An algebraic approach to symmetry detection

Yosi Keller
Yale University
Mathematics Department
New-Haven, Connecticut, USA
yosi.keller@yale.edu

Yoel Shkolnisky
Tel-Aviv University
Computer Science Department
Tel-Aviv, Isreal
yoel@post.tau.ac.il

Abstract

We present an algorithm for detecting cyclic and dihedral symmetries of an object. Both symmetry types can be detected by the special patterns they generate in the object's Fourier transform. These patterns are effectively detected and analyzed using the "angular difference function" (ADF), which measures the difference in the angular content of images. The ADF is accurately computed by using the pseudo-polar Fourier transform, which rapidly computes the Fourier transform of an object on a near-polar grid. The algorithm detects all the axes of centered and non-centered symmetries. The proposed algorithm is algebraically accurate and uses no interpolations.

1 Introduction

The two most common types of symmetries are rotational and reflectional symmetries. An object is said to have a rotational symmetry of order $N$ if it is invariant under rotations of $\frac{2\pi}{N}$, $n = 0, \ldots, N - 1$. An object is said to have a reflectional symmetry if it is invariant under a reflection transformation about some line. Most existing algorithms usually detect either rotational or reflectional symmetry. The algorithm presented in this paper is based on the angular difference function (ADF), which measures the difference of two objects in a given angular direction. For symmetric objects the value of this function is shown to be zero in points that correspond to the symmetry axes. The zeros of the ADF identify both rotational and reflectional symmetries. The algorithm characterizes rotational symmetries by the set of rotation angles that keep the object unchanged. Similarly, it characterizes reflectional symmetries by the set of reflection axes.

Our idea is related to the work presented in [3]. Both algorithms detect the patterns that symmetries induce in the frequency domain. However, the algorithm we present in this paper uses an algebraically exact method for detecting these patterns. Specifically, it computes the ADF using the pseudo-polar Fourier transform and then uses the zeros of the ADF to detect minima ridges in the Fourier domain. It also uses a simpler scheme to infer the reflectional symmetry from the rotational symmetry.

The paper is organized as follows. In section 2 we describe previous work related to symmetry detection. In section 3 we describe the pseudo-polar Fourier transform, which evaluates the Fourier transform of an object on a near-polar grid. This transform is the basis for our symmetry detection algorithm. In section 4 we introduce the Angular difference function (ADF) as a tool for analyzing polar properties of images and utilize it to detect and analyze rotational and reflectional symmetries. In sections 6 we present experimental results.

2 Previous work

Symmetry is thoroughly studied in the literature from both theoretical, algorithmic and applicative perspectives. Theoretical treatment of symmetry can be found in [4]. The algorithmic approach to symmetry detection can be divided into several categories based on its characteristics. The first characteristic of a symmetry detection algorithm is whether it considers symmetry as a binary or continuous feature that measures the amount of symmetry. A second characteristic is the type of symmetry detected by the algorithm. Most algorithms detect either rotational or reflectional symmetry but not both. A third characteristic is the assumptions on the image. For example, whether the algorithm assumes that the image is symmetric or detects it itself, or whether the algorithm assumes that the symmetric feature is located at the center of the image. A fourth characteristic is whether the algorithm operates in the image domain or transforms the problem into a different domain, like the Fourier domain. A fifth characteristic is the robustness of the algorithm to noise and its ability to operate on real-life non-synthetic images. The last characteristic of an algorithm is its complexity. This characteristic is important for symmetry detection algorithms since most algorithms typically require an exhaustive search over all potential symmetry axes. Such a
search requires excessive computation even for small images.

In the light of these characteristics, we will examine the existing work on symmetry detection. Some of the work we describe refers to 3D symmetry detection algorithms. We describe such algorithms if they are applicable to 2D problems.

[2] presents a low-level, context free operator for detecting points of interest within an image, which relies on the assumption that context free attention is directed by symmetry. The suggested symmetry operator constructs the symmetry map of the image by assigning symmetry magnitude and symmetry orientation to each pixel. This map is an edge map where the magnitude and orientation of each edge depends on the symmetry associated with each of its pixels. The proposed operator allows processing different symmetry scales, enabling it to be used in multi-resolution schemes. Generally, the transform iterates over all pixels in the image, and for each pixel $p$ it inspects all pairs of points in a neighborhood with midpoint $p$ and radius $r$. It then computes the contribution of each pair according to its gradient and distance from $p$. The symmetry value of a point $p$ is obtained by summing all contributions of the individual pairs. The direction of the symmetry at a point $p$ is obtained by averaging the directions of the pair with the highest symmetry contribution to $p$. The proposed operator is demonstrated to be effective in detecting points of interests in natural images.

