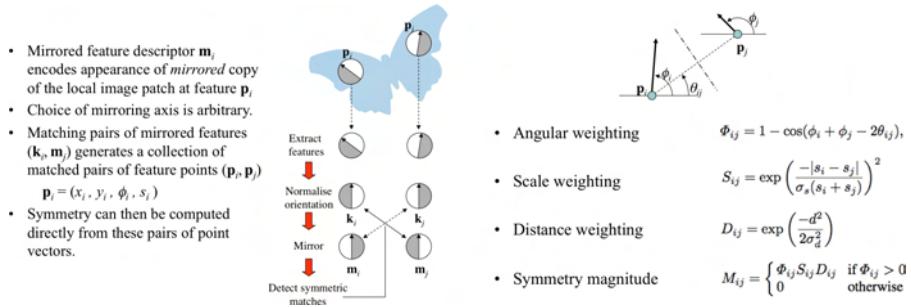


Fig. 3.15 The basic algorithm flow for reflection symmetry detection proposed in [177], a feature-based symmetry detection method.

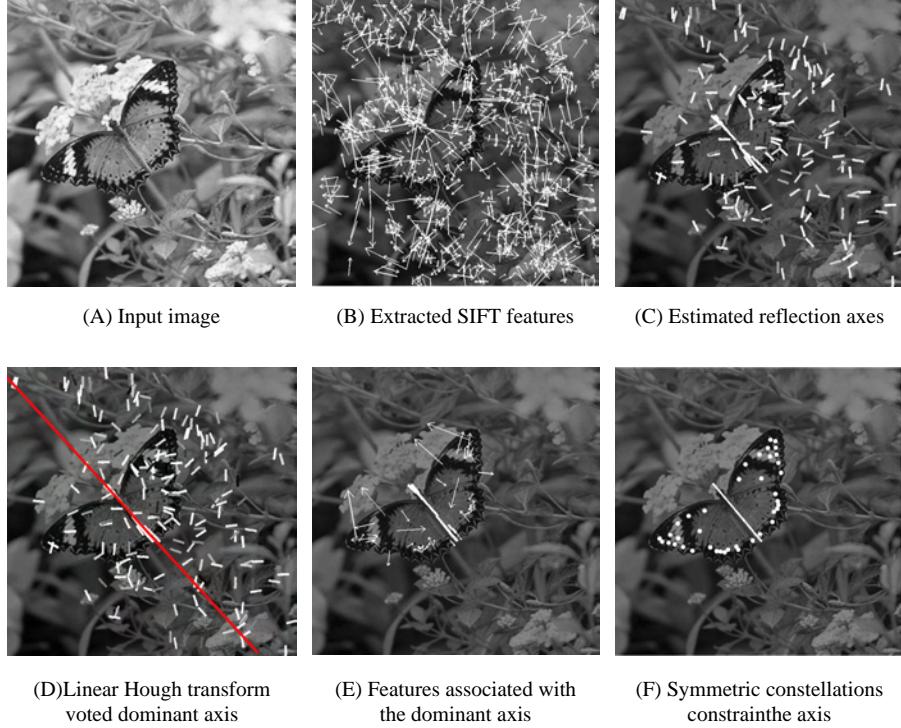
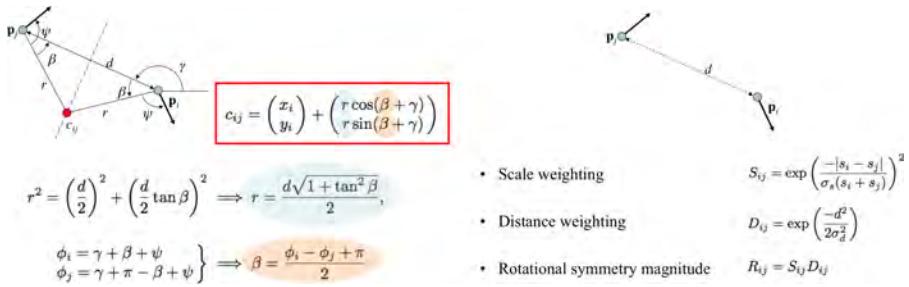


Fig. 3.16 A sample result from [177], a feature-based reflection symmetry detection method.

Fig. 3.17 The basic algorithm flow for rotation symmetry detection proposed in [177], a feature-based symmetry detection method. Every non-parallel feature pair defines a centre  $c_{ij}$  of rotational symmetry.

the fold (cardinality) of the rotation symmetry group. However, it only detects single reflection symmetries without recognizing any potential dihedral groups where both rotation and reflection symmetries co-exist.

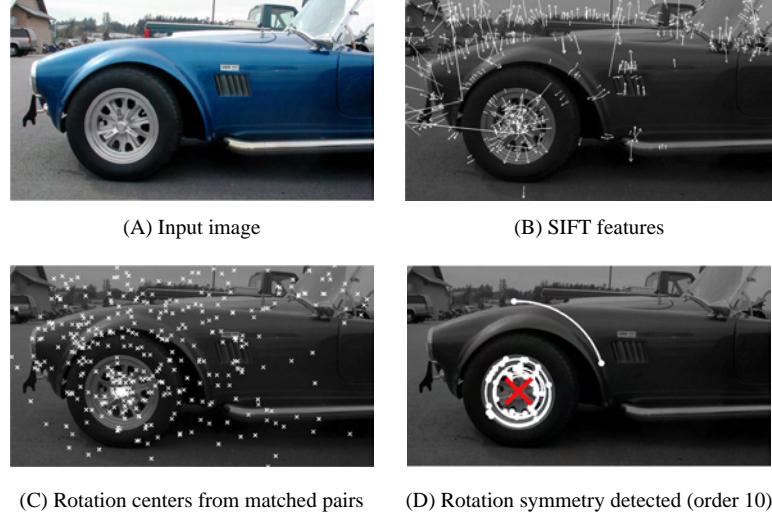


Fig. 3.18 Sample result from [177], a feature-based rotation symmetry detection method.

Figures 3.16 and 3.18 show two sample results of [177] for reflection and rotation symmetry detection from real images. This is one of the fastest and relatively more effective symmetry detection algorithms among the reflection/rotation symmetry detection algorithms evaluated in [203] (Section 3.3).

### 3.2.1.3 Rotation Symmetry Groups ( $C_n, D_{2n}, O(2)$ ) Detection via Frieze Expansion [132]

In 2D Euclidean space there are only three types of possible rotation symmetry groups about a fixed point (Figure 2.3). They are cyclic group  $C_n$ , dihedral group  $D_n$  and the orthogonal group of degree two  $O(2)$  (Table 2.3). Rarely, all three rotation symmetry groups can be detected from unsegmented images by a single symmetry algorithm, while a recently proposed rotation symmetry detection algorithm by Lee et al. [132] does it all. A rotation symmetry group is detected when the following four properties are identified by the algorithm automatically: (1) center of the rotation, (2) number of fold (cardinality of the symmetry group), (3) type of symmetry group (dihedral/cyclic/ $O(2)$ ), and (4) local supporting region (annulus).

A key idea behind this algorithm is to discover rotation symmetries in the Cartesian space by a freeze-expansion to discover translation symmetries of a freeze pattern in a corresponding polar coordinate space (Figure 3.19). A key observation is close relation between rotation symmetry groups and frieze groups (Figure 3.20). Given the location of a candidate rotation center, a diameter and a polar angle-step size, the algorithm maps each diameter to parallel lines, and re-aligns them from left to right, to form a horizontal pattern while advancing about the center angularly in the original image (Figure 3.19). This operation is called *frieze expansion*. Once the rotation symmetry pattern opens up, a frequency-based method is used to figure out the fold of the rotation and the supporting local regions, by performing a one-dimensional horizontal discrete Fourier transform (DFT) on each row of the frieze-expansion.

The proposed algorithm detects these four properties of a rotation symmetry group in two steps: (1) rotation center detection and (2) symmetry group analysis. A rotation symmetry-strength (RSS) map represents the symmetry strength at each pixel location of the image by a 2D weighted map. Rotation center detection finds peaks in the RSS map, followed by a symmetry analysis step to find the rest of the three symmetry group properties.

Let  $p_{x,y}(r, n)$  be an  $N \times R$  frieze-expansion pattern (FEP) (Figure 3.19) expanded at an image position  $(x, y)$  where  $r \in [1, R]$ ,  $n \in [1, N]$ ,  $R$  and  $N$  are the height and width of the FEP respectively. A one-dimensional horizontal discrete Fourier transform (DFT) is performed on each row of the FEP. The  $k$ th coefficient of the DFT of the  $r$ th row,  $P_{x,y}(r, k)$  is

$$\begin{aligned} P_{x,y}(r, k) &= a_{x,y}(r, k) + i b_{x,y}(r, k) \\ &= \sum_{n=1}^N p_{x,y}(r, n) e^{-i \frac{2\pi}{N} (n-1)(k-1)} \end{aligned} \quad (3.1)$$

where  $a_{x,y}(r, k)$  is the real part and  $b_{x,y}(r, k)$  is the imaginary part of the DFT coefficient.  $P_{x,y}(r, k)$  represents the complex value of each frequency component of the spatial domain information  $p_{x,y}(r, n)$ .

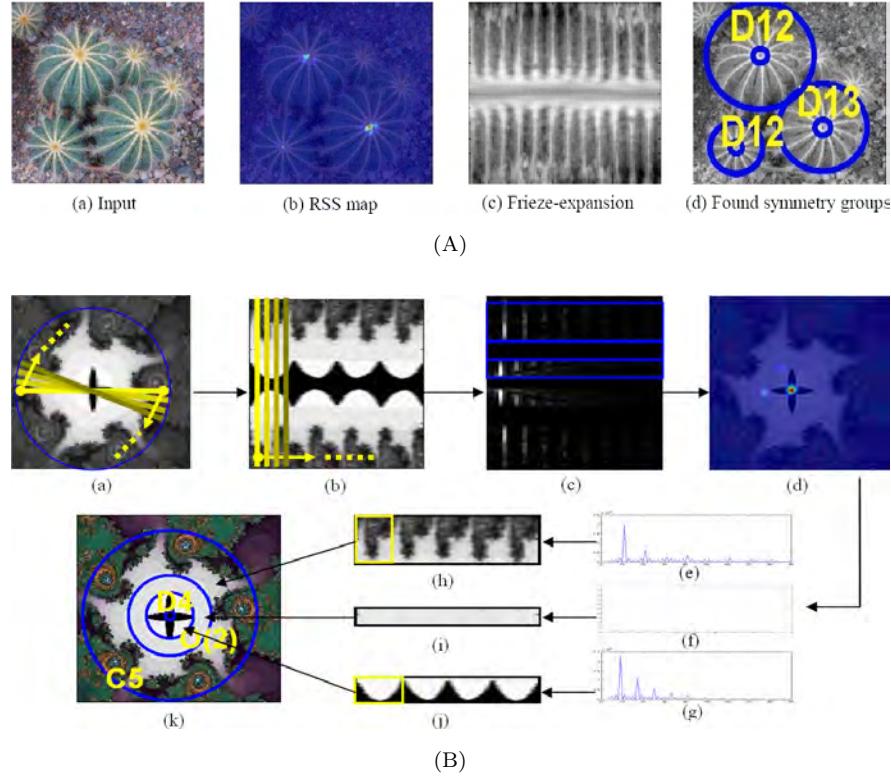


Fig. 3.19 Figures from [133] and [132] describing the rotation symmetry detection method: frequency analysis on the frieze expansion patterns. (A) A sample rotation symmetry detection process from a real image to detected rotation centers, rotation symmetry group types (all are dihedral group in this case) and their respective cardinality, and the supporting regions. (B) The proposed **frieze expansion** and symmetry analysis process on a sample image. Top-row: finding the rotation center; bottom-row: finding the corresponding symmetry groups dihedral group  $D_4$ , cyclic group  $C_5$  and orthogonal group  $O(2)$  are detected at different bands (annulus).

$S_{x,y}(r, k)$  is the energy spectral density of  $p_{x,y}(r, n)$ :

$$S_{x,y}(r, k) = \overline{P_{x,y}(r, k)} P_{x,y}(r, k) = a_{x,y}(r, k)^2 + b_{x,y}(r, k)^2 \quad (3.2)$$

The rotation symmetry-strength (RSS) equals to  $S_{x,y}(r, k)$ , which reflects the relative strength of a translational symmetry on the FEP with the  $k$ th potential frequency. Originally, RSS is a function of center position  $(x, y)$ , the radius  $r$  and the angular step size  $\theta$ . In the rotation center detection step of our algorithm, the largest circle with radius  $r$

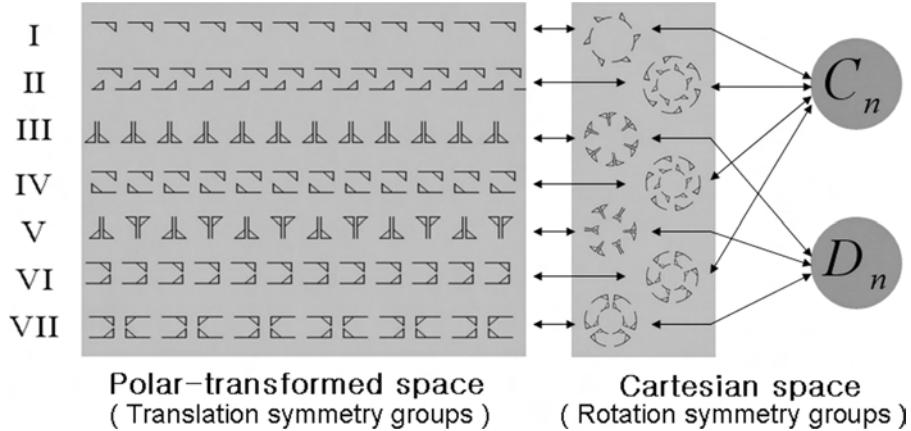


Fig. 3.20 One key observation of [132] and [134] (Figure 4 of [134]) is the unique relation between the discrete rotation symmetry groups and frieze patterns: if there is a rotation symmetry group ( $C_n$ : cyclic group of order  $n$ ,  $D_n$ : dihedral group of order  $2n$ ) in an image, its frieze-expansion should fall into one of the seven frieze groups.

inside the given image for each center location  $(x, y)$  is used and the angular step size  $\theta$  is fixed to  $\frac{2\pi}{N}$ , where  $N = 90$  [132]. RSS then becomes a function of position  $(x, y)$  only and can be represented as a two-dimensional RSS map. Higher values of RSS imply higher likelihood of rotation symmetry centered at that location. By calculating RSS values at each pixel of an image one can construct a pointwise RSS map, which reveals the rotation symmetry saliency of the given image (Figure 3.19). Sample results from this algorithm can be found in Figure 3.21. Lee and Liu [134] extends the algorithm reported in [132] by including skewed rotation symmetry groups, a novel rotation symmetry measure and an extensive quantitative comparison with state of the art rotation symmetry detection algorithms [132] and [177].

### 3.2.2 Curved Glide-reflection Symmetry

To the best of our knowledge, little attention has been paid to glide-reflection symmetry detection algorithms, except in [159] where the authors evaluate the existence of specific glide-reflection symmetries to verify certain types of wallpaper and frieze symmetry groups.

Synthetic image				Natural image			
Desc.	Image	RSS map	Result	Desc.	Image	RSS map	Result
[I-1] C3				[I-10] D4 No symmetry inside			
[I-2] D12				[I-11] C14			
[I-3] D4,C5 SO(2) Concentric symmetries				[I-12] D22 Occlusion High texture			
[I-4] C3 Different color				[I-13] D33 High fold High texture			
[I-5] D4 No texture				[I-14] D12,D13 Multiple skewed			
[I-6] C3,C6 C4,C5 Concentric symmetries				[I-15] D13 skewed			
[I-7] SO(2) No texture				[I-16] D5 Multiple centers			
[I-8] D4,D8 Wall paper (P4M) Multiple centers				[I-17] D High texture Occlusion			
[I-9] C6,C3 Wall paper (P6) Multiple centers				[I-18] Random noise (No Symmetry)			

Fig. 3.21 Sample results of the proposed rotation symmetry detection [132].

In 2009, Lee and Liu [133] propose a new *curved glide reflection symmetry* that captures many real world symmetries without a straight reflection axis (Figure 3.24). This type of symmetry generalizes the popular reflection symmetry and the not-so-popular glide reflection symmetry as two of its special cases, as follows (Figure 3.22).

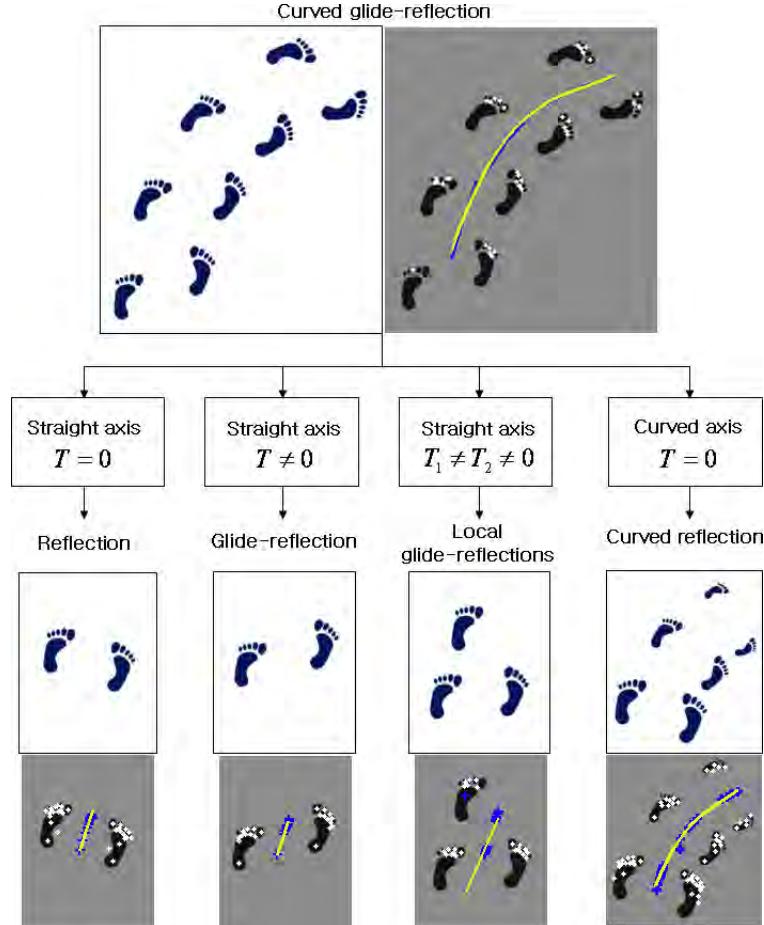


Fig. 3.22 Four special cases of the curved glide-reflection symmetry and detection results of the proposed algorithm (yellow lines): blue dots represent the middle points of the supporting local feature pairs and yellow lines are detected curved reflection axes.

Glide-reflection symmetry is a transformation composed of a translation  $T$  along and a reflection  $R$  about the same axis (Figure 2.1). If a pair of image patches  $P_i$  and  $P_j$  has a glide-reflection symmetry, it can be expressed as:  $P_i = T + R(P_j)$ . Thus, a pure reflection is a special case of a glide-reflection when  $T = 0$ . A *curved glide-reflection symmetry* is defined in [133] as: a sequential collection of local glide-reflection symmetries whose reflection axes are connected and tangent

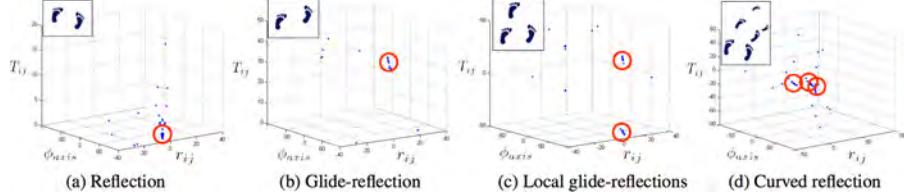


Fig. 3.23 3D parameter space examples of the four sub-types of curved glide-reflection symmetries: red circles show the characteristic patterns detected in the 3D parameter space location.

to a smooth curve. Thus a curved, glide-reflection symmetry can be represented as a sequence of  $(T_i, R_i)$ s where  $T_i \neq T_j$  and  $R_i \neq R_j$  in general. The four special cases are (Figure 3.22):

- (1) *Reflection* when all  $T_i = T_j = 0$  and  $R_i = R_j$ ;
- (2) *Glide-reflection* when  $T_i = T_j = T \neq 0$ ;
- (3) *Local glide-reflection* when multiple glide-reflections exist, and  $0 \neq T_i \neq T_j \neq 0$ ;
- (4) *Curved reflection* when multiple reflections exist,  $T_i = T_j = 0, R_i \neq R_j$ .

The proposed method is feature-based that groups statistically dominant local reflection axis in a 3D parametric space: translation is represented by one parameter  $T_{ij}$ , reflection  $R$  is indicated by a pair of parameters  $(r_{ij}, \phi_{axis})$  representing its reflection axis (Figure 3.23). They also develop a curved glide reflection symmetry detection algorithm and applied on 40 real world images (Figure 3.24).

### 3.2.3 Wallpaper and Frieze Groups Classification: Euclidean Case

In computer vision, the introduction of a fully automated, computational treatment of the 17 wallpaper groups and the seven frieze groups embedded in real, unsegmented images, appeared in a sequence of papers: [154, 155, 156, 159]. The authors argue that the discrete (finitely generated) symmetry groups for periodic patterns, discovered by mathematicians a century ago, but largely ignored by the computer vision community, are indeed important, relevant and useful for computer

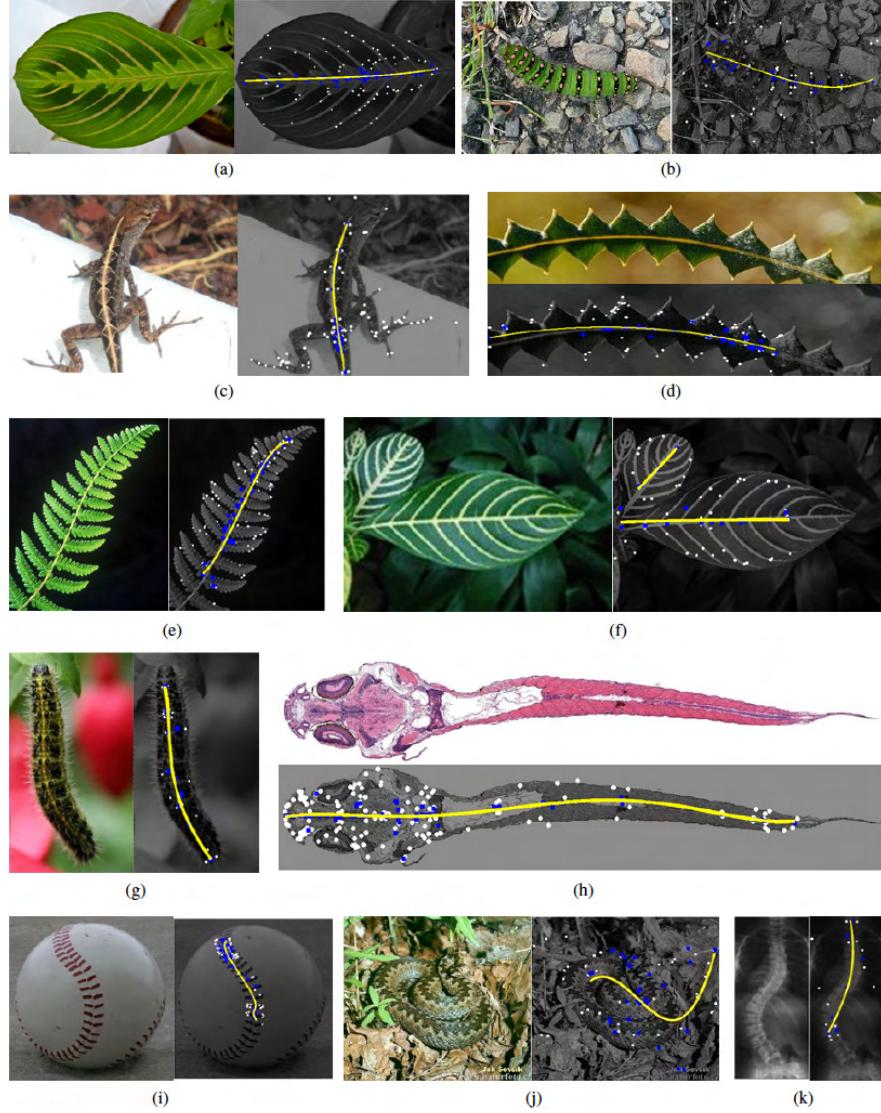


Fig. 3.24 Results from [133]. The bottom row shows some of the failure cases.

vision, computer graphics and pattern recognition research. Various computer vision/graphics applications of such symmetry groups, especially its translation subgroups and its associated lattice structures, subsequently followed (Section 4), including: texture replacement in

real photos [262], human gait analysis [158], real world near-regular texture analysis, synthesis and manipulation [171, 163, 164, 172], dynamic near-regular texture tracking and dynamic texture replacement/superimposing [147, 148], image ‘de-fencing’ [153], automatic geotagging [240], fabric categorization [89], architectural facard in-painting [127], multi-target tracking of time-varying spatial patterns [150] and a quantification of the firing fields of the rats’ grid cells in computational neurosciences [32].

### 3.2.3.1 Lattice detection

A non-trivial translation subgroup is common to all crystallographic groups (Figure 2.7). For example, the translation subgroup of each wallpaper group is generated using a pair of *translation generators*, customarily called  $t_1, t_2$  (Figure 3.25). In order to classify the wallpaper (frieze) symmetry group of a real periodic pattern, the first step is to detect the *lattice* that is equivalent to locating the translation subgroup of the symmetry group by finding its  $t_1, t_2$  generators (Figures 3.26, 3.27). Figures 3.28, 3.29, and 3.30 demonstrate the underlying lattice

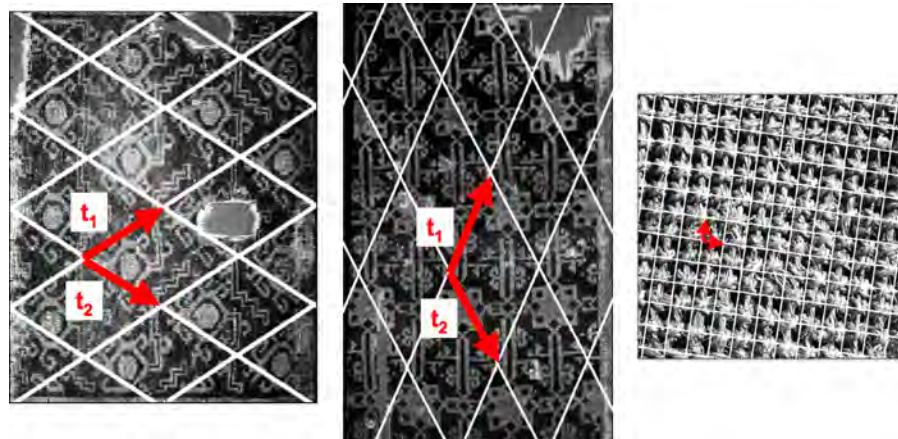


Fig. 3.25 The 2D lattice and its respective *translation generators*  $t_1$  and  $t_2$  [159].

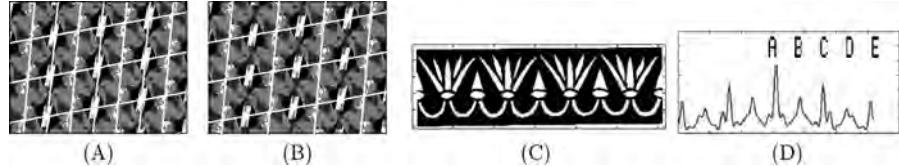


Fig. 3.26 (A) An incorrect lattice found by the algorithm of Lin et al. [145] (B) Correct lattice found by the authors' algorithm. (C) Frieze pattern and (D) its 1D autocorrelation response, used to explain how spurious peaks can form.

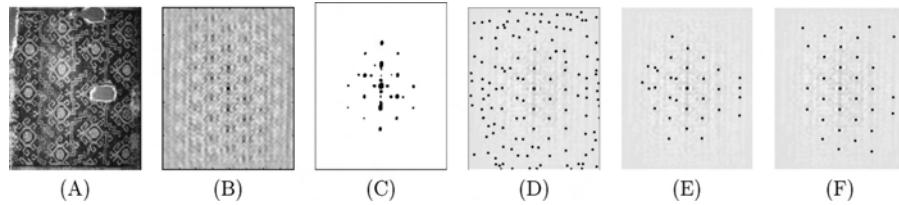


Fig. 3.27 (A) Original image of a rug. (B) An autocorrelation surface. (C) Peaks found using a global threshold. (D) Peaks extracted using the threshold-free method of [145]. (E) The highest 32 peaks from those return by [145]. (F) The 32 most-dominant peaks found using the region-of-dominance idea.

automatically discovered and the wallpaper symmetry group classification results [159].

The proposed algorithm in [159] requires each of the input images dominated by some periodic pattern, thus the auto-correlation method is feasible while the Fourier method is not, due to the lack of a large number of sample cycles (usually 2–3 cycles only on each input image). Realizing that the peak-finding problem is non-trivial, the authors proposed a *region-of-dominance* idea based on the observation that the absolute height of a peak is not as important as the size of its *region of dominance*, defined as the *largest circle centered on the candidate peak such that no higher peaks are contained in the circle*. A peak with a low height, but located far from any larger neighbors, is much more perceptually important than a high peak that is close to an even higher one. Referring to Figure 3.26(D), the true peak E is lower than the spurious peak B, but is located twice as far away from any higher peak than B is,

and thus dominates a larger region. Revisiting Figure 3.27(F), the first 32 most-dominant peaks found using this method are well distributed over the whole image, with very few spurious peaks.

### 3.2.3.2 Wallpaper group classification

For the 17 wallpaper groups, once the translation generators  $t_1, t_2$  are identified, i.e., the translation subgroup is found, the algorithm proceeds to verify the existence of the rest of symmetries, rotation, reflection and glide-reflections, based on the group definitions (Figure 2.4) and Table 3.1 (from [159]). The seven Frieze groups are classified in an analogous manner. The algorithm in [159] successfully classifies the symmetry group of a given pattern into one of the 7 frieze groups or one of the 17 wallpaper groups (Figures 3.28, 3.29, and 3.30).

The limitations of the work are: (1) the periodic pattern has to dominate the input image; and (2) geometrically, the pattern has to be close to periodic. Later deformed lattice detection algorithms (Section 4, [95, 204, 202]) are not restricted by these constraints.

Table 3.1. Wallpaper group classification: the group associated with a wallpaper pattern can be determined by checking the small number of symmetries: 180, 120, 90 or 60 degree rotation symmetry, reflection symmetry and glide-reflection symmetry about axes parallel to lattice unit parallelogram boundary vectors  $T_1$  and  $T_2$  or diagonal vectors  $D_1$  and  $D_2$ .

	p1	p2	pm	pg	cm	pmm	pmg	pgg	cmm	p4	p4m	p4g	p3	p3m1	p31m	p6	p6m
180	Y				Y	Y	Y	Y	Y	Y	Y	Y			Y	Y	
120													Y	Y	Y	Y	Y
90										Y	Y	Y					
60															Y	Y	
$T_1$	Y	Y(g)			Y	Y(g)	Y(g)			Y	Y(g)			Y		Y	
$T_2$					Y	Y	Y(g)			Y	Y(g)			Y		Y	
$D_1$			Y					Y		Y	Y		Y	Y		Y	
$D_2$								Y		Y	Y					Y	

“Y” means the symmetry exists for that symmetry group; empty space means no. Y(g) denotes a glide reflection.

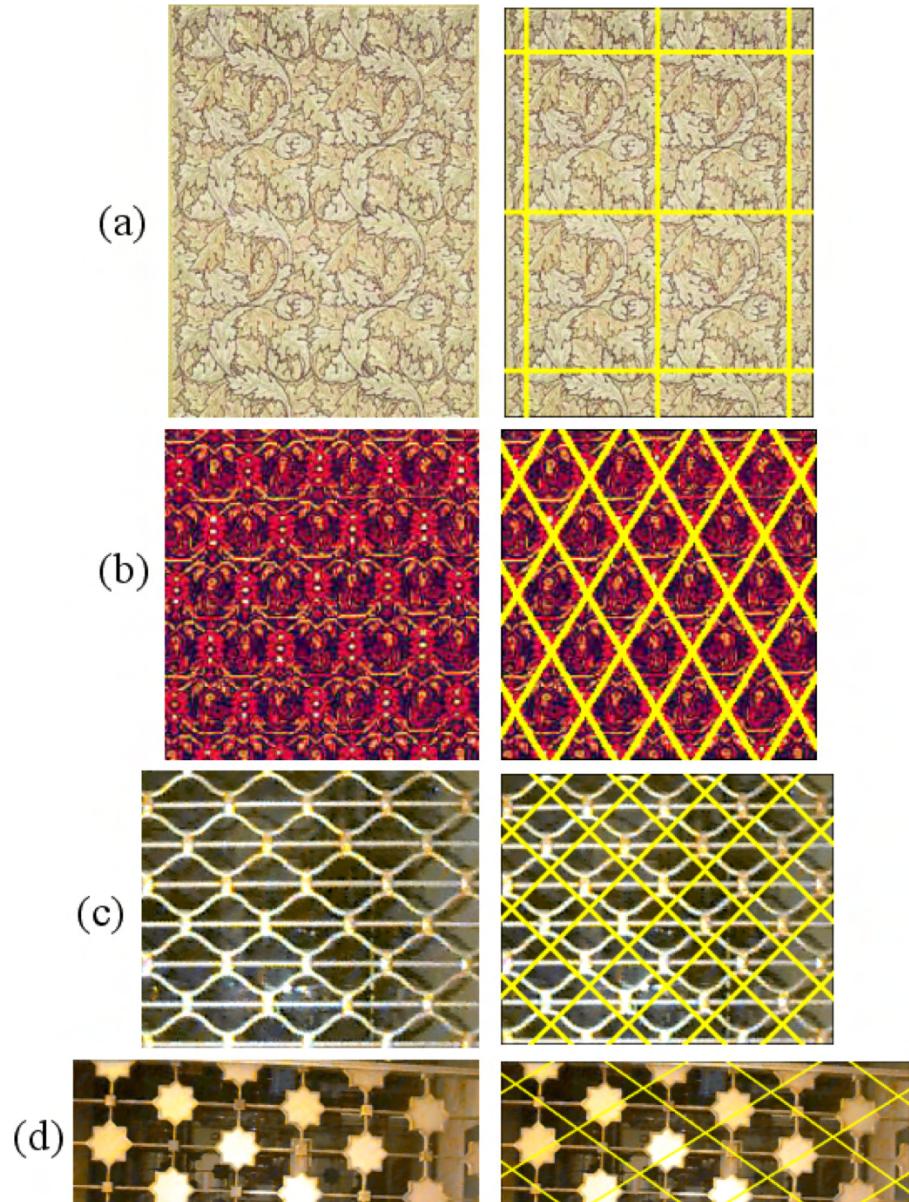


Fig. 3.28 The detected lattice and corresponding wallpaper symmetry group [159].

