Image Matching using Local Symmetry Features

Daniel Cabrini Hauagge Noah Snavely
Cornell University
{hauagge,snavely}@cs.cornell.edu

Abstract

We present a new technique for extracting local features from images of architectural scenes, based on detecting and representing local symmetries. These new features are motivated by the fact that local symmetries, at different scales, are a fundamental characteristic of many urban images, and are potentially more invariant to large appearance changes than lower-level features such as SIFT. Hence, we apply these features to the problem of matching challenging pairs of photos of urban scenes. Our features are based on simple measures of local bilateral and rotational symmetries computed using local image operations. These measures are used both for feature detection and for computing descriptors. We demonstrate our method on a challenging new dataset containing image pairs exhibiting a range of dramatic variations in lighting, age, and rendering style, and show that our features can improve matching performance for this difficult task.

1. Introduction

Symmetry, at many different scales, and in many different forms, is inherent in the structure of our world, and evident in the shape and appearance of many natural and man-made scenes. Humans have an innate ability to perceive symmetries in objects and images, and tend to construct buildings and other objects that exhibit a range of local and global symmetries. In computer vision, analysis of symmetry has been a long-standing problem, and is attractive as a way of representing images for many reasons. In particular, symmetries are a potentially stable and robust feature of an image, yet, when considered at multiple scales and locations, symmetries can also be quite descriptive.

For instance, consider the image pairs shown in Figure 1. Even though each pair is geometrically registered, the pairs exhibit large changes in appearance due to varying illumination, age, or style of depiction. These factors can lead to large differences in low-level cues such as intensities or gradients. Hence, while existing local features, such as SIFT [10], are highly invariant to a range of geometric and photometric transformations, we find that they often perform poorly in feature matching given the kinds of dramatic variations shown in Figure 1. On the other hand, the structures depicted in the images can be described at a higher level in terms of a nested hierarchy of local symmetries (as well as repeated elements). Our work begins with the hypothesis that such symmetries are better preserved across a wider range of appearances than are lower-level cues.

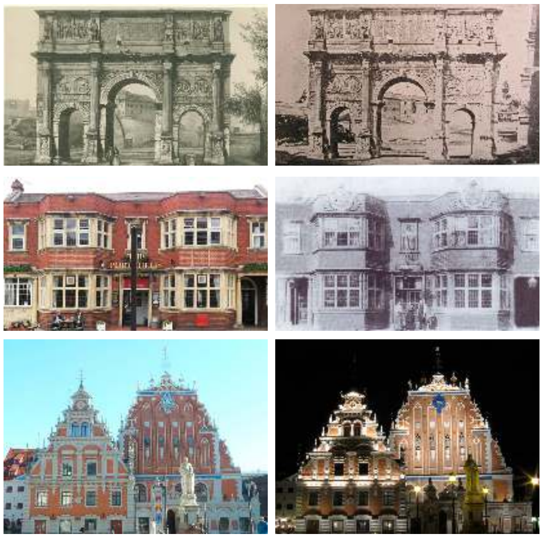


Figure 1: *Challenging pairs for feature matching.* Each pair of images show registered views of a building. Despite the geometric consistency, these images are difficult for feature matching algorithms because of dramatic changes in appearance, due to different depictions (drawing vs. photo), different time periods (modern vs. historical), and different illumination (day vs. night). While these images are dissimilar at a pixel level, each image pairs exhibit similar local symmetries, which we seek to exploit for matching. These pairs of images are drawn from our new dataset.

In this paper, we seek to exploit such symmetries for robust image matching through local features derived from local symmetries. To that end, we propose both a feature detector and a descriptor, designed primarily for architectural scenes, based on scoring local symmetries at all locations and scales in an image. Our symmetry features retain advantages of local features, but aim for a more "mid-level" description relative to standard features.

To compute these features, we propose a simple measure of local symmetry based on analyzing image differences across symmetry axes. We compute this measure densely on an image, and at different scales, and score each image patch based on three types of symmetries: horizontal, vertical, and rotational. We propose a method for detecting scales for our local symmetry measure, and use this to define a feature detector that returns a set of maximally symmetric positions and scales. We also develop a feature descriptor based on these same symmetry measures.

We evaluate our method on a challenging new dataset consisting of pairs of images with significant appearance changes, including the images in Figure 1. Some pairs are rephotographs of the same scene many years apart, others represent extreme differences in illumination, while others differ in rendering style (e.g., photograph vs. drawing). These types of images are of interest in applications such as large-scale 3D reconstruction from heterogeneous image sources [1], especially for datasets that include imagery with a wide range of illumination or time periods [17]. We evaluate two versions of our method on this dataset, one based on raw image intensities, and another based on gradient histograms. Our experimental results show that our detectors are more robust than standard techniques, and our descriptors, while less discriminating than SIFT overall, can significantly improve matching performance when used in combination with SIFT.