3 The pseudo-polar Fourier transform

The proposed registration algorithm is based on a fast and algebraically accurate discrete pseudo-polar FFT (PPFFT) [1]. A FFT where the evaluated frequencies lie on an oversampled set of non-angularly equispaced points which we call the pseudo-polar (PP) grid. This grid, shown in Fig. 1, resembles the polar grid. Its samples are given on rays through the origin (DC frequency). The computation of the PPFFT involves only 1-D FFT’s. In particular, there is no need for re-gridding or interpolation.

4 The angular difference function

The angular difference function (ADF) measures the difference of two images in the angular direction. In section 4.1 we present a method for 1D shift estimation using a difference function. In section 4.2 we present the application of the ADF to the frequency domain of 2D images. We conclude the presentation of the ADF by presenting in section 4.3 a fast and accurate algorithm for its computation.

4.1 Translation estimation using difference functions

We begin the derivation of the ADF with a 1D example. Difference functions (DF) enable us to derive a naive algorithm for 1D shift estimation. Let $f_1(x)$ and $f_2(x)$ $x \in [0,N]$ be two shifted versions of the same function. Specifically, $f_1(x) = f_2(x + \Delta x)$. We denote by $g_2(x)$ the flipped and shifted version of $f_2(x)$

$$g_2(x) = f_2(-x+N). \quad (1)$$

We define the difference function (DF) $\Delta f$ by

$$\Delta f(x) = f_1(x) - g_2(x) = f_2(x + \Delta x) - f_2(-x+N)$$

and consider its zeros $\Delta f(x) = 0$. One of its zeros necessarily satisfies $\Delta x = \frac{N}{2} - x_0$ which means that we can estimate the relative translation from the location of the zero of $\Delta f$.

4.2 The difference function in the Fourier domain

Given two images $I_1$ and $I_2$, we denote by $M_1(r,\theta)$ and $M_2(r,\theta)$ the magnitudes of the Fourier transforms of $I_1$ and $I_2$, respectively. We define the difference function of $M_1(r,\theta)$ and $M_2(-r,\theta)$ in the angular direction by

$$\Delta M(\theta) = \int_0^\infty |M_1(r,\theta) - M_2(-r,\theta)| dr, \quad \theta \in [0,\pi]. \quad (2)$$

The value of $\Delta M(\theta_0)$ is zero if

$$\theta_0^{(1)} = -\frac{\Delta \theta}{2}, \quad \theta_0^{(2)} = -\frac{\Delta \theta}{2} + \frac{\pi}{2}. \quad (3)$$

We see from Eq. 3 that the zeros $\theta_0^1$ and $\theta_0^2$ are $\pi/2$ radians apart. Therefore we define the angular difference function (ADF) by

$$ADF(\theta) = \Delta M(\theta) + \Delta M\left(\theta + \frac{\pi}{2}\right), \quad \theta \in [0,\frac{\pi}{2}], \quad (4)$$
4.3 Computing the ADF for discrete images

The reversal of the angular axis, indicated by Eq. 2, is accurately implemented by flipping the input image either along the x or the y axes.

We use the PPFT to compute the ADF as follows: Given input images $I_1$ and $I_2$, defined on a Cartesian grid

1. Flip $I_1$ in the left-right direction.
2. Compute $M_I^d$ and $M_J^d$, where $M_I^d$ is the magnitude of the PPFT of $I_j$, $j = 1, 2$.
3. Evaluate Eq. 2 using numerical integration
   \[ \Delta M^d(\theta_i) = \sum_{0 \leq r_j \leq \pi} |M_I^d(\theta_i, r_j) - M_J^d(-\theta_i, r_j)| \Delta r_i, \theta_i \in [0, \pi]. \]
4. Compute the ADF by
   \[ ADF(\theta_i) = \Delta M^d(\theta_i) + \Delta M^d(\theta_i + N) \]
   where $N$ is the size of the PPFT domain.