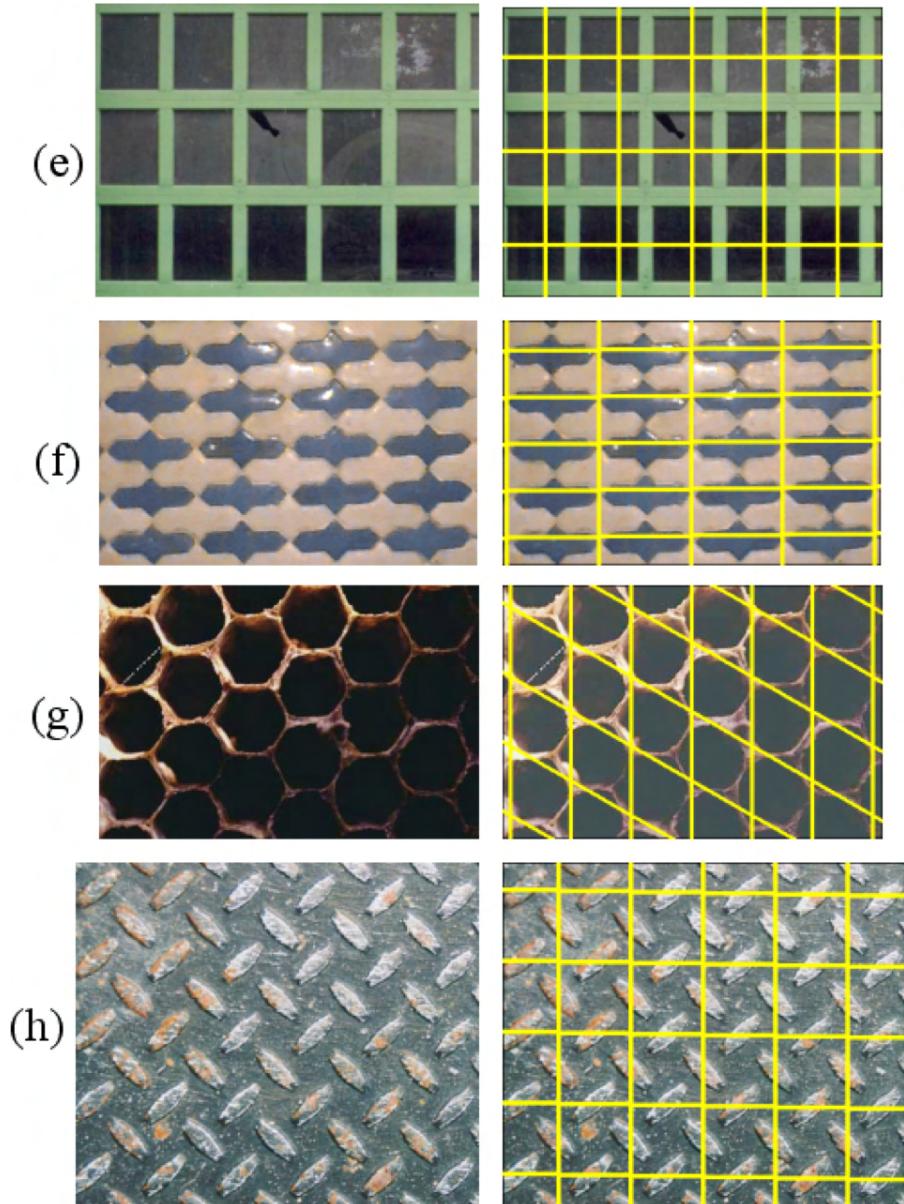


Fig. 3.29 The detected lattice and corresponding wallpaper symmetry group [159].

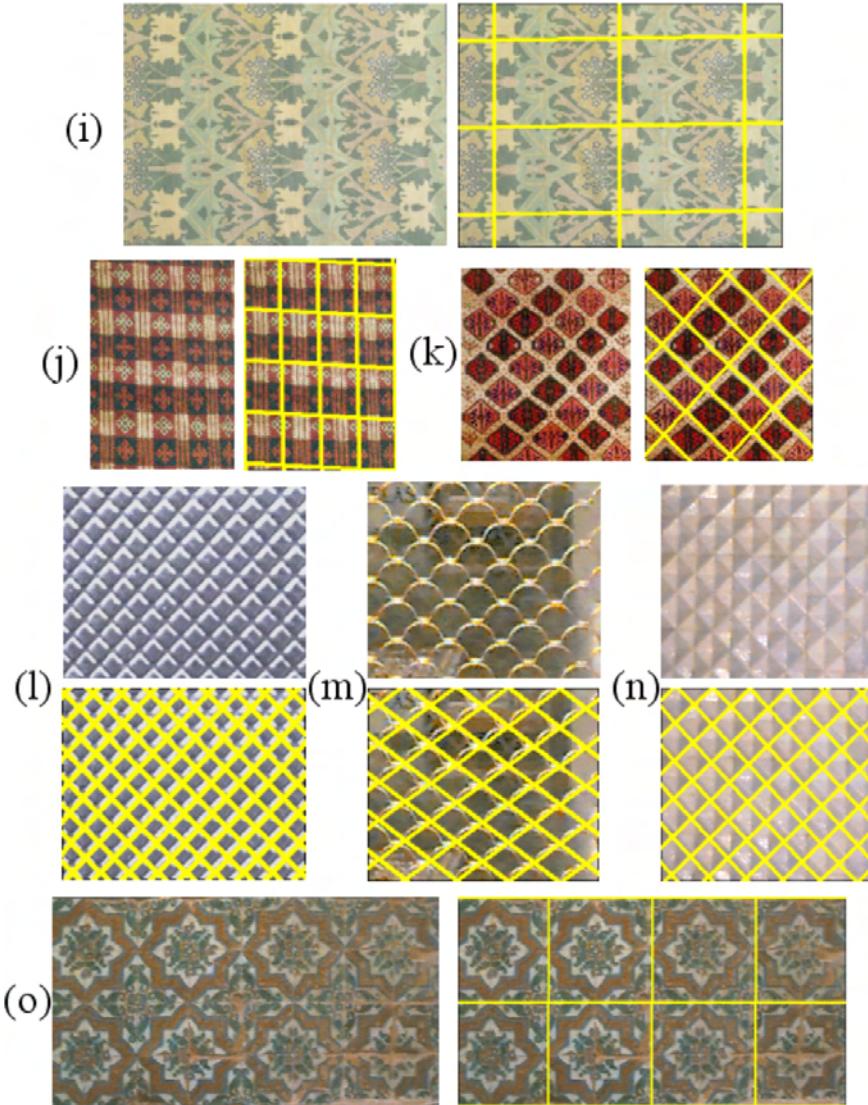


Fig. 3.30 The detected lattice and corresponding wallpaper symmetry group [159].

### 3.2.4 Discovering Structural Regularity in 3D Models [207]

The advantages of discovering regular structures in 3D graphical models are recently demonstrated in [207]. Pauly et al. developed an algorithm to capture a set of repeated structures of 3D models in Euclidean space.

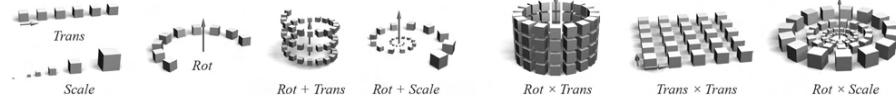


Figure 3 of [210]: the types of 2D (2-parameter) symmetries it handles.

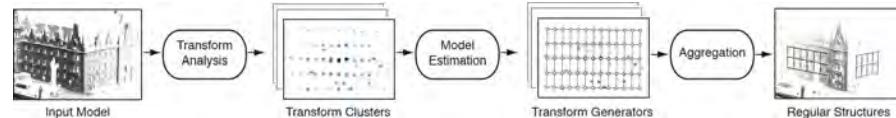


Figure 2 of [210]: the processing pipe line – the lattice is found in a 2D transformation space.

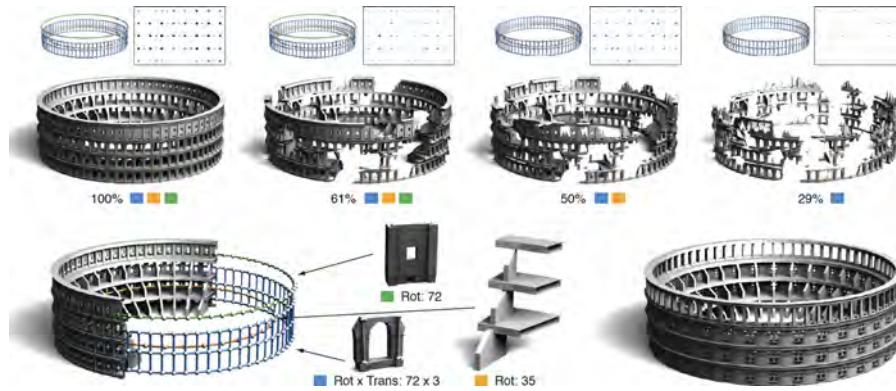


Fig. 3.31 Regular structures discovered by [207] involve combinations of rotation, translation, and scaling of the repetitive elements. Bottom: structure discovery and procedural design on a 3D model of an amphitheater. The top row indicates which of the three regular structures (blue, green and yellow) can be reconstructed in their entirety when successively removing geometry. The density plots show a section of the distribution of transformations of the structure depicted in blue. The recovered structures are used to procedurally design a new model in the bottom right.

The algorithm searches for a finite set (three 1-parameter and five commutative 2-parameter groups) of discrete symmetry groups composed of translation, rotation and scaling (Figure 3.31 top row), extending the Euclidean symmetries to Similarity transformations while excluding reflection and glide-reflection symmetries (Figure 1.3, Table 2.3).

The idea is to study the structures of a set of 3D points by clustering its symmetries in a 2D transformation space. The authors pick a finite set of possible 1D and 2D discrete symmetry groups. Therefore, the central computational problem is to discover a set of 2D symmetries such like 1D translation, rotation, helical, spiral, and 2D translation

symmetries in Euclidean space that appear as regular grids in transformation space (Figure 3.31 pipeline). Different from the noisy real world image data faced by computer vision researchers, the input is composed of random sampled point pairs from a 3D computer model (a large point set) . However, this work faces similar challenges as in computer vision algorithms (e.g., [95, 159, 202, 204]), namely the algorithm does not know either the implied symmetry group (transformations) or the *unit tile or generators* to start with.

Using a quadratic objective function inspired by the *iterative closest point* (ICP) algorithm, the authors construct an energy-based model to minimize the sum of squared distances of the grid locations to the cluster centers, where a Gauss–Newton solver is applied. The user need to input a parameter  $h$  to determine the smallest scale at which regular structures are discovered. Impressive results are shown, in particular for subsampled data (Figure 3.31 bottom), where the regular structure prevails even when 71% of the data is missing. This example serves as yet another evidence that symmetry features, if exist, are resilient to statistical sampling perturbations.

### 3.2.5 Translation and Reflection Symmetries under Perspective Skew [267]

Tuytelaars et al. propose a framework for detecting regular repetitions of planar (but not necessarily coplanar) patterns with translation and reflection symmetries [267]. They aim at a uniform framework, by picking up the fixed structures under different types of (perspectively skewed) symmetries. Most of the paper focuses on the cases where the symmetry presents itself as a planar homology in the image plane. This means that the symmetry leaves all points on a line and all lines through a point fixed. Skewed mirror symmetries and skewed periodicities are special cases.

A key goal of the paper is to find such fixed structures efficiently, i.e., without lengthy combinatorial search. Therefore, corresponding points are searched through the matching of affine-invariant interest points. These matches then yield the input for a so-called Cascaded Hough Tranform (CHT). This involves an interated application of the

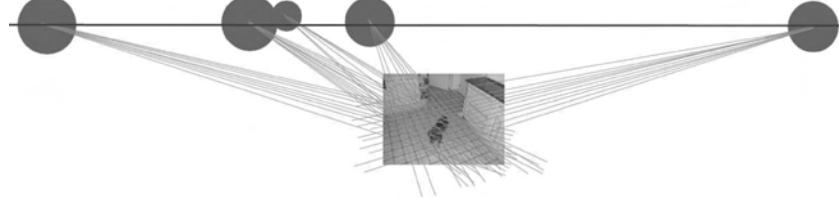


Fig. 3.32 Figure 2.4 in [267]. The candidate pencils of fixed lines (due to vanishing points) and line of fixed points, as detected by the Cascaded Hough Transform (CHT, an iterated application of the traditional Hough transform for straight lines). Different sizes of the vertices indicate different supports.

Hough transform for straight lines, but with the special parameterization  $ax + b + y = 0$  for the lines. This expression is perfectly symmetric in the point coordinates  $(x, y)$  and the line coordinates  $(a, b)$ , thereby bringing out the duality between the two types of structures in the projective plane (Figure 3.32). Identically the same Hough transform can then be repeated, where the first detects co-linear points, the second the points where multiple such lines intersect, and the third collinear arrangements of such intersections. These structures then potentially correspond to fixed structures of (skewed) symmetries. The system is also capable to reason about the consistency of multiple homologies that are detected, to further weed out false detections. Figure 3.33 shows some regularities picked up by CHT analysis. Note how for the carpet multiple, consistent symmetries have been found.

### 3.3 Quantitative Comparison of Symmetry Detection Algorithms [203]

Within the past few years, we have observed a surge of new symmetry detection papers in the related fields of computer vision and computer graphics [95, 162, 177, 190, 215, 219, 259]. While each paper demonstrates certain experimental results of the proposed algorithm, without a systematic evaluation of different symmetry detection algorithms against a common image set under a uniform standard, our understanding of the power and limitations/pitfalls in state of the art symmetry detection algorithms remains partial. This situation seriously hinders a wide applicability of existing symmetry detection algorithms and any concrete, measurable improvements thereafter.



Fig. 16. Two books in front of a mirror. **Middle and right:** the resulting segmentations.

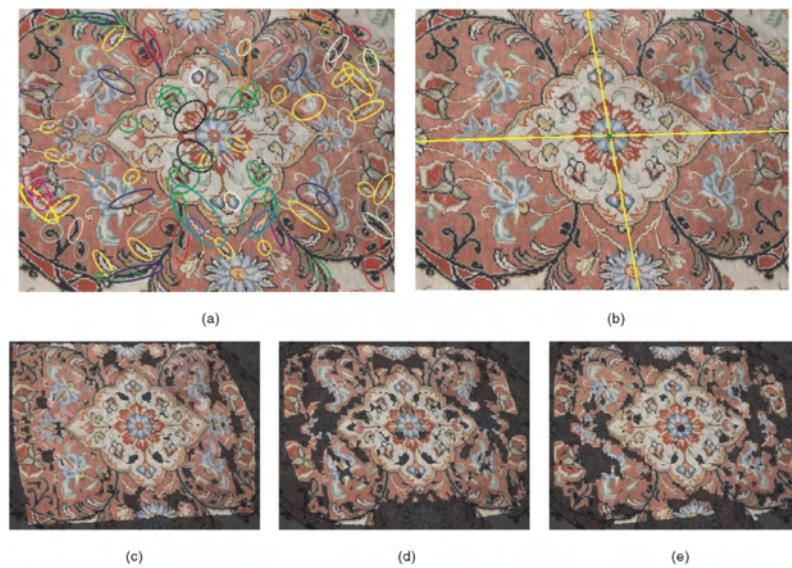


Fig. 17. A hand-woven carpet. (a) Detected repetitions, (b) some of the detected fixed structures, (c), (d), and (e) the resulting segmentations for the two mirror symmetries and the reflection about a point, respectively.

Fig. 3.33 Figures 2.11 and 2.12 in [267].

The first quantitative benchmark dataset for evaluation of different symmetry detection algorithms on the same set of images is reported in [33, 203]. The authors make a specific effort on

- collecting a test image set and making it publicly available;
- providing hand-labeled groundtruth by a group of raters;
- proposing and implementing a performance evaluation standard; and
- selecting publicly available (code or executable) state of the art discrete symmetry detection algorithms for evaluation.

The primary reasons the authors choose the following three algorithms [162, 177, 219] to evaluate are: (1) recently published; (2) have gone beyond single reflection/rotation symmetry detection. In fact, they all claim to detect multiple symmetries in an image<sup>1</sup>; (3) directly applicable to un-segmented, real images; and (4) the authors make their source code publicly available.

**Reflection/Rotation Symmetry Detection Algorithms.** We briefly describe each symmetry detection algorithm evaluated in this paper below.

**1. Detecting Symmetry and Symmetric Constellations of Features** [177] (ECCV06) for both rotation and reflection symmetry detection (Figure 3.34 (1)):

This is a feature-based reflection and rotation symmetry detection algorithm, which takes advantage of local oriented features expressed as SIFT keys [176]. The basic symmetry detection technique uses pairwise matching and voting for symmetry foci (single reflections and  $C_n$  symmetries) in a Hough transform fashion. It also estimates the  $n$  in cyclic group  $C_n$  but it does not make the distinction between  $C_n$  and  $D_{2n}$  type symmetry groups.

**2. Digital Papercutting (Papercut)** [162] (SIGGRAPH 2005) for reflection symmetry detection (Figure 3.34 (2)):

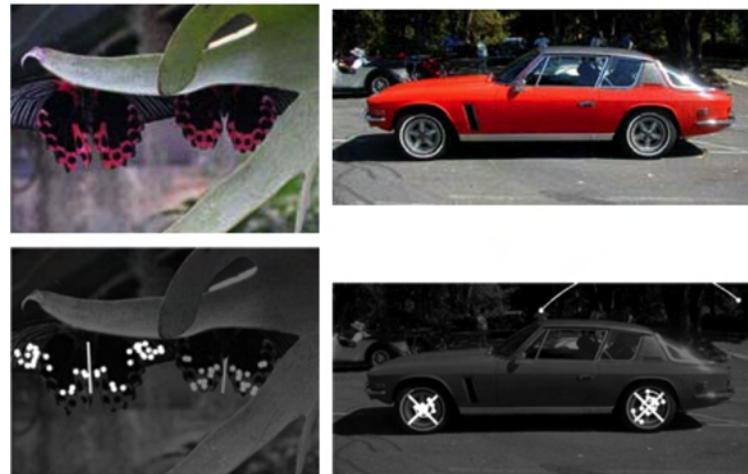
This algorithm is originally designed for the analysis of images of artistic papercutting patterns. Thus, it uses edge-based features. The algorithm exhaustively searches through the parameter space of potential reflection axes (in polar coordinates  $\rho, d$ ) to identify single reflection symmetries by voting for pairwise matches, and structures of reflection axes to discover  $D_{2n}$  (if reflection axes intersect in one point) and frieze (reflection axes are placed in parallel, with equal distance) symmetries.

**3. Detecting Rotational Symmetries** [219] (ICCV05) (Figure 3.34 (3)):

This algorithm filters an input color image into a gradient vector flow (GVF) field and conducts the extraction and matching of local features in the GVF field. The symmetry detection is formulated, once again, as a voting scheme for the centroids of  $C_n$  symmetries.

---

<sup>1</sup>Even though [162] is designed to work on images of papercut-patterns only.



(1) Reflection (left) and rotation (right) symmetry detection [180]

Top row: input images. Bottom row: detected reflection symmetry axes and rotation symmetry centers with symmetric constellations of feature points.



(2) Reflection symmetry group detection [165]: multiple symmetry axes of local regions are detected one-by-one.



(3) Rotation symmetry detection [222].

Left image: input image.

Fig. 3.34 Sample results for multiple-symmetry detection without segmentation and pre-processing from each of the three algorithms evaluated in [203].

## Reection Symmetry Detection Test Set

Image Number	Original Image	Ground Truth	Description /Challenges	Loy and Eklundh 2006	Liu et al 2005	Loy and Eklundh 2006		Liu et al. 2005	
						GT/TP/FP	GT/TP/FP	GT/TP/FP	GT/TP/FP
Ref_SS12			Single bilateral symmetry; Highly textured lower half			1/1/0	1/1/0		
Ref_SM5			Multi-reflection symmetry axes with global and local supporting regions			16/2/0	16/1/0		
Ref_RS22			Real photo, off-centered single symmetry; Cluttered BG			1/0/4	1/0/0		
Ref_RM27			Real photo, multiple local symmetry axes; Skewed			5/0/1	5/1/0		

## Rotation Symmetry Detection Test Set

Image Number	Original Image	Ground Truth	Description /Challenges	Loy and Eklundh 2006	Prasad and Davis 2005	Loy and Eklundh 2006		Prasad and Davis 2005	
						GT/TP/FP	GT/TP/FP	GT/TP/FP	GT/TP/FP
Rot_SS9			Synthetic image, single symmetry: C3; Textured			1/1/2	1/1/0		
Rot_SM1			Synthetic multiple symmetries: D8, D16; Color variation; Global and local symmetries			5/3/0	5/0/2		
Rot_RS4			Real photo, single symmetry: D24; Cluttered; Background inside the rotation region			1/1/0	1/0/1		
Rot_RM5			Real photo, multiple symmetries: D13, D15; Highly textured FG and BG			14/3/0	14/1/0		

Fig. 3.35 Sample images and results from our test image set. We also provide labeled ground truth, descriptions of computational challenges, and the numbers of ground truth (GT), and detected true positive (TP) and false positives (FP). The complete test image set can be found in <http://vision.cse.psu.edu/evaluation.htm>.

**Test image data set:** the authors tested these symmetry detection algorithms on a total of more than 800 images, including 176 images with hand-labeled ground truth and images from three publicly

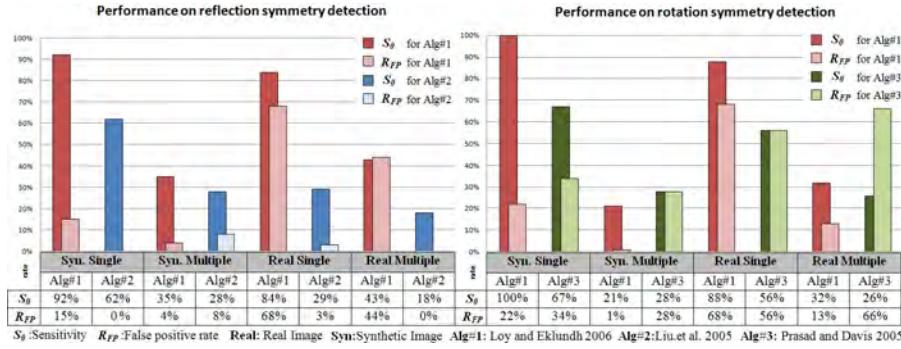


Fig. 3.36 The pairwise reflection and rotation symmetry detection algorithms evaluation on our 176 test-images with labeled ground truth.

available databases: PASCAL VOC'07 [60], MSRC Object class recognition database [278] and Caltech-256 [80] data sets. The 176-image test set provided by the authors is selected to demonstrate unambiguous object-level symmetries, with diverse visual properties: synthesized versus real images, single symmetry versus multiple, clean versus textured regions, frontal versus skewed views, similar versus contrasting color intensities. The whole 176-test-image set is divided according to two standards: (1) synthetic versus real images; and (2) images containing a single dominant symmetry versus multiple symmetries. Given two types of symmetries (rotation and reflection), there is a total of eight different subcategories of test images (Figure 3.36).

Furthermore, they explored the potential role a symmetry detection algorithm might play in object and object class recognition/categorization by testing and evaluating the algorithm with best performance on selected images from PASCAL VOC'07 [60], MSRC Object class recognition database [278] and CALtech-256 [80].

**Evaluation standard:** to quantify algorithm performance, the following formula is proposed to compute the success rate on a test image set [33, 203]:

$$S_{K_p} = \frac{(N_t - K_p \times N_f)}{N_{GT}} \quad (3.3)$$

where  $N_t$  is the number of true positive (symmetries in the image that are detected by the algorithm),  $N_f$  is the number of false positive

(non-symmetries detected by the algorithm as symmetries), and  $N_{GT}$  is the number of ground truth (symmetries in the image that should be detected).  $K_p$  is a constant weight that determines how strongly we penalize the false positives. When  $K_p = 0$ ,  $S_0 = \frac{N_t}{N_{GT}}$  is the commonly known *sensitivity* [131] that is independent of the false positives. When  $K_p = 1$ ,  $S_1 = \frac{(N_t - N_f)}{N_{GT}}$  reflects a combination of rewarding true positives and penalizing false positives in a 1-to-1 ratio.

**Results:** the evaluation results (Figure 3.36) show that for all types (synthetic or real, single or multiple symmetries) of images the best mean sensitivity is 63% and 58% by [177] for reflection and rotation symmetry detection, respectively. For computer vision applications, the detection of multiple symmetries in one real image should be most relevant. We have found the best mean sensitivity rates to be 42% (for reflection) and 32% (for rotation), once again by [177]. However, for a true understanding of multiple existing symmetries in an image, we should be more concerned with the overall success rate  $S_1$  (including the false positives). The best values of  $S_1$  are much worse: 25–27% [162, 177] for reflection/rotation symmetry detection in the overall image set, and 16–19% [162, 177] on real images. The drastic drop from  $S_0$  (42%) to  $S_1$  (1%) (Table 1, real image, multiple symmetries, of [203]) reveals a fatal weakness of the SIFT-type feature-based method used in [177], indicating high false positive rates on real images with multiple symmetries.

Ideally, these algorithms should be compared either under a precision-recall or ROC curve framework by varying certain internal parameters of the corresponding algorithm. Instead, the current results are shown on the default parameter settings given, for which the original authors consider that the algorithm performs optimally.

These quantitative evaluation results of reflection/rotation symmetry detection algorithms from this initial benchmark effort are alarming: namely, the best symmetry detection algorithm tested fails more than 70% of the time on all-type symmetry/images, and more than 80% on multiple reflection/rotation-symmetry detection in real images!

# 4

---

## Near Regular Texture (NRT)

---

Texture has been an intersecting research interest of computer vision, human vision, computer graphics and psychology for decades. There exist ample computational models and methods for texture analysis and synthesis. Directly related to computational symmetry is an alternative view of textures, that is in terms of textures' regularities. Using regularity as an essential measure, we can now view all textures in a spectrum of varying regularity (Figure 4.1 [164]). *Near Regular Textures* (NRT) are characterized in [164] as: “*The commonality behind the varied appearance of near-regular patterns is their strong tendency towards regularity or symmetry, even though the regularity is often imperfectly presented and intertwined with stochastic signals and random noise*” (Figure 1.1). Finding repeating structures in a scene attracts the attention of many since it has a direct relation to perceptual grouping in computer vision and texture sampling in computer graphics. Researchers encounter a great need for and difficult in dealing with NRTs, textures that not only repeat but also has an obvious underlying topological structure that relates all its texture elements.

The central theme of a sequence of recent papers on NRTs [95, 127, 146, 147, 148, 153, 163, 164, 171, 172, 204] is to address that real world NRTs (1) are important and ubiquitous (Figures 1.1

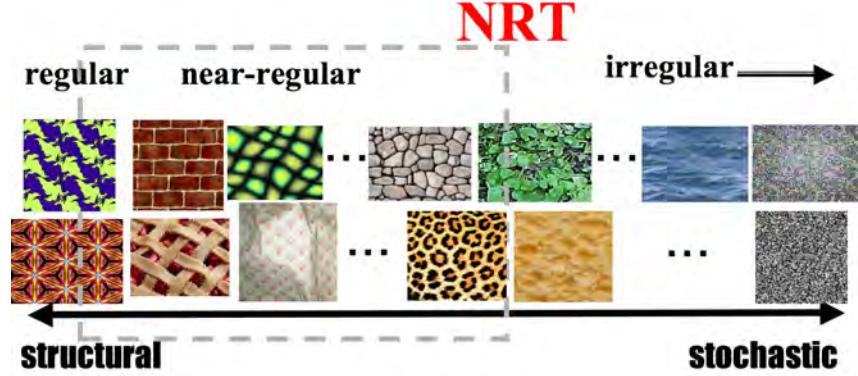


Fig. 4.1 A texture spectrum in terms of texture regularity [164].

and 4.1); (2) present severe computational challenges to the state of the art computer vision and computer graphics algorithms (Figure 4.2); and thus (3) deserve a special computational treatment for their analysis, synthesis and manipulation (Figure 4.3). Quantitative evaluations also show statistically significant results that human observers are indeed sensitive to the irregularities in the synthesized textures if they are inconsistent to input textures in terms of regularity [146].

#### 4.1 NRT Formalization

The term *near-regular texture* (NRT) is formally introduced by Liu et al. in [164] as

... a statistical distortion of a regular, wallpaper-like congruent tiling, possibly with individual variations in tile shape, size, color and lighting. Near-regular textures can depart from regular tiling along different axes of appearance, and thus could have (1) a regular structural layout but irregular color appearance in individual tiles; (2) distorted spatial layout but topologically regular alterations in color; or (3) deviations from regularity in both structural placement and color intensity. We call these **Type I**, **II** and **III** near-regular textures, respectively (Table 4.1).

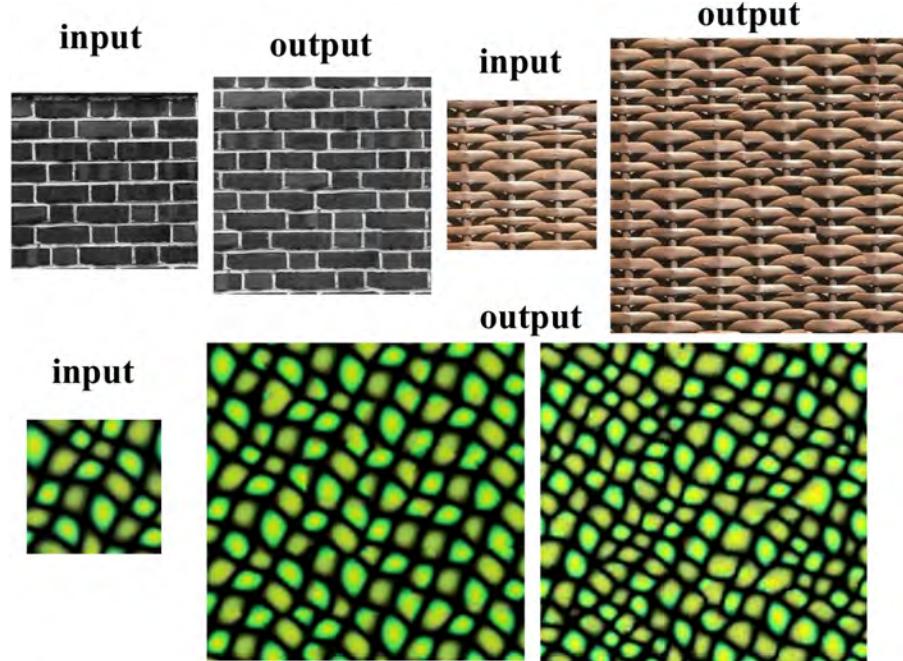


Fig. 4.2 NRT synthesis results of general texture synthesis algorithms, where the geometry (top row) and color (bottom row) regularities are not faithfully preserved in the synthesized results.

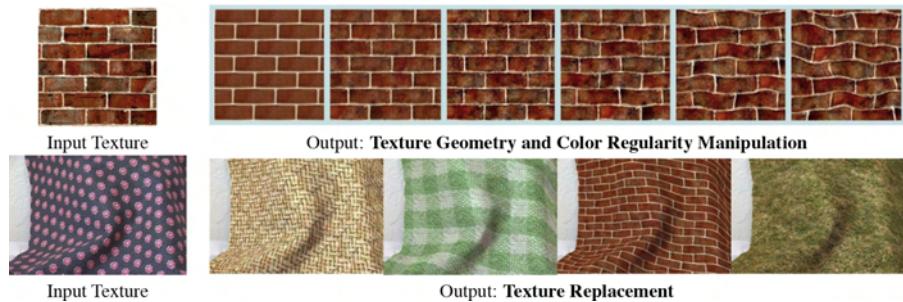


Fig. 4.3 Near-regular texture analysis and manipulation [164].

*Regular textures* are defined as wallpaper-like, congruent 2D tiling whose structural regularity can be completely characterized by the 17 wallpaper groups [84, 235]. The underlying lattice structure of each regular texture can thus be represented and generated by a pair of

Table 4.1. A categorization of near-regular textures [164].

Type	Geometry	Color	Symbols
0	Regular	Regular	GRCR
I	Regular	Irregular	GRCI
II	Irregular	Regular	GICR
III	Irregular	Irregular	GICI

linearly independent translations  $\vec{t}_1, \vec{t}_2$ . Under the translational subgroup of each 2D regular texture  $p_r$ , the smallest bounded region that produces simultaneously a covering (no gaps) and a packing (no overlaps) of the texture pattern on the 2D plane is called a *tile* [84]. A *near-regular* texture is defined as  $p = d(p_r)$  where  $d = d_{\text{geo}} \times d_{\text{light}} \times d_{\text{color}}$  is a pixel-wise mapping representing the composition of geometric transformations, lighting changes and color alterations, respectively. Using regular tiling [84] as our anchor point, we are able to define and capture these statistical departures from regularity using a multi-modal (geometry  $d_{\text{geo}}$ , lighting  $d_{\text{light}}$ , color  $d_{\text{color}}$ ), multi-dimensional mapping, which we call a *deformation field*.