To summarize, our main contribution is a simple method for densely characterizing local symmetries across image and scale space, and defining a corresponding local symmetry feature detector and descriptor. We also introduce a new dataset that we hope will help push the envelope in feature matching methods. Our code and dataset are available on our project webpage.[1]

2. Related Work

Researchers in computer vision have made significant progress in representing and detecting symmetries in images and other types of data (Liu *et al.* [9] present an excellent survey). However, there has been relatively little work on using local symmetries as explicit features for matching tasks. The closest work to ours is probably the self-similarity descriptor of Shechtman and Irani [19]; they propose to use patterns of self-similarity of an image patch as a robust descriptor for matching across images and videos. In our case, we use a variety of *symmetries*, rather than repetitions, as cues , and use these to define both a feature detector and a descriptor. Schindler *et al.* [18] also detect and match repetitive patterns on building facades, and Wu *et al.* [21] detect large repeating elements of facade images for use as robust support regions for computing features. In contrast, we do not make hard decisions about symmetric structures, but instead compute many local features based on a softer definition of a symmetry score.

Loy and Zelinsky also propose a feature detector based on radial symmetries [12], using a fast radial symmetry transform that accumulates votes for symmetric regions from gradient information. This was demonstrated to work well for finding radially symmetric features such as eyes, but was not demonstrated for matching applications. In an earlier approach, Reisfeld *et al.* [16] used a similar voting scheme to detect interest points using radial symmetries via a generalized symmetry transform.

Our local symmetry detector is also related to other low-level measures of symmetry. Kovesi observed that local bilateral symmetry in image intensity relates to the response of filters of different phase [6], and later extended this concept of "phase-congruency" for detecting features such as edges and corners [7]. Di Gesù *et al.* proposed the discrete symmetry transform [4], based on axial moments related to the medial axis transform. Other work is based not on image intensities or gradients, but instead on computing an initial set of sparse local features (e.g., SIFT) and finding relationships between these features consistent with local bilateral or radial symmetries [11]. In contrast, we define a symmetry score based on a general measure of image similarity across reflection axes, and compute this densely over the image and across scale space. Our symmetry score is related to the reflective symmetry transform proposed for 3D shape analysis [15], but we compute scores locally (and over scales), rather than globally. Finally, our detector bears some similarity to the recently proposed edge foci interest regions of Zitnick and Ramnath [22], although ours is explicitly based on symmetries.

Recent work by Shrivastava *et al.* [20] also focuses on matching of difficult images (e.g., paintings), but using global features (in their case, a global HOG descriptor) and using linear classification techniques to learn how to weight the dimensions of the descriptor. In contrast, we focus on local-feature-level matching so as to derive feature correspondence, rather than on global image similarity metrics.

3. Local Symmetry

As a brief overview, our method takes an image as input, and computes local symmetry scores over the image and across scale space. These scores are then used to produce a set of feature detections, as well as to define a descriptor for each detected feature. We begin by presenting our framework for computing symmetry scores, then describe a detector and descriptor based on these scores.

[1] http://www.cs.cornell.edu/projects/symfeat/

In order to "score" local symmetries in an image, we use a simple analysis of image similarity across different types of reflection axes. We begin by treating a grayscale image as a 2D function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ that maps pixel locations (x, y) (or (ρ, θ), in polar coordinates) to intensities. We consider two common types of symmetry: bilateral (a symmetry about a given line in the image) and $2n$-fold rotational symmetry (symmetry defined by reflection across an image point). If f exhibits exact bilateral symmetry about a line passing through the origin, then for all (ρ, θ) in the image:

$$f(\rho, \theta_s + \theta) = f(\rho, \theta_s - \theta),$$

where θ_s is the angle of the axis of symmetry. Similarly, if f exhibits $2n$-fold rotational symmetry about the origin,

$$f(\rho, \theta) = f(-\rho, \theta)$$

for all (ρ, θ). These two kinds of symmetry can be understood intuitively by considering slices of $\mathbb{R}^2$, as shown in Figure 2. For each symmetry type, certain 1D slices will be symmetric. In the case of bilateral symmetry, these slices will be perpendicular to the axis of symmetry; for $2n$-fold rotational symmetry, the slices all pass through a single point. For both symmetry types, the property that holds is:

$$f_s(t) = f_s(-t),$$

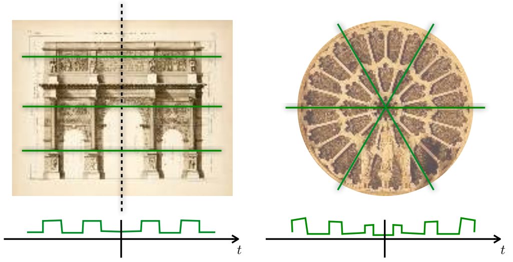
where $f_s(t)$ is a 1D slice of f parameterized by variable t, and centered on the axis or point of symmetry. For a given 1D function f_s, one could check if it is symmetric by comparing each value $f_s(t)$ to $f_s(-t)$, either in a window (for local symmetry), or everywhere (for global symmetry). For a 2D patch of an image, one could check for symmetry by checking the appropriate family of 1D slices.