5 Symmetry detection algorithm

We present a symmetry detection algorithm consisting of three stages. First, the algorithm determines $N$, the number of ADF minima in section 5.1. Then, in section 5.2, it uses $N$ to determine the symmetry axes. Finally, for non-centered symmetries it locates the center of symmetry as described in section 5.3.

5.1 Computing the number of minima of the ADF

For a given input image $I(x, y)$, we detect reflectional symmetry by computing the ADF of $I$ and flipping $I$ according to section 4.3. Figure 2 illustrates the ADF of a symmetric image. We can clearly observe that the number of minima corresponds to the degree of symmetry in the input image. We will robustly estimate the number of minima in the ADF by using its spectrum, denoted by $S_{ADF}$. If the ADF has $N$ minima, then the spectrum $S_{ADF}$ has a maximum at $\omega_N$. The number of minima $N$ is given by

\[ N = \max_i S_{ADF}(\theta_i) \quad (5) \]

In the example shown in Figure 2 there are 3 symmetry axes, the maxima of the spectrum $S_{ADF}$ will be detected at $\omega_3$. Natural and synthetic objects usually exhibit low symmetry orders, e.g., $(N < 15)$, which makes $\omega_N$ a very low frequency. Thus, the ADF can be pre-processed by low-pass filtering.

5.2 Computing the symmetry axes

The symmetry axes are given by
\[ \theta_s(i, N_s) = \frac{2\pi}{N_s} i + \theta_0, \quad i = 0, \ldots, N_s - 1 \quad (6) \]
where $\theta_0$ is the angle of any of the symmetry axes and $N_s$ is the number of symmetry axes. From Eq. 6 we can see that $\theta_s(i, N_s)$ is periodic with respect to $i$, having $N_s$ minima. We integrate the ADF over intervals of length $T = \frac{L_{ADF}}{N}$
\[ ADF_p(i) = \sum_{j=1}^{N} ADF(i+jT), \quad i = 0, \ldots, T-1 \quad (7) \]
where $L_{ADF}$ is the length of the ADF. This summation creates a dominant minimum at the point $\Delta \theta$. The angle $\theta_0$ is given by
\[ \theta_0 = \frac{1}{2} \arg \min_{\theta} ADF_p(\theta) \quad (8) \]

If the number of minima of the ADF is odd, then the number of symmetry axes $N_s$ is equal to the number of minima $N_s$ given by Eq. 5. If the number of minima $N$ is even, then either $N_s = N$ or $N_s = 2N$ [4]. This is resolved by comparing the registration errors related to $\theta_s(1, k)$ and $\theta_s(1, 2k)$. 

Figure 2: The ADF of a symmetric image. The input image (a) is flipped left-to-right to create (b). $\Delta M$ in (c) is the difference function of the Pseudo-Polar FFTs of (a) and (b). The ADF in (d) is computed by averaging $\Delta M$ with an offset of $\frac{\pi}{2}$. 

0-7695-2128-2/04 $20.00 \text{ (C) 2004 IEEE}$
5.3 Symmetry Center detection

For non-centered symmetries, we detect the center of symmetry by computing two symmetry axes and finding their intersection point \((x_0, y_0)\). Each symmetry axis \(l_1\) is located as follows. We rotate the image \(I\) by \(\theta_0\), given in Eq. 8, around the center of the image. We denote the rotated image \(I_1\). The symmetry axis of \(I_1\) is now parallel to the \(y\) axis. Hence, we can estimate its location by flipping \(I_1\) around the \(y\) axis and estimating the 1D translation \(\Delta x\) along the \(x\) axis, between \(I_1\) and its flipped version.

6 Experimental results

The proposed algorithm was extensively tested using synthetic and real images. For each image we present its \(ADF\), its lowpass filtered \(ADF\), and its spectrum \(S_{ADF}\). For reflectional symmetries, the detected symmetry axes are overlayed on the input image. For all images, the spectrum \(S_{ADF}\) was computed by a four dimensional MUSIC algorithm without zero padding.

In the synthetic images in Figs. 3 and 4, we can clearly see the periodic nature of the \(ADF\). Thus, for synthetic images lowpass filtering of the \(ADF\) is not crucial. The number of symmetry axes and their exact location are clearly detected. In Fig. 5 we applied the algorithm to real images. We can observe that the algorithm detected the correct number of axes.

References