For any near-regular texture  $p$ , there exists an underlying lattice  $L$  that is a geometric distortion of a regular lattice  $L_r$  from a regular texture  $p_r$ . Though there may be many potential regular lattices that a near-regular lattice  $L$  can deform to, there is a well-defined regular lattice  $L_r$  that has a minimum distance from  $L$  [122, 159]. The *Geometric deformation field*  $d_{\text{geo}}$  is defined as a function that maps  $L$  to  $L_r$  and warps all the input texture pixels to a geometrically regular texture.

As it turns out, the idea of formulating a given texture as a deformed version of a regular texture is not new. In 1976, Zucker proposed a texture model [308] for the so-called *structural texture* models with texture primitives. The texture primitive placement rule is based on a graph that is isomorphic to a set of regular tessellations (only three types), and eight star-shaped patterns. This class of algorithms was considered “*in general, limited in power unless one is dealing with very regular textures*” [263]. At the time, the straightforward link to the translation subgroup of wallpaper patterns with a pair of linearly independent generators was not made, nor did an effective algorithm existed that could handle both topological regularity/invariance and

geometric/photometric deformations/variations in real images (Section 4.4). The recent development, by making the connection between regular textures and wallpaper patterns (and wallpaper groups) explicitly, has led to the generalization to a single degree-4 graph for all NRTs and computational efficiency for real world applications [159, 95, 147, 148, 202].

## 4.2 NRT Quantification

A pair of quantitative regularity measurements is proposed in [164] to measure and compare the *degrees of regularity* of various NRTs (Figure 4.4). They are **G**eometry regularity and **A**ppearance regularity scores or the so-called *GA-scores* (Figure 4.4). Figure 4.4 demonstrates some sample Type I, II, and III near-regular textures placed in the 2D GA-score regularity space. This pair of measures is used subsequently to evaluate:

- (1) the synthesis results for NRT input [140] (Figure 4.5) [164];
- (2) the appearance model of dynamic near-regular texture tracking [147, 205];
- (3) the quality of automatically detected lattices in [95, 204]; and
- (4) the clustering of grid cells in rats' (Figure 4.10) [32].

### 4.2.1 Measuring the Firing Fields of Grid Cells

Surprisingly strong hexagonal symmetry is observed in the the firing fields of grid cells found in the rat dorsolateral medial entorhinal cortex [88] (top of Figure 4.10). Chastain and Liu apply computational wallpaper group theory [159] and the NRT regularity quantification method [164]. They demonstrate qualitatively that the firing fields have a nearest symmetry group of p6m (one of the 17 wallpaper groups) and, quantitatively, the cells cluster around two types of regularities (bottom of Figure 4.10): one is more regular in geometry and intensity (Type I NRT), and one tends to be less regular in both (Type III NRT). These analysis of the patterns are important because they may serve as indicators of the associated cells function in navigation, with the regular set acting as a reference for the deviating set which indicates unique orientations [32].

### Geometric regularity:

$$G = \sum_{i=1}^{N_i} \frac{(l_i - \|\vec{r}_1\|)^2}{\|\vec{r}_1\|^2} + \sum_{j=1}^{N_j} \frac{(l_j - \|\vec{r}_2\|)^2}{\|\vec{r}_2\|^2} + \sum_{k=1}^{N_k} \frac{(l_k - \|\vec{r}_1 + \vec{r}_2\|)^2}{\|\vec{r}_1 + \vec{r}_2\|^2} + \sum_{m=1}^{N_m} \frac{(l_m - \|\vec{r}_1 - \vec{r}_2\|)^2}{\|\vec{r}_1 - \vec{r}_2\|^2}$$

**Appearance regularity:**  $A = \frac{1}{m} \sum_{i=1}^m \text{std}([T_1(i), T_2(i), \dots, T_n(i)])$  where  $T_i$  is a reshaped tile (a column vector) of a Type I texture,  $m$  is the number of pixels within a tile, and  $n$  is the number of tiles.

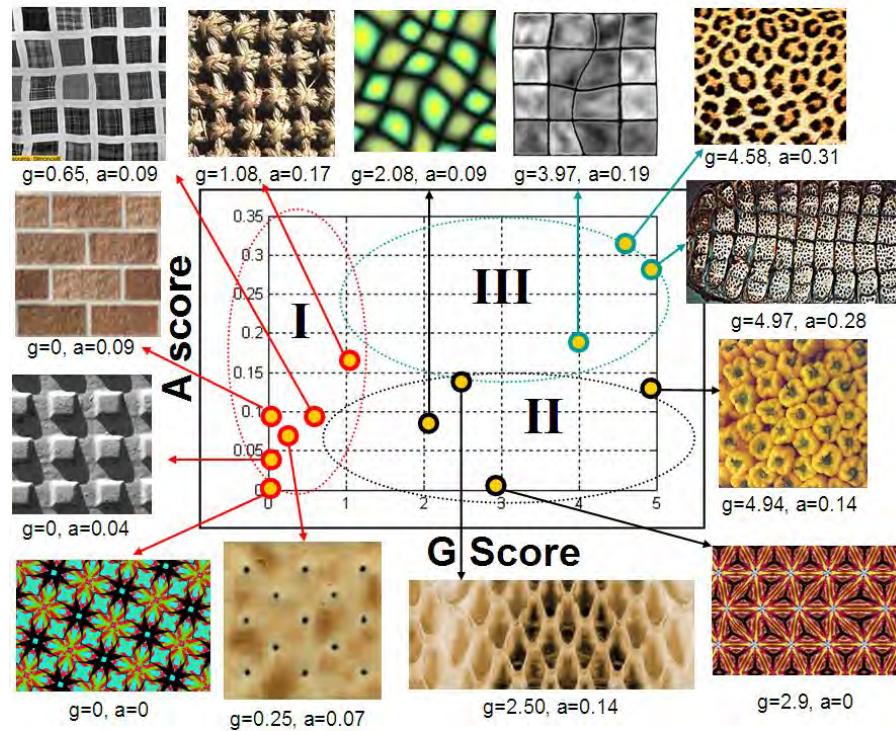


Fig. 4.4 Regularity quantification: geometry (G) and appearance (A) scores.

## 4.3 NRT Applications

### 4.3.1 Texture Replacement in Real Images [262]

Tsin et al. [262] are among the first to take advantage of planar texture regularity to estimate a statistical lighting model from a real photo, automatically segment out the texture in the image and achieve texture

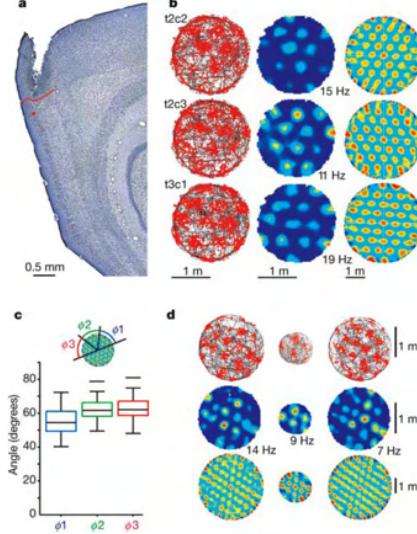


Figure 1 | Firing fields of grid cells have a repetitive triangular structure.

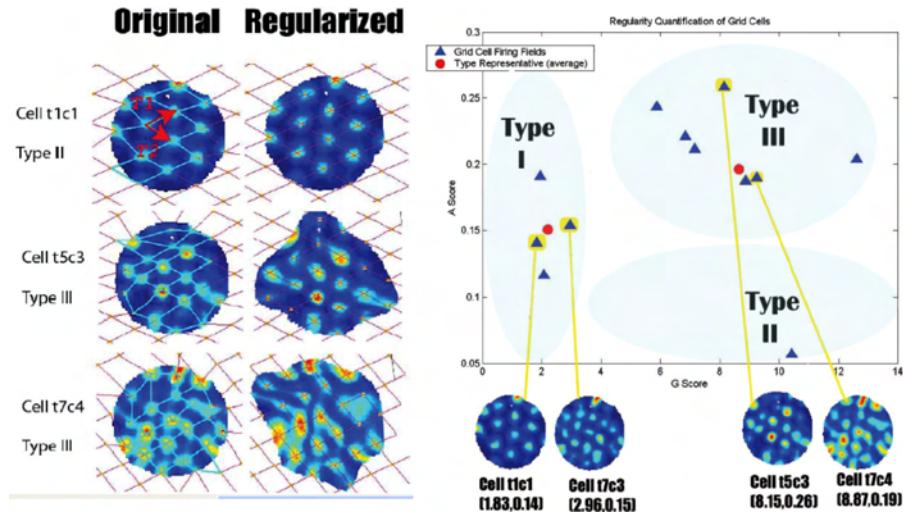


Fig. 4.5 Top: courtesy of Nature [88]. Bottom: clustering of the firing fields from the rats grid cells [32] using the GA-score developed in [164].

replacement with similar lighting effects (Figure 4.6). This method is further extended in [164] to NRT synthesis, replacement and analogy for non-planar NRTs (Figure 4.9).

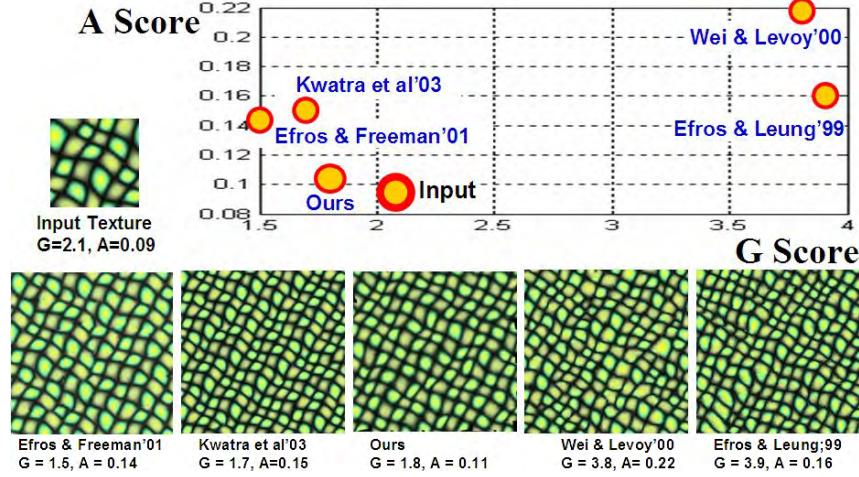


Fig. 4.6 Figure 2.3 from [164], using GA-scores to compare, quantitatively, the output of one of the NRT synthesis results from different texture synthesis algorithms. This result shows that the near-regular alterations in color (...yellow, green, yellow, green, ...) of the input texture is more faithfully preserved by the texture synthesis method proposed in [164] (Figure 4.8 when gain = 1): in the quantified geometry-appearance regularity space represented by the  $G$ - $A$  regularity scores, its distance to the input texture is the shortest.

#### 4.3.2 NRT Synthesis and Manipulation [164, 171, 172]

The authors of [147, 148, 164, 171, 172] illustrate that in general the popular, state of the art texture synthesis algorithms cannot handle faithful reproduction of input NRTs (Figure 4.2). Therefore, the authors proposed a set of texture analysis and synthesis algorithms that respect texture regularities. In [171, 172], an NRT synthesis algorithm for Type I NRTs is proposed. The idea is to add an analysis step before synthesis to locate the  $t_1$  and  $t_2$  generators while during synthesis the choice of ‘tiles’ is made randomly such that the geometric regularity and the randomized appearance are both preserved (Figure 4.7). In [164], a complete set of NRT synthesis algorithms is proposed for Type I, II and III NRTs. The idea is to first extract the geometric, lighting and color *deformation* fields from an input NRT, then treating these deformation fields themselves as ‘textures’ for synthesis, and finally the synthesized deformation fields are mapped back to their functional (as opposed to their appearance) side of the dual role and applied to regular textures to accomplish the deformation process (Figure 4.8).

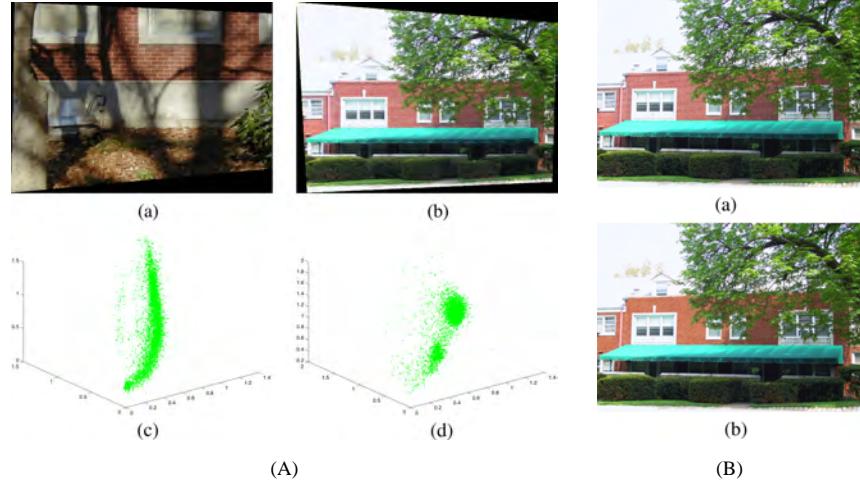


Fig. 4.7 (A) Estimated lighting distributions of two scenes taken by two different cameras. (a) A scene with many soft shadows on the textured region. (b) An image whose textured regions have little soft shadows. (c) Lighting samples estimated from (a). They are distributed along a curved structure. (d) Lighting samples are concentrated on two clouds in the lighting space. [262]. (B) Texture replacement in real photos [262].

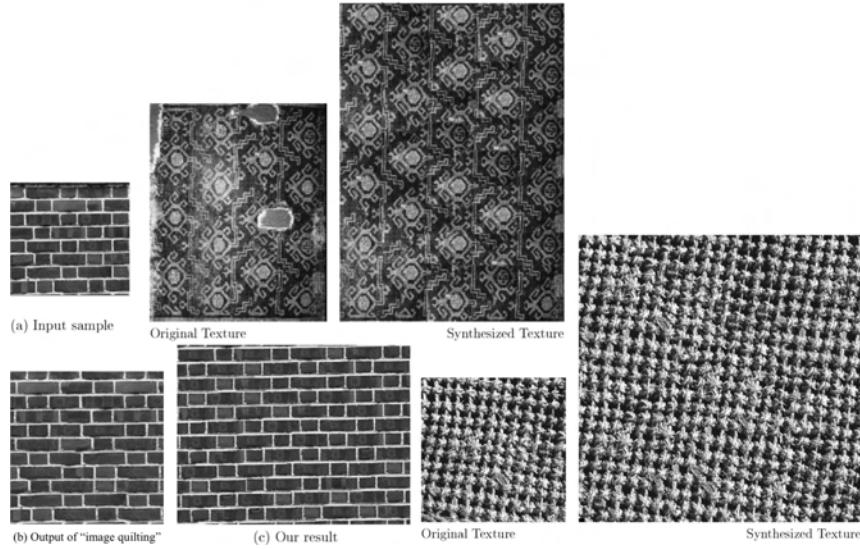


Fig. 4.8 Few samples results from [172] where Type I NRT is analyzed and synthesized. The algorithm respects both the regular spatial arrangement and the random visual effect of the input texture — an example of Type I NRT.

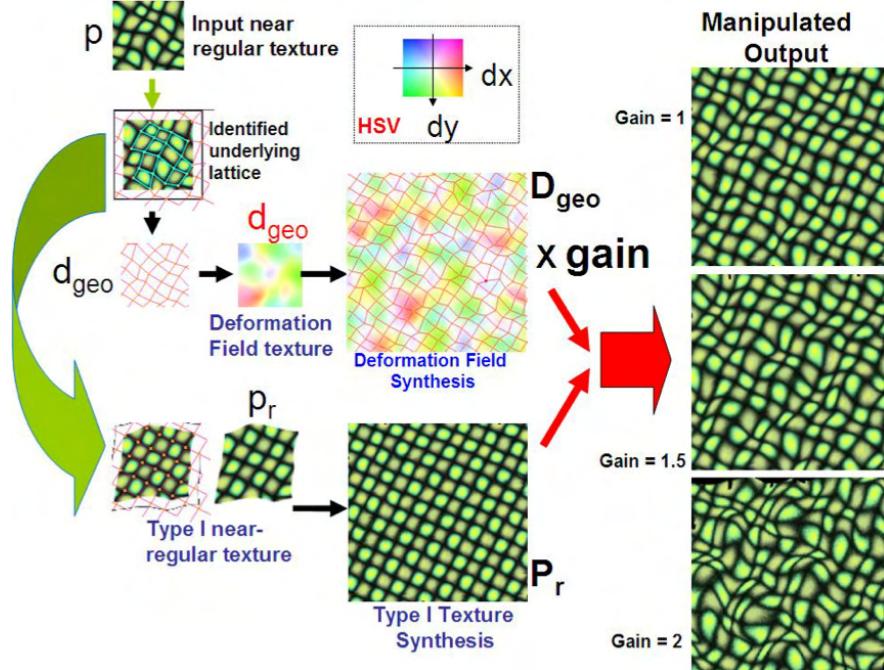


Fig. 4.9 NRT analysis, synthesis and manipulation algorithm for Type II/III NRTs, and results from [164].

Using the fact that the geometric deformation field is capable of warping an arbitrary near-regular texture back to a Type I texture, by indicating a portion of the texture that is regular (after rectification) and *canonically* lighted, the proposed algorithm simultaneously finds the lighting map (shadows, occlusion) while segmenting out the texture region in the image. The algorithm for extracting the lighting deformation field works as follows: (1) straighten the lattice of the input NRT using the geometric deformation field  $d_{geo}$ ; (2) apply Tsin et al. [262] algorithm for lighting map extraction in the plane; and (3) apply the inverse geometric deformation field to map the lighting deformation field back to the original input texture.

#### 4.3.3 NRT for Gait Analysis [158, 159]

Human and animal gaits are known to be approximately periodic [39, 46, 149, 244]. Therefore, the gait patterns generated by human



Figure 12: Given a near-regular texture (Type III), we show its texture synthesis and then texture replacement results [164].



Fig. 4.10 NRT synthesis and replacement results from [164].

or animal motions appear to be *near regular*. The authors of [158, 159] investigate two types of spatiotemporal image representations of gait. The first is a wallpaper pattern-like NRT composed of the correlation scores between all pairs of video frames  $i, j$  from a gait sequence. Figure 4.11 shows that the gait patterns of a running dog and a walking human have different symmetry groups,  $cm$  and  $p4m$ , respectively. This result reinforces that human gait is more bilaterally symmetric than a dog's gallop gait (similar to a horse), and thus symmetry of the gait patterns serves as a discriminative feature when shape cues are hard to extract. The second type of spatiotemporal representation of gaits is a frieze pattern. Figure 4.12(A) illustrates how a frieze pattern is generated from a time sequence of projections of 2D silhouettes along the  $X$  or  $Y$  axis. Note that each frieze pattern shown in Figure 4.12(A) is a 1D (along the time axis) near-periodic pattern in a 2D space: time versus  $X$ -axis for column projection  $f_C$  and time versus  $Y$ -axis for row-projection  $f_R$ . Figure 4.12(B) shows frieze pattern variation of  $f_C$  for a walking avatar viewed from different orientations. Figure 4.12(C) shows the trajectory and  $f_C$  frieze pattern of a 30-second human walking sequence, in which one can observe the variations of frieze pattern  $f_C$  with the change of the viewing angles.

This frieze pattern based gait analysis idea was explored further in [158], where gait angle estimation is carried out using  $f_C$  frieze

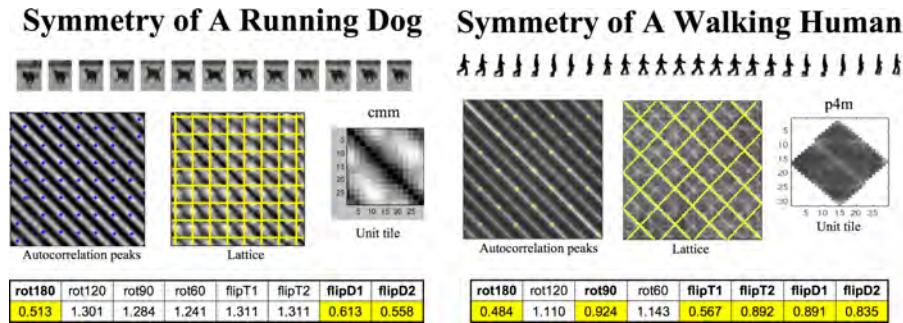


Fig. 4.11 The symmetry groups of human walking gait ( $p4m$ ) and dog running gait patterns ( $cmm$ ) differ. Human is more symmetrical than a dog's gait pattern. Dog running sequence courtesy of [46].

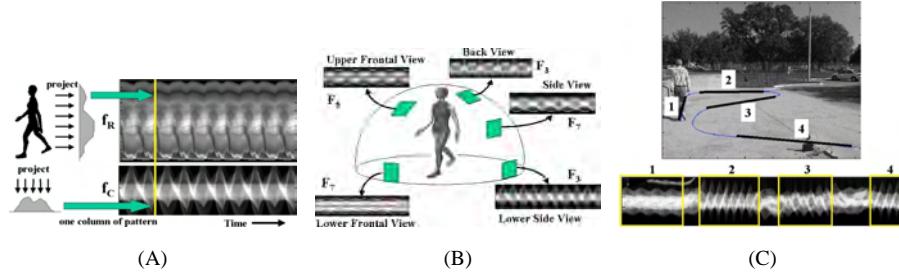


Fig. 4.12 (A) Spatio-temporal gait representations are generated by projecting the body silhouette along its columns and rows, then stacking these 1D projections over time to form frieze-like patterns  $f_C, f_R$  repeating along the time dimension. (B) Observation #1:  $f_C$  of a humanoid avatar exhibits a wide variety of symmetric structures when viewed from different orientations. (C) Frieze pattern extracted from a 30-second long human walking sequence. Observation #2: the symmetry group variations of  $f_C$  of a human echo those of the avatar when viewed from different orientations.

patterns while human identification from gaits observed from similar orientations is carried out using  $f_R$ . Classification rates range from 81% to 100% when training and testing on four different types of human gaits: slow walk, fast walk, inclined walk and walk holding a ball.

#### 4.4 Discovery of NRTs in the Real World: Deformed-Lattice Extraction

Texture analysis has been a long-standing and surprisingly difficult problem. Interest in visual texture predates computer vision, going back at least to Gibson [72], who pointed out its importance for the perception of surface orientation (i.e., shape-from-texture). Later, Bela Julesz developed a theory of human texture discrimination based on matching  $N$ th order texture statistics [108]. Both researchers speak of a **texture element** (texel), as a unit of texture that is repeated. The one exception would be in talking about texture discrimination which is the context in which Julesz uses the term *texton*.

Ample evidence can be found that real world repeated patterns (textures) are not composed of merely random collections of individual texture elements. Instead, they often exhibit some geometric, topologic and/or statistical regularities among the elements. The causes

of these inter-relations are deeply rooted in the original design intentions, natural formations or biological laws. Thus, instead of treating texture elements as isolated individuals, the underlying topological lattice structure of a near-regular texture (NRT) [164] is sought after to gain insights of the textures formation and robust manipulation. In particular, dynamic (moving) NRTs present additional challenges to classic tracking algorithms, while can be handled with considerable robustness when tracked as multiple targets connected by one or more underlying lattice structures [147, 150].

A wide range of image manipulations become possible as shown in Figures 4.9, 4.10, 4.11, 4.12, once the underlying lattice structure (geometric deformation field) of an NRT in a real image/video is found either interactively [164] or automatically [95, 148, 202, 204].

There are several existing algorithms for repeated pattern/texture analysis [68, 137, 147, 159, 164, 232, 251, 264]. We distinguish these algorithms from an automatic deformed-lattice detection algorithm like Hays et al. [95] and Park et al. [202, 204] because (1) they [68, 137, 173, 251] place more emphasis on the appearance of individual texels rather than the spatial relationships among the texels, thus no lattice is detected; (2) for those algorithms where a lattice is extracted, their initialization is not fully automatic [147, 164, 251] or no significant geometric deformations are allowed [89, 159]; (3) Schafalitzky et al. [232] and Turina et al. [264] assume that the texture has undergone a global projective transformation without significant local geometric distortions.

Previous work on discovering repeated texture elements can be viewed as two extremes: one focusing on individual texels (or local neighborhoods) with no regard to their global organization or regularity [138, 173]; the other placing very strict requirements on the overall image structure (i.e., planar regular texture under global perspective transformation) [232, 264]. The former can handle a wide range of textures but at the cost of relying entirely on the appearance of individual texels. The latter uses the overall arrangement of the texels to its advantage, but is only applicable to a limited, globally transformed textures.

Though many previous attempts have been made in finding the underlying lattice structure of a given wallpaper-like structure [145, 159, 308], algorithms for automatic detection of deformed-lattices from non-segmented images is rare. Hays et al. [95] are perhaps the first to develop a completely automated lattice extraction algorithm for an arbitrarily distorted (local and global) near-regular texture in an image without segmentation. Meanwhile Lin and Liu [147, 148] developed the first deformed-lattice tracking algorithm for *dynamic NRTs* (Section 4.5) where the lattice in the first frame of the video is initialized semi-automatically: the user inputs three points indicating a pair of  $t_1$  and  $t_2$  vectors, then the algorithm discovers the whole, mostly deformed-lattice automatically under a spatial tracking framework.

Two important observations on the lattice are worth noting: (1) *topological consistency* — the lattice topology for all different types of 2D repeating patterns (wallpaper patterns) remains the same: quadrilateral; (2) *topological invariance* — while NRT may suffer large geometry and appearance variations locally or globally, its lattice topology remains invariant. Therefore, automatically discovering the lattice of an NRT is a well-defined and conceptually feasible task.

Understanding and explicitly searching for lattice structures in real world textures enables more powerful texture analysis algorithms that are aware of the global structure of a texture thus less dependent on specific, local image features [138, 173]; and it leads to a much broader set of textures than could be covered before in [232, 264].

The key insight for translation symmetry detection in NRT (Sections 4.4.1 and 4.4.2) and dynamic NRT (Section 4.5) is to capture the underlying quadrilateral lattice of a 2D texture, generated by translating its texture element (texel) using the two generating vectors:  $t_1$  and  $t_2$  (bounding a texel region) [159]. Real world textures can be considered as globally and locally deformed wallpaper patterns (NRTs) [164], where  $t_1$  and  $t_2$  vectors become a function of location — differing from a regular pattern where  $t_1$  and  $t_2$  remain constant across the whole pattern. Therefore, translation symmetry detection from real-world images is equivalent to finding the underlying deformed-lattice of a piece of texture in an image.

#### 4.4.1 Regularity Discovery in Real Images: A Spectral Method [95]

In this work, the authors formulate lattice discovery as an instance of the general *correspondence problem*. Different from the usual correspondence problems in computer vision when two disjoint sets of features require a one-to-one assignment (e.g., stereo correspondence, non-rigid shape matching, etc.), this assignment of neighbor relationships is from a set of potential texels *to itself*, with constraints to avoid self-assignments and other degeneracies. By phrasing lattice finding as a correspondence problem the authors can leverage powerful matching algorithms to reason *globally* about lattice regularity.

Finding texture regularity is inherently a non-local problem. It requires global consensus, thus assignments that are geometrically consistent with each other are encouraged, leading to a higher order problem. Finding the globally optimal assignments under second-order constraints is an instance of the Integer Quadratic Programming problem and is NP-complete. The authors therefore adapt an alternative method to approximate the optimal assignments for higher order correspondence based on a spectral method proposed in [136] given its speed and simplicity.

Two semi-independent pairwise correspondence assignments are carried out, resulting in each texel being paired with two of its neighbors along two directions (one at a time), which turn out to be precisely the two translation symmetry subgroup generating vectors  $t_1$  and  $t_2$ .

The lattice discovery algorithm proceeds in four stages: (1) *Proposal of texels*, in which new candidate texels will be proposed based on an interest point detector, correlation with a random template, or a lattice from a previous iteration. (2) *Lattice assignment*, in which potential texels will be assigned  $t_1$  and  $t_2$  neighbors based on their pairwise relationships to each other as well as higher-order relationships between assignments. (3) *Lattice refinement*, in which the assignments are interpreted so as to form a meaningful lattice and discard the outlier texels. (4) *Thin-plate spline warping*, in which the texture is regularized. The algorithm iterates through these four stages based on several random initializations and pick the best overall lattice. These four stages

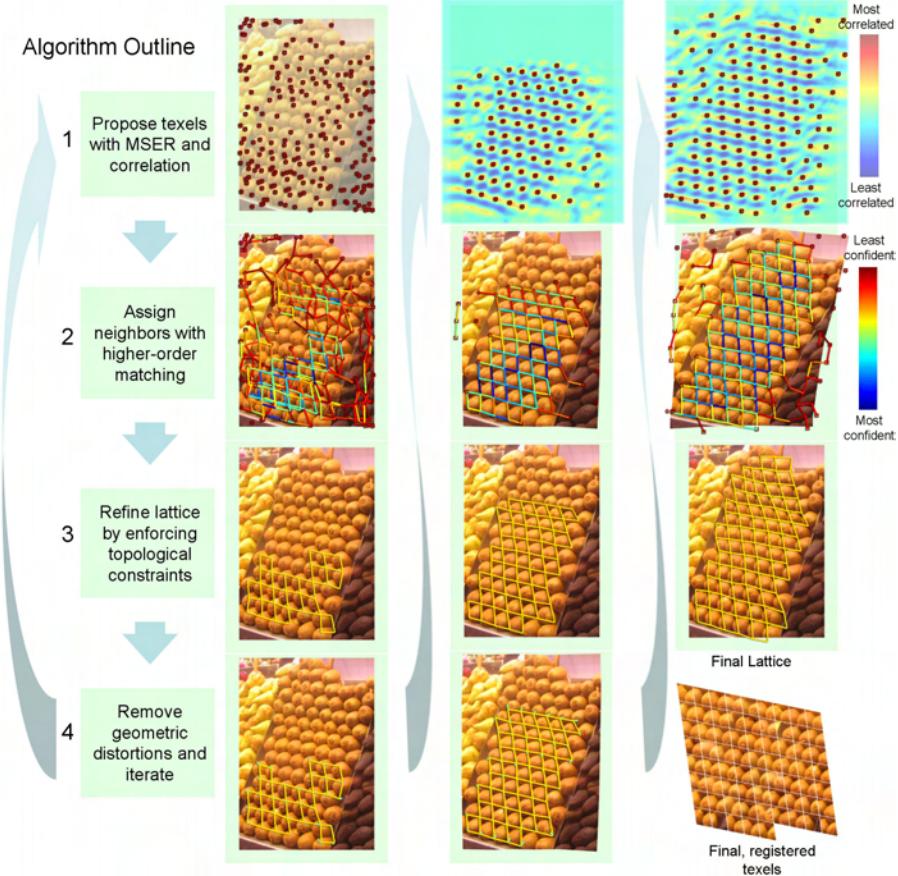


Fig. 4.13 Flow chart of the proposed lattice extraction algorithm (from [95]). The leftmost column shows the first iteration of the algorithm after initialization from the maximally stable extremal region interest point detector [184]. The middle column is the second iteration and the rightmost column is the final iteration.

are visualized in Figure 4.13 with each column representing a single iteration through these four stages. Formulating the lattice detection problem as a *higher order correspondence* problem adds computational robustness against geometric distortions and photometric artifacts in real images. Figure 4.14 shows the lattice detection results from a set of real world images.



Fig. 4.14 Input and extracted lattice pairs (from [95]).

#### 4.4.2 Regularity Discovery in Real Images: A Graphical Model Approach [204]

Although the deformed-lattice detection algorithm proposed by Hays et al. [95] produces impressive results (Figure 4.14), there are several limitations preventing its wider applicability. First, local correlation-based peak finding is used as a last resort for finding points of interest, which is both time consuming and sensitive to noise, occlusion and transformation discontinuity in the image. Second, the method is based on finding the eigenvalues of a  $n^2 \times n^2$  sparse matrix ( $n$  is the number of potential texture elements), which is cumbersome computationally. Third, the algorithm only examines one of the  $t_1$  and  $t_2$  vectors at a time, and is thus less robust against misleading repetitions and prone to wasting time on interest points that do not lead to legitimate  $t_1$  and  $t_2$  neighbors. The authors of [204] proposed an alternative method that overcomes these weaknesses.

The novelty of [204] is to formulate the extraction of an underlying deformed-lattice as a spatial, *multi-target tracking* problem using a new and efficient Mean-Shift Belief Propagation (MSBP) method [205] as an inference tool on an Markov Radom Field (a degree-4 graph). Compared to [95, 204] has shown effectiveness in (1) incorporating higher order constraints early-on to propose highly plausible lattice generator pairs; (2) growing a lattice in multiple directions simultaneously instead of one at a time sequentially; and (3) achieving significant speed-up and accuracy improvement over [95]. The paper [204] also differs from [147, 148] in that the MRF model proposed is used for automatic deformed-lattice detection via spatial tracking, while the first-frame lattice detection in [147, 148] is done semi-automatically and the MRF model is primarily used for tracking the dynamic NRTs.

A Markov Random Field (MRF) specifies a factorization of the joint distribution of a set  $\mathbf{X}$  of random variables. An MRF can be represented as an undirected graph  $\mathbf{G} = (\mathbf{N}, \mathbf{E})$ , where each node in  $\mathbf{N}$  represents a random variable in set  $\mathbf{X}$  and each edge in  $\mathbf{E}$  represents a statistical dependency between random variables in  $\mathbf{X}$ . In [204], the value of a random variable is the location of a texture element, and the spatial dependency between variables represents how one element's

position constrains the positions of its neighboring elements in the pattern. Since the lattice indicates a repeated pattern, the appearance of each of the texture elements can be described by a single reference appearance model. Determining the location of an element thus requires combining these two sources of information: the compatibilities of the texture element locations and their appearance. Determining the location of one particular element given estimated positions of all other elements would be infeasible if it requires brute force evaluation of the marginal distribution of that single random variable's value, since that will lead to  $O(n \times n^{k-1})$  computation time where  $k$  is the number of nodes in the graph and  $n$  is the size of the hidden variable space (for simplicity we can think of it as 2D location in the current example). Thus determining where each of the elements is in the pattern requires  $O(kn^k)$  computation time. Fortunately, the joint probability over the pattern state  $x$  and image measurement  $z$  in an MRF can be factored as

$$p(x_1, \dots, x_N, z_1, \dots, z_N) = \prod_{(i,j)} \psi(x_i, x_j) \prod_s \phi(x_s, z_s) \quad (4.1)$$

where  $\psi$  and  $\phi$  are functions specifying pairwise compatibility (spatial constraints between elements) and joint compatibility (appearance similarity of elements at given image locations), respectively. The belief propagation algorithm takes advantage of this factorization to perform inference on the graph efficiently. Cost of computation for estimating the state of all NRT elements is reduced from  $O(kn^k)$  to  $O(kn^2)$ . However, if the hidden variable state space is large, BP can still be very expensive. Figure 4.15 illustrates a simplified flowchart of the overall algorithm for deformed lattice detection.