3.1. Scoring local symmetries

We now use this intuition to define a local symmetry score. We begin by describing this score in terms of raw image intensities; extending to other per-pixel features is straightforward, as explained below. Ideally, such a score would be efficient to compute and would fire strongly on symmetric regions, yet would be robust to small asymmetries in a region due to noise, occlusion, or illumination.

Our local symmetry score is defined for each location $\mathbf{p} = (x, y)$ in an image, and characterizes the degree of symmetry at that location based on the intuition described above. To define our score, we require three components:

Symmetry type. As described above, we consider bilateral and $2n$-fold rotational symmetries. Given a symmetry type s, and a point $\mathbf{p}$ at which we want to compute a symmetry score, we define a function $M_{s,\mathbf{p}} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that maps an image point $\mathbf{q} = (x, y)$ onto its symmetrically corresponding point with respect to the point of interest $\mathbf{p}$ and the symmetry type s. If the image f exhibits perfect symmetry type s at location $\mathbf{p}$, then $f(\mathbf{q}) = f(M_{s,\mathbf{p}}(\mathbf{q}))$ for all $\mathbf{q}$.



(a) Bilateral symmetry (b) Rotational symmetry

Figure 2: By taking slices of a 2D image f, we see that the problem of symmetry detection can be posed as that of determining if a set of 1D slices through f are symmetric. Left: an image with approximate bilateral symmetry about the dotted vertical line. All horizontal slices (green) are nearly even functions (as shown in the function profile at bottom). Right: approximate $2n$-fold rotational symmetry. Most slices through the image center are even functions.

Distance function. Next, we define a distance function $d(\cdot, \cdot)$ that measures how well a given pair of corresponding symmetric points $(\mathbf{q}, M_{s,\mathbf{p}}(\mathbf{q}))$ match each other in appearance. For instance, for a symmetry score based on raw intensities, d could be defined as the absolute difference in intensity between two points: $d(\mathbf{q}, \mathbf{r}) = |f(\mathbf{q}) - f(\mathbf{r})|$.

Weight mask. Finally, we define a weighting function $w_\sigma(r)$ that gives the *importance* of each set of corresponding point pairs around the center point $\mathbf{p}$ in determining the symmetry score at $\mathbf{p}$. If we were interested in global symmetries, the weight mask would have infinite support (i.e., would be 1 everywhere). To detect local symmetries, one might use a Gaussian mask, giving more importance to points close to $\mathbf{p}$. For simplicity, we assume that the weight mask is radially symmetric, and hence is a function just of the distance r from the center point $\mathbf{p}$. The subscript σ denotes a scale for the weight mask that modulates the size of its support region, allowing us to create a scale space for symmetries, as we describe in Section 4. In the case of Gaussian weights, σ is the standard deviation.

Putting these three components together, we define a function that we call the *local symmetry distance*, SD:

$$SD(\mathbf{p}) = \sum_{\mathbf{q}} w_\sigma(\|\mathbf{q} - \mathbf{p}\|) \cdot d(\mathbf{q}, M_{s,\mathbf{p}}(\mathbf{q})). \qquad (1)$$

where $\mathbf{q}$ varies over all pixel locations with valid reflected pairs. Note that this function is a distance, so lower values indicate greater symmetry. As described above, a simple symmetry distance function we consider is one built on raw image intensities, where d is the absolute difference in intensity, and w_σ is a Gaussian. This definition of SD measures, at a given location $\mathbf{p}$, how similar the image is to itself when flipped across a symmetry axis (or point) through $\mathbf{p}$, accumulating intensity differences across a Gaussian support region. We refer to this family of symmetry distances as SYM-I (for "intensity"). In practice, we consider three versions, for three symmetry types: horizontal-bilateral (denoted SYM-IH), vertical-bilateral (SYM-IV), and $2n$-fold rotational symmetry (SYM-IR).

