A SIFT Descriptor with Global Context

Eric N. Mortensen
Oregon State University
enm@eecs.oregonstate.edu

Hongli Deng
Oregon State University
deng@eecs.oregonstate.edu

Linda Shapiro
University of Washington
shapiro@cs.washington.edu

Abstract

Matching points between multiple images of a scene is a vital component of many computer vision tasks. Point matching involves creating a succinct and discriminative descriptor for each point. While current descriptors such as SIFT can find matches between features with unique local neighborhoods, these descriptors typically fail to consider global context to resolve ambiguities that can occur locally when an image has multiple similar regions. This paper presents a feature descriptor that augments SIFT with a global context vector that adds curvilinear shape information from a much larger neighborhood, thus reducing mismatches when multiple local descriptors are similar. It also provides a more robust method for handling 2D nonrigid transformations since points are more effectively matched individually at a global scale rather than constraining multiple matched points to be mapped via a planar homography. We have tested our technique on various images and compare matching accuracy between the SIFT descriptor with global context to that without.

1. Introduction

Given two or more images of a scene, the ability to match corresponding points between these images is an important component of many computer vision tasks such as image registration, object tracking, 3D reconstruction, and object recognition. Each point to be matched must be identified by describing it and its surroundings so that it can be matched to descriptions of points in another image. It is important that a point's description be as unique as possible while also allowing for various image transformations due to differences in lighting, object movement, and changes in camera pose.

This paper presents a feature descriptor that combines a local SIFT descriptor [9] with a global context vector similar to shape contexts [2]. The global context helps discriminate between local features that have similar local appearance. As such, we believe that this technique more closely matches human feature matching in that humans are able to augment local regions with the "big picture" that provides a overall reference to help disambiguate multiple regions with locally similar appearance.

We developed this descriptor as part of a environmental monitoring and assessment project that classifies insects according to their species. One such insect is stonefly larvae extracted from streams and imaged under a microscope (an example stonefly larva image is shown in Fig. 2(a)). In this project, we match feature points between different views of the insect for tracking, extraction of classification features, and ultimately 3D reconstruction.

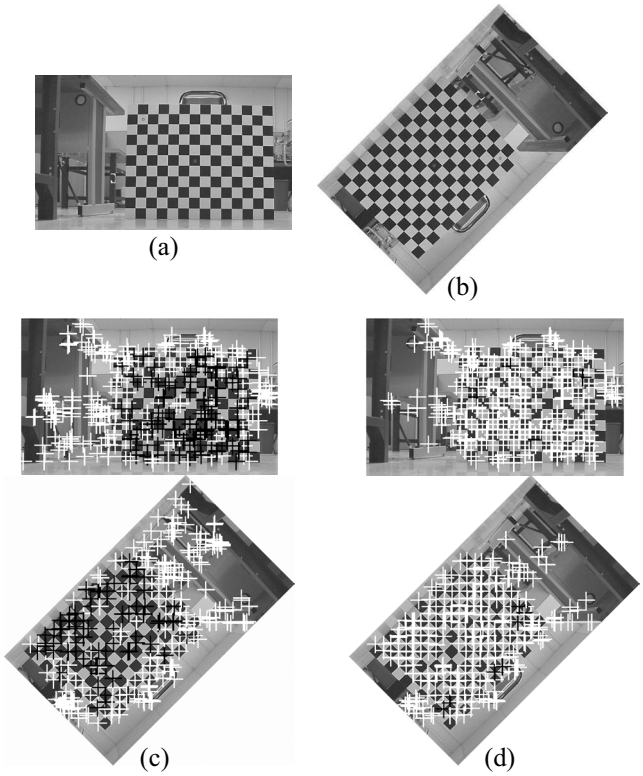

Figure 1: Comparison of matching results. (a) Original checkerboard image. (b) Rotated 135°. (c-d) Matches (white) and mismatches (black) using ambiguity rejection with (c) SIFT alone—268/400 correct matches (67%)—and (d) SIFT with global context—391/400 correct (97.75%).