Quantitatively, the detection rate of Lin and Liu [147, 148] is  $20 \pm 21\%$ , of Hays et al.<sup>1</sup> is  $47 \pm 38\%$ , and of [204] is  $81 \pm 19\%$ . The detection rate is computed by the ratio of the number of detected texels

---

<sup>1</sup>The algorithm of [95] includes an element of randomness; it runs for several iterations, and takes the best result according to a modified A-score. A-score, originally introduced in [164], is the average per-pixel standard deviation among the final, aligned texels. The modification is the inclusion of  $\sqrt{n}$  in the divisor in order to bias the A-score toward more complete lattices [95].

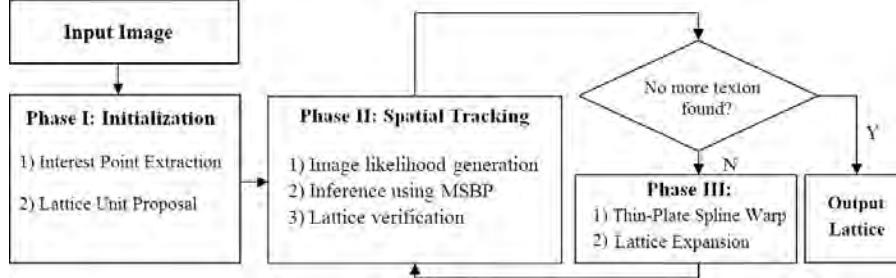


Fig. 4.15 Flowchart for the deformed lattice detection algorithm proposed in [204]. There are two main components: initialization and spatial tracking, plus a transition phase for spatial warping (rectifying the lattice into an integer coordinate system).

over the number of ground truth texels. The ground truth is manually obtained by two human coders. The average ratio of run times of the Hays' algorithm [95] versus [204] is  $10.66 \pm 9.6$ . The paper [95] failed on five images which are excluded when computing average time ratio. Running time ratio is defined by the ratio of the time used by [95] over the time used by [204] to detect the lattice on each of the 32 test images (Figure 4.16).

#### 4.4.3 See-Through NRTs: Image De-fencing [153]

This work delivers both a positive and a negative message. On the one hand, it demonstrates the promise of translation symmetry (or a deformed version of it) for machine perception of *layers*, and a relevant graphics application: successful *image de-fencing* examples are obtained using the automatically discovered foreground mask as the *holes* for an state of the art in-painting algorithm (Figure 4.17). On the other hand, the paper reports more negative results than positive ones, by demonstrating the limits of both the lattice extraction algorithm [95] and the in-painting algorithm [45].

Assuming that an automatic NRT extraction algorithm can identify accurately the fence-like structure with approximate translational symmetry in a real image, the authors implement a foreground/background classifier. In conjunction with texture-based inpainting [45], a mesh-occluded background (as much as 53% occluded area) can be recovered to achieve the effect of image and photo de-fencing (Figure 4.17).

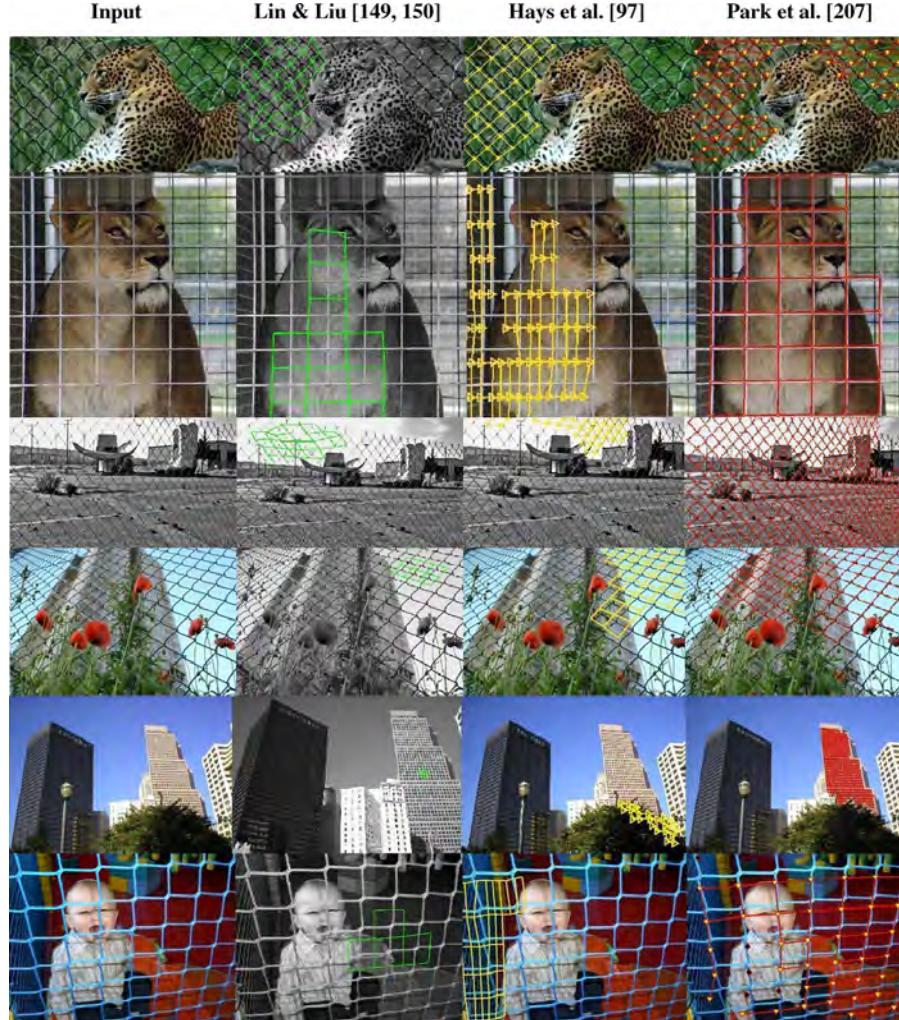


Fig. 4.16 Figure 1.4 from [204]: Sample lattice detection results: the input images (leftmost), the results of [147, 148] (second left), the results of [95] (second right), and our results (rightmost). Note that the comparison with [147, 148] is not a fair comparison since they are designed to start interactively. For a complete set of images tested, see our website <http://vision.cse.psu.edu/MSBPLattice.htm>.

The algorithm consists of three distinct phases — (1) automatically finding the skeleton structure of a potential frontal layer (fence) in the form of a deformed-lattice, (2) separating foreground/background

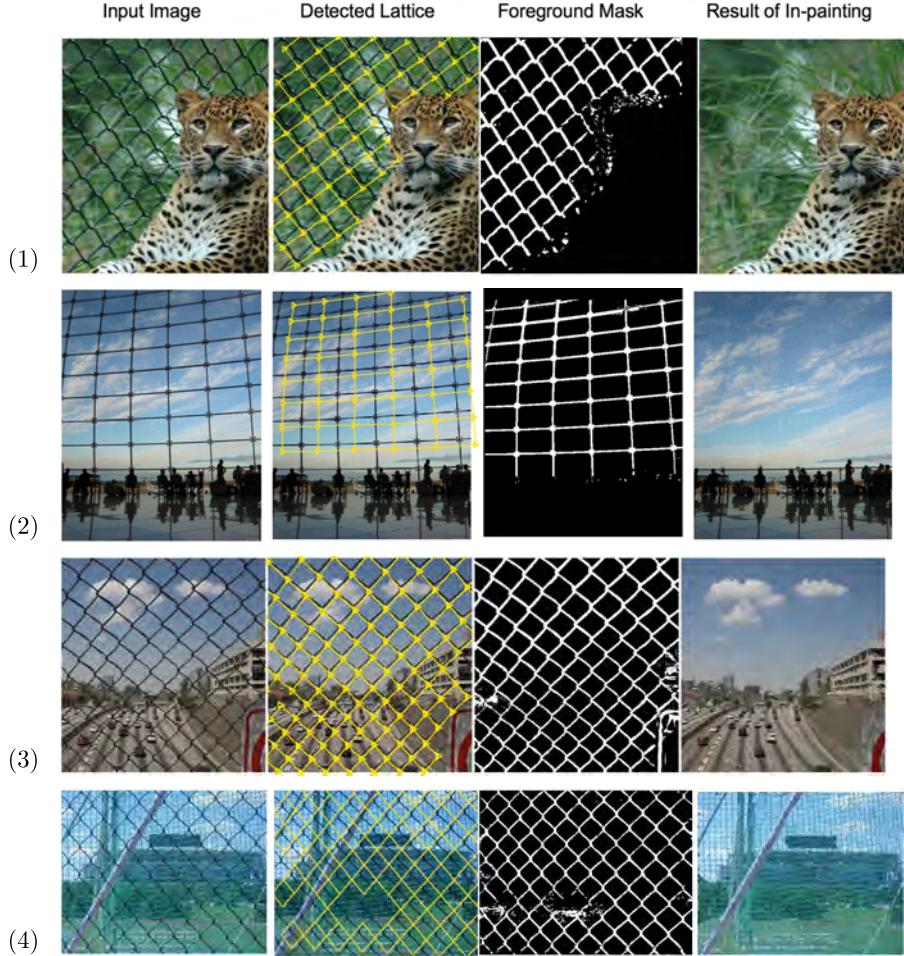


Fig. 4.17 Figure 2.3 of [153]. Several relatively promising image de-fencing results demonstrate the effectiveness of the proposed, translation symmetry-based detection-classification-inpainting method. The lattice detection in (4) is imperfect, thus a piece of the fence remains after inpainting.

layers using appearance regularity, and (3) occluded foreground inpainting to reveal a complete, non-occluded image. Each of these three tasks presents its own special computational challenges that are not encountered in previous, general image de-layering or texture inpainting applications.

This paper reports results to show the limitations of both the lattice detection algorithms and the inpainting algorithms (<http://vision.cse.psu.edu/defencing.htm>) by pointing out:

Automatic lattice detection from real images [95] has met some serious challenges in this application: detection of see-through, near-regular structures from adverse background clutters. We have observed the failure cases often are accompanied by sudden changes of colors in the background (e.g., peacock, moose); obscuring objects in front of the fence (e.g., building), and irregular background geometry.

... contrary to our initial expectations, we observe that the mesh-like regions are actually more difficult to texture fill than large, circular regions of similar area. This is because the mesh-like regions are wide enough to show errors with incorrect structure propagation, but they have dramatically larger perimeter than a single large region and thus there are many more structures which need to be correctly propagated and joined. ... The high ratio of foreground area to background area as well as the fragmented background source textures present special challenges for existing inpainting methods.

These results provide a new NRT application in foreground/ background perception, and concrete evidence on the limitations of state of the art lattice detection and in-painting algorithms, thus should inspire further research in both areas.

#### 4.4.4 NRTs for Architectural Images [127, 240]

Algorithms designed specifically for the analysis of different types of NRTs have appeared in recent computer vision and computer graphics conferences, for example, for architectural images/models [127, 185, 207, 233, 234, 240, 266, 283], or for fabric patterns [89]. The two alternative methods for lattice extraction from photos of

architectural structures, where global perspective distortions represent the major form of *deformation*, are detailed below.

#### 4.4.4.1 Lattice Detection for Automatic Geo-tagging (CVPR 2008) [240]

The results from the affinely skewed symmetry group analysis [156] and the motif analysis [159] are applied recently for automatic geo-tagging [240] (Figure 4.18) where each facade of the building (from different view/time, image or 3D models) is rectified to its ‘canonical’ representation and the unique set of motifs are matched without ambiguity (Figure 4.18). From this experimental set of architecture photos of Atlanta, Georgia, the authors observed that the most frequently occurring symmetry groups among different building facades were *pmm* and *p4m* (Figure 2.4).

The authors of [240] use a variation of the RANSAC-based planar grouping method introduced in [231] to detect perspectively distorted lattices of SIFT feature points [176], which allows the identification of the two basic translation vectors ( $t_1$  and  $t_2$ ) of the underlying repeated wallpaper pattern (Section 2). Given an image of a scene containing multiple buildings, the goal is to detect all the repeated patterns (or lattices) present in the image (Figure 4.19).

The main *lattice extraction* procedure is as follows: (1) clustering the SIFT features into  $N$  groups based on their appearance descriptor;

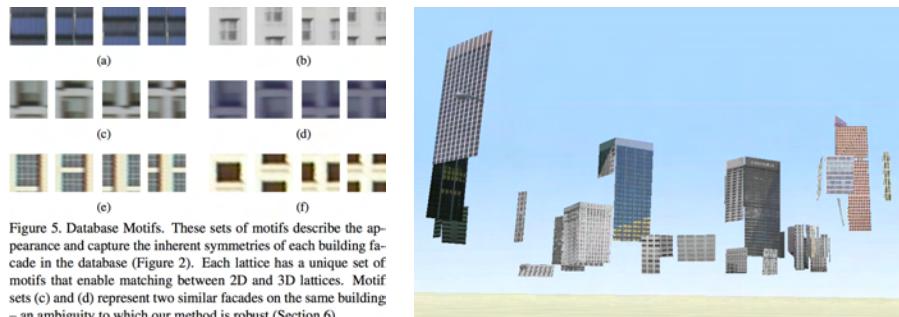


Fig. 4.18 Figures from [240]. Left: the canonical motifs [156, 159] used for matching. Right: one of the 3D reconstructed results using the skewed symmetry group idea developed in [156] to find correspondences from different views of the architectural facades.

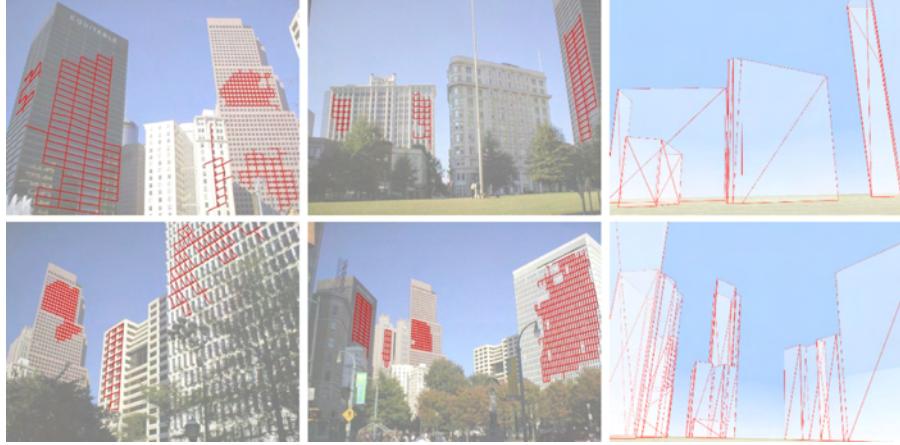


Fig. 4.19 Figure 2.3 of [240]. Multiple lattices on different facades are detected from different views of the same buildings. These lattices are subsequently matched with each other, and with a 3D model of the city (rightmost) using skewed symmetry groups as a theoretical and computational basis (Section 2.2.1).

(2) randomly sampling four points (within each cluster)  $\{a, b, c, d\}$  and computing the homography  ${}_lH_i$  which maps these four points from image space into the lattice basis  $\{(0,0), (1,0), (1,1), (0,1)\}$ ; (3) transforming all remaining points from image space into their equivalent lattice positions via the homography  ${}_lH_i$ , and count as an inlier each point whose lattice space coordinates are within some threshold of an integer position  $(i, j)$ . This method takes advantage of the repeated nature of such NRTs. Indeed, if the four chosen points  $\{a, b, c, d\}$  really do define a *tile* corresponding to the generating region of a wallpaper group, the accumulated effect should be reflected by the repetitions of identical SIFT features.

#### 4.4.4.2 Analysis of Building Textures for Reconstructing Partially Occluded Facades (ECCV 2008) [127]

The goal of [127] is to understand and manipulate images of buildings. The primary motivation is to automatically detect and seamlessly remove unwanted foreground elements from urban scenes (Figure 4.20). The authors draw the analogy that building facades are often examples

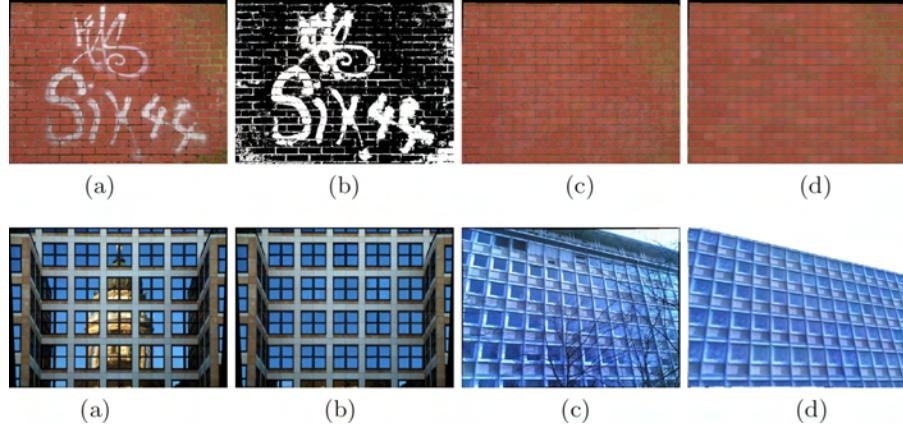


Fig. 4.20 Top: Figure 1.1 of [127]. Virtual graffiti removal. (a) Original image; (b) automatically detected foreground pixels; (c) tile-aligned exemplar-based inpainting; (d) Eigenimage reconstruction. Tiles with  $>25\%$  outliers were sampled. While there is some loss of detail in (d), many local characteristics are retained. Bottom: Figure 2.3 of [127]. Foreground removal by eigenimage reconstruction for tile aligned images (a) and (c).

of Near-Regular Textures, and show that discovering these textures could provide valuable insight into the rest of the facade. The method is designed specifically for extracting 2D lattice from building images using an MRF/MCMC-based approach.

Similar to [153], and given the assumption that the background is visible in a majority of the tiles, the foreground elements or reflections are treated as outliers of the building tile pixel values in corresponding locations. The median absolute deviation (MAD) [49] is used to assess which pixel values vary enough across the tiles to arouse suspicion of foreground. The reconstruction of the background is carried out by tiling sampled from the PCA space [164] as well as using existing inpainting algorithms. The authors observed that “the more complex the tile interior is, the less effective inpainting would be in avoiding geometric and photometric artifacts.”

It is interesting to note that, though both works use the concept of NRT and both deal with the foreground/background separation problem from a single photo image, the authors of [127] are trying to reveal the background that is NRT (Figure 4.20) by NRT texture

synthesis or inpainting of NRT, while Liu et al. [153] are trying to reveal the background that is obscured by the foreground, see-through NRT (Section 4.4.3, Figure 4.17) by inpainting of the cluttered background. Both works report the insufficiency of state of the art inpainting algorithms for different reasons.

#### 4.5 Dynamic NRTs [147, 148, 150]

A regular 2D periodic pattern can be characterized as a congruent wallpaper pattern formed by 2D translations of a single tile: given by a tile  $T$  and a pair of translation generators ( $t_1$  and  $t_2$ ) (Section 2.1.3). A near-regular texture (NRT) is a geometric and photometric deformation from its regular origin, thus the tile  $T$  becomes a function of location  $T(x, y)$  (Section 4.1). A dynamic NRT is an NRT under motion and its tile becomes a function of time  $t$  in addition to location  $(x, y)$ :  $T(x, y, t)$  (Figure 4.21). Lin and Liu [147, 148] point out that: despite various

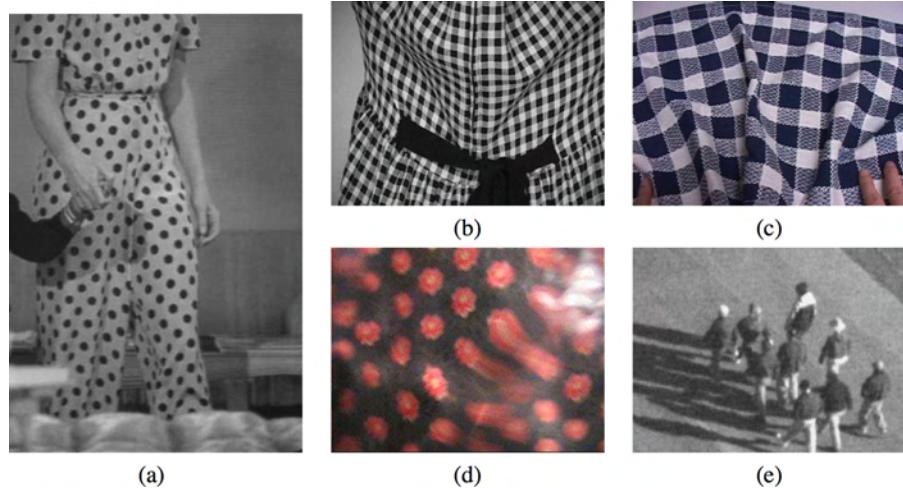


Fig. 4.21 Figure 1.1 in [148]. Examples of dynamic near-regular textures. These images illustrate challenges of dynamic NRT tracking: (a, b) ambiguous correspondences; (b, c) occlusions; (d) appearance (motion blur) and illumination variations. The texture in (d) is a pattern seen through disturbed water. A remote (long zoom from an air plane) video of a small group of people marching in formation on a non-straight path (e) exhibits drastic viewing angle and lighting change.

forms of dynamic NRTs, they have two common properties:

- **Statistical appearance regularity:** even though the geometry and the appearance of individual texels in a dynamic NRT vary, they bear strong similarity among themselves (and across time) that can be considered as statistically similar, especially neighboring texels.
- **Topology invariance:** the topological structure of a dynamic near-regular texture remains invariant during motion, even though its geometry and appearance may vary from frame to frame.

#### 4.5.1 Topological Invariant Dynamic NRTs [147, 148]

Lin and Liu [147, 148] propose a lattice-based Markov-Random-Field (MRF) model for modeling dynamic NRTs in a 3D spatiotemporal space (top of Figure 4.22). Their dynamic NRT model consists of a global lattice structure that characterizes the topological constraint among multiple texels and an image observation model that handles local geometry and appearance variations. The model behaves like a network of statistically varied springs. Based on this dynamic NRT model, they develop a tracking algorithm that can effectively handle the special challenges of dynamic NRT tracking, including: ambiguous correspondences, occlusions, geometry and illumination variations. Using the result of dynamic NRT tracking, the authors are able to further develop a dynamic NRT manipulation system that can replace new patterns for the same visual (geometry and photometry) effects, and superimpose an augmented image on a dynamic NRT in any unknown scene (Figure 4.22).

Subsequently, Park et al. [205] developed an efficient Mean-Shift Belief Propagation (MSBP) algorithm and applied it to graphical model inferencing for dynamic near-regular texture tracking. They demonstrate the robustness and efficiency of the algorithm on tracking low quality marching band videos as a multi-target tracking problem, even when the pattern is disturbed by outside targets that interleave the original target pattern (see video at <http://vision.cse.psu.edu/NRTtracking.html>).

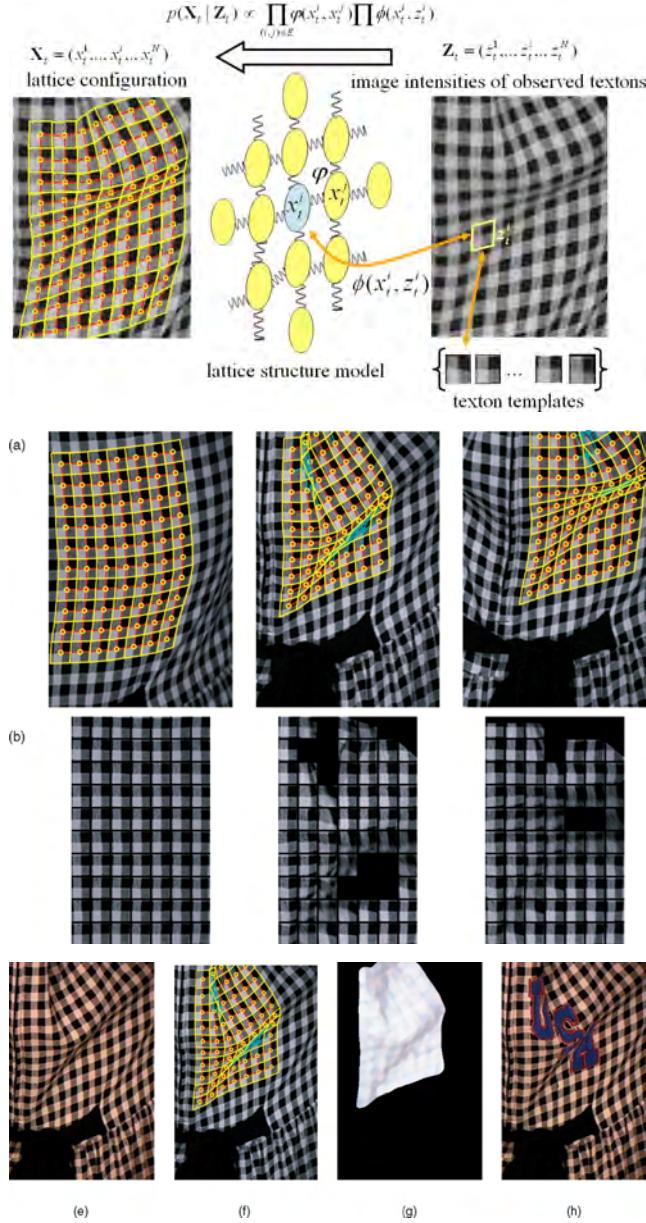


Fig. 4.22 Top: Figure 2.1 of [148] illustrating the temporal lattice-based MRF model, which consists of (1) a global lattice structure that characterizes the topological constraint among multiple texels and (2) an image observation model that handles local geometry and appearance variations. Bottom: Figure 2.13 of [148]. dNRT tracking and manipulation: (e) An input frame. (f) Tracked lattice and visibility map. (g) Extracted lighting. (h) Superimposed text. For videos, please see <http://www.cs.cmu.edu/~wclin/dnrtPAMI/dnrt.html>.

#### 4.5.2 Topological Variant Dynamic NRTs [150]

Recently, Liu and Liu [150] have relaxed the invariant topology condition used in the dynamic NRT tracking [147, 148, 205] for *Multi-Target Tracking of Time-varying Spatial Patterns*. To track multiple targets that have time-varying spatial patterns, there are two intertwined problems: (1) pattern perception: to recognize the (dis)appearance of spatial patterns from target locations; (2) position localization: to utilize the found pattern constraints for inferring target locations. By treating the patterns as latent variables that guide the tracking, the authors adopt an iterative two-step framework similar to the dynamic expectation–maximization (EM) method [107], addressing both problems simultaneously (Figure 4.23). In the first (E-like) step the system detects possible patterns and evaluate their posterior probabilities based on previous tracking results. In the second (M-like) step, on a mixture of MRFs, the tracking algorithm infers optimal target locations under pattern constraints controlled by posteriors as well as cues from image observations.

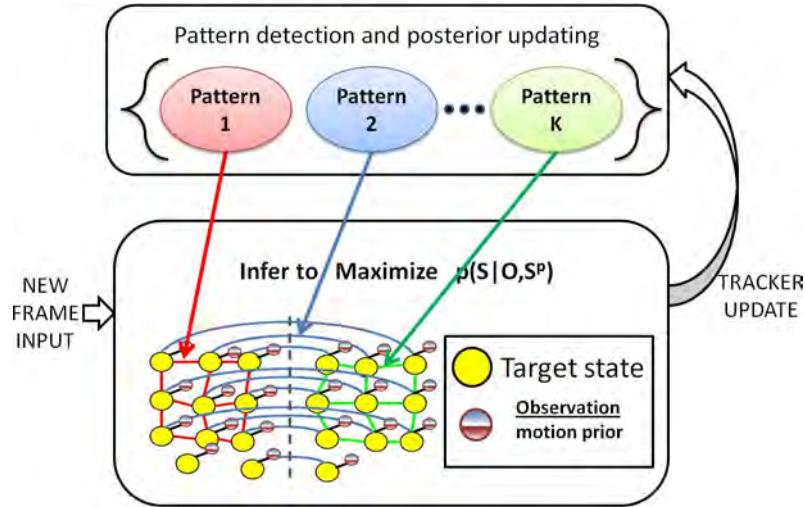


Fig. 4.23 Figure 1.4 from [150]. The proposed multi-target tracking framework. At E-like step, the algorithm detects patterns and/or evaluate pattern posteriors based on previous tracking results. At M-like step, the algorithm estimates the best target states given detected patterns, image observation and motion prior via inference on the mixture of Markov networks.

In their work, each potential pattern is specified with a Markov network enabling both local (short range) and global (long range) relations among multiple targets. Through inference on such a mixture of Markov networks, the authors exploit spatial domain information and its interplay with temporal cues to achieve robust tracking and a high level of understanding of the underlying, and varying, spatial structures of the targets being tracked. By treating spatial patterns as Bernoulli variables, the spatial pattern constraints are adjusted according to their posterior probability, so that the algorithm can handle pattern transitions softly and adaptively. Experimental results on various challenging videos demonstrate the effectiveness of the approach (Figure 4.24) and superior performance of the proposed algorithm over existing methods (Figure 4.25).

#### 4.5.3 Application of Symmetry in Multiple-View Geometry

In this article, we have mainly surveyed 2D symmetric patterns in 2D images. We have considered affine projection of these 2D symmetries, and the associated detection, reconstruction, and synthesis problems. In a much related line of work, people have also studied how 2D or 3D symmetry interacts with perspective projection. Perspective projection may significantly distort a symmetric pattern or structure in a 2D image. In this case, in order to correctly recover their geometry, one must resort to additional constraints offered by multiple-view geometry. Interested readers may refer to [102] for a more complete survey on that subject.

In particular, [102] has provided a clear characterization of the relationship between 3D symmetric structures and their 2D perspective images. The key observation there is that the symmetry group actions associated with any symmetric structure allows us to interpret a single perspective image of the structure as multiple images, called “hidden images” in [102], taken from viewpoints related by the same group actions. Thus, the 3D information about the structure is naturally encoded in the multiple-view geometric constraints among these (perspectively projected) images, characterized by the multiple-view rank conditions [180]. Based on these constraints, [102] has derived the

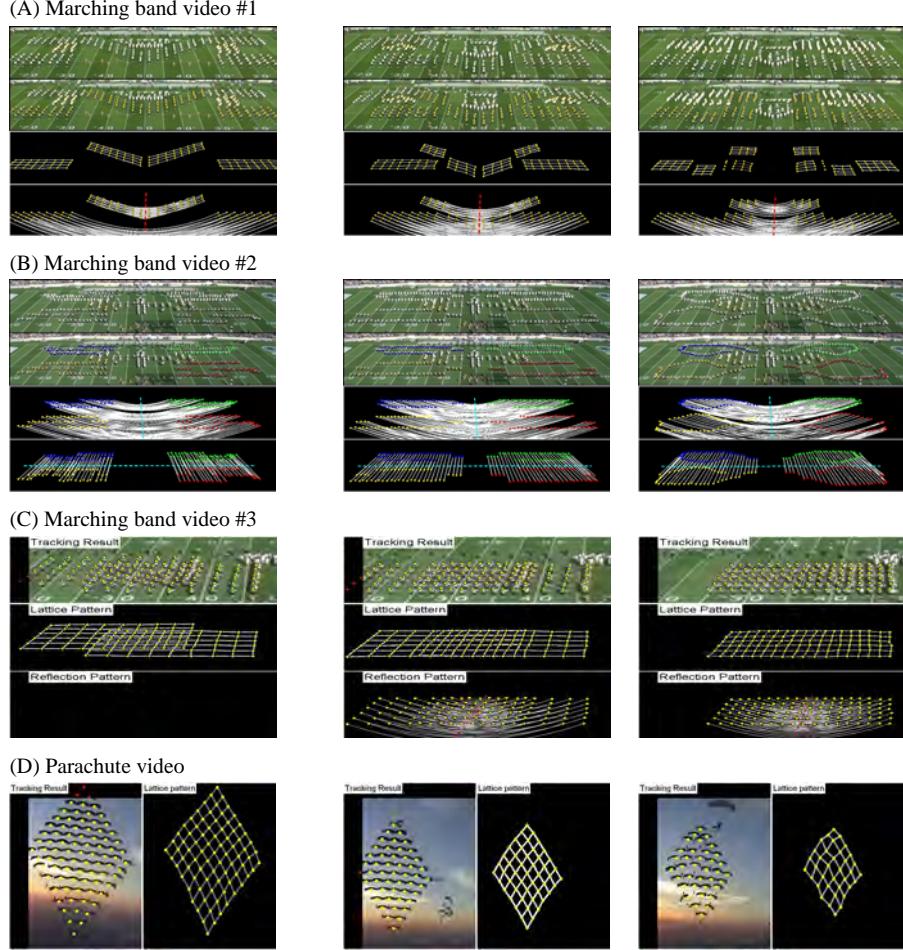


Fig. 4.24 Figures 5, 6, 7, 8 from [150]. (A) Detection and tracking lattice and reflection patterns on video #1. Left: 200th frame, middle: 400th frame, right: 650th frame. (B) Detection and tracking reflection patterns on video #2. Left: 1th frame, middle: 200th frame, right: 300th frame. (C) Detection and tracking lattice and reflection patterns on video #3. Left: 100th frame, middle: 200th frame, right: 400th frame. (D) Tracking and monitoring a breaking away lattice pattern of a parachute video. Left: 230th frame, 81 targets; middle: 400th frame, 49 targets; right: 530th frame, only 36 targets are left. Please refer to project website for related videos: <http://vision.cse.psu.edu/NRTtracking.html>.

necessary and sufficient conditions for a unique 3D reconstruction from all three basic types of Euclidean symmetry: rotation, reflection, and translation. It has also been shown that with symmetry, the 2D to 3D

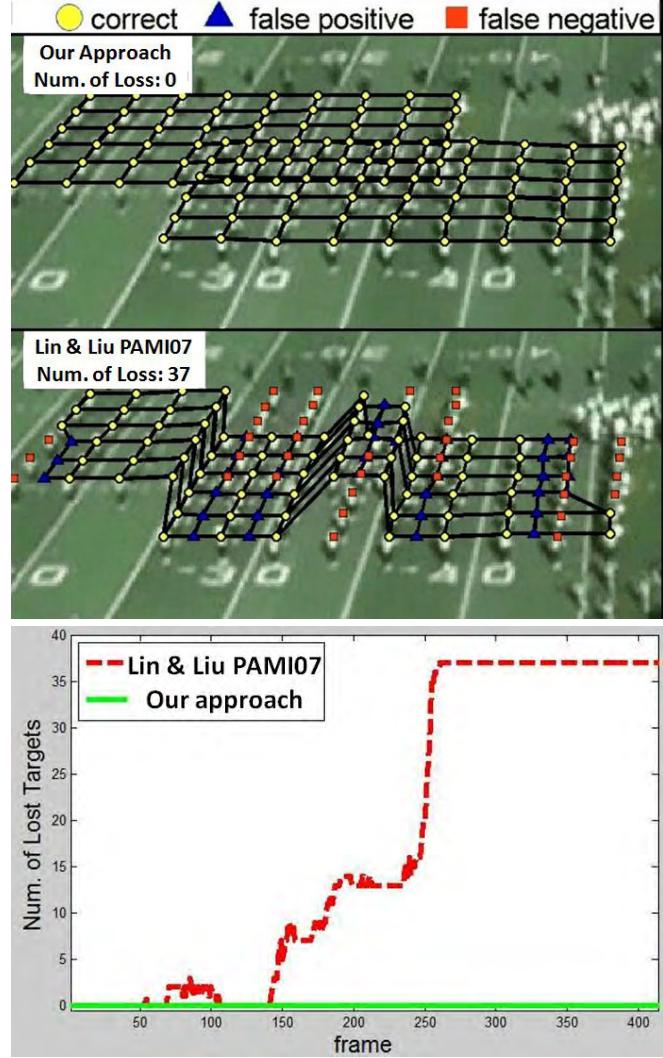


Fig. 4.25 Figure 11 from [150]. Comparison of tracking results (top) and statistics (bottom) of the method proposed in [150] and that of Lin & Liu's in PAMI07 [148] during the tracking of the entire marching band video #3.

reconstruction problem becomes well conditioned and the associated algorithms are remarkably simple, accurate, and robust.