One potential problem with SD in Eq. 1 is that uniform image regions are trivially symmetric, and thus have low symmetry distance. However, a characteristic of these regions that differentiates them from "interesting" symmetric regions is that their symmetry is not well localized, i.e., SD is low in a wide region. To address this, we find "edges" or "blobs" in the raw SD map, by convolving SD with a filter L based on a Laplacian-of-Gaussian (LoG) kernel [8]. For the rotational-symmetry-based SYM-IR distance, we use the standard LoG kernel, while for bilateral symmetries (SYM-IH and SYM-IV) we use a kernel that has a LoG profile perpendicular to the axis of symmetry and a Gaussian along the axis of symmetry. With the correct sign on the kernel L, this convolution converts the symmetry distance SD into a symmetry score function, denoted SS:

$$SS = L * SD.$$

3.2. Gradient histogram-based score

In addition to raw intensities, we also consider a symmetry score based on gradient orientations, as these tend to be more stable to photometric changes [10, 22], and can also be more discriminative than raw intensities (e.g., oriented edges may accidentally coincide less frequently). At each image location $\mathbf{q}$ we define a histogram of local gradient orientations $\mathbf{h}(\mathbf{q})$. We then use this vector-valued function in place of the scalar function f in the local symmetry transform described above (using the dot product of two vectors, rather than the absolute difference in intensities, to compare the appearance of two symmetric points). This vector-valued function is related to SIFT or HOG features [3], but computed densely at each pixel of a given scale. We call this variant of our symmetry score SYM-G (for "gradient"); example SYM-G scores are shown in Figure 4.

To compute the gradient orientation histogram function $\mathbf{h}(\mathbf{q})$ at each pixel $\mathbf{q}$ of an image I, we first compute the gradient magnitude and orientation of I using finite differences, first blurring I to remove noise (we use a Gaussian with $\sigma = 0.5$). We compute the gradient orientation as usual, but apply the local constrast enhancement method of Zitnick [22] to the gradient magnitudes, which helps normalize edge magnitudes between high- and low-constrast regions. We then bin the gradient orientations in a small region around $\mathbf{p}$ into an orientation histogram, weighted by (constrast-enhanced) gradient magnitude, and Gaussian weighted by distance to $\mathbf{p}$. We again use a Gaussian with $\sigma = 0.5$, and an orientation histogram with eight bins, softly binning each edge orientation, and treating orientations as "unsigned" (e.g., orientation is only defined up to a 180 degree rotation). This results in a local histogram $\hat{\mathbf{h}}(\mathbf{q})$, which we normalize to form the final vector-valued function:

$$\mathbf{h}(\mathbf{q}) = \frac{\hat{\mathbf{h}}(\mathbf{q})}{||\hat{\mathbf{h}}(\mathbf{q}))|| + \epsilon}$$

where $\epsilon = 0.05$ is added to the norm for robustness to noise.

To compare histograms at two reflected pixel positions $\mathbf{q}$ and $M_{s,\mathbf{p}}(\mathbf{q})$, we compute their dot product after "flipping" one of the histograms by permuting the bins as necessary to form a histogram of reflected orientations. This dot product is large if the two histograms are similar, and so represents a similarity, rather than a distance; hence, this directly gives us a symmetry score function SS. This symmetry score is related to the method of Loy and Eklundh [11] (who match SIFT features within an image), but our score is computed densely across image (and scale) space. As with SYM-I, we define three versions of this score, SYM-GH, SYM-GV, and SYM-GR, for horizontal-bilateral, vertical-bilateral, and rotational symmetry types, respectively.

4. Scale Space

The previous section defined a dense local symmetry score for an image I; we now want to use this score to detect features. To create good detections, we must reliably compute interest points in scale space, as well as image space. To do so, we compute the symmetry score on a Gaussian image pyramid densely sampled in scale space, and then compute stable locations, as we describe in this section.

Which symmetry score should we use for feature detection? The bilateral symmetries, such as SYM-IH and SYM-IV, are problematic, because bilateral symmetries are usually only well-localized in the direction perpendicular to the axis of symmetry. In contrast, the rotational-symmetry-based SYM-IR or SYM-GR score functions tend to have well-localized peaks in the centers of symmetric regions. However, for SYM-IR, we found that using a Gaussian weight function w_σ gives good localization in x and y, but poor localization in scale. This is because a Gaussian has too much mass close to its center, and the support region of the weight mask changes slowly across scales. Hence, the Gaussian-weighted score responds slowly to the inclusion of asymmetric or uniform image regions under its support, and it becomes hard to precisely localize features in scale space. Thus, for the purpose of feature detection, we choose a different function for w_σ, a mask resembling a smooth ring, defined as $w_\sigma(r) = Ae^{-\frac{(r-r_0)^2}{2\phi^2}}$, where r is the distance to the point of interest, A is a normalization constant, r_0 is the ring radius, and ϕ controls the width of the ring (an example ring mask is shown in Figure 3).