Figure 1 illustrates the primary difficulty that this paper addresses. In particular, an image may have many areas that are locally similar to each other (such as the checkerboard pattern). Further, an object—such as the stonefly larva in Fig. 2(a-b)—may have a complex shape and thus exhibits non-affine distortions due to out-of-plane rotations or other articulated or non-rigid object movements. Multiple locally similar areas produce ambiguities when matching local descriptors while non-rigid distortions produce difficulties when matching groups of feature points with assumed 2D rigid body or affine transformations. The global shape context addresses both these problems by integrating global scope to resolve ambiguities while allowing for non-rigid shape transformations.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

2. Background

Point matching involves three steps: First, determine stable features that are of interest and/or are to be matched, second, describe each point, and third, match each point to those in another image by comparing descriptors. For example, dense stereo matching estimates the distance of every pixel from the camera by computing a pixel's epipolar disparity to triangulate its distance. In this case, every pixel is a interest point and the descriptor can be simply the pixel value itself or the pixel's neighborhood—sampled relative to epipolar geometry for rotation invariance and normalized to account for brightness and contrast changes. Since many areas produce nearly identical descriptors, global matching is often employed to resolve ambiguity by constraining epipolar matches.

The last few years has seen a lot of work in detecting, describing, and matching sparse feature points. Harris and Stephens [6] develop a corner detector that is robust to changes in rotation and intensity but is very sensitive to changes in scale. The Harris detector finds points where the local image geometry has high curvature in the direction of both maximal and minimal curvature, as provided by the eigenvalues of the Hessian matrix. They develop an efficient method for determining the relative magnitude of the eigenvalues without explicitly computing them.

Schmid and Mohr [16] detect key points with the Harris detector and describe them using local differential gray-level invariants. While the invariants are invariant to scale, rotation, and intensity changes, the Harris detector is not scale invariant, thus limiting the effectiveness of their technique to scale changes. Mikolajczyk and Schmid [11] develop a scale-invariant Harris detector that keeps key points at each scale only if it's a maximum in the Laplacian scale-space [8]. More recently, Mikolajczyk, Zisserman, and Schmid [14] integrate edge-based features with local feature-based recognition using a structure similar to shape contexts [2] for general object-class recognition.

David Lowe [9] uses a scale-invariant detector that finds extrema in the difference of Gaussian scale space. In [10], he fits a quadratic to the local scale-space neighborhood to improve accuracy. He then creates a Scale Invariant Feature Transform (SIFT) descriptor to match key points using a Euclidean distance metric in an efficient best-bin first algorithm where a match is rejected if the ratio of the best and second best matches is greater than a threshold.

In [2,3], Belongie, Malik, and Puzicha start with a collection of shape points (identified using the Canny edge detector, for example) and, for each point, build the relative distribution of the other points in log-polar space. The shape context is scale and rotation invariant and point differences are measured using the χ^2 statistic between the log-polar histograms. They use the Hungarian algorithm to find the best global one-to-one assignment followed by a thin-plate spline fit to warp one shape to the other.

There has been a lot of other work in detecting, describing, and matching feature points [1,12,15,19,20]. However, for the most part, the descriptors fail to account for global context and can therefore produce ambiguity when matching. To remedy this, many techniques assume 2D planar homographies (such as rigid or affine transformations) and reject bad matches using more computationally expensive group-wise or global consistency checks. The descriptor presented in this paper resolves the ambiguity problem by augmenting the local descriptor with global scope. As such, simple nearest neighbor matching is typically sufficient to resolve local ambiguity. Thus, our matching algorithm allows for reasonable non-rigid transformations since we do not need to constrain groups of points to match under some restricted transformation.

3. Interest Point Detector

As noted, the first step in point correspondence is feature (or interest) point detection. We have experimented with various feature detectors including the Harris corner detector [6], curvilinear structure detector [18], and extrema in difference of Gaussian (DoG) scale space [10]. To more accurately quantify performance of the feature *descriptors* without introducing variability due to differences in interest point *detectors*, we use the scale-space DoG extrema detection code available from David Lowe[1] that provides both interest points and the SIFT descriptor at each point.

The DoG is an approximation to the normalized Laplacian, which is needed for true scale invariance [8]. DoG scale space is sampled by blurring an image with successively larger Gaussian filters and subtracting each blurred image from the adjacent (more blurred) image. In this case, three levels of scale are created for each octave by blurring the image with incrementally larger Gaussian filters with scale steps of $\sigma = 2^{1/3}$. After completing one octave, the image with twice the initial scale is resampled by taking every other row and column and the process is repeated for the next octave, thus reducing computation.

Interest points are characterized as the extrema (maxima or minima) in the 3D (x, y, σ) space. As such, each pixel is compared with its 26 neighbors in scale space and a pixel is selected as a feature point if its value is larger or smaller than all of its neighbors. Subsample accurate position and scale is computed for each extrema point by fitting a quadratic polynomial to the scale space function $D(x, y, \sigma)$ and finding the extremum, giving

$$\hat{\boldsymbol{x}} = -\frac{\partial^2 D^{-1}}{\partial \boldsymbol{x}^2}\frac{\partial D}{\partial \boldsymbol{x}} \tag{1}$$

where $\boldsymbol{x} = (x, y, \sigma)$ and $\hat{\boldsymbol{x}}$ is the extremum position providing accurate position and scale.