This unified theory has later seen many important extensions and applications recently. In [281], Yang et al. has proposed a complete

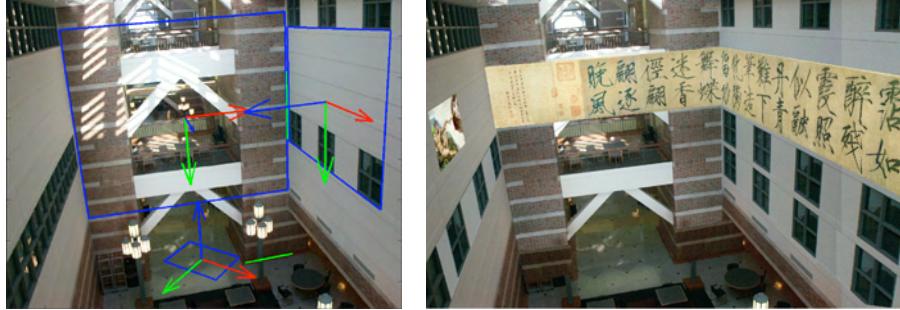


Fig. 4.26 An example of photo editing based on symmetric structures. Left: the original picture with some symmetric regions registered. Notice that the symmetric structures are highly distorted by the perspective projection and their precise orientation cannot be accurately recovered by an affine approximation. Right: a doctored picture. The shadows of the roof on the frontal wall and the occlusions by the lamp poles are removed using symmetry-based “copy-and-paste.” Some paintings are pasted on the side walls according to the correct perspective projection. Additional windows are added on the side walls. All parts of the resulting photo obey correct 3D geometry and perspective projection, without any noticeable distortion.

system pipeline for the detection and reconstruction of multiple symmetric objects from multiple perspective 2D images, which can be readily used for practical applications such as editing photos [104] or map building for autonomous navigation [280]. Figure 4.26 shows one such example. This work was later extended to continuous symmetric objects that do not have discrete features [101]. Symmetry has also been shown to be an extremely effective cue that helps significantly simplify difficult computer vision problems such as image segmentation [282] and large-baseline feature correspondence [105].

# 5

---

## Continuous Symmetry

---

In most of the above mentioned techniques (specifically, the Global Methods), symmetry is treated as a binary feature: either it exists or it does not exist in an object. In general, however, they include some form of thresholding in order to overcome noise and imprecisions in the input data. However the notion of quantifying or measuring symmetry is very limited. In an early work, Grünbaum [82] reviews methods of geometrically measuring symmetry of convex sets. Yodogawa [287] has presented an evaluation of symmetry (namely “Symmetropy”) in single patterns which uses information theory to evaluate the distribution of symmetries in a pattern. Marola [183] presents a coefficient of mirror-symmetry with respect to a given axis. Global mirror-symmetry of an object (image) is found by roughly estimating the axis location and then fine tuning the location by minimizing the symmetry coefficient. Gilat [73], Hel-Or et al. [97] and Avnir et al. [11] present the idea of a *Measure of Chirality* (a measure of deviation from mirror-symmetry). Similar to Marola, Gilat’s chirality measure is based on minimizing the volume difference between the object and its reflection through a varying plane of reflection. Hel-Or et al. present a measure of chirality for 2D objects based on rotational effects of chiral bodies

on the surroundings. Additional discussion on measuring chirality can be found in chemistry research and is reviewed in [295]. These symmetry evaluation methods are each limited to a certain type of symmetry (mirror or circular symmetry) and are generally of high complexity.

### 5.1 Measuring Symmetry as a Continuous Feature

The basic assumption that symmetry is a binary feature arises naturally from the exact mathematical definition of symmetry [186, 277]. However, this definition is inadequate when describing symmetric shapes and objects found in nature. A classic example is that of faces (Figure 5.1): although considered symmetric, faces actually deviate from perfect symmetry. Even perfectly symmetric objects lose their exact symmetry during image acquisition when projected onto the image plane or the retina due to occlusion, perspective transformations, digitization, etc. Psychological studies show that our percept of symmetry as a characteristic of shapes and objects is inherently continuous [292, 293]. Symmetry as a quantitative measure is naturally expressed as a relativistic feature. Considering the shapes in Figure 5.2 one would describe Shape ‘a’ as *more* symmetric than Shape ‘b’ and that Shape ‘c’ is *more* rotationally symmetric than mirror symmetric. Studies in natural sciences are continuously demonstrating correlation between the ‘amount’ of symmetry and other characteristics such as beauty, attraction, natural stability, etc. [78, 128, 192, 239, 306]. Thus, although symmetry is usually considered a binary feature, (i.e., an object is either symmetric or it is not symmetric), it would seem

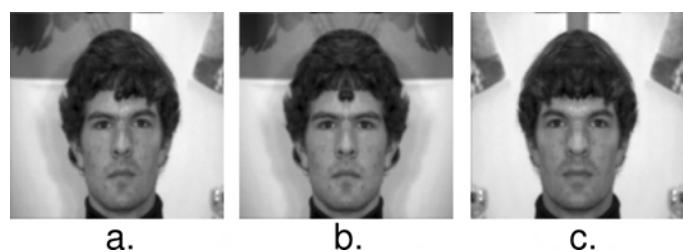


Fig. 5.1 Faces are not perfectly symmetrical. (a) Original image. (b) Left half of original image and its reflection. (c) Right half of original image and its reflection.

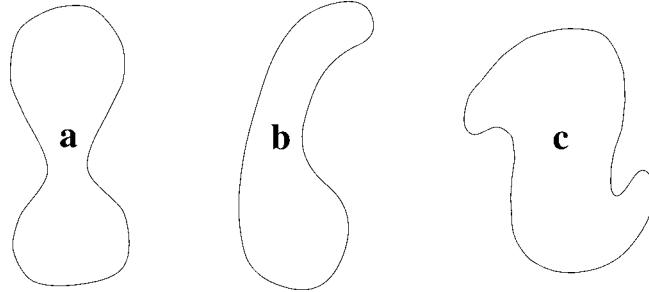


Fig. 5.2 The quantitative nature of Symmetry: Shape a is *more* mirror symmetric than Shape b and Shape c is *more* rotationally symmetric than mirror symmetric.

more beneficial to view symmetry as a continuous feature where the *amount* of symmetry is evaluated.

The concept of symmetry as a *continuous feature* was introduced in a series of papers [188, 293, 294, 295, 296, 297, 298, 299, 300, 302, 303, 304, 305] where the notion of a Continuous Symmetry Measure (CSM) was established. These studies have developed a measure of symmetry as well as algorithms for applying the measure to various input structures ranging from sets of points, polygonal shapes, 3d structures, images, and recently dealing with complex bifurcating structures found in nature. The continuous nature of the measure allows a comparison between symmetry characteristics of different shapes and comparison of different types of symmetries of a single shape (mirror-symmetries, rotational symmetries, etc.).

### 5.1.1 CSM — Basics

The underlying mathematics of symmetry relies on the Symmetry Groups [228, 277] (Section 2). In this section, we define a *symmetric object* as a set of points which is invariant under the actions of a symmetry group. A point  $x$  of an object and all its transformations under the group members is called an *orbit*. This implies that an object is symmetric if it can be partitioned into orbits each of which is invariant under the operations of the symmetry group. For example, an equilateral triangle in 2D is rotationally symmetric (order 3) and can be partitioned into orbits containing three points each. The vertices of the

triangle form one such orbit. One should note that, given a point  $x$ , every point in its orbit is naturally associated with an element of the symmetry group. For example, in Figure 5.4d, the orbit of point  $P_0$  consists of the points  $\hat{P}_0, \hat{P}_1$  and  $\hat{P}_2$  which are associated with  $R_3^0, R_3^1$  and  $R_3^2$ , respectively.

The essence of the Continuous Symmetry Measure approach involves determining the amount of ‘effort’ required to transform a given shape (composed of orbits) into a symmetric shape. This ‘effort’ is measured by the mean of the square distances each point must move from its original position in order for the shape to become symmetric. Note that no a priori symmetric reference shape is assumed. Denote by  $\Omega$  the space of all shapes of a given dimension, where each shape  $P$  is represented by a sequence of  $n$  points  $\{P_i\}_{i=0}^{n-1}$ . A metric  $d$  is defined on this space as follows:

$$d: \Omega \times \Omega \rightarrow \mathbb{R}$$

$$d(P, Q) = d(\{P_i\}, \{Q_i\}) = \frac{1}{n} \sum_{i=0}^{n-1} \|P_i - Q_i\|^2$$

This metric defines a distance function between any two shapes in  $\Omega$ .

We define the **Symmetry Transform** of a shape  $P$  as the symmetric shape  $\hat{P}$ , closest to  $P$  in terms of metric  $d$ .

The **Continuous Symmetry Measure (CSM)** of a shape  $P$  is defined as the distance between  $P$  and its Symmetry Transform:

$$\text{CSM} = d(P, \hat{P})$$

This definition of the CSM implicitly implies invariance to rotation and translation. Normalization of the original shape prior to the transformation additionally allows invariance to scale. An additional feature of the CSM is that we obtain the symmetric shape which is ‘closest’ to the given one, enabling visual evaluation (Figure 5.3c).

A geometric algorithm for deriving the Symmetry Transform of a shape and, accordingly, the CSM value was developed initially in [303]. For a specific symmetry group, the process of evaluating the CSM involves two steps:

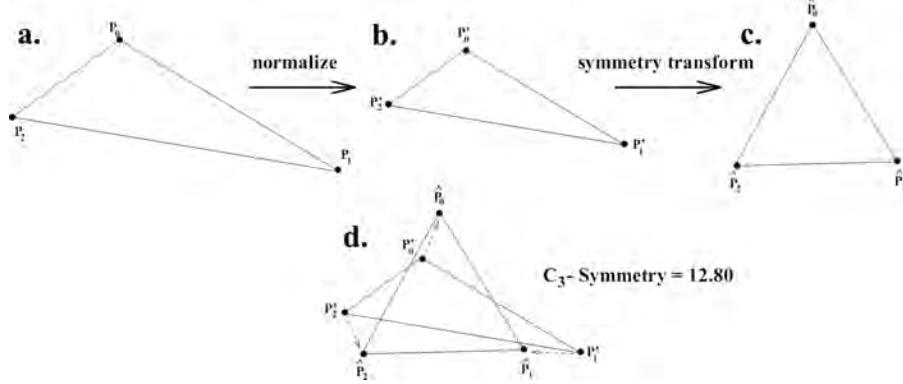


Fig. 5.3 Calculating the Symmetry Distance of a shape: (a) original shape  $\{P_0, P_1, P_2\}$ . (b) Normalized shape  $\{P'_0, P'_1, P'_2\}$ , such that maximum distance to the center of mass is a given constant (10). (c) Applying the symmetry transform to obtain a symmetric shape  $\{\hat{P}_0, \hat{P}_1, \hat{P}_2\}$ . (d)  $SD = \frac{1}{3}(\|P'_0 - \hat{P}_0\|^2 + \|P'_1 - \hat{P}_1\|^2 + \|P'_2 - \hat{P}_2\|^2)$ .

- (1) Partitioning the object into orbits, i.e., determining the points or features of the input that will form each orbit and associating an element of the symmetry group with each point.
- (2) Evaluating the symmetry transform and CSM value given the orbits.

Typically, there are numerous possibilities for partitioning the input points into orbits. The partition that provides the minimal distance between the original set and its symmetry transform is sought. The algorithm implementing Step 2 of the above process is referred to as the *Folding–Unfolding Method* due to its geometric visualization. Given an orbit of points  $\{P_i\}_{i=0}^{k-1} \subseteq \{P_i\}_{i=0}^{n-1}$  with associated symmetry group members  $\{g_i\}_{i=0}^{k-1}$ , the Folding–Unfolding Methods involves the following steps:

- (1) **Fold** — apply the inverse symmetry element to each point, i.e., apply  $g_i^{-1}$  to  $P_i$  for  $i = 0, \dots, k - 1$ . The *folded* points  $\{\tilde{P}_i\}_{i=0}^{k-1}$  are obtained.
- (2) **Average** — if the original points were perfectly symmetric, then by definition of symmetry, the points  $\{\tilde{P}_i\}_{i=0}^{k-1}$  would

coincide, however in the general case they do not. Thus, the folded points  $\{\tilde{P}_i\}_{i=0}^{k-1}$  are *averaged* obtaining  $\hat{P}_0 = \frac{1}{k} \sum_{i=0}^{k-1} \tilde{P}_i$ .

- (3) **Unfold** the average point  $\hat{P}_0$  by applying back the symmetry element  $g_i$ . The *unfolded* points  $\{\hat{P}'_i\}_{i=0}^{k-1}$  are obtained where  $\hat{P}'_i = g_i \hat{P}_0$ .
- (4) The former steps are applied independently to each orbit. The CSM is calculated by averaging the deviation of each orbit from perfect symmetry:  $\text{CSM} = \frac{1}{n} \sum_{i=0}^{n-1} \|P'_i - \hat{P}_i\|^2$ .

An example of implementing the algorithm is shown in Figure 5.4, for a single orbit of the  $C_3$  symmetry group, namely the rotational symmetry group of order 3. For a collection of orbits, the CSM is calculated independently for every orbit with the appropriate symmetry operation applied about the center of mass of all points rather than about the origin. Proof and details can be found in [303, 296, 300]. Examples of ST and their associated CSM values are given in Figure 5.5 and 5.6.

### 5.1.2 CSM Develops

The CSM approach has been implemented and shown to be viable on structures of increasing complexity. The major impediment in all cases

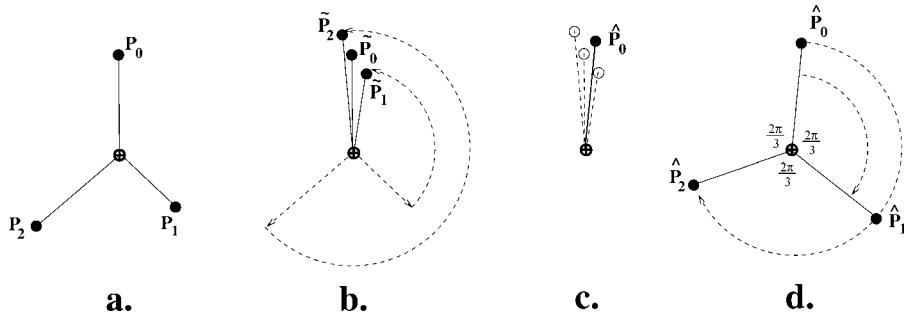


Fig. 5.4 The Folding–Unfolding algorithm applied to a single orbit. CSM with respect to rotational symmetry of order three is calculated. The points  $P_0, P_1$  and  $P_2$  are associated with the rotations  $R_3^0, R_3^1$  and  $R_3^2$  respectively. (a) Original 3 points  $\{P_i\}_{i=0}^2$ . (b) **Fold** — apply the inverse symmetry element to each point, i.e., rotate  $P_i$  by  $R_3^{-i}$ . Points  $\{\tilde{P}_i\}_{i=0}^2$  are obtained. (c) **Average** — points  $\{\tilde{P}_i\}_{i=0}^2$  are *averaged* obtaining  $\hat{P}_0 = \frac{1}{3} \sum_{i=0}^2 \tilde{P}_i$ . (d) **Unfold** — apply the symmetry elements to the average point  $\hat{P}_0$  (i.e., rotate by  $R_3^i$ ) obtaining points  $\{\hat{P}'_i\}_{i=0}^2$ . CSM is calculated:  $\text{CSM} = \frac{1}{3} (\|P'_0 - \hat{P}_0\|^2 + \|P'_1 - \hat{P}_1\|^2 + \|P'_2 - \hat{P}_2\|^2)$ . The centroid  $\omega$  is marked by  $\oplus$ .

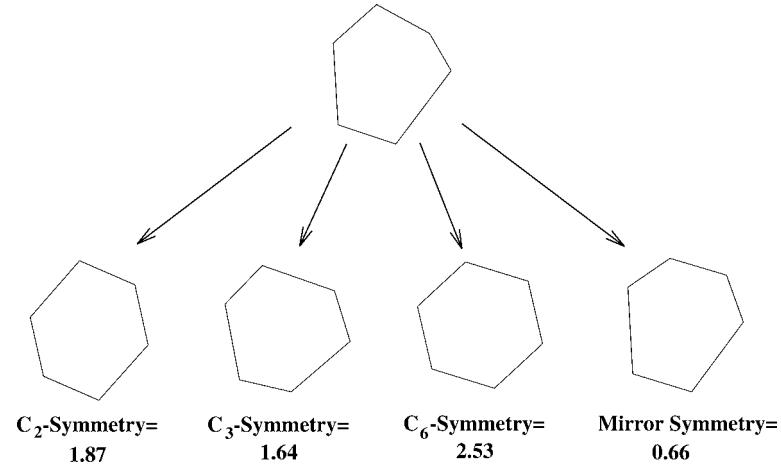


Fig. 5.5 Hexagon and its CSM value and symmetry transform for various symmetry groups.

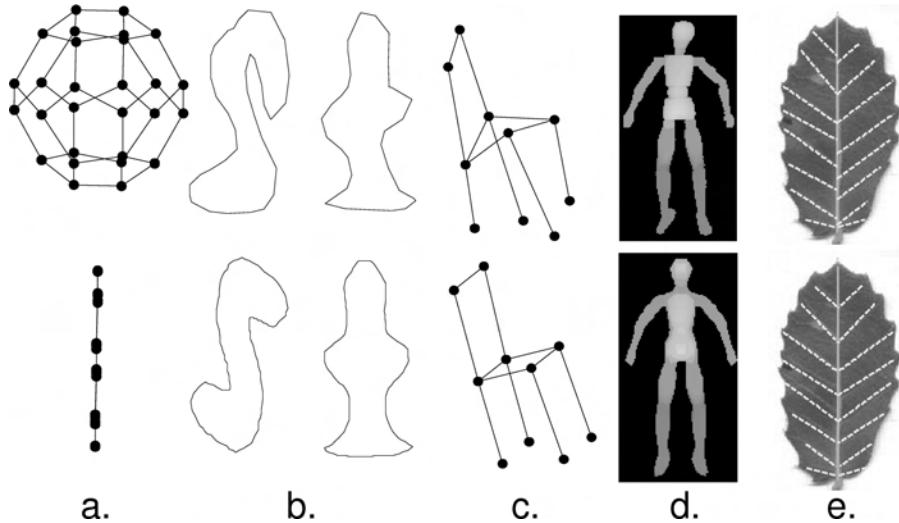


Fig. 5.6 Objects (top) and their symmetry transform (bottom).

is the determination of the partition into orbits composed of points or features of the input structure. Without any prior knowledge or constraints, determining the optimal division of input points into orbits is an intractable, highly complex problem (exponential with the number of points). However, as shown in the series of studies, certain classes

of structures can easily be dealt with. Connectivity and orderings of the input can constrain the possible partitioning into orbits and reduce complexity. For example, shapes and objects defined by polygonal or continuous closed contours inherently induce constraints on the partitioning into orbits as shown in Figure 5.7 [298, 301, 303, 296]. Higher dimensional objects such as polytopes and 3D objects induce partial ordering on the input points as well [294, 296]. This applies also to images when viewed as intensity elevation above baseline [298, 303] allowing not only the evaluation of CSM but also the detection of the optimal mirror symmetry plane and local symmetries (Figure 5.8) [297]. In the general case of objects described as graph structures (vertices and connecting edges), the connectivity, as well as the degree of each of the vertices induces constraints on the possible orbits [295]. For example, in Figure 5.9a points  $P_2$  and  $P_3$  cannot be in the same orbit as they differ in degree. Moreover, points  $P_3$  and  $P_6$  both of degree 3 cannot

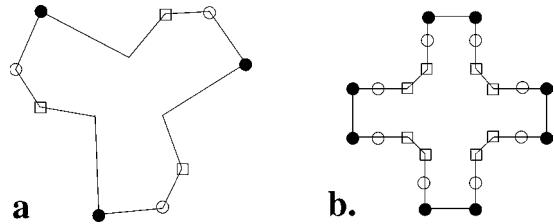


Fig. 5.7 A  $C_3$ -symmetric configuration of connected points. Orbits are interlaced: each orbit is differently marked ( $\circ \bullet \diamond$ ). (a)  $C_3$ -symmetric and (b) mirror-symmetric.

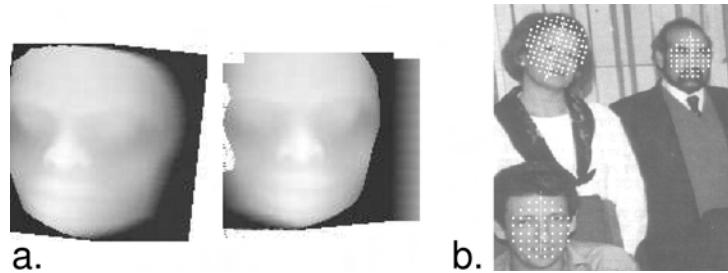


Fig. 5.8 Images viewed as intensity elevation above baseline allows evaluation of CSM as well as: (a) detection of the optimal mirror symmetry plane (and rotation of object to frontal vertical view based on the symmetry plane). (b) Detection of local symmetries in images.

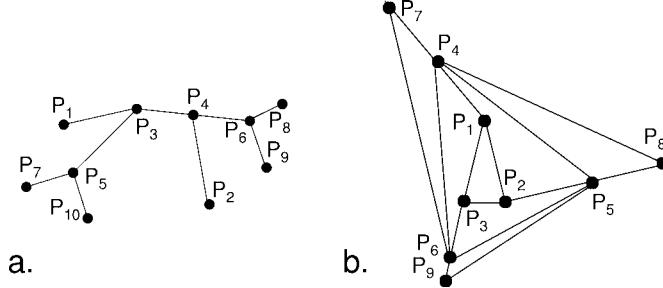


Fig. 5.9 Connectivity as in graph structures induce constraints on possible partitioning of points into orbits.

be in the same orbit since  $P_6$  has two connecting points of degree 1 whereas  $P_3$  only has one. This reasoning does not terminate at the second order connectivity ( $P_2$  and  $P_7$  do not agree in their third order connectivity) but must be taken to the maximal connectivity of the configuration. Accordingly, the structure in Figure 5.9b allows only one possible partition into orbits (up to a permutation of the assigned symmetry elements).

When input data consists of a collection of indistinguishable points (graphs with vertices of degree 0) no a priori constraint can be assumed to direct the division into orbits. In order to evaluate the CSM in these cases hierarchical symmetry seeking approaches have been used [297], as well as iterative approaches that approximate the optimal orbit decomposition of the points [9, 120].

### 5.1.3 CSM on Real Data

Measured and acquired data cannot be assumed to be noise-free. In addition to deviation from perfect symmetry, errors may be introduced in the acquisition process itself. Localization errors are specifically of importance in measuring CSM since the measure evaluates deviation from perfect symmetry as a function of the distances each point must move from its original position in order for the shape to become symmetric. This measure is thus strongly affected by measurement errors — namely errors in the location of points in the original structure. In [295, 302] CSM was developed to deal with localization

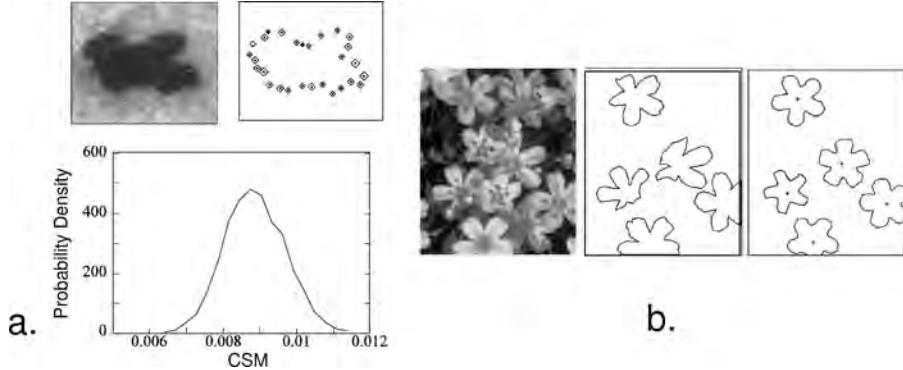


Fig. 5.10 CSM and noise: (a) fuzzy boundaries (left) and normally distributed localization errors (right) results in a probability distribution of CSM values (bottom). (b) Occluded flowers (top) and their boundaries (left). The Symmetry Transform of the occluded flowers (right).

errors modeled as probability distributions associated with point locations. The outcome is a probability distribution on CSM values. An example is shown in Figure 5.10a for a fuzzy image of a skin mark (Melanoma). Boundary points are assumed to have localization errors.

Occlusion of objects is a common artifact in imaging, strongly affecting the perceived symmetry of an object. Estimating the symmetry of a partially occluded object is challenging. In dealing with occlusion, one must be careful to ignore the occluding edge when evaluating CSM. This is performed by excluding outliers in the computation of the average in Step 2 in the Folding–Unfolding algorithm described above [299]. A search for the optimal center of symmetry improves the results. An example is shown in Figure 5.10b.

Even perfectly symmetric objects lose their exact symmetry when projected onto the image plane. This is dealt with in [304, 305] where CSM is evaluated in 2D projections of 3D objects. Furthermore, the symmetry transform is shown to improve the reconstruction of 3D objects from their 2D projections (Figure 5.6c).

#### 5.1.4 CSM in Nature

One of the basic features of shapes and objects in nature is symmetry. Symmetry is closely associated with the chemical and physical

characteristics of materials whether natural or man-made substances [10]. The study of symmetry is highly important in chemistry and physics where the symmetry of molecular structure is shown to correlate with chemical and physical properties [111, 295, 300]. The CSM has been used in the study of structures ranging from single molecules [6, 212, 295, 296, 300] through clusters [9, 120] and up to crystals [288]. Correlation has been found between CSM values and nuclear-magnetic resonance values [254], energy changes along molecular isomerizations [213], optical rotations [290], pressure and temperature structural effects [289] as well as enzymatic activity [123] and more [124].

Deviation of organisms or populations in nature, from perfect symmetry may provide information regarding both developmental failures and the success in buffering these deviations. The study of symmetry, and specifically mirror symmetry, as an indicator for stress and abnormal growth in plants and animals is widespread (see review in [77, 128, 192, 201]), as well as the study of correlation of symmetry with other characteristics such as attractiveness, dominance, natural selection, etc. [78, 193, 201, 239, 306]. The typical experiment involves measuring symmetry of a collection of specimens of a certain plant or animal and evaluating the distribution of symmetry values and/or correlation with other characteristic of form, of the natural habitat or of animal behavior. Symmetry evaluation is typically performed by detecting *landmarks* in the specimens and evaluating symmetry based on their location or other measurements evaluated from them [201]. In most cases, specimens are chosen in which landmarks are consistent (i.e., appearing in all specimens) and are easily detected (Figure 5.11a). In some cases these landmarks are variable in their number across the specimens in the experiment (Figure 5.11b). The CSM approach is easily applicable in these cases since the determination of the orbit points is consistent over the specimens in the test group and, under the assumption that topological symmetry is maintained, determination of orbits can be simplified by introducing constraints on possible matching points. Examples include the evaluation of symmetry of various molecules [6, 212, 295, 296, 300].

Of greater interest are the experiments involving specimens that demonstrate variability in their structural elements themselves

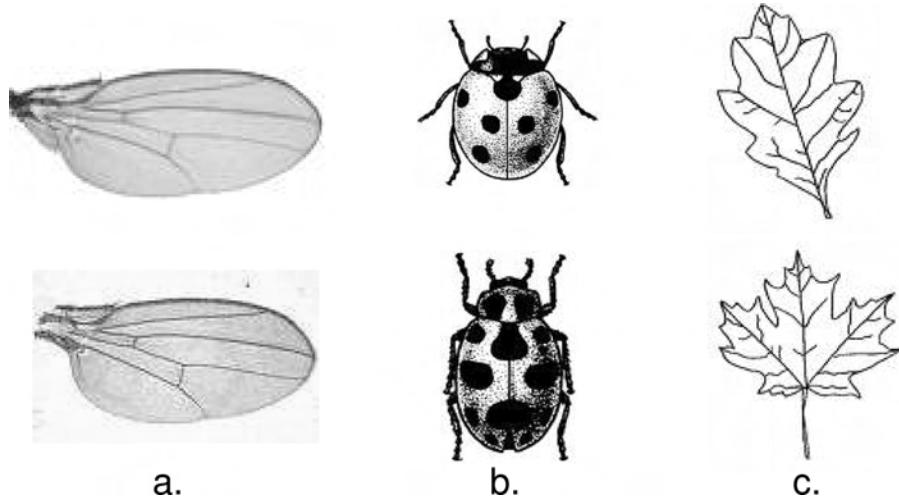


Fig. 5.11 Three types of natural objects for quantification of symmetry. (a) Structures that show consistency in topology and in number of landmarks. (b) Structures that show consistency in topology, but vary in number of corresponding landmarks among specimens. (c) Variable structures, no consistent topology, no quantitative consistency and sometimes no matching points.

(e.g., leaf veins — Figure 5.11c). In this case, there are no predefined landmarks, no consistent topology, no quantitative consistency, and at times missing points or elements (producing incomplete orbits), for example, a missing secondary vein in a leaf structure (Figure 5.6e).

The main challenge in evaluating symmetry in these cases is in determining the orbits (or matching points in case of mirror symmetry) in these structures since neither the number nor location of the matching points can be assumed. Additional constraints or assumptions must be adopted. In a series of studies on such objects, namely veins of leaves (Figure 5.11c), the CSM has been shown to be viable. In the case of bifurcating structures, namely, leaf veins, mirror symmetry evaluation is based on maintaining the structure rules while allowing topological and geometrical variations. Since no landmarks can be assumed in these structures, symmetry evaluation is based on the Biological Growth model of the structure (leaf) [187, 188]. The measure was used in an evolution study and has been shown to correctly classify leaves originating in challenging and non-challenging growth-environments [188].

This study is currently being extended in a similar manner to other species and animals including bacteria, beetles, and flowers.

### 5.1.5 CSM — What Next

The CSM as originally defined by Zabrodsky et al. in [303] can be implemented for any finite point symmetry group in any dimension. This implies that orbits of a finite number of elements or points are found in the input structure. A natural extension of the CSM would be to non-point symmetry groups including translatory groups and crystallographic symmetry groups [186, 277]. Theoretically, orbits of these groups are infinite and the input structure should be spatially unbounded. In practice, a bounded input structure is typically assumed. The orbits (or rather the bounded orbits), are accordingly, finite. A thorough study of patterns expressing these symmetry groups, is developed in [159] and described in Section 2.

Another direction in which CSM can be extended is in the symmetry analysis of time-varying systems. For example, measuring symmetry of moving objects such as molecules (as performed in [6, 10, 111, 123, 124, 212, 213, 254, 289, 290, 294, 295]), bodily movement (gait [158]), limb movement, facial expressions, etc. On a longer time scale, measuring symmetry as a function of time along the developmental time-line of plants and animals is of interest.

Finally, symmetry with respect to CSM is associated with geometrical properties of the input structure being measured. Further studies will incorporate additional characteristics into the measurement of CSM. These might include, color, width, weight, local direction and other features of the structure. Of interest, in these cases, is how to weight these characteristics against the geometric characteristics.

## 5.2 Facial Asymmetry as a Biometric [165, 167, 168, 191]

Human facial asymmetry has long been a critical factor for evaluation of facial attractiveness [258] and expressions [225] in psychology and anthropology, albeit most studies are carried out qualitatively using human observers as judges, or locally where features are measured individually (e.g., length of ears). Asymmetrical faces are considered

less attractive [258], and it has been reported that facial attractiveness for men is inversely related to recognition accuracy [200]. For face recognition by humans, a small yet statistically significant decrease in recognition performance is observed when facial asymmetry is removed from images [260], which suggests that facial asymmetry may play a role in human identification by humans.