To better understand the advantages of using this weight mask for scale space localization, we compare it to a Gaus-

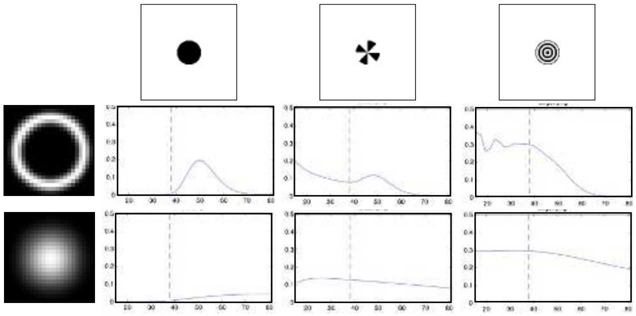


Figure 3: Response of the LoG filter as a function of scale for the central pixel of three synthetic images (top row), using SYM-IR. Two weight masks w_σ are shown: ring (middle row) and Gaussian (bottom row). The horizontal axis shows the kernel size, and the vertical axis the fraction of maximum response of the LoG filter. The ring mask gives more localized peaks, especially for the first two images.

sian mask in Figure 3. For a set of three simple images, we show how the SYM-IR score at the center of the image varies as a function of the scale σ. The middle row shows results for the ring-shaped weight mask, while the bottom row shows results for the Gaussian weights. The dashed red line shows the radius of the “interesting” symmetric portion in the center. The plots show that the ring weighting gives a much more pronounced peak around the dashed line than the Gaussian, sharp enough to fire for each individual ring in the third image consisting of concentric circles.

For SYM-G (the gradient-based score), we found that the Gaussian weights worked well, possibly because the overlap between symmetric pairs of gradient histograms generally results in a much sparser signal in the support region of the weight mask. Example SYM-G scores at two different scales are shown in Figure 4.

5. Local Symmetry Features

We now describe how we use our symmetry scores to define local features; we consider both feature detection, by finding local maxima of the score, and feature description, by building a feature vector from local symmetry scores.

5.1. Feature detector

As mentioned above, the maxima of the SYM-IR (with ring weighting) and SYM-GR functions are good candidates for feature detection, due to their good localization, while bilateral symmetries (e.g. SYM-IH and SYM-IV) tend not to be well-localized isolated points, as their responses tend to look edge-like—an entire line of pixels along an axis of symmetry will tend to fire strongly at once. However, we found that another stable detector can be constructed by finding maxima of the *product* of SYM-H and SYM-V, as this will be large only at locations that exhibit both horizontal and vertical symmetries (e.g., the centers of windows or other architectural features). In our evaluation, we consider using the SYM-IR function as a detector,

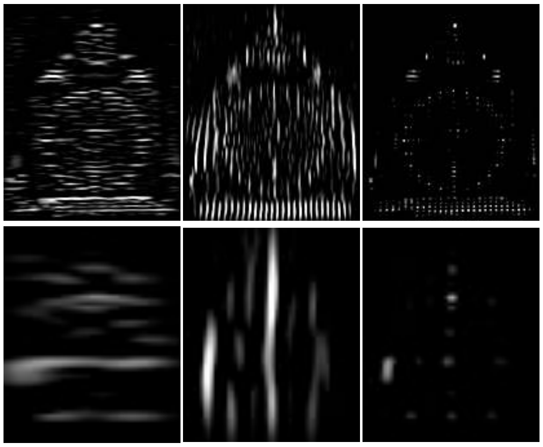


Figure 4: Two levels of the scale space pyramid for SYM-G for the image in Figure 5. The top row shows the symmetry score computed at a fine scale, the bottom row a coarse scale. Columns correspond to horizontal, vertical, and the product of the two symmetries.

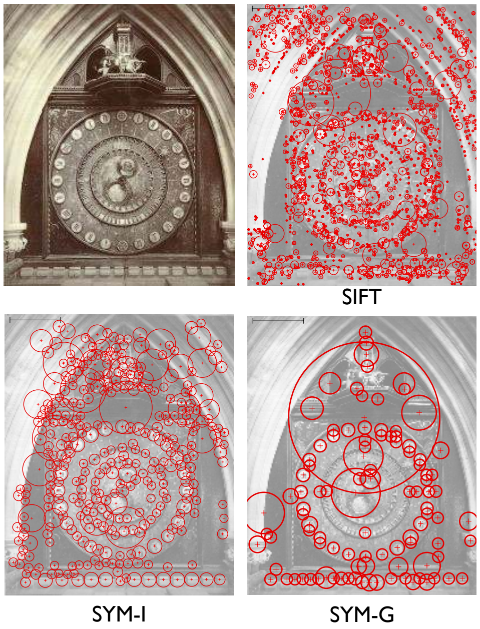


Figure 5: Detected features for SIFT, SYM-I and SYM-G. Each circle shows a feature with scale. For ease of visualization, the non-maxima overlap threshold was set to a stricter 0.1 for SYM-I. Note how the symmetry-based detectors more reliably fire on features such as the ring of circles around the clock’s circumference. Our symmetry features also tend to fire in larger regions on average than SIFT.