1. PC linux binary code for detecting interest points and creating SIFT descriptors is available at http://www.cs.ubc.ca/~lowe/keypoints/

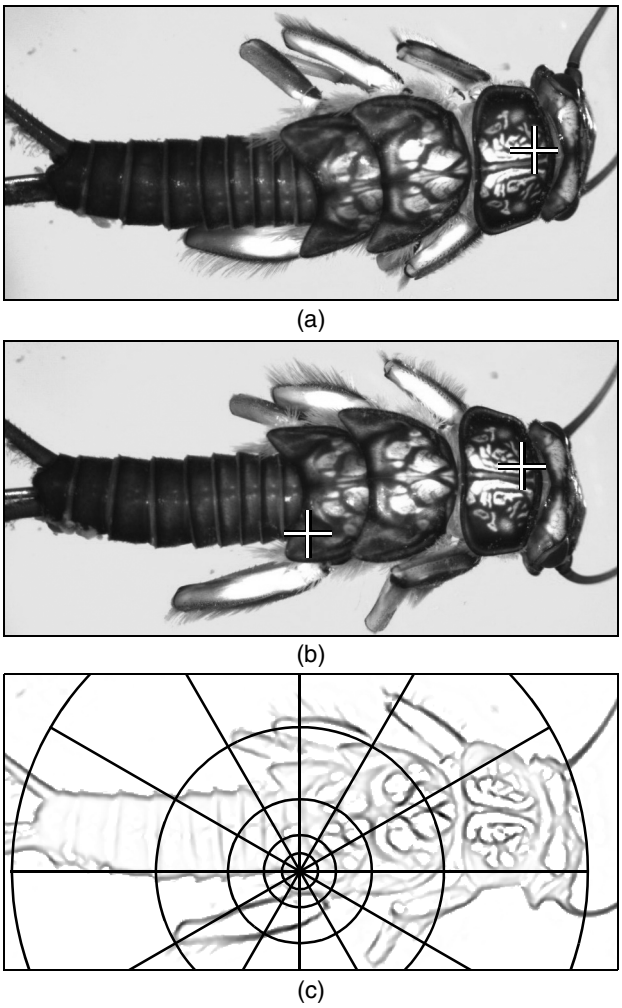

Figure 2: (a-b) Original images with selected feature points marked. (c) Reversed curvature image of (b) with shape context bins overlaid. (d) SIFT (left) and shape context (right) of point marked in (a). (e) SIFT and shape context of matching point in (b). (f) SIFT and shape context of random point in (b).

Finally, an orientation is assigned to each interest point that, combined with the scale above, provides a scale and rotation invariant coordinate system for the descriptor. Orientation is determined by building a histogram of gradient orientations from the key point's neighborhood, weighed by a Gaussian and the gradient magnitude. Every peak in the histogram with a height of 80% of the maximum produces a key point with the corresponding orientation. A parabola is fit to the peak(s) to improve accuracy.

4. Feature Descriptor

For every interest point detected, we built a two-component vector consisting of a SIFT descriptor representing local properties and a global context vector to disambiguate locally similar features. Thus, our vector is defined as

$$\boldsymbol{F} = \begin{bmatrix} \omega \boldsymbol{L} \\ (1-\omega)\boldsymbol{G} \end{bmatrix} \quad (2)$$

where $\boldsymbol{L}$ is the 128-dimension local SIFT descriptor, $\boldsymbol{G}$ is a 60-dimension global context vector, and ω is a relative weighting factor.

4.1 SIFT

The SIFT (Scale Invariant Feature Transform) [9,10] has been shown to perform better than other local descriptors [13]. Given a feature point, the SIFT descriptor computes the gradient vector for each pixel in the feature point's neighborhood and builds a normalized histogram of gradient directions. The SIFT descriptor creates a 16×16 neighborhood that is partitioned into 16 subregions of 4×4 pixels each. For each pixel within a subregion, SIFT adds the pixel's gradient vector to a histogram of gradient directions by quantizing each orientation to one of 8 directions and weighting the contribution of each vector by its magnitude. Each gradient direction is further weighted by a Gaussian of scale $\sigma = n/2$ where n is the neighborhood size and the values are distributed to neighboring bins using trilinear interpolation to reduce boundary effects as samples move between positions and orientations. Figure 2 shows the SIFT descriptor created for a corresponding pair of points in two stonefly images and a non-matching point.