Seitz and Dyer successfully produced believable 3D morphings of Mona Lisa's face by assuming that it is bilaterally symmetrical [243],<sup>1</sup> while facial asymmetry is readily observable on individuals in the general population (Figure 5.1). Asymmetry is a structural descriptor that cannot be captured by a single local measure (e.g., either left or right face alone). In [167, 168] the authors define two types of multi-dimensional quantified facial asymmetry measures and use them as candidate discriminative features (biometrics) for human identification. Following a face normalization process analogous to [12, 265], three anatomical feature points, the inner canthus of each eye and the philtrum are the three anchor-points to define an affine deformation, a *face-midline* centered vertically in each face image, and a set of uniform metric coordinates for all faces (Figure 5.12). The facial asymmetry features are defined as:

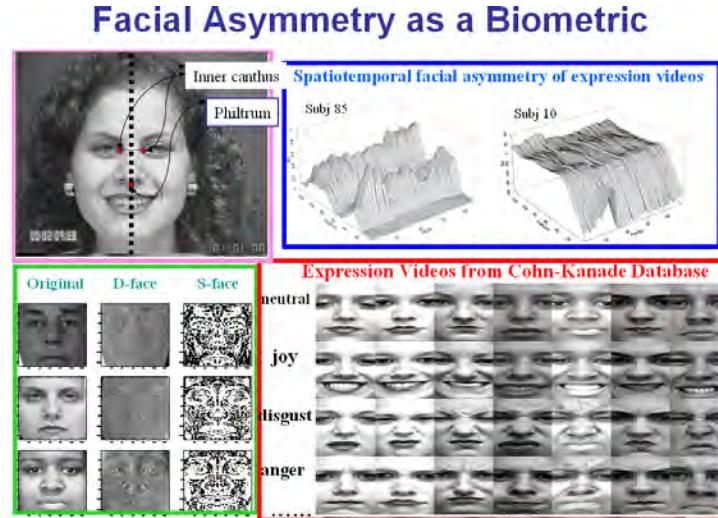
**Density Difference *D-face*:**  $D(x, y) = I(x, y) - I'(x, y)$

**Edge Orientation Similarity *S-face*:**  $S(x, y) = \cos(\phi_{I_e(x, y), I'_e(x, y)})$

where  $I_e, I'_e$  are the respective “edge” images of the normalized face density image  $I$  and its vertically reflected image  $I'$ ;  $\phi_{I_e(x, y), I'_e(x, y)}$  is the angle between the two edge orientations of images  $I_e, I'_e$  at the same pixel point  $x, y$  (Figure 5.12A). Using a learning-based feature selection scheme for human identification, and *augmented variance ratio* [167, 168] to measure the discriminative power of single features, they found the most discriminative single feature is at the nose-bridge region (Figure 5.12B). Classification rates show the power of quantified facial asymmetry using FisherFace as a baseline

---

<sup>1</sup>See the movie at <http://www.cs.cmu.edu/~seitz/vmorph/monas.mpg>.



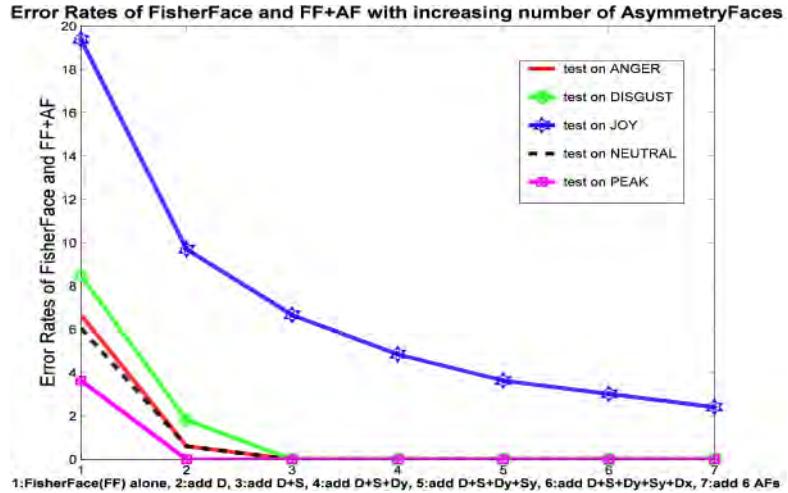
(A) Normalization, feature definition and data set



(B) Most discriminative single facial asymmetry feature for human identification

Fig. 5.12 (A, B) describe the method from [167].

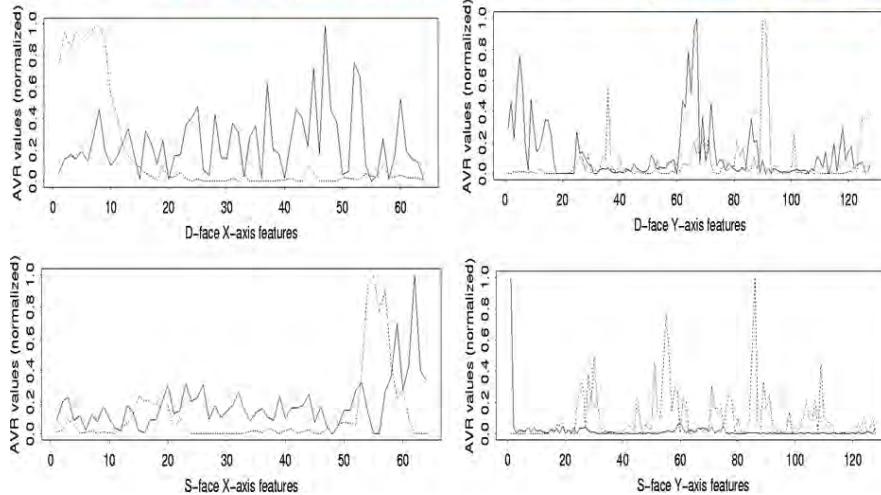
classifier (Figure 5.13C). The authors report: “*a synergy is achieved by combining facial asymmetry information with conventional Eigen-Face and FisherFace methods. We have assessed the generality of these findings across two publicly available face databases: Using a random*



(C) Classification errors

## Discriminative Feature Subspace Selection

Solid lines --- Human Identification, Dotted lines --- Expression Classification



(D) AVR values of all facial asymmetry features for human identity vs expression discrimination.

Fig. 5.13 (C) Classification results from [167]. (D) Part of Figure 1.4 from [191] indicating the mutual exclusiveness of the automatically selected discriminative facial asymmetry features (locations) for human identification versus expression classification.

*subset of 110 subjects from the FERET database, a 38% classification error reduction rate is obtained. Error reduction rates of 45–100% are achieved on 55 subjects from the Cohn Kanade AU-Coded Facial Expression Database. These results suggest that facial asymmetry may provide complementary discriminative information to human identification methods, which has been missing in automatic human identification.”* Similar types of facial asymmetry measures and discriminative feature-subset learning method are used for expression classification of 2D images [191] and gender classification from 3D faces (3D mesh data) [165]. Interestingly, and logically, the facial asymmetry features automatically selected for human identification and for expression classification appear to be mutually exclusive (Figure 5.13(D)).

### **5.3 Statistical Brain Asymmetry from Volumetric Images [157, 160, 161, 169, 170, 257]**

Human (and animal) brains have an approximate bilateral symmetry. The imaging technology, especially high resolution, structural MR images, provides a great data source for computational quantification, analysis and cross-subject comparison. To compare and compensate for spatially and anatomically different brain images, the first step is to bring different images into a common coordinate system (Figure 5.14(A)). The midsagittal plane (MSP) [157] of the brain serves as a good landmark for this alignment purpose. Liu et al. developed one of the first robust 3D (MSP) extraction algorithm that works on normal brains as well as brains with large lesions, it also works for CT images with large between slice gaps (up to 8 mm). Figure 5.14(B) shows the validation results of the MSP extraction algorithm on clinical and noise/tumor added images [157]. The difference between symmetry detection and MSP extraction is that the ultimate goal of MSP extraction is to find where the approximate bilateral symmetry would have been before the onset of the lesion. The idea is to compute and locate the peak in the correlation scores of the rotated, reflected image and its original image, resulting in a set of 3D midlines in space, then use the statistical robust fit to determine an optimal 3D plane that has least distance to all 2D mid-lines. Quantitative evaluations of this

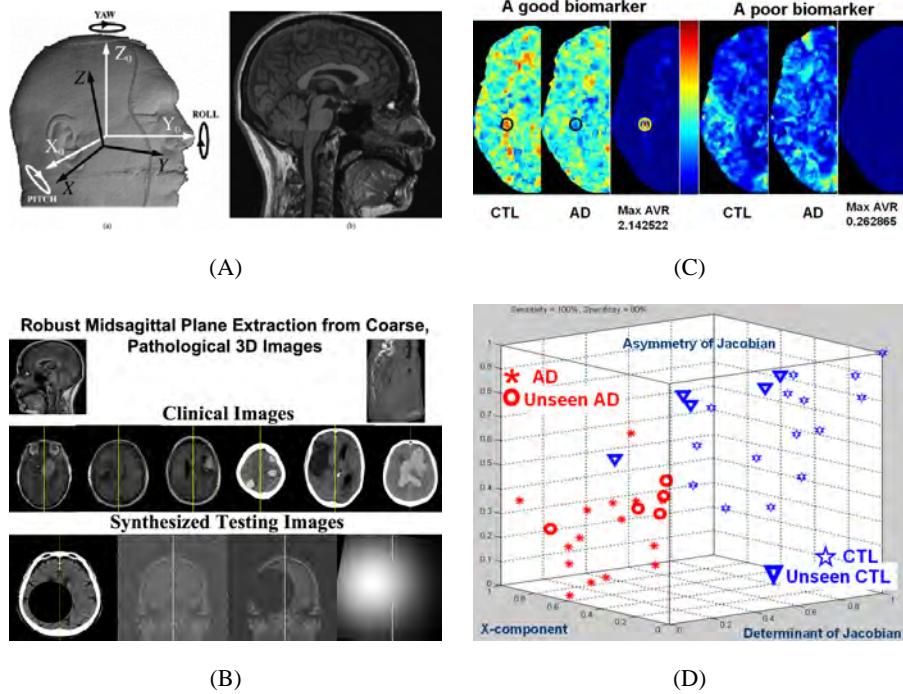


Fig. 5.14 (A, B) are figures from [157] which demonstrate a robust midsagittal plane extraction algorithm. (C), (D) are results from [170] which show how the statistical asymmetry features are used in learning the most discriminative anatomical features for Alzheimer's Disease classification from MR structural images.

method show no statistical difference with the MSPs determined by human experts (two) from neuroimages [157].

Using the MSPs as a basis, various types of brain asymmetry features are computed either directly from the MR/CT images or from the deformation fields obtained after a deformable registration of all images of the subject to a reference brain [160, 161, 169, 170, 257]. These features are used for 3D pathological image retrieval [160], and for computer aided diagnosis of schizophrenia and Alzheimer's Disease (AD) [169, 170]. One example for classification of AD and controls (CTL) using quantified brain asymmetry features is shown in Figure 5.14(C, D). Quantified brain asymmetry is also used for age estimation [257] where the significant difference between genders is observed.

# 6

---

## Symmetry in Graphics

---

Symmetry has also found extensive use in the computer graphics world, particularly in the generation of patterns and textures. Some aspects of symmetry are so natural and intuitive that they can be found implicitly in many graphics techniques. Our goal in this section is to survey papers that make explicit use of symmetry and symmetry groups. We will focus on kaleidoscopic drawing. Tilings, ornamental design, design in non-Euclidean geometry, sculptural forms, and chaos and fractals.

### 6.1 Kaleidoscopes

In the second SIGGRAPH conference, Alexander presented a system for drawing wallpaper patterns using a pen plotter [4]. He adopted a straightforward approach to breaking down the drawing problem. A motif is selected to act as an asymmetrical base for the drawing. Copies of the motif are composed via rotations and reflections into what Alexander calls a “medallion”, a figure with cyclic or dihedral symmetry. The medallion can be used to fill the plane using only translations and glide reflections. Alexander observes that appropriate choices of cyclic and dihedral medallions can be combined with translations and glide reflections to produce pictures in all 17 wallpaper groups.

Loreto et al. developed an interactive, multi-window system for visualizing wallpaper patterns [174]. Their multiple views permit side-by-side comparisons of patterns from different symmetry groups, as well as patterns related through continuous transformations they offer for exploration purposes.

Several popular software tools exist for creating planar symmetric drawings. Kali was originally written by Amenta at the Geometry Center, and later refined by Weeks and Phillips.<sup>1</sup> It can apply finite, frieze, or wallpaper symmetry to collections of colored line segments. It replicates line segments using a hierarchy of recursive calls bottoming out in translation: every direct and reflected orientation of a segment is drawn in every translation before proceeding to the next. (Obviously, this drawing order would have been inefficient in Alexander’s plotter-based work.) Another well known tool is Artlandia, a plugin for Adobe Illustrator.<sup>2</sup>

## 6.2 *Tilings*

In the past 100 years, tiling theory has flourished into a rich topic in geometry with connections to almost every other branch of Mathematics. Most modern tiling theory, such as the study of aperiodicity and substitution systems, is well beyond the scope of this survey. On the other hand, the tilings used commonly in art (most famously in Escher’s regular divisions of the plane [237]), design, and ornament rely on theory that is thoroughly investigated and well understood [84].

We will leave most of the formal definitions of tiling theory to the reader’s intuition, and introduce terminology only as needed to discuss the papers that follow. A practical introduction to the use of tiling theory within computer graphics can be found in the book by Kaplan [114]. The authoritative reference on tiling theory is still the treatise by Grünbaum and Shephard [84]. For the purposes of computer-generated imagery we may usually restrict ourselves to the so-called “normal tilings” discussed in both of those references, eliminating many pathological cases.

---

<sup>1</sup> <http://www.geom.uiuc.edu/java/Kali/>.

<sup>2</sup> <http://www.artlandia.com/>.

A *monohedral* tiling is one in which all tiles are congruent to a single shape, called the *prototile*. Of course, a tiling is a design in the plane, with an associated symmetry group. We may define the *orbit* of a tile as the set of its images under the symmetries of the tiling, and define a *k-isoedral* tiling to be one with  $k$  distinct orbits. When  $k = 1$ , the symmetries act transitively on the tiles and the tiling is simply called *isoedral*.

The isoedral tilings are especially well suited to use in computer graphics. They are sufficiently versatile to account for a wide range of commonly used monohedral tilings (all but one of Escher’s monohedral regular division drawings are isoedral). On the other hand, they have a simple structure and have been fully classified. They fall into 93 distinct combinatorial types, based on the relationships between a tile and its neighbors, and admit an efficient software implementation. An earlier formulation based on the structure of a tile’s edges is due to Heesch and Kienzle [96, 237]. Each class of “Heesch tiling” corresponds to a set of related isoedral tiling types (the isoedral types are further distinguished by the internal symmetries of tiles).

The earliest use of tiling theory in computer graphics was due to Chow [36, 37]. He observes that in any Heesch tiling, exactly half the prototile’s boundary is redundant. He created a system in which the user describes edges that make up half the boundary, and optionally adds decorative markings. The computer fills in the other half of the tile and covers a region of the plane with it. See Figure 6.1 for two examples. Chow’s results are primarily decorative tilings in the style of Escher, though he also draws connections to manufacturing. He claims that copies of a part that tiles the plane can be cut from a single sheet with no waste material.

Several research projects produced tiling software during the 1980s and 1990s, though they were concerned more with the mathematical theory than the production of images [50]. In 2000, Akleman et al. presented a Java-based system for drawing symmetric tilings. Any periodic tiling has a translational subgroup with a fundamental region isomorphic to a torus. Their system is based on interactively editing a planar graph embedded on a torus.

Also in 2000, Kaplan and Salesin presented Escherization, in which the computer automatically discovered tilings in the style of Escher’s

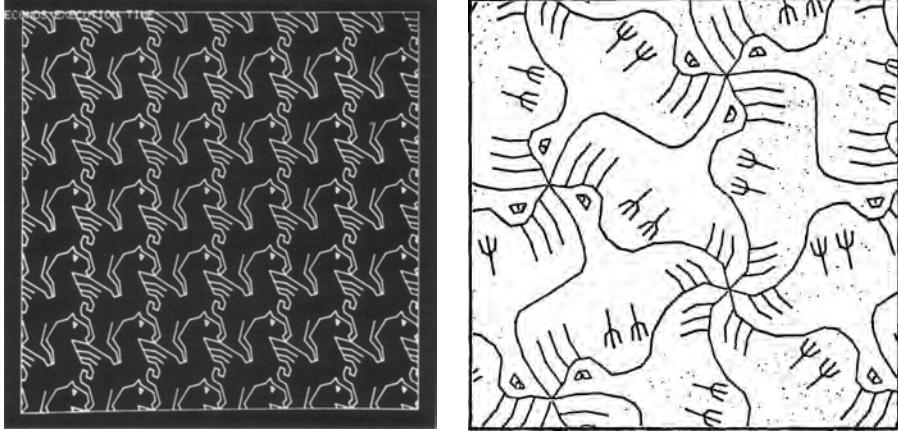


Fig. 6.1 Two tilings created by Chow.

drawings [116]. They show that the isohedral tilings may be parameterized in such a way that an optimization algorithm can search the space of tilings for one whose prototile most closely resembles a user-supplied goal shape. Candidate tiles are compared to the goal shape using an efficient least-squares metric. They later extended their technique to the dihedral (two shape) case and to Penrose tilings [117]. See Figure 6.2 for examples of monohedral and dihedral tilings produced via Escherization.

Several software packages are available freely or commercially for tiling design. They are primarily oriented towards educational use. The most popular are KaleidoMania! by Lee,<sup>3</sup> Tess by Pedagoguery Software,<sup>4</sup> and KaleidoTile by Weeks.<sup>5</sup> The latter two support tilings in non-Euclidean geometry (see Section 2.3).

### 6.3 Ornamental Design

Ornamentation has been practiced around the world since before recorded history. The style of ornament, and the manner in which it is executed, is often tied to a particular culture and time period [274].

<sup>3</sup> <http://www.keypress.com/x6173.xml>.

<sup>4</sup> <http://peda.com/tess/Welcome.html>.

<sup>5</sup> <http://www.geometrygames.org/KaleidoTile/>.

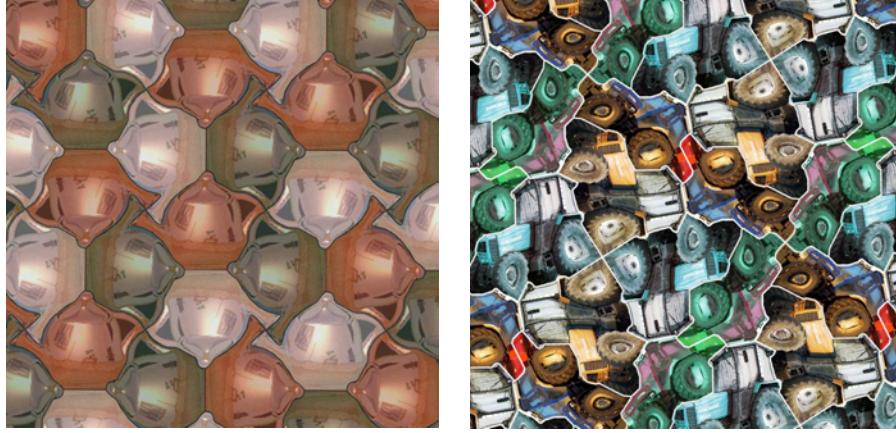


Fig. 6.2 Monohedral and dihedral Escherized tilings created by Kaplan and Salesin.

Ornament is usually distinguished from art in part by its tendency towards structure, repetition, and order. This order need not necessarily be accounted for by symmetry theory; some researchers have dogmatically taken symmetry as the defining characteristic of ornament, an attitude referred to by Grünbaum as “the group theory cult” [83]. Though not all-encompassing, the frieze and wallpaper groups are a powerful set of tools for analyzing and producing symmetric designs in the plane.

Several researchers in computer graphics have studied the design and rendering of Islamic geometric patterns (also called Islamic star patterns). These abstract patterns, found most famously on the walls of the Alhambra palace in Granada, Spain, are usually highly symmetric. They represent an interesting challenge because most of the original design techniques are lost to history. Researchers are therefore forced to “reverse engineer” those techniques, or invent new ones altogether that account for the structure of the many available historical examples. Grünbaum and Shephard carry out a group-theoretic analysis of geometric patterns with  $p6m$  symmetry [85]. They show that tools from group theory, such as the Cayley diagram, can be used to understand the number and types of distinct strands in a pattern. Ostromoukhov extends this analysis framework to all the wallpaper groups [199], providing a valuable framework for ornamental design

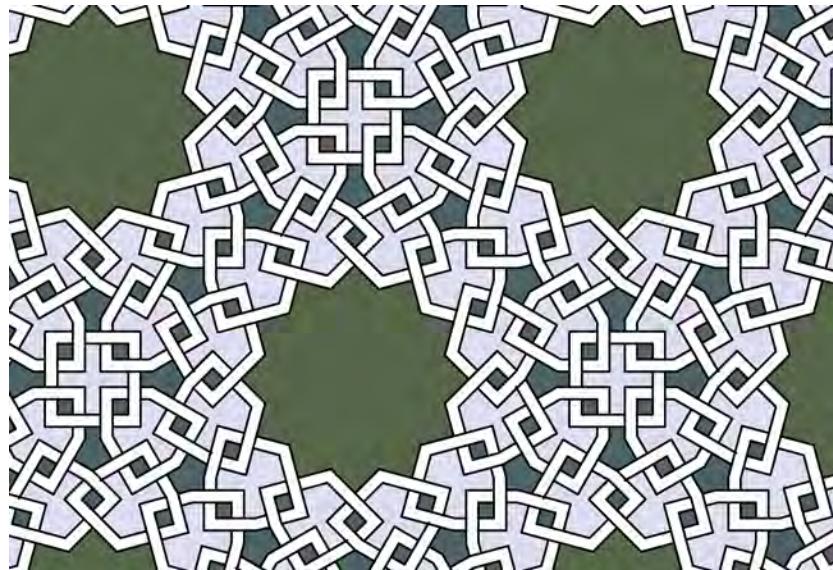


Fig. 6.3 An Islamic star pattern created by Kaplan using the “polygons in contact” technique.

in computer graphics that extends beyond Islamic art. The straightforward replication of Islamic patterns from asymmetric fundamental regions is carried out by Karam and Nakajima [119], and in the many examples collected by Abas and Salman [1]. In other work, Kaplan uses symmetric tilings as a basis for assembling star patterns using a simple “polygons in contact” technique [113] (Figure 6.3).

Several other forms of symmetric ornament have been investigated by computer graphics researchers. Kaplan created abstract symmetric designs by computing Voronoi diagrams of symmetric arrangements of points, for example overlapping lattices [112] (Figure 6.4). Glassner built textures by recursively replacing square tiles with four scaled copies, transformed according to simple local symmetry rules [76] (Figure 6.5). The results usually do not have global wallpaper symmetry, but contain a high degree of order and usually some cyclic or dihedral symmetry. Liu et al. analyze the symmetries of Chinese paper-cut patterns [162]. They extract the cyclic and dihedral portions of a scanned-in pattern, and then synthesize new instances with different symmetries (Figure 6.6).

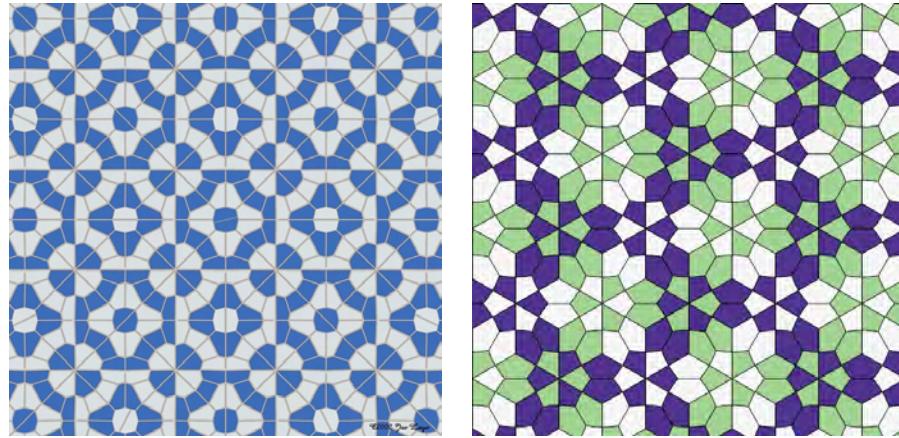


Fig. 6.4 Two ornamental Voronoi diagrams. The image on the left, by Leys, is formed from overlapping grids of points. The image on the right, by Kaplan, is the Voronoi diagram of overlapping circles.

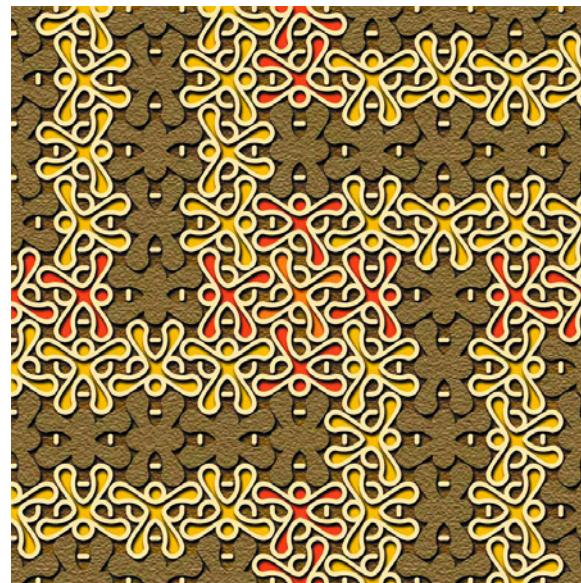


Fig. 6.5 One of Glassner's “hierarchical textures”.

Kaplan recently demonstrated how to transfer symmetric ornamental patterns from the Euclidean plane to an arbitrary mesh surface [115]. Given a pattern with symmetry group  $p4$ , it is possible

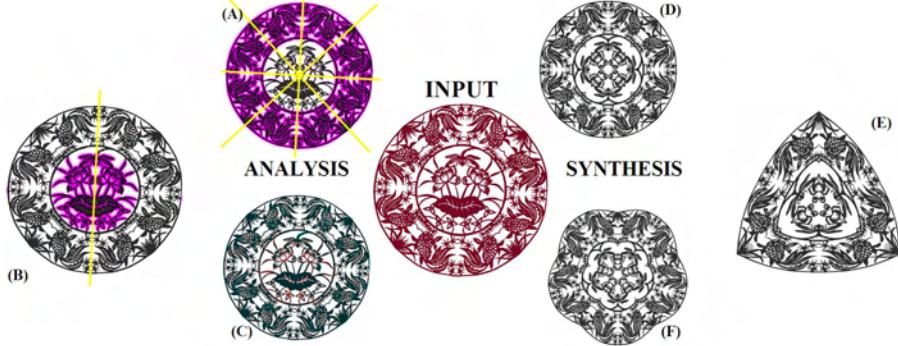


Fig. 6.6 Given a scanned-in papercut-pattern, three types of symmetry structures automatically found by the analysis process of [162]: (A) a region with dihedral symmetry group  $D_{2x4}$  supported by edges from the outer-ring of the pattern (magenta color); (B) a nested bilateral symmetry supported by edges from the center pattern; and (C) asymmetry region, primarily covering the tail of the dragonfly (cyan color). Synthesis results: (D), (E) and (F) show three of many synthesized new papercut patterns, with dihedral symmetry groups  $D_{2x4}$ ,  $D_{2x3}$ , and  $D_{2x5}$ , respectively, using the motif captured in the input papercut pattern. A set of fold-then-cut action plans to generate these patterns are also generated accordingly.



Fig. 6.7 Examples of symmetric patterns mapped onto surfaces using the method of Kaplan [115].

to identify a square region of the pattern that is a union of four fundamental regions, with centers of rotation at the corners of the square. Contemporary mesh parameterization techniques such as Spectral Surface Quadrangulation [51] can subdivide any mesh surface into coarse square regions, and a copy of the pattern region can be copied into each square to create a seamless covering of the surface. This technique extends easily to pattern types  $p4g$  and  $p4m$ , which contain  $p4$  as a subgroup, as well as to  $p6$  and  $p6m$ , using a triangular surface parameterization method such as Globally Smooth Parameterization [125].

## 6.4 Non-Euclidean Geometry

In Section 2.3, we introduced the notion of symmetry groups in non-Euclidean geometry. In this section, we describe some of the techniques that can be used to draw hyperbolic designs, and survey graphics research concerned with such drawings.

In a Euclidean wallpaper pattern, the simple structure of the translational subgroup makes it easy to fill space with copies of a motif. Replication in hyperbolic symmetry groups is more challenging. Dunham et al. offer a straightforward recursive implementation based on building layers of motifs outward from the centre of the Poincaré disk, with motifs being placed relative to the centres of the regular  $p$ -gons in  $\{p, q\}$  tilings [56]. More recently, Epstein et al. developed the theory of *automatic groups* [59]. According to this theory, these hyperbolic symmetry groups have an *automatic structure*, a table-driven system that can generate isometries associated with copies of a fundamental region in a hyperbolic pattern [142]. The isometries are generated in a fixed order that grows outward from the centre of the disc, and each one is produced exactly once.

Over the years, Dunham has created hyperbolic interpretations of several different styles of ornamental design, including Escher's Euclidean drawings [52, 57], Celtic knotwork [53], and Islamic patterns [54] (Figure 6.8). Kaplan and Salesin presented a system that could draw Islamic star patterns in the Euclidean plane, the hyperbolic

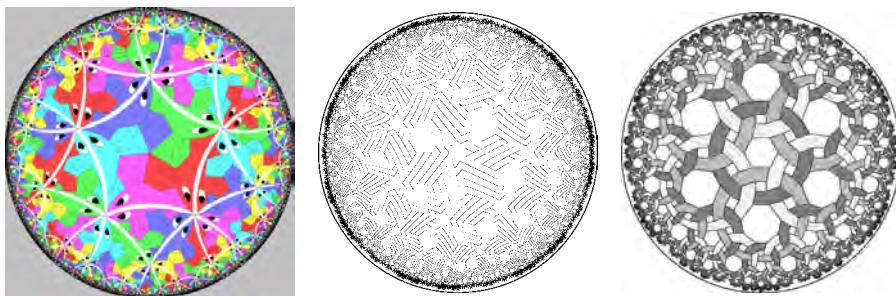


Fig. 6.8 Three hyperbolic designs by Dunham: a rendition of an alternate version of Escher's *Circle Limit III* on the left, an Islamic geometric design in the center, and a Celtic design of linked rings on the right.



Fig. 6.9 An Islamic star pattern design interpreted in the Euclidean plane, the surface of a sphere, and the hyperbolic plane.

plane, and the surface of a sphere [118]. Their algorithm relies on a construction in *absolute geometry*, in which neither the parallel postulate nor its negation are taken as true (see Figure 6.9). The construction is then equally valid in any of the three target planes.

A new paper by von Gagern and Richter-Gebert describes a technique they call Hyperbolization, a complete solution for transferring Euclidean symmetric patterns to the hyperbolic plane [273]. They begin by extracting the necessary information from an image of a symmetric pattern by detecting the pattern's Euclidean symmetry group and computing an “average” fundamental region. They use recent computer graphics work on discrete conformal geometry [252] to compute a low-deformation conformal mapping of the Euclidean fundamental region into the hyperbolic plane, and a reverse lookup algorithm for sampling the colors of the final hyperbolic design. This technique is very general, and can handle Euclidean ornaments belonging to every symmetry group except  $p1$ .

## 6.5 Polyhedra and Sculpture

The polyhedral symmetry groups offer a fertile ground for the creation of symmetric sculptures in three dimensions. These works of art have a definite mathematical flavor, and invite the viewer to appreciate what Hart calls the “geometric aesthetic” [90]. That aesthetic is certainly evident in the fascination many hobbyists have with the construction of models of symmetric polyhedra — see, for example, the books by



Fig. 6.10 Three Escher-like spheres created using the technique of Yen and Séquin and manufactured by an FDM rapid prototyping machine.

Holden [100] and Wenninger [276]. Blinn shows how the symmetries of the Platonic solids may be used as a method of drawing them or related three dimensional shapes [19].

In a few instances, computer graphics research has played a role in the construction of symmetric sculptures. Yen and Séquin developed a system for drawing Escher-like tilings on the surface of a sphere [286] (Figure 6.10). They offer the user a choice of several isohedral tilings of the sphere, and the ability to modify the boundaries of the tiles interactively. The user can also specify a set of bas-relief markings for the interior of the tile. Tiles are then extruded into three-dimensional models that can be manufactured via rapid prototyping and assembled into a sphere.

Séquin has also collaborated extensively with artist Brent Collins, who creates geometric sculptures based on profound intuition rather than mathematical training. They have described several different families of sculptures, based on minimal surfaces and curves on spheres [246]. Most of these sculptures have simple cyclic symmetry, having been designed as a stack of repeating units bent into a circle. The geometry is quite complex, and is visualized in an interactive tool that permits free exploration of spaces of hypothetical sculptures. An example is shown in Figure 6.11.

Akleman et al. presented a suite of shape modelling tools that collectively enable the creation of many symmetric sculptures [3]. Beginning with Platonic solids, they interactively add geometry to faces, attach pieces of the model with handles, carve holes using “rind modelling”,



Fig. 6.11 A photograph of Brent Collins displaying a symmetric minimal surface sculpture produced in collaboration with Séquin.