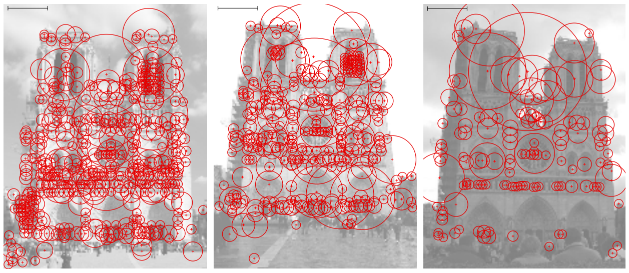


Figure 6: Detected features for SYM-G for images of the Notre Dame Cathedral. Note how some symmetric features are repeatably detected across each image, including the central rose window.

as well as a SYM-GH × SYM-GV detector built from the product of SYM-GH and SYM-GV.

Non-maxima suppression. To detect features given a score function computed on a Gaussian pyramid, we first find maxima in each scale independently (thresholding small score values), resulting in a set of initial detections at locations (x, y, s), where x and y represent positions in the full resolution image, and s is the detected scale. We represent the support of each feature as a circle of radius s centered at (x, y) in the original image. We then select a subset of features that are locally strong across scales. In particular, for each detected feature F, we keep F if it has the highest symmetry score of any feature whose support overlaps that of F by more than a threshold α (we use $\alpha = 0.4$ for SYM-G and $\alpha = 0.2$ for SYM-I; SYM-I tends to be noisier, demanding more stable detections). This results in a final set of partially overlapping features; we note that symmetric regions often overlap, due to repeated patterns and nested symmetric regions. This approach to non-maxima suppression is similar to that commonly used in object detection [5]. Example detections using SYM-I and SYM-G are shown in Figure 5, and example detections for a set of images of Notre Dame are shown in Figure 6.

5.2. Feature descriptor

We also devise a simple feature descriptor based on our local symmetry score, which we refer to as SYMD. SYMD encodes the distributions of the three SYM-I scores (SYM-IH, SYM-IV, and SYM-IR) around a feature location at the detected scale, in essence describing patterns of local symmetry around the keypoint. We use Gaussian weighting for SYM-I here, as we are not concerned with detection. To compute a descriptor for a keypoint (x, y, s), we use a strategy similar to that of shape context [2]: we impose a log-polar grid on the image, centered at (x, y), at scale s of the Gaussian pyramid, with diameter four times s. For each symmetry type and each cell we store the maximum value of the corresponding symmetry score within that cell; the max gives robustness to small deformations. We concatenate the grid values for the three different symmetry types, and normalize the resulting vector to unit norm. In our experiments we use a log-polar grid with 20 angular cells and 4 radial cells, resulting in a 240-dimensional descriptor.

6. Experimental Results

We designed a set of experiments to evaluate our features. We first evaluate the detector, comparing its repeatability to that of the common DoG and MSER detectors [13]. We then show how different combinations of detector and descriptor perform on a feature matching task.

For evaluation we collected a set of overlapping image pairs, mainly of architectural scenes, that we believe challenge modern feature matching methods. The pairs exhibit an array of dramatic appearance changes, due to illumination, age, and rendering style (paintings, drawings, etc.) Some pairs are closely aligned, and we registered these further with a homography so as to factor out geometry and focus on appearance changes. Other pairs were left unaligned, and exhibit both geometric and photometric variation. In total we gathered 48 image pairs, 12 of which come from the benchmark in [22] (the "Notre Dame" and "Painted Ladies" sets). Two pairs come from the Mikolajczyk *et al.* [14], one each from "Graffiti" and "Cars," exhibiting viewpoint and exposure changes, respectively. For each pair we manually compute a homography for use in computing ground truth matches. A few pairs are shown in Figure 1.

6.1. Evaluating detections

For each image pair (I_1, I_2) in the dataset, and each detector, we detect sets of keypoints K_1 and K_2, and compare these detections using the known homography H_{12} mapping points from I_1 to I_2. A common repeatibility metric is the fraction of keys in K_1 that have a similar detection in K_2, when warped by H_{12}. However, this measure is biased towards detectors that produce large numbers of keypoints. To alleviate this problem, we compare subsets of detections of varying size k, selected according to some "goodness" criterion (in our case, detector response or feature size), rather than the full sets of detections. In particular, for each k we compute the fraction of the top-k detections in K_1 with a similar detection in the top-k detections in K_2 (1.0 is a perfect score). By varying k we can produce a curve showing repeatability for all sets of top k features. By ordering keypoints according to detector response, we measure how invariant the detector is to the changes observed in the pair of images. By ordering according to scale (in decreasing order) we measure the repeatability of larger features; this is an interesting measure, as for difficult images we observe that features with larger support are often easier to match.