4.2 Global Context

We use an approach similar to shape contexts [3] to describe the global context of each feature point. Like SIFT, shape contexts also create a histogram, but in this case they count the number of sampled edge points in each bin of a log-polar histogram that extends over a large portion of the image. Rather than count distinct edge points—detection of which can be sensitive to changes in contrast and threshold values—we compute the maximum curvature at each pixel. Given an image point (x, y), the maximum curvature is the absolute maximum eigenvalue of the Hessian matrix

$$H(x, y) = \begin{bmatrix} r_{xx} & r_{xy} \\ r_{xy} & r_{yy} \end{bmatrix} = I(x, y) * \begin{bmatrix} g^{\sigma}_{xx} & g^{\sigma}_{xy} \\ g^{\sigma}_{xy} & g^{\sigma}_{yy} \end{bmatrix} \tag{3}$$

where r_{xx} and r_{yy} are the second partials of the image in x and y, respectively, and r_{xy} is the second cross partial. The second derivatives are computed by convolving the image with the corresponding second derivative of a Gaussian, g_{xx}, g_{xy}, and g_{yy}, with scale σ—in this work we use a scale of $\sigma = 2$ pixels. Thus, the curvature image is defined as

$$C(x, y) = |\alpha(x, y)| \tag{4}$$

where $\alpha(x, y)$ is the eigenvalue of (3) with the largest absolute value. As noted in [18], $\alpha(x, y)$ can be computed in a numerically stable and efficient manner with just a single Jacobian rotation of H to eliminate the r_{xy} term. Figure 2(c) shows the curvature image (reversed for printing), $C(x, y)$, resulting from the insect image in Fig. 2(b).

For each feature, the global shape context accumulates curvature values in each log-polar bin. The diameter is equal to the image diagonal and, like [3], our shape context is a 5×12 histogram. Our implementation is not exactly log-polar since the radial increment of the center two bins are equal—thus, the bins have radial increments

$$\frac{r}{16}, \frac{r}{16}, \frac{r}{8}, \frac{r}{4}, \text{ and } \frac{r}{2}, \tag{5}$$

where r is the shape context's radius. Each pixel's curvature value is weighted by an inverted Gaussian and then added to the corresponding bin. The larger a pixel's curvature measure (shown as darker pixels in Fig. 2), the more it adds to its bin. The Gaussian weighting function is

$$w(x, y) = 1 - e^{-((x - x_f)^2 + (y - y_f)^2)/(2\sigma^2)} \tag{6}$$

where (x_f, y_f) is the feature point position and σ is the same scale used to weight the SIFT feature's neighborhood in Section 4.1. In this way, the weighting functions places more importance on features beyond the neighborhood described by SIFT and provides a smooth transition from the local SIFT descriptor to the global shape context.

To reduce boundary effects as pixels shift between bins and to improve computational efficiency, the curvature image is reduced by a factor of 4 with a low-pass Harr wavelet filter and the resulting image is further smoothed with a Gaussian filter of scale $\sigma = 3$ pixels. The shape context samples this reduced and smoothed image. Finally, the global context vector is normalized to unit magnitude so that it is invariant to changes in image contrast.

More specifically, if $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})^T$ is the feature point position with orientation θ, then

$$a = \left\lfloor \frac{6}{\pi}\left(\operatorname{atan}\left(\frac{y - \tilde{y}}{x - \tilde{x}}\right) - \theta \right) \right\rfloor \tag{7}$$

and

$$d = \max\left(1, \left\lfloor \log_2\left(\frac{\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|}{r}\right) + 6 \right\rfloor\right) \tag{8}$$

are the angular and radial-distance bin indices, respectively, for a point $\boldsymbol{x} = (x, y)^T$, where $\|\cdot\|$ is the L_2 norm and r is the shape context radius as used in (5). Let $\boldsymbol{N}_{a,d}$ be the neighborhood of points with bin indices a and d, then bin $\dot{\boldsymbol{G}}_{a,d}$ of the unnormalized histogram is computed by

$$\dot{\boldsymbol{G}}_{a,d} = \sum_{(x,y) \in \boldsymbol{N}_{a,d}} C'(x, y) \tag{9}$$

where C' is the reduced and smoothed curvature image from Eq. (4) as described in the previous paragraph. Finally, the normalized global shape context is given by

$$\boldsymbol{G} = \frac{\dot{\boldsymbol{G}}}{\|\dot{\boldsymbol{G}}\|} \tag{10}$$

In practice, $\boldsymbol{G}$ is computed by scanning the shape context's bounding box, computing the indices a and d for each pixel and incrementing the corresponding bin by $C'(x, y)$, and finally normalizing it to unit magnitude.