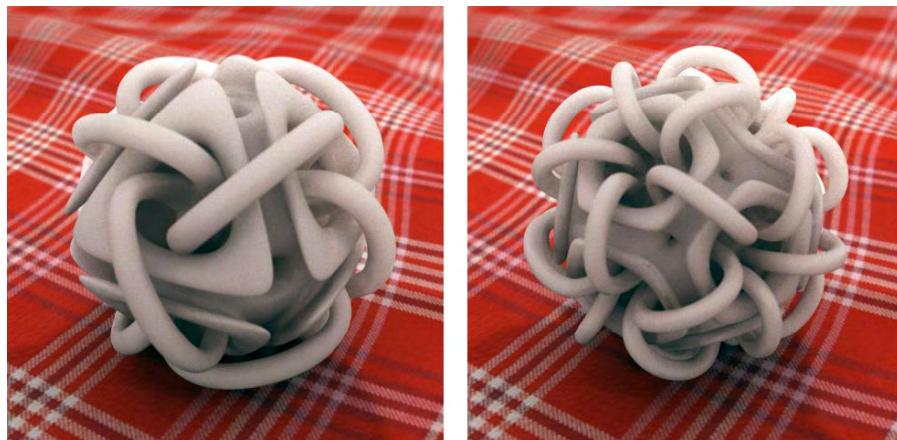


Fig. 6.12 Two symmetric sculptures constructed using the method of Akleman et al.

and smooth the sculpture using surface subdivision. See Figure 6.12 for examples.

Hart has a long history of building sculptures based on polyhedral symmetry [90, 91]. His recent experiments have involved modular design of sculptures based on pieces lying in the face planes of polyhedra [93]. In his design software, the user can draw modules on top of a polyhedron's stellation diagram, in order to see how it will interact with

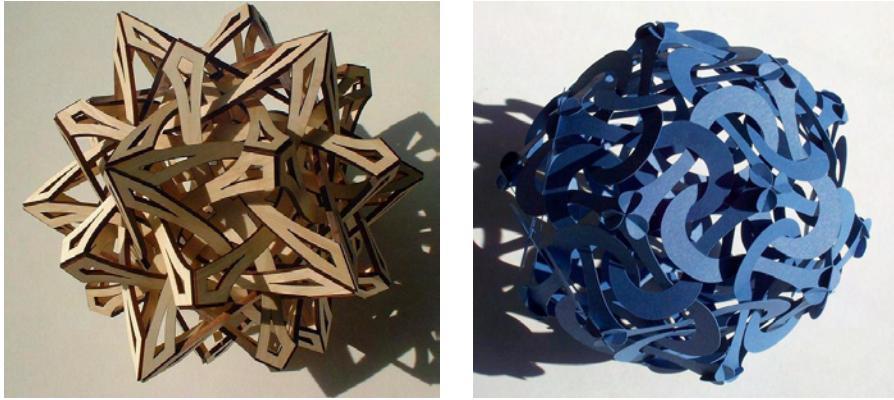


Fig. 6.13 Two symmetric sculptures by Hart. “Compass Points” on the left is assembled from laser-cut wooden pieces. “Meanders” on the right is made from interlocking pieces of paper.

symmetric copies of itself. For example, this allows Hart to construct sculptures out of pieces that interlock at edges of the stellation diagram, yielding an object that is simultaneously a mathematical model, a work of art, and a test of manual dexterity and puzzle-solving skills [92]. Figure 6.13 shows two sculptures by Hart.

Other creators of symmetric sculpture use computer graphics regularly as part of their artistic process. Rinus Roelofs and Bathsheba Grossman both work this way. Grossman typically begins with a clay maquette, which she replicates in rough form using 3D modelling software. She then uses Brakke’s SurfaceEvolver [25] to minimize the surface area subject to position and symmetry constraints.

## 6.6 Chaos and Fractals

Images of chaos and fractals have been a popular use of computer graphics ever since fast computers and raster displays first made it possible to visualize these complex systems. Their limitless variety, endless detail and unpredictability make them an open-ended source of computer-generated imagery.

Some degree of symmetry has always been evident in these images. Setting aside the “symmetry” of approximate self-similarity, the simple

equation that governs the Mandelbrot set guarantees that the result will have bilateral symmetry. Similarly, all of its Julia sets have (at least)  $c_2$  symmetry. Fractals derived from Newton iteration can easily be given any  $d_n$  symmetry through an appropriate choice of polynomial.

An interesting problem from the point of view of symmetry theory is to construct chaotic dynamical systems whose attractors exhibit the symmetries of a desired group. Field and Golubitsky have produced many drawings of attractors with finite, frieze, and wallpaper symmetries [64, 65, 66]. Given a symmetry group  $G$ , their technique relies on iterating “ $G$ -equivariant maps”, functions whose action commutes with the application of elements of  $G$ . Suitably chosen  $G$ -equivariant maps will be designs with precisely the symmetries of  $G$ . Figure 6.14 shows three examples of symmetric attractors. A more recent approach to finding invariant functions and coloring was described by Lu et al. [178]. They also investigated the use of orbit traps (a popular coloring method in fractal art) to render symmetric attractors [179]. Both techniques are illustrated in Figure 6.15.

Some work has also been done on rendering fractals in other spaces or based on other symmetry groups. Adcock et al. build iterated function systems from isometries in the hyperbolic plane, producing attractors in the Poincaré disk model with symmetries based on the standard hyperbolic groups  $[p, q]$  [2]. Chung et al. use transformations of the plane inspired by Escher’s drawings to create images that look as if they had been rendered in the Poincaré half-plane and disk models [38]. (Their paper claims to be inspired by Escher’s “Smaller and

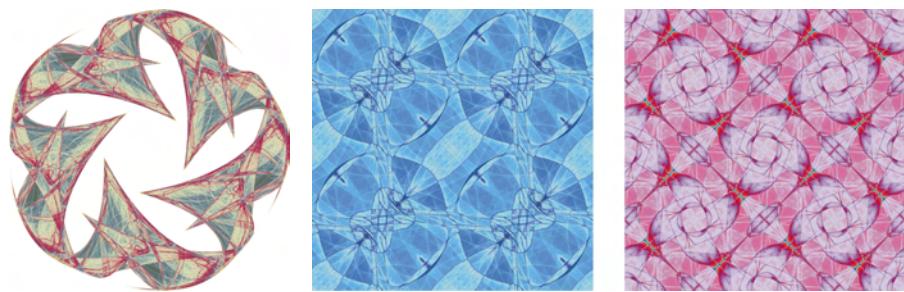


Fig. 6.14 A chaotic attractor with finite cyclic symmetry, and two attractors with wallpaper symmetry. All images were produced by Field.

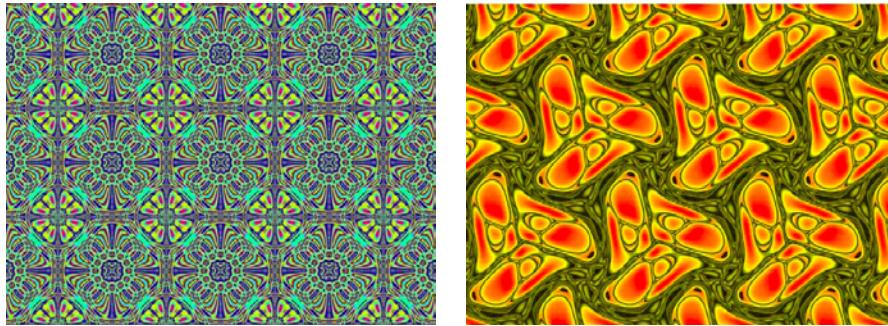


Fig. 6.15 Two additional chaotic attractors with wallpaper symmetry, both by Lu et al. The image on the left demonstrates their technique based on finding invariant functions; the image on the right is colored using orbit traps.

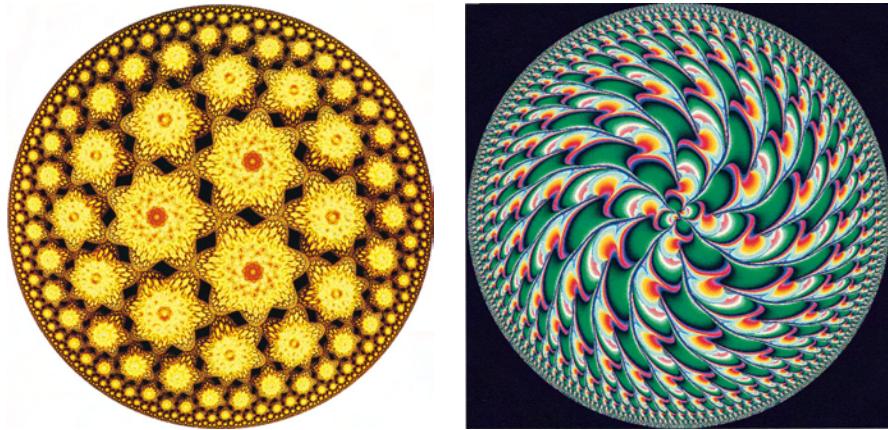


Fig. 6.16 Two additional fractal images with hyperbolic symmetry. The design on the left is an iterated function system by Adcock et al., and the one on the right is a “Smaller and Smaller” design by Chung et al.

“Smaller”, though it would be more accurate to point to “Square Limit” and an untitled woodcut usually referred to as “Regular Division of the Plane VI” as sources.) Examples based on these two techniques are shown in Figure 6.16. Ye et al. produce fractal drawings based on the modular group in two dimensions [285] and the extended Picard group in three dimensions [284].

# 7

---

## Summary

---

Symmetry, or deviation from symmetry, has been a fascinating topic across many different fields including science, engineering and the arts. Detection of symmetry, as either a pure intellectual pursuit of programmable perception or as a practical tool for data compression, has also fascinated computer scientists for decades, leading to the effort that we call *Computational Symmetry* [152]. In this survey, we have provided the first, up-to-date (albeit partial) account of research activities in the emerging field of *Computational Symmetry* in both computer vision and computer graphics. Beyond continuous groups, we found great potential in the computational formalism, efficiency and applications of discrete symmetries and symmetry groups. Recent papers from related research fields echo this observation (Figure 1.4), and the following extract from a 2006 SIGGRAPH paper [190] resonates our appreciation of the importance and relevance of symmetry in the computational sciences:

Symmetry is an essential and ubiquitous concept in nature, science, and art. For example, in geometry, the Erlanger program of Felix Klein has fueled for over

a century mathematicians interest in invariance under certain group actions as a key principle for understanding geometric spaces. Numerous biological, physical, or man-made structures exhibit symmetries as a fundamental design principle or as an essential aspect of their function. Whether by evolution or design, symmetry implies certain economies and efficiencies of structure that make it universally appealing. Symmetry also plays an important role in human visual perception and aesthetics. Arguably much of the understanding of the world around us is based on the perception and recognition of shared or repeated structures, and so is our sense of beauty [47].

The state of the art in computational symmetry, although demonstrating potential in recent years, has yet to reach a level to make fundamental contributions to the fields of computer vision and computer graphics. The progress made has been mostly scattered, un-sustained and less than systematic. As pointed out in our Introduction (in particular, Sections 1.2 and 1.3), this is an inherently difficult subfield, challenged by the alarming discrepancies between the formal elegance of group theory and the imperfect/noisy/ambiguous/distorted data that it must be applied to, as well as the limited representation power of computers. Therefore, to make real progress, we need to be particularly creative and systematic. Here is an immediate action list for those of us who are interested in making solid advances in computational symmetry:

- (1) have a consistent and unified taxonomy of symmetry and symmetry group types;
- (2) provide publicly available data sets with ground truth (e.g., the publicly available *PSU NRT image database*: <http://vivid.cse.psu.edu/texturedb/gallery/>);
- (3) make research code or executables publicly available;
- (4) establish benchmarks for symmetry detection and analysis algorithms;

- (5) organize symmetry detection algorithm competitions, workshops and tutorials.

Obviously, developing robust symmetry detection algorithms for various real world data sets tops the list of immediate tasks. Here is a short list of additional tangible research directions:

- symmetry-based perceptual organization, grouping and segmentation (although many ideas were suggested in 1970–1980s, few practical algorithms exist);
- symmetry-based registration, especially for biomedical objects;
- symmetry-based object recognition, an obvious direction that is hindered by many (computational) obstacles.

Strategies needed for handling real world complexity have to be developed to deal with:

- hidden structures: distorted, disguised symmetries in noisy data (Figure 1.5),
- ambiguity: clearly categorized mathematical symmetries (Figure 2.1) can become incredibly confusing in digitized real data, and
- symmetry model selection: in particular, the issue of subgroup relations among symmetry groups, raised by Kanatani in [110] and addressed only for gait frieze patterns/groups by Liu et al. [158, 159].

Given the tremendous insights and wisdom that have been gained from the concept of symmetry throughout the history of science, we are hopeful that *computational symmetry* will make a similar impact to the computational sciences, particularly in computer vision and computer graphics research.

## Acknowledgments

---

We would like to thank Yi Ma and James Hays for their respective contributions to this survey. Thanks also go to Jingchen Liu, Kyle Brocklehurst, and Seungkyu Lee for collecting citations on symmetry related papers, and Jingchen Liu for creating Figure 1.4. Thanks to Michael Field, Ergun Akleman, and Doug Dunham. Thanks also to Doris Schattschneider, D. W. Crowe, Marjorie Senechal, Takeo Kanade, Jeff Cohn, D. K. Washburn, the late Robin Popplestone and the late H. S. M. Coxeter.

## References

---

- [1] S. J. Abas and A. S. Salman, *Symmetries of Islamic Geometrical Patterns*. World Scientific, 1995.
- [2] B. M. Adcock, K. C. Jones, C. A. Reiter, and L. M. Vislocky, “Iterated function systems with symmetry in the hyperbolic plane,” *Computers & Graphics*, vol. 24, no. 5, pp. 791–796, October 2000.
- [3] E. Akleman, O. Ozener, and C. Yuksel, “On a family of symmetric, connected and high genus sculptures,” in *Proceedings of Bridges 2006*, 2006.
- [4] H. Alexander, “The computer/plotter and the 17 ornamental design types,” in *SIGGRAPH '75: Proceedings of the 2nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 160–167, New York, NY, USA: ACM Press, 1975.
- [5] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl, “Congruence, similarity and symmetries of geometric objects,” *ACM Journal of Computing*, vol. 4, pp. 308–315, 1987.
- [6] S. Alvarez, P. Alemany, D. Casanova, J. Cirera, M. Llunell, and D. Avnir, “Shape maps and polyhedral interconversion paths in transition metal chemistry,” *Coordination Chemistry Review*, vol. 249, pp. 1693–1708, 2005.
- [7] R. Arnheim, *Art and Visual Perception: A Psychology of the Creative Eye*. University of California Press, 2004.
- [8] M. J. Atallah, “On symmetry detection,” *IEEE Transaction on Computers*, vol. c-34, no. 7, pp. 663–666, July 1985.
- [9] D. Avnir, O. Katzenelson, and H. Hel-Or, “Further comments on the chirality of large random objects,” *Chemistry — A European Journal*, vol. 2, pp. 744–746, 1996.

- [10] D. Avnir, O. Katzenelson, S. Keinan, M. Pinsky, Y. Pinto, Y. Salomon, and H. Hel-Or, “The measurement of symmetry and chirality: Conceptual aspects,” in *Concepts in Chemistry*, (D. H. Rouvray, ed.), pp. 283–324, Research Studies Press, Somerset, 1997.
- [11] D. Avnir and A. Y. Meyer, “Quantifying the degree of molecular shape deformation. A chirality measure,” *Journal of Molecular Structure (Theochem)*, vol. 94, pp. 211–222, 1991.
- [12] P. N. Belhumeur, J. P. Jespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *PAMI*, vol. 19, no. 7, pp. 711–720, July 1997.
- [13] L. Bieberbach, “Über die Bewegungsgruppen der n-dimensionalen Euklidischen Räume mit einem endlichen Fundamentalbereich,” *Göttinger Nachrichten*, pp. 75–84, 1910.
- [14] I. Biederman, “Human image understanding: Recent research and a theory,” *Computer vision, graphics, and image processing*, vol. 32, pp. 29–73, 1985.
- [15] J. Bigün, “Pattern recognition by detection of local symmetries,” in *Pattern Recognition and Artificial Intelligence*, (E. S. Gelsema and L. N. Kanal, eds.), pp. 75–90, Elsevier, 1988.
- [16] J. Bigün and G. Granlund, “Central symmetry modelling,” in *Third European Signal Processing Conference*, pp. 883–886, The Hague, September 1986.
- [17] J. Bigün and G. Granlund, “Optimal orientation detection of linear symmetry,” in *First International Conference on Computer Vision*, pp. 433–438, London, June 1987.
- [18] G. D. Birkhoff, *Aesthetic Measure*. Cambridge, MA: Harvard University Press, 1932.
- [19] J. F. Blinn, “The 3d kaleidoscope,” *IEEE Computer Graphics and Applications*, vol. 8, no. 5, pp. 9–14, 1988.
- [20] H. Blum, “Biological shape and visual science (part i),” *Journal of Theoretical Biology*, vol. 38, pp. 205–287, 1973.
- [21] H. Blum and R. N. Nagel, “Shape description using weighted symmetric axis features,” *Pattern Recognition*, vol. 10, pp. 167–180, 1978.
- [22] A. F. Blumenthal, L. S. Davis, and A. Rosenfeld, “Detecting natural “plateaus” in one-dimensional patterns,” *IEEE Transactions on Computers*, vol. 26, no. 2, pp. 178–179, 1977.
- [23] F. L. Bookstein, “The line-skeleton,” *Computer Graphics and Image Processing*, vol. 11, pp. 123–137, 1979.
- [24] M. Brady and H. Asada, “Smoothed local symmetries and their implementation,” *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 36–61, Fall 1984.
- [25] K. Brakke, “The surface evolver,” *Experimental Mathematics*, vol. 1, no. 2, pp. 141–165, 1992.
- [26] R. A. Brooks, “Symbolic reasoning among 3D models and 2D images,” *Artificial Intelligence*, vol. 17, pp. 285–348, 1981.
- [27] V. Bruce and M. Morgan, “Violations of symmetry and repetition in visual patterns,” *Perception*, vol. 4, pp. 239–249, 1975.

- [28] R. P. Burn, *Groups: A path to geometry*. Cambridge, New York, Melbourne: Cambridge University Press, 1985.
- [29] F. W. Burton, J. G. Kollins, and N. A. Alexandridis, “An implementation of the exponential pyramid data structure with application to determination of symmetries in pictures,” *Computer Vision, Graphics, and Image Processing*, vol. 25, pp. 218–225, 1984.
- [30] S. Carlsson, “Symmetry in perspective,” in *ECCV 1998*, vol. 1, pp. 249–263, 1998.
- [31] T. J. Cham and R. Chipolla, “Symmetry detection through local skewed symmetries,” *Image and Vision Computing*, vol. 13, no. 5, pp. 439–450, 1995.
- [32] E. Chastain and Y. Liu, “Quantified symmetry for entorhinal spatial maps,” *Neurocomputing*, vol. 70, no. 10–12, pp. 1723–1727, June 2007.
- [33] P. Chen, J. H. Hays, S. Lee, M. Park, and Y. Liu, “A quantitative evaluation of symmetry detection algorithms,” Technical Report PSU-CSE-07011 (also listed as Tech. report CMU-RI-TR-07-36), The Pennsylvania State University, State College, PA, 2007.
- [34] K. Cho and S. M. Dunn, “Hierarchical local symmetries,” *Pattern Recognition Letters*, vol. 12, pp. 343–347, 1991.
- [35] S. L. Chou, J. C. Lin, and W. H. Tsai, “Fold principal axis-a new tool for defining the orientations of rotationally symmetric shapes,” *Pattern Recognition Letters*, vol. 12, pp. 109–115, 1991.
- [36] W. W. Chow, “Automatic generation of interlocking shapes,” *Computer Graphics and Image Processing*, vol. 9, no. 4, pp. 333–353, April 1979.
- [37] W. W. Chow, “Interlocking shapes in art and engineering,” *Computer Aided Design*, vol. 12, no. 1, pp. 29–34, January 1980.
- [38] K. W. Chung, H. S. Y. Chan, and B. N. Wang, ““smaller and smaller” from dynamics,” *Computers & Graphics*, vol. 22, no. 4, pp. 527–536, August 1998.
- [39] J. J. Collins and I. N. Stewart, “Coupled nonlinear oscillators and the symmetries of animal gaits,” *Journal of Nonlinear Science*, vol. 3, pp. 349–392, 1993.
- [40] R. W. Conners and C. T. Ng, “Developing a quantitative model of human preattentive vision,” *SMC*, vol. 19, no. 6, pp. 1384–1407, 1989.
- [41] J. H. Conway, H. Burgiel, and C. Goodman-Strauss, *The Symmetries of Things*. A K Peters, Wellesley, Massachusetts, 2008.
- [42] H. S. M. Coxeter, *Introduction to Geometry*. Wiley, New York, 2nd Edition, 1980.
- [43] H. S. M. Coxeter, *Personal Communication*. 2000.
- [44] H. S. M. Coxeter and W. O. J. Moser, *Generators and Relations for Discrete Groups*. New York: Springer-Verlag, 4th Edition, 1980.
- [45] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based inpainting,” in *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2003.
- [46] R. Cutler and L. Davis, “Robust real-time periodic motion detection, analysis and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Video Surveillance*, vol. 22, no. 8, August 2000.
- [47] D’Arcy W. Thompson, *On Growth and Form*. Cambridge University Press, 1961.

- [48] L. S. Davis, “Understanding shape: Angles and sides,” *IEEE Transactions on Computers*, vol. 26, no. 3, pp. 236–242, 1977.
- [49] F. De la Toore and M. J. Black, “A framework for robust subspace learning,” *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.
- [50] O. Delgado-Friedrichs, “Data structures and algorithms for tilings i,” *Theoretical Computer Science*, vol. 303, no. 2–3, pp. 431–445, 2003.
- [51] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart, “Spectral surface quadrangulation,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, vol. 25, no. 3, pp. 1057–1066, 2006.
- [52] D. Dunham, “Artistic patterns in hyperbolic geometry,” in *Bridges 1999 Proceedings*, (R. Sarhangi, ed.), pp. 139–149, 1999.
- [53] D. Dunham, “Hyperbolic celtic knot patterns,” in *Bridges 2000 Proceedings*, (R. Sarhangi, ed.), 2000.
- [54] D. Dunham, “Hyperbolic Islamic patterns — a beginning,” in *Bridges 2001 Proceedings*, (R. Sarhangi, ed.), pp. 247–254, 2001.
- [55] D. Dunham, “An algorithm to generate repeating hyperbolic patterns,” in *Proceedings of ISAMA 2007*, pp. 111–118, 2007.
- [56] D. Dunham, J. Lindgren, and D. Witte, “Creating repeating hyperbolic patterns,” in *SIGGRAPH '81: Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 215–223, New York, NY, USA: ACM Press, 1981.
- [57] D. J. Dunham, “Creating hyperbolic escher patterns,” in H. S. M. Coxeter and M. Emmer and R. Penrose and M. L. Teuber and *M.C. Escher: Art and Science*, pp. 241–247, Elsevier Science Publishers B.V., 1986.
- [58] P. Eades, “Symmetry finding algorithms,” in *Computational Morphology*, (G. T. Toussaint, ed.), pp. 41–51, Elsevier, 1988.
- [59] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston, *Word Processing in Groups*. Natick, MA, USA: A. K. Peters, Ltd., 1992.
- [60] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [61] E. S. Fedorov, “The elements of the study of figures. [Russian] (2) 21,” in *Zapiski Imperatorskogo S. Peterburgskogo Mineralogichesko Obshchestva [Proc. S. Peterb. Mineral. Soc.]*, pp. 1–289, 1885.
- [62] E. S. Fedorov, “Symmetry in the plane. [Russian] (2) 28,” in *Zapiski Imperatorskogo S. Peterburgskogo Mineralogichesko Obshchestva [Proc. S. Peterb. Mineral. Soc.]*, pp. 345–390, 1891.
- [63] E. S. Fedorov, “Symmetry of finite figures. [Russian] (2) 28,” in *Zapiski Imperatorskogo S. Peterburgskogo Mineralogichesko Obshchestva [Proc. S. Peterb. Mineral. Soc.]*, pp. 1–146, 1891.
- [64] M. Field, “Designer chaos,” *Computer-Aided Design*, vol. 33, no. 3, pp. 349–365, April 2001.
- [65] M. Field, “Dynamics, chaos, and design,” in *The Visual Mind II*, (M. Emmer, ed.), pp. 473–493, Cambridge, Massachusetts: The MIT Press, 2005.

- [66] M. Field and M. Golubitsky, *Symmetry in Chaos: A Search for Pattern in Mathematics, Art, and Nature*. Oxford University Press, 1992.
- [67] M. Fleck, “Local rotational symmetries,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 332–337, 1986.
- [68] D. A. Forsyth, “Shape from texture without boundaries,” in *ECCV*, pp. 225–239, 2002.
- [69] J. A. Gallian, *Contemporary Abstract Algebra*. D.C. Heath and Company, Lexington, MA, 1986.
- [70] J. M. Gauch and S. M. Pizer, “Image description via the multiresolution intensity axis of symmetry,” in *International Conference on Computer Vision*, pp. 269–274, 1988.
- [71] J. M. Gauch and S. M. Pizer, “The intensity axis of symmetry and its application to image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence Archive*, vol. 15, no. 8, pp. 753–770, August 1993.
- [72] J. J. Gibson, *The Perception of the Visual World*. Houghton Mifflin, Boston, Massachusetts, 1950.
- [73] G. Gilat, “Chiral coefficient — a measure of the amount of structural chirality,” *Journal of Physics A: Mathematical and Theoretical*, vol. 22, pp. 545–545, 1989.
- [74] M. Giurfa, B. Eichmann, and R. Menzel, “Symmetry perception in an insect,” *Nature*, vol. 382, no. 6590, pp. 458–461, August 1996.
- [75] R. Glachet, M. Dhome, and J. T. Lapreste, “Finding the perspective projection of an axis of revolution,” *Pattern Recognition Letters*, vol. 12, pp. 693–700, 1991.
- [76] A. Glassner, “Hierarchical textures,” *IEEE Computer Graphics and Applications*, vol. 20, no. 4, pp. 104–109, 2000.
- [77] J. H. Graham, S. Raz, H. Hel-Or, and E. Nevo, “Fluctuating asymmetry: Methods, theory, and applications,” *Symmetry*, vol. 2, no. 2, pp. 466–540, 2010. Special Issue “Feature Papers: Symmetry Concepts and Applications,” doi: 10.3390/sym2020466, www.mdpi.com/journal/symmetry (ISSN 2073-8994; CODEN: SYMMAM).
- [78] K. Grammer and R. Thornhill, “Human facial attractiveness and sexual selection: The role of symmetry and averageness,” *Journal of Comparative Psychology*, vol. 108, pp. 233–242, 1994.
- [79] M. J. Greenberg, *Euclidean and Non-Euclidean Geometries: Development and History*. W. H. Freeman and Company, 3rd Edition, 1993.
- [80] G. Griffin, A. Holub, and P. Perona, “Caltech-256 object category dataset,” Technical Report 7694, California Institute of Technology, 2007.
- [81] A. D. Gross and T. E. Boult, “Analyzing skewed symmetries,” *International Journal of Computer Vision*, vol. 13, no. 1, pp. 91–111, September 1994.
- [82] B. Grünbaum, “Measures of symmetry for convex sets,” *Proceedings of Symposia in Pure Mathematics: American Mathematical Society*, vol. 7, pp. 233–270, 1963.
- [83] B. Grünbaum, “Periodic ornamentation of the fabric plane: Lessons from Peruvian fabrics,” *Symmetry*, vol. 1, no. 1, pp. 45–68, 1990.

- [84] B. Grünbaum and G. C. Shephard, *Tilings and Patterns*. New York: W. H. Freeman and Company, 1987.
- [85] B. Grünbaum and G. C. Shephard, “Interlace patterns in Islamic and Moorish art,” in *The Visual Mind*, (M. Emmer, ed.), pp. 147–155, Cambridge, Massachusetts: The MIT Press, 1993.
- [86] C. Gunn, “Visualizing hyperbolic geometry,” in *Computer Graphics and Mathematics*, pp. 299–313, Springer-Verlag, 1992.
- [87] C. Gunn, “Discrete groups and visualization of three-dimensional manifolds,” in *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 255–262, New York, NY, USA: ACM Press, 1993.
- [88] H. Hafting, M. Fyhn, S. Molden, E. I. Moser, and M. Moser, “Microstructure of a spatial map in the entorhinal cortex,” *Nature*, vol. 436, pp. 801–806, August 2005.
- [89] J. Han, S. J. McKenna, and R. Wang, “Regular texture analysis as statistical model selection,” in *ECCV08*, 2008.
- [90] G. W. Hart, “The geometric aesthetic,” in *The Visual Mind II*, (M. Emmer, ed.), pp. 215–234, Cambridge, Massachusetts: The MIT Press, 2005.
- [91] G. W. Hart, “Orderly tangles revisited,” in *Proceedings of Bridges 2005*, 2005.
- [92] G. W. Hart, “Modular kirigami,” in *Proceedings of Bridges 2007*, 2007.
- [93] G. W. Hart, “Symmetric sculpture,” *Journal of Mathematics and the Arts*, vol. 1, no. 1, pp. 21–28, March 2007.
- [94] B. Hausmann, B. Slopinaka, and H.-P. Seidel, “Exploring plane hyperbolic geometry,” in *Visualization and Mathematics*, (H.-C. Hege and K. Polthier, eds.), pp. 21–36, Springer, 1997.
- [95] J. Hays, M. Leordeanu, A. Efros, and Y. Liu, “Discovering texture regularity as a higher-order correspondence problem,” in *European Conference on Computer Vision (ECCV'06)*, 2006.
- [96] H. Heesch and O. Kienzle, *Flächenschluss*. Springer Verlag, 1963.
- [97] Y. Hel-Or, S. Peleg, and D. Avnir, “Characterization of right handed and left handed shapes,” *Computer Vision, Graphics, and Image Processing*, vol. 53, no. 2, 1991.
- [98] N. F. M. Henry and K. Lonsdale, eds., *International Tables for X-ray Crystallography*, vol. 1, Symmetry Groups, The International Union of Crystallography. England: The Kynoch Press, 1969.
- [99] P. T. Highnam, “Optimal algorithms for finding the symmetries of planar point sets,” *Information Processing Letters*, vol. 22, pp. 219–222, 1986.
- [100] A. Holden, *Shapes, Space, and Symmetry*. Dover, 1991.
- [101] W. Hong, Y. Ma, and Y. Yu, “Reconstruction of 3-d symmetric curves from perspective images without discrete features,” in *Proceedings of the European Conference on Computer Vision*, 2004.
- [102] W. Hong, A. Y. Yang, and Y. Ma, “On symmetry and multiple view geometry: Structure, pose and calibration from a single image,” *International Journal of Computer Vision*, vol. 60, no. 3, pp. 241–265, 2004.
- [103] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. IT-20, pp. 179–187, February 1962.

- [104] K. Huang, W. Hong, and Y. Ma, “Symmetry-based photo editing,” *Pattern Recognition*, vol. 27, no. 12, pp. 1945–1959, 2005.
- [105] K. Huang, A. Y. Yang, W. Hong, and Y. Ma, “Large baseline matching and reconstruction from symmetry cells,” in *Proceedings of the International Conference on Robotics and Automation*, 2004.
- [106] O. Jones (1809–1874), *The Grammar of Ornament*. London: Quaritch, 1910.
- [107] M. Jorgensen, “A dynamic em algorithm for estimating mixture proportions,” *Statistics and Computing*, vol. 9, no. 4, pp. 299–302, 1999.
- [108] B. Julesz, “Visual pattern discrimination,” *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 84–92, 1962.
- [109] T. Kanade, “Recovery of the 3-dimensional shape of an object from a single view,” *Artificial Intelligence*, vol. 17, pp. 75–116, 1981.
- [110] K. Kanatani, “Comments on “Symmetry as a Continuous Feature”,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 246–247, 1997.
- [111] D. R. Kanis, J. S. Wong, T. J. Marks, M. A. Ratner, H. Zabrodsky, S. Keinan, and D. Avnir, “Continuous symmetry analysis of hyperpolarizabilities. Characterization of second order non-linear optical response of distorted benzene,” *Journal of Physical Chemistry*, vol. 99, pp. 11061–11066, 1995.
- [112] C. S. Kaplan, “Voronoi diagrams and ornamental design,” in *ISAMA '99: The First Annual Symposium of the International Society for the Arts, Mathematics, and Architecture*, pp. 277–283, 1999.
- [113] C. S. Kaplan, “Islamic star patterns from polygons in contact,” in *GI '05: Proceedings of the 2005 conference on Graphics Interface*, pp. 177–185, Canadian Human-Computer Communications Society, 2005.
- [114] C. S. Kaplan, *Introductory Tiling Theory for Computer Graphics*. Morgan & Claypool, 2009.
- [115] C. S. Kaplan, “Semiregular patterns on surfaces,” in *NPAR '09: Proceedings of the 7th International Symposium on Non-photorealistic Animation and Rendering*, pp. 35–39, New York, NY, USA: ACM Press, 2009.
- [116] C. S. Kaplan and D. H. Salesin, “Escherization,” in *SIGGRAPH '00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 499–510, New York, NY, USA: ACM Press/Addison-Wesley Publishing Co, 2000.
- [117] C. S. Kaplan and D. H. Salesin, “Dihedral escherization,” in *GI '04: Proceedings of Graphics Interface 2004*, pp. 255–262, Canadian Human-Computer Communications Society. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004.
- [118] C. S. Kaplan and D. H. Salesin, “Islamic star patterns in absolute geometry,” *ACM Transactions on Graphics*, vol. 23, no. 2, pp. 97–119, 2004.
- [119] H. Karam and M. Nakajima, “Islamic symmetric pattern generation based on group theory,” in *CGI '99: Proceedings of the International Conference on Computer Graphics*, p. 112, Washington, DC, USA: IEEE Computer Society, 1999.
- [120] O. Katzenelson, H. Hel-Or, and D. Avnir, “Chirality of large supramolecular structures,” *Chemistry — A European Journal*, vol. 2, pp. 174–181, 1996.