To determine if two detections $k_1 \in K_1$ and $k_2 \in K_2$ are repeated, we use the overlap metric of [14]. To summarize,

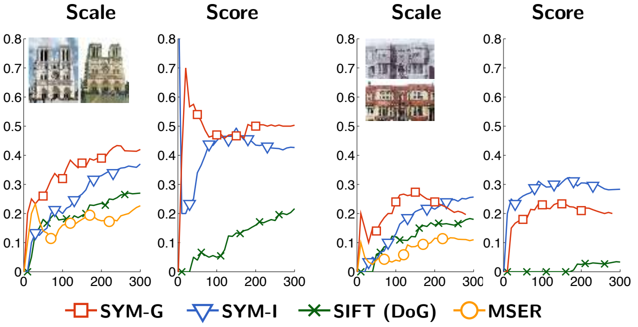


Figure 7: Example results for the detector repeatability experiment for two pairs of images (shown as insets), for each of four detectors, SIFT, MSER, Sym-I, and Sym-G. The horizontal axis shows the number of features k in the selected subsets, and the vertical axis shows the fraction of keypoints that have a repeated detection (higher means better repeatability). The titles indicate the ordering criterion. Please see the project webpage for a full set of plots.

	Scale		Score	
	100	200	100	200
MSER	0.087	0.103	–	–
SIFT(DoG)	0.144	0.153	0.050	0.078
Sym-I	0.135	0.184	0.173	0.206
Sym-G	**0.173**	**0.228**	**0.227**	**0.281**

Table 1: Avg. repeatability score for the top $k = \{100, 200\}$ detections according to scale and detection score. In cases where a detector produced fewer than k keypoints we report the repeatability score for the full feature set.

k_1 is rescaled by a factor s so that it has a fixed area A; we denote this scaled detection k_1^s. The same scale s is applied to k_2.[2] Finally, the relative overlap of the support regions of $H_{12}k_1^s$ (k_1^s remapped by the homography H_{12}) and k_2^s gives an overlap score. If this score is above a threshold (0.6 in our experiments), we declare a match. This overlap metric allows for some geometric uncertainty for smaller detections, while not being overly forgiving of larger detections.

We compare the DoG detector (as implemented in SIFT) and MSER with our Sym-I and Sym-G detectors. We use the SIFT implementation from OpenCV, and the MSER detector available at `http://www.featurespace.org/`; this implementation does not expose the detector response, so we only order MSER keypoints by scale. Example repeatability curves for top subsets of features ordered by score and scale are shown in Figure 7. Table 1 shows average repeatability scores over all dataset pairs for $n = 100$ and $n = 200$, for both score and scale orderings. In most cases, the repeatability scores are higher for our Sym-I and Sym-G detectors than for SIFT and MSER, with Sym-G performing particularly well. This suggests that our symmetry scores are better preserved under the types of variation exhibited in the dataset. On the other hand, we observed that for image pairs with few symmetries (such as the standard Graffiti pairs), DoG outperforms our detectors.

6.2. Evaluating descriptors

We now evaluate our local symmetry descriptor. For each pair of images we extract keypoints and descriptors and match descriptors using the standard ratio test [10] on the top two nearest neighbor distances. By varying the threshold on the ratio score, and comparing the matched set of keypoints to ground truth (known in advance from the homography) we can obtain a precision-recall (PR) curve that summarizes the quality of the match scores.

We measure the impact of descriptor and detector separately as follows. First, we generate two sets of perfectly matched synthetic detections by creating a set of keypoints K_1 on a grid in I_1 (in our experiments the spacing between points is 25 pixels and the scale of each keypoint is set to 6.25). We then map these keypoints to I_2 using H_{12}, creating a matched set of keys K_2. We discard keypoints whose support regions are not fully within the image. We also extract SIFT (DoG), Sym-I, and Sym-G keypoints from each image, and describe all four types of detections with four feature descriptors: SIFT, the self-similarity descriptor of [19], our symmetry descriptor (SymD), and a combination of SIFT and our symmetry descriptor (SIFT-SymD) formed by concatenating the descriptors (after normalizing each to have unit norm). Evaluating this combined descriptor gives us a measure of the complementarity of the information provided by SIFT (a gradient-based descriptor), and SymD (a local symmetry-based descriptor).