- [121] D. C. Kay, *College Geometry*. Holt, Rinehart and Winston, Inc., 1st Edition, 1969.
- [122] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser, “A reflective symmetry descriptor,” in *Proceedings of Seventh European Conference on Computer Vision*, pp. 642–656, Springer, May 2002.
- [123] S. Keinan and D. Avnir, “Quantitative symmetry in structure-activity correlations. The near c2 symmetry of inhibitor/hiv-protease complexes,” *Journal of the American Chemical Society*, vol. 122, pp. 4378–4384, 2000.
- [124] S. Keinan and D. Avnir, “Continuous symmetry analysis of tetrahedral/planar distortions. Copper chlorides and other ab4 species,” *Inorganic Chemistry*, vol. 40, pp. 318–323, 2001.
- [125] A. Khodakovsky, N. Litke, and P. Schröder, “Globally smooth parameterizations with low distortion,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, vol. 22, no. 3, pp. 350–357, July 2003.
- [126] D. Knuth, J. Morris, and V. Pratt, “Fast pattern matching in strings,” *SIAM Journal of Computing*, vol. 6, pp. 322–350, 1977.
- [127] T. Korah and D. Rasmussen, “Analysis of building textures for reconstructing partially occluded facades,” in *ECCV08*, 2008.
- [128] R. Kowner and R. Thornhill, “The imperfect organism: On the concept of asymmetry and its significance in humans, non-human animals and plants,” *Symmetry, Culture and Science*, vol. 10, pp. 227–243, 2002.
- [129] J. L. Krahe, “Detection of symmetric and radial structures in images,” in *International Conference on Pattern Recognition*, pp. 947–950, 1986.
- [130] A. Kuehnle, “Symmetry-based recognition of vehicle rears,” *Pattern Recognition Letters*, vol. 12, pp. 249–258, 1991.
- [131] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*. McGraw-Hill Irwin, 2005.
- [132] S. Lee, R. T. Collins, and Y. Liu, “Rotation symmetry group detection via frequency analysis of frieze-expansions,” in *IEEE Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8, 2008.
- [133] S. Lee and Y. Liu, “Curved glide-reflection symmetry detection,” in *Computer Vision and Pattern Recognition (CVPR09)*, pp. 1–8, 2009.
- [134] S. Lee and Y. Liu, “Skewed rotation symmetry group detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, Accepted for publication, 2009.
- [135] Y. Lei and K. Wong, “Detection and localisation of reflectional and rotational symmetry under weak perspective projection,” *Pattern Recognition*, vol. 32, no. 2, pp. 167–180, 1999.
- [136] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *ICCV*, 2005.
- [137] T. Leung and J. Malik, “Detecting, localizing and grouping repeated scene elements,” in *ECCV LNCS 1064*, vol. 1, pp. 546–555, 1996.
- [138] T. K. Leung and J. Malik, “Detecting, localizing and grouping repeated scene elements from an image,” in *Proceedings of European Conference on Computer Vision (ECCV)*, pp. 546–555, 1996.

- [139] G. Levi and U. Montanari, "A grey-weighted skeleton," *Information Control*, vol. 17, pp. 62–91, 1970.
- [140] A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Multiscale symmetric part detection and grouping," in *ICCV*, 2009.
- [141] T. S. Levitt, "Domain independent object description and decomposition," in *Proceedings of American Association of Artificial Intelligence*, pp. 207–211, 1984.
- [142] S. Levy, "Automatic generation of hyperbolic tilings," in *The Visual Mind*, (M. Emmer, ed.), pp. 165–170, Cambridge, Massachusetts: The MIT Press, 1993.
- [143] M. Leyton, "A process-grammar for shape," *Artificial Intelligence*, vol. 34, pp. 213–247, 1988.
- [144] M. Leyton, *Symmetry, Causality, Mind*. Cambridge, Massachusetts: The MIT Press, 1992.
- [145] H.-C. Lin, L.-L. Wang, and S.-N. Yang, "Extracting periodicity of a regular texture based on autocorrelation functions," *Pattern Recognition Letters*, vol. 18, pp. 433–443, 1997.
- [146] W. C. Lin, J. H. Hays, C. Wu, V. Kwatra, and Y. Liu, "Quantitative evaluation on near regular texture synthesis," in *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR'06)*, vol. 1, pp. 427, 434, New York City: IEEE Computer Society Press, June 2006.
- [147] W. C. Lin and Y. Liu, "Tracking dynamic near-regular textures under occlusion and rapid movements," in *9th European Conference on Computer Vision (ECCV'06)*, vol 2, pp. 44–55, 2006.
- [148] W. C. Lin and Y. Liu, "A lattice-based mrf model for dynamic near-regular texture tracking," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 777–792, May 2007.
- [149] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 722–733, 1996.
- [150] J. Liu and Y. Liu, "Multi-target tracking of time-varying spatial patterns," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'10)*, pp. 1–8, IEEE Computer Society Press, 2010.
- [151] Y. Liu, "Symmetry groups in robotic assembly planning," PhD thesis, University of Massachusetts, Amherst, MA., September 1990.
- [152] Y. Liu, "Computational Symmetry," in *Symmetry 2000*, vol. 80, chapter 21, (I. Hargittai and T. C. Laurent, eds.), pp. 231–245, Wenner-Gren International Series, Portland, London, ISBN I 85578 149 2, 2002.
- [153] Y. Liu, T. Belkina, H. Hays, and R. Lublinerman, "Image de-fencing," in *IEEE Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8, 2008.
- [154] Y. Liu and R. T. Collins, "A Computational Model for Repeated Pattern Perception using Frieze and Wallpaper Groups," in *Computer Vision and Pattern Recognition Conference (CVPR'00)*, pp. 537–544, Hilton Head, SC, June 2000. IEEE Computer Society Press. ([http://www.ri.cmu.edu/pubs/pub\\_3302.html](http://www.ri.cmu.edu/pubs/pub_3302.html)).

- [155] Y. Liu and R. T. Collins, "Frieze and wallpaper symmetry groups classification under affine and perspective distortion," Technical Report CMU-RI-TR-98-37, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1998.
- [156] Y. Liu and R. T. Collins, "Skewed Symmetry Groups," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pp. 872–879, Kauai, HI: IEEE Computer Society Press. ([http://www.ri.cmu.edu/pubs/pub\\_3815.html](http://www.ri.cmu.edu/pubs/pub_3815.html)), December 2001.
- [157] Y. Liu, R. T. Collins, and W. E. Rothfus, "Robust Midsagittal Plane Extraction from Normal and Pathological 3D Neuroradiology Images," *IEEE Transactions on Medical Imaging*, vol. 20, no. 3, pp. 175–192, March 2001.
- [158] Y. Liu, R. T. Collins, and Y. Tsin, "Gait sequence analysis using frieze patterns," in *ECCV 02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pp. 657–671, 2002.
- [159] Y. Liu, R. T. Collins, and Y. Tsin, "A computational model for periodic pattern perception based on frieze and wallpaper groups," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 354–371, March 2004.
- [160] Y. Liu and F. Dellaert, "A Classification-based Similarity Metric for 3D Image Retrieval," in *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR'98)*, pp. 800–807, Santa Barbara: IEEE Computer Society Press, June 1998.
- [161] Y. Liu, F. Dellaert, W. E. Rothfus, A. Moore, J. Schneider, and T. Kanade, "Classification-driven pathological neuroimage retrieval using statistical asymmetry measures," in *International Conference on Medical Imaging Computing and Comptuer Assisted Intervention (MICCAI 2001)*, pp. 655–665, Springer, October 2001.
- [162] Y. Liu, J. Hays, Y.-Q. Xu, and H.-Y. Shum, "Digital papercutting," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, p. 99, New York, NY, USA: ACM Press, 2005.
- [163] Y. Liu and W. C. Lin, "Deformable texture: the irregular-regular-irregular cycle," in *Texture 2003, The 3rd International Workshop on Texture Analysis and Synthesis*, pp. 65–70, Nice, France, October 2003.
- [164] Y. Liu, W. C. Lin, and J. Hays, "Near-regular texture analysis and manipulation," *ACM Transactions on Graphics (SIGGRAPH)*, vol. 23, no. 3, pp. 368–376, August 2004.
- [165] Y. Liu and J. Palmer, "A quantified study of facial asymmetry in 3d faces," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 222–229, IEEE, 2003.
- [166] Y. Liu and R. Popplestone, "A Group Theoretical Formalization of Surface Contact," *International Journal of Robotics Research*, vol. 13, no. 2, pp. 148–161, April 1994.
- [167] Y. Liu, K. Schmidt, J. Cohn, and S. Mitra, "Facial asymmetry quantification for expression invariant human identification," *Computer Vision and Image Understanding Journal*, vol. 91, no. 1/2, pp. 138–159, July/August 2003. Special issue on face recognition, (Martinez, Yang and Kriegman, Eds.).

- [168] Y. Liu, K. Schmidt, J. Cohn, and R. L. Weaver, "Facial asymmetry quantification for expression invariant human identification," in *International Conference on Automatic Face and Gesture Recognition (FG'02)*, May 2002.
- [169] Y. Liu, L. Teverovskiy, O. Carmichael, R. Kikinis, M. Shenton, C. S. Carter, V. A. Stenger, S. Davis, H. Aizenstein, J. Becker, O. Lopez, and C. Meltzer, "Discriminative mr image feature analysis for automatic schizophrenia and alzheimer's disease classification," in *7th International Conference on Medical Imaging Computing and Comptuer Assisted Intervention (MICCAI 2004)*, pp. 378–385, Springer, October 2004.
- [170] Y. Liu, L. Teverovskiy, O. Lopez, H. Aizenstein, J. Becker, and C. Meltzer, "Discovery of "biomarkers" for Alzheimer's Disease prediction from structural mr images," in *2002 IEEE International Symposium on Biomedical Imaging: Macro to Nano*, pp. 1344–1347, April 2007.
- [171] Y. Liu and Y. Tsin, "The promise and perils of near-regular texture," in *Texture 2002*, pp. 657–671, Copenhagen, Denmark: in conjunction with ECCV'02, May 2002.
- [172] Y. Liu, Y. Tsin, and W. Lin, "The promise and perils of near-regular texture," *International Journal of Computer Vision*, vol. 62, no. 1–2, pp. 145, 159, April 2005.
- [173] A. Lobay and D. A. Forsyth, "Recovering shape and irradiance maps from rich dense texton fields," in *CVPR*, pp. 400–406, 2004.
- [174] L. Loreto, R. Farinato, and M. Tonetti, "Crystallography and plane ornaments: Interactive multi-window computer graphics," in *The Visual Mind*, (M. Emmer, ed.), pp. 187–192, Cambridge, Massachussetts: The MIT Press, 1993.
- [175] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Kluwer Academic, 1985.
- [176] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91, 110, 2004.
- [177] G. Loy and J. Eklundh, "Detecting symmetry and symmetric constellations of features," in *European Conference on Computer Vision (ECCV'04), Part II, LNCS 3952*, pp. 508, 521, May 2006.
- [178] J. Lu, Z. Ye, and Y. Zou, "Automatic generation of colorful patterns with wallpaper symmetries from dynamics," *Visual Computing*, vol. 23, no. 6, pp. 445–449, 2007.
- [179] J. Lu, Z. Ye, Y. Zou, and R. Ye, "Orbit trap rendering methods for generating artistic images with crystallographic symmetries," *Computers & Graphics*, vol. 29, no. 5, pp. 787–794, October 2005.
- [180] Y. Ma, K. Huang, R. Vidal, J. Kosecka, and S. S. Sastry, "Rank conditions on the multiple view matrix," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 115–137, 2004.
- [181] S. MacLane and G. Birkhoff, *Algebra*. New York: MacMillan Publishing Co., Inc., 2nd Edition, 1979.
- [182] D. Malin and K. Roucoux, *Heaven & Earth: Unseen by the Naked Eye*. Phaidon Press Limited, October 1 2004.

- [183] G. Marola, “On the detection of the axes of symmetry of symmetric and almost symmetric planar images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 1, pp. 104–108, 1989.
- [184] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” in *British Machine Vision Conference*, 2002.
- [185] B. Micusik, H. Wildenauer, and M. Kosecká, “Detection and matching of rectilinear structures,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’08)*, IEEE Computer Society Press, June 2008.
- [186] W. Miller, *Symmetry Groups and their Applications*. London: Academic Press, 1972.
- [187] D. Milner, S. Raz, H. Hel-Or, D. Keren, and E. Nevo, “Measuring symmetry without landmarks,” In preparation.
- [188] D. Milner, S. Raz, H. Hel-Or, D. Keren, and E. Nevo, “A new measure of symmetry and its application to classification of bifurcating structures,” *Pattern Recognition*, vol. 40, no. 8, pp. 2237–2250, 2007.
- [189] J. Milnor, “Hilbert’s problem 18,” in *Proceedings of symposia in pure mathematics, 28, American Mathematical Society*, ISBN 0-8218-1428-1 (Browder, Felix E., Mathematical developments arising from Hilbert problems), 1976.
- [190] N. J. Mitra, L. Guibas, and M. Pauly, “Partial and approximate symmetry detection for 3d geometry,” in *ACM Transactions on Graphics*, vol. 25, pp. 560–568, 2006.
- [191] S. Mitra and Y. Liu, “Local facial asymmetry for expression classification,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’04)*, pp. 889–894, Washington, DC: IEEE Computer Society Press, June 27–July 2 2004. ([http://www.ri.cmu.edu/pubs/pub\\_4640.html](http://www.ri.cmu.edu/pubs/pub_4640.html)).
- [192] A. Moller and J. Swaddle, *Asymmetry, Developmental Stability, and Evolution*. Oxford University Press, 1997.
- [193] A. P. Moller, R. Thornhill, and S. W. Gangestad, “Direct and indirect tests for publication bias: Asymmetry and sexual selection,” *Animal Behaviour*, vol. 70, pp. 497–506, 2005.
- [194] D. Mukherjee, A. Zisserman, and J. Brady, “Shape from symmetry — detecting and exploiting symmetry in affine images,” 1995.
- [195] G. L. Naber, *The Geometry of Minkowski Spacetime*. New York: Springer-Verlag, 1992.
- [196] L. R. Nackman and S. M. Pizer, “Three-dimensional shape description using the symmetric axis transform i: Theory,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 187–202, 1985.
- [197] R. Nevatia and T. O. Binford, “Description and recognition of curved objects,” *Artificial Intelligence*, vol. 8, no. 1, pp. 77–98, 1977.
- [198] H. Ogawa, “Symmetry analysis of line drawings using the hough transform,” *Pattern Recognition Letters*, vol. 12, pp. 9–12, 1991.
- [199] V. Ostromoukhov, “Mathematical tools for computer-generated ornamental patterns,” in *Electronic Publishing, Artistic Imaging and Digital Typography*,

number 1375 in Lecture Notes in Computer Science, pp. 193–223, Springer-Verlag, 1998.

[200] A. J. O'Toole, K. A. Deffenbacher, D. Valentini, K. McKee, D. Huff, and H. Abdi, “The perception of face gender: The role of stimulus structure in recognition and classification,” *Memory and Cognition*, vol. 26, no. 1, pp. 146–160, 1998.

[201] A. R. Palmer, “Fluctuating asymmetry analyses: A primer,” in *Developmental instability: Its origins and evolutionary implications*, (T. A. Markow, ed.), pp. 335–364, Kluwer, Dordrecht, Netherlands, 1994.

[202] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, “Deformed lattice detection in real-world images using mean-shift belief propagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, vol. 31, no. 10, 2009.

[203] M. Park, S. Lee, P. Chen, S. Kashyap, A. A. Butt, and Y. Liu, “Performance evaluation of state-of-the-art discrete symmetry detection algorithms,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8, 2008.

[204] M. Park, Y. Liu, and R. T. Collins, “Deformed lattice detection via mean-shift belief propagation,” in *Proceedings of the 10th European Conference on Computer Vision (ECCV'08)*, 2008.

[205] M. Park, Y. Liu, and R. T. Collins, “Efficient mean shift belief propagation for vision tracking,” in *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR'08)*, IEEE Computer Society Press, June 2008.

[206] S. K. Parui and D. D. Majumder, “Symmetry analysis by computer,” *Pattern Recognition*, vol. 16, pp. 63–67, 1983.

[207] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas, “Discovering structural regularity in 3D geometry,” *ACM Transactions on Graphics*, vol. 27, no. 3, pp. #43, 1–11, 2008.

[208] S. Peleg and A. Rosenfeld, “A min-max medial axis transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 2, pp. 208–210, 1981.

[209] R. Penrose, “Role of aesthetics in pure and applied research,” *Bulletin of the Institute of Mathematics and its Applications*, vol. 10, no. 266, 1974.

[210] A. P. Pentland, “Perceptual organization and the representation of natural form,” *Artificial Intelligence*, vol. 28, pp. 293–331, 1986.

[211] N. Pillow, S. Utcke, and A. Zisserman, “Viewpoint-invariant representation of generalized cylinders using the symmetry set,” *Image and Vision Computing*, vol. 13, no. 5, 1995.

[212] M. Pinsky and D. Avnir, “Continuous symmetry measures, v: The classical polyhedra,” *Inorganic Chemistry*, vol. 37, pp. 5575–5582, 1998.

[213] Y. Pinto, P. W. Fowler, D. Mitchell, and D. Avnir, “Continuous chirality analysis of model stone-wales rearrangements in fullerenes,” *Journal of Physical Chemistry*, vol. 102, pp. 5776–5784, 1998.

[214] S. Pizer, W. Oliver, and S. Bloomberg, “Hierarchical shape description via the multiresolution symmetric axis transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 505–511, 1987.

- [215] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. A. Funkhouser, “A planar-reflective symmetry transform for 3d shapes,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 549–559, 2006.
- [216] J. Ponce, “On characterizing ribbons and finding skewed symmetries,” in *Proceedings of International Conference on Robotics and Automation*, pp. 49–54, 1989.
- [217] J. Ponce, “On characterizing ribbons and finding skewed symmetries,” *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 328–340, 1990.
- [218] S. Posch, “Detecting skewed symmetries,” in *International Conference on Pattern Recognition*, pp. 602–606, The Hague, August 1992.
- [219] V. S. N. Prasad and L. S. Davis, “Detection rotational symmetries,” in *IEEE International Conference on Computer Vision (ICCV)*, pp. 346–352, 2005.
- [220] M. Proesmans and L. Van Gool, “Grouping based on coupled diffusion maps,” in *Shape, Contour and Grouping in Computer Vision*, LNCS vol. 1681, (D. Forsyth, J. Mundy, V. Gesu, and R. Cipolla, eds.).
- [221] M. Proesmans, L. Van Gool, and A. Oosterlinck, “Grouping through local, parallel interactions,” in *Proceedings Applications of digital image processing XVIII*, vol. 2564, July 1995, pp. 458–469, SPIE, 1995.
- [222] D. Reisfeld, H. Wolfson, and Y. Yeshurun, “Detection of interest points using symmetry,” in *International Conference on Computer Vision*, pp. 62–65, Osaka, 1990.
- [223] D. Reisfeld, H. Wolfson, and Y. Yeshurun, “Robust detection of facial features by generalized symmetry,” in *International Conference on Pattern Recognition*, pp. A:117–120, June 1992.
- [224] D. Reisfeld, H. Wolfson, and Y. Yeshurun, “Context free attentional operators: the generalized symmetry transform,” *The International Journal of Computer Vision*, vol. 14, pp. 119–130, 1995.
- [225] C. K. Richardson, D. Bowers, R. M. Bauer, K. M. Heilman, and C. M. Leonard, “Digitizing the moving face during dynamic displays of emotion,” *Neuropsychologia*, vol. 38, no. 7, pp. 1028–1039, 2000.
- [226] I. Rodríguez, A. Gumbert, N. Hempel de Ibarra, J. Kunze, and M. Giurfa, “Symmetry is in the eye of the ‘beeholder’: innate preference for bilateral symmetry in flower-naïve bumblebees,” *Naturwissenschaften*, vol. 91, no. 8, pp. 374–377, August 2004.
- [227] A. Rosenfeld, “Axial representation of shape,” *Computer Vision, Graphics, and Image Processing*, vol. 10, pp. 156–173, 1986.
- [228] T. Rowland and E. W. Weisstein, Group. <http://mathworld.wolfram.com/Group.html>.
- [229] P. Saint-Marc and G. Medioni, “B-spline contour representation and symmetry detection,” in *European Conference on Computer Vision*, pp. 604–606, 1990.
- [230] Y. Sato, J. Ohya, and K. Ishii, “Smoothed local generalized cones: An axial representation of 3D shapes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 56–62, Champaign, June 1992.
- [231] F. Schaffalitzky and A. Zisserman, “Geometric grouping of repeated elements within images,” in *Proceedings of 9th British Machine Vision Conference, Southampton*, pp. 13–22, 1998.

- [232] F. Schaffalitzky and A. Zisserman, “Geometric grouping of repeated elements within images,” in *Shape, Contour and Grouping in Computer Vision*, LNCS, (D. A. Forsyth, V. Di Gesu, J. L. Mundy, and R. Cipolla, eds.), pp. 165–181, Springer-Verlag, 1999.
- [233] F. Schaffalitzky and A. Zisserman, “Planar grouping for automatic detection of vanishing lines and points,” *IVC*, vol. 18, pp. 647–658, 2000.
- [234] F. Schaffalitzky and A. Zisserman, “Multi-view matching for unordered image sets, or “How do I organize my holiday snaps?”,” in *ECCV*, pp. 414–431, Springer-Verlag, 2002.
- [235] D. Schattschneider, “The plane symmetry groups: their recognition and notation,” *American Mathematical Monthly*, vol. 85, pp. 439–450, 1978.
- [236] D. Schattschneider, *Visions of Symmetry*. New York: Freeman, 1998.
- [237] D. Schattschneider, *M. C. Escher: Visions of Symmetry*. Harry N. Abrams, 2nd Edition, 2004.
- [238] D. Schattschneider, “Coxeter and the artists: Two-way inspiration,” in *The Coxeter Legacy: Reflections and Projections*, number 46, (C. Davis and E. W. Ellers, eds.), pp. 255–280, Fields Institute Communications, 2006.
- [239] J. E. Scheib, S. W. Gangestad, and R. Thornhill, “Facial attractiveness, symmetry and cues of good genes,” *Proceedings of the Royal Society of London, B*, vol. 266, pp. 1913–1918, 1999.
- [240] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert, “Detecting and matching repeated patterns for automatic geo-tagging in urban environments,” in *IEEE Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8, 2008.
- [241] G. L. Scott, S. C. Turner, and A. Zisserman, “Using a mixed wave/diffusion process to elicit the symmetry set,” *Image Vision Computing*, vol. 7, no. 1, pp. 63–70, 1989.
- [242] P. Seitz, “The robust recognition of object primitives using local axes of symmetry,” *Signal Processing*, vol. 18, pp. 89–108, 1989.
- [243] S. M. Seitz and C. R. Dyer, “View morphing,” *SIGGRAPH*, pp. 21–30, 1996.
- [244] S. M. Seitz and C. R. Dyer, “View-invariant analysis of cyclic motion,” *International Journal of Computer Vision*, vol. 25, pp. 1–23, 1997.
- [245] J. Semple and G. Kneebone, *Algebraic Projective Geometry*. Clarendon, 1979.
- [246] C. H. Séquin, “Parameterized sculpture families,” in *The Visual Mind II*, (M. Emmer, ed.), pp. 527–545, Cambridge, Massachusetts: The MIT Press, 2005.
- [247] D. Shechtman, I. Blech, D. Gratias, and J. W. Cahn, “Metallic phase with long-range orientational order and no translational symmetry,” *Physical Review Letters*, vol. 53, pp. 1951–1953, November 1984.
- [248] K. Siddiqi and B. B. Kimia, “A shock grammar for recognition,” in *CVPR 96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 507–513, 1996.
- [249] K. Siddiqi and S. Pizer, eds., *Medial Representations: Mathematics, Algorithms and Applications*. Springer, Series: Computational Imaging and Vision, vol. 37, 2008.

- [250] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker, “Shock graphs and shape matching,” *International Journal of Computer Vision*, vol. 30, pp. 1–24, 1999.
- [251] K. Sookocheff and D. Mould, “One-click lattice extraction from near-regular texture,” in *GRAPHITE '05: Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pp. 265–268, New York, NY, USA: ACM Press, 2005.
- [252] B. Springborn, P. Schröder, and U. Pinkall, “Conformal equivalence of triangle meshes,” in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pp. 1–11, New York, NY, USA: ACM, 2008.
- [253] J. S. Stahl and S. S. Wang, “Globally optimal grouping for symmetric closed boundaries by combining boundary and region information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 30, no. 3, pp. 395–411, 2008.
- [254] M. A. Steinberg and D. Avnir, “Continuous symmetry analysis of nmr chemical shielding anisotropy,” *Chemistry — A European Journal*, vol. 12, pp. 8534–8538, 2006.
- [255] S. Tari, J. Shah, and H. Pien, “Extraction of shape skeletons from grayscale images,” *Computer Vision and Image Understanding*, vol. 66, no. 2, pp. 133–146, 1997.
- [256] D. Terzopoulos, A. Witkin, and M. Kass, “Symmetry-seeking models and 3d object reconstruction,” *International Journal of Computer Vision*, vol. 1, pp. 211–221, 1987.
- [257] L. Teverovskiy, J. Becker, O. Lopez, and Y. Liu, “Quantified brain asymmetry for age estimation of normal and AD/MCI subjects,” in *2008 IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pp. 1509–1512, 2008.
- [258] R. Thornhill and S. W. Gangestad, “Facial attractiveness,” *Transactions in Cognitive Sciences*, vol. 3, no. 12, pp. 452–460, December 1999.
- [259] S. Thrun and B. Wegbreit, “Shape from symmetry,” in *ICCV*, pp. 1824–1831, 2005.
- [260] N. F. Troje and H. H. Buelthoff, “How is bilateral symmetry of human faces used for recognition of novel views?,” *Vision Research*, vol. 38, no. 1, pp. 79–89, 1998.
- [261] W. H. Tsai and S. L. Chou, “Detection of generalized principal axes in rotationally symmetric shapes,” *Pattern Recognition*, vol. 24, pp. 95–104, 1991.
- [262] Y. Tsin, Y. Liu, and V. Ramesh, “Texture replacement in real images,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pp. 539–544, Kauai: IEEE Computer Society Press, December 2001.
- [263] M. Tuceryan and A. K. Jain, “Texture Analysis, The Handbook of Pattern Recognition and Computer Vision (2nd Edition),” chapter 2.1, pp. 207, 248, World Scientific Publishing Co., (C. H. Chen, L. F. Pau, and P. S. P. Wang, eds.), 1998.
- [264] A. Turina, T. Tuytelaars, and L. Van Gool, “Efficient grouping under perspective skew,” in *Proceedings of IEEE Computer Society Conference on Computer*

*Vision and Pattern Recognition (CVPR'01)*, Kauai: IEEE Computer Society Press, December 2001.

- [265] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [266] T. Tuytelaars, M. Proesmans, and L. Van Gool, "The cascaded hough transforms," in *ICIP*, pp. 736–739, 1998.
- [267] T. Tuytelaars, A. Turina, and L. Van Gool, "Noncombinatorial detection of regular repetitions under perspective skew," *PAMI*, vol. 25, no. 4, pp. 418–432, 2003.
- [268] C. W. Tyler, ed., *Human Symmetry Perception and its Computational Analysis*. The Netherlands: VSP, Utrecht, 1996.
- [269] L. Van Gool, "Projective subgroups for grouping," *Philosophical Transactions of the Royal Society London*, vol. A(356), pp. 1252–1266, 1998.
- [270] L. Van Gool, T. Moons, and M. Proesmans, "Mirror and point symmetry under perspective skewing," in *The Proceedings of CVPR*, pp. 285–292, IEEE Computer Society, 1996.
- [271] L. Van Gool, T. Moons, D. Ungureanu, and A. Oosterlinck, "The characterization and detection of skewed symmetry," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 138–150, 1995.
- [272] A. A. Vasilier, "Recognition of symmetrical patterns in images," in *International Conference on Pattern Recognition*, pp. 1138–1140, Montreal, July 1984.
- [273] M. von Gagern and J. Richter-Gebert, "Hyperbolization of euclidean ornaments," *The Electronic Journal of Combinatorics*, vol. 16, no. 2, 2009.
- [274] D. K. Washburn and D. W. Crowe, *Symmetries of Culture: Theory and Practice of Plane Pattern Analysis*. University of Washington Press, 1991.
- [275] J. D. Watson and F. Crick, "The molecular structure of nucleic acids: A structure for deoxyribose nucleic acid," *Nature*, vol. 171, pp. 737–738, 1953.
- [276] M. J. Wenninger, *Polyhedron Models*. Cambridge University Press, 1974.
- [277] H. Weyl, *Symmetry*. Princeton, New Jersey: Princeton University Press, 1952.
- [278] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *IEEE International Conference on Computer Vision (ICCV)*, 2005.
- [279] J. Wolter, T. Woo, and R. Volz, "Optimal algorithms for symmetry detection in two and three dimensions," *The Visual Computer*, vol. 1, pp. 37–48, 1985.
- [280] A. Y. Yang, W. Hong, and Y. Ma, "Structure and pose from single images of symmetric objects with applications to robot navigation," in *Proceedings of the International Conference on Robotics and Automation*, 2003.
- [281] A. Y. Yang, K. Huang, S. Rao, and Y. Ma, "Symmetry-based 3-d reconstruction from perspective images," *Computer Vision and Image Understanding*, vol. 99, no. 2, pp. 210–240, 2005.
- [282] A. Y. Yang, S. Rao, W. Hong, and Y. Ma, "Geometric segmentation of perspective images based on symmetry group," in *Proceedings of the International Conference on Computer Vision*, 2003.

- [283] A. Y. Yang, K. Huang, S. Rao, W. Hong, and Y. Ma, "Symmetry-based 3-d reconstruction from perspective images," *Computer Vision and Image Understanding*, vol. 99, no. 2, pp. 210–240, 2005.
- [284] R. Ye, J. Ma, and H. Li, "Symmetric tiling patterns with the extended picard group in three-dimensional space," in *Computer Graphics International*, pp. 486–493, 2006.
- [285] R. Ye, Y. Zou, and J. Lu, "Fractal tiling with the extended modular group," in *Computational and Information Science*, volume 3314 of *Lecture Notes in Computer Science*, pp. 286–291, Berlin/Heidelberg: Springer, 2004.
- [286] J. Yen and C. Séquin, "Escher sphere construction kit," in *I3D '01: Proceedings of the 2001 Symposium on Interactive 3D Graphics*, pp. 95–98, New York, NY, USA: ACM Press, 2001.
- [287] E. Yodogawa, "Symmetropy, an entropy-like measure of visual symmetry," *Perception and Psychophysics*, vol. 32, no. 3, pp. 230–240, 1982.
- [288] D. Yogeve-Einot and D. Avnir, "Quantitative symmetry and chirality of the molecular building blocks of quartz," *Chemistry of Materials*, vol. 15, pp. 464–472, 2003.
- [289] D. Yogeve-Einot and D. Avnir, "Pressure and temperature effects on the degree of symmetry and chirality of the molecular building blocks of low quartz," *Acta Crystallographica*, vol. B60, pp. 163–173, 2004.
- [290] D. Yogeve-Einot and D. Avnir, "The temperature-dependent optical activity of quartz: from le chatelier to chirality measures," *Tetrahedron: Asymmetry*, vol. 17, pp. 2723–2725, 2006.
- [291] S. Y. K. Yuen, "Shape from contour using symmetries," in *European Conference on Computer Vision*, pp. 437–453, Antibes, April 1990.
- [292] H. Zabrodsky, "Symmetry — A review," Technical Report TR-90-16, Dept. of Computer Science, Hebrew University, Jerusalem, Israel, May 1990.
- [293] H. Zabrodsky and D. Algom, "Continuous symmetry: A model for human figural perception," *Spatial Vision*, vol. 8, no. 4, pp. 455–467, 1994.
- [294] H. Zabrodsky and D. Avnir, "Measuring symmetry in structural chemistry," in *Advanced Molecular Structure Research*, vol. 1, (I. Hargittai, ed.), Greenwich, CT: JAI Press, 1993.
- [295] H. Zabrodsky and D. Avnir, "Continuous symmetry measures, iv: Chirality," *Journal of the American Chemical Society*, vol. 117, pp. 462–473, 1995.
- [296] H. Zabrodsky, S. Peleg, and D. Avnir, "Continuous symmetry measures," *Journal of the American Chemical Society*, vol. 114, pp. 7843–7851, September 1992.
- [297] H. Zabrodsky, S. Peleg, and D. Avnir, "Hierarchical symmetry," in *International Conference on Pattern Recognition*, vol. C: Image, Speech, and Signal Analysis, pp. 9–12, The Hague, August–September 1992.
- [298] H. Zabrodsky, S. Peleg, and D. Avnir, "A measure of symmetry based on shape similarity," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 703–706, Champaign, June 1992.
- [299] H. Zabrodsky, S. Peleg, and D. Avnir, "Completion of occluded shapes using symmetry," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 678–679, New York, June 1993.

- [300] H. Zabrodsky, S. Peleg, and D. Avnir, "Continuous symmetry measures II: Symmetry groups and the tetrahedron," *Journal of the American Chemical Society*, vol. 115, pp. 8278–8289, 1993.
- [301] H. Zabrodsky, S. Peleg, and D. Avnir, "Continuous symmetry for shapes," in *Aspects of Visual Form Processing*, (C. Arcelli, L. Cordella, and G. Sanniti di Baja, eds.), pp. 594–613, World Scientific, 1994. Proceedings of the 2nd Int. Work. Visual Form, Capri, Italy.
- [302] H. Zabrodsky, S. Peleg, and D. Avnir, "Symmetry of fuzzy data," in *International Conference on Pattern Recognition*, pp. 499–504, Tel-Aviv, Israel, October 1994.
- [303] H. Zabrodsky, S. Peleg, and D. Avnir, "Symmetry as a continuous feature," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, pp. 1154–1165, December 1995.
- [304] H. Zabrodsky and D. Weinshall, "3D symmetry from 2D data," in *European Conference on Computer Vision*, Stockholm, Sweden, May 1994.
- [305] H. Zabrodsky and D. Weinshall, "Using bilateral symmetry to improve 3D reconstruction from image sequences," *Computer Vision and Image Understanding*, vol. 67, no. 1, pp. 48–57, 1997.
- [306] D. W. Zaidel, S. M. Aarde, and K. Baig, "Appearance of symmetry, beauty, and health in human faces," *Brain and Cognition*, vol. 57, no. 3, pp. 261–263, 2005.
- [307] T. Zielke, M. Brauckmann, and W. von Seelen, "Intensity and edge-based symmetry detection applied to car-following," in *European Conference on Computer Vision*, pp. 865–873, Santa Margherita, May 1992.
- [308] S. W. Zucker, "Toward a model of texture," *CGIP*, vol. 5, no. 2, pp. 190–202, June 1976.