PR curves for feature matching between sample image pairs are shown in Figure 8, and Table 2 reports mean average precision (mAP) for each combination of detector and descriptor over all pairs in the dataset. In general the SIFT descriptor outperforms SymD, although in some cases local symmetries give more robust matches, such as the pair in the top row of Figure 8. However, the combined SIFT-SymD descriptor often outperforms either descriptor alone, and the mAP score for the combined descriptors is higher than any single descriptor, as shown in Table 2, suggesting that the two descriptors provide complementary information. Among the detectors, the synthetic GRID detections unsuprisingly yield much higher mAP scores than the real detectors. Sym-I and Sym-G both generally yield a higher mAP score than the SIFT (DoG) detector given the same descriptor (except, somewhat surprisingly, when the Sym-I detector is combined with the SymD descriptor). Looking at individual pairs, we observe that our symmetry

[2] k_2^s will not necessarily have area A because its initial scale might differ from that of k_1.

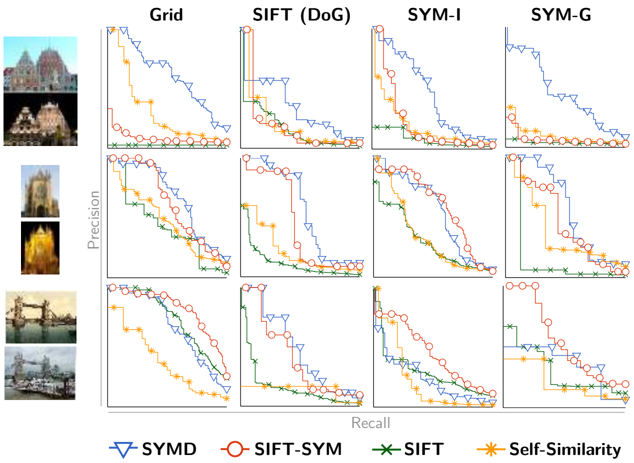


Figure 8: Precision-recall curves for a feature matching task on selected image pairs. Each column shows a different detector, and each row a different pair of images.

	GRID	SIFT(DoG)	SYM-I	SYM-G
Self-Similarity	0.29	0.14	0.12	0.16
SIFT	0.49	0.21	0.28	0.25
SYMD	0.41	0.22	0.20	0.25
SIFT-SYMD	0.58	0.28	0.35	0.36

Table 2: Mean average precision for different combinations of detector and descriptor. Each column corresponds to a different detector, and each row to a different descriptor.

descriptor performed especially well for dramatic illumination changes, but also boosted performance for some difficult pairs taken years apart. We also found that some pairs are so challenging that no technique we tried works well.

7. Conclusion

We have presented a new feature detector and descriptor based on local symmetries detected densely across image and scale space; these features are designed for finding correspondence between difficult image pairs rich in symmetries. To evaluate our method, we created a new dataset of image pairs with dramatic appearance changes, and showed that our features are more repeatable, and yield complementary information, compared to standard features such as SIFT. However, our dataset is still extremely challenging for state-of-the-art matching matches, and so we believe that image features are still a wide-open area. In the future, we plan to explore more comprehensive descriptors that describe complex patterns of symmetries across multiple scales, and build detectors that leverage more types of symmetry than the ones we use here.

Acknowledgements. This work is supported by the National Science Foundation under IIS-0964027 and the Intel Science and Technology Center for Visual Computing.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009. 2

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 2002. 6

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 4

[4] V. Di Gesù, V. Di Gesu, and C. Valenti. The Discrete Symmetry Transform in Computer Vision. Technical report, Universita di Palermo, 1995. 2

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 6

[6] P. Kovesi. Symmetry and asymmetry from local phase. *Australian Joint Conf. on Artificial Intelligence*, 1997. 2

[7] P. Kovesi. Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1999. 2

[8] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994. 4

[9] Y. Liu, H. Hel-Or, C. Kaplan, and L. V. Gool. Computational Symmetry in Computer Vision and Computer Graphics. *Foundations and Trends in Computer Graphics and Vision*, 2009. 2

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 4, 7

[11] G. Loy and J.-O. Eklundh. Detecting symmetry and symmetric constellations of features. In *ECCV*, 2006. 2, 4

[12] G. Loy and A. Zelinsky. Fast radial symmetry for detecting points of interest. *PAMI*, 2003. 2

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. 6

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool. A comparison of affine region detectors. *IJCV*, 2005. 6

[15] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A planar-reflective symmetry transform for 3D shapes. In *SIGGRAPH*, 2006. 2

[16] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: the generalized symmetry transform. *IJCV*, 1995. 2

[17] G. Schindler and F. Dellaert. Probabilistic temporal inference on reconstructed 3d scenes. In *CVPR*, 2010. 2

[18] G. Schindler, P. K. R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, 2008. 2

[19] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, June 2007. 2, 7

[20] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *SIGGRAPH Asia*, 2011. 2

[21] C. Wu, J.-M. Frahm, and M. Pollefeys. Detecting large repetitive structures with salient boundaries. In *ECCV*, 2010. 2

[22] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *ICCV*, 2011. 2, 4, 6