4.3 Rotation and Scale Invariance

Our combined feature descriptor, $\boldsymbol{F}$, in Eq. (2) is rotation invariant since both the SIFT descriptor and the shape context are constructed relative to the key point's orientation. Further, the SIFT descriptor is scale invariant since it is constructed in the key point's scaled coordinate frame. However, the size of the global context vector is a function of the image size rather than the interest point's scale and, as such, is not fully scale invariant—although some scale invariance is afforded by the smoothing and by the log-polar construction in that the log radial bins allow for increasing uncertainty as relative distance increases.

There are two reasons why the shape context size is not relative to the interest point scale. First, in our insect ID project, we only have minor scale changes. As such, we don't have the need for large scale invariance. Second, the range of scales returned by the feature detector is on the

order of a couple of pixels up to hundreds of pixels. To capture enough global scope for the feature's with the smallest scale, the radius of the shape context would need to be many (perhaps a hundred or more) times larger than the feature's scale. This would be impractical for the feature's with large scale since (a) a shape context that large would extend well beyond the image boundary and (b) the larger features don't really need the global context, as they already describe a large neighborhood.

As it is, the weighting function in Eq. (6) balances the contributions of the fixed-size shape context with the variable-size SIFT descriptor. When the SIFT scale is small, the shape context extends well beyond the SIFT descriptor's neighborhood to give the small neighborhood a global scope. For large local features that already describe large portions of the image, the shape context size is proportionally much smaller and Eq. (6) further reduces its relative contribution.

For our insect recognition project, we have explored a more robust option for achieving rotation and scale invariance. Since we already segment the insect prior to feature matching (the blue background simplifies automatic segmentation), we compute the principal axis of the segmented insect using principal component analysis (PCA) and build our feature descriptor relative to this global orientation. The principle axis is much more robust to noise and local transformations that would otherwise effect the local orientation computation described in Section 3. We also achieve scale invariance by constructing our shape context relative to the magnitude of the principal axis.

5. Matching

Given two or more images, a set of feature points that can be reliably detected in each image, and robust descriptors for those features, we next match feature points between images. Since our descriptor already includes global shape information, we don't need to perform expensive groupwise or global consistency checks when matching. Consequently, we compare descriptors with a simple nearest neighbor distance or nearest neighbor with ambiguity rejection metric with a threshold on the match. If two or more points match to a single point in an other image, we keep the pair with the best match and discard the other(s).

Given the definition of our feature descriptor in Eq. (2) and two descriptors, $\boldsymbol{F}_i$ and $\boldsymbol{F}_j$, our distance metric is a simple Euclidean distance metric

$$d_L = \left|L_i - L_j\right| = \sqrt{\sum_k (L_{i,k} - L_{j,k})^2} \tag{11}$$

for the SIFT component, $\boldsymbol{L}$, of the feature vector and a χ^2 statistic

$$d_G = \chi^2 = \frac{1}{2}\sum_k \frac{(h_{i,k} - h_{j,k})^2}{h_{i,k} + h_{j,k}} \tag{12}$$

for the shape context component, $\boldsymbol{G}$. The χ^2 measure is appropriate since it normalizes larger bins so that small differences between large bins—which typically have much greater accumulated values—produce a smaller distance than a small difference between the small bins (which have small values to begin with) [3]. The final distance measure value is given by

$$d = \omega d_L + (1 - \omega) d_G \tag{13}$$

where ω is the same weight used in Eq. (2). For the results presented here, we use a value of $\omega = 0.5$.

Finally, we discard matches with a distance above some threshold T_d. Since the components of our feature vector, $\boldsymbol{F}$, are normalized, we can apply a meaningful threshold that will be consistent across multiple images and transformations. In this work, we use $T_d = 0.5$.

6. Results

To assess matching rate, we artificially transform images so as to automatically determine if a match is correct. Figures 1, 3-5 compare the matching rate between SIFT alone and SIFT with global context (SIFT+GC). For a given descriptor (SIFT or SIFT+GC), we match each feature point in the original image with feature points in the transformed image using both nearest neighbor (NN) and ambiguity rejection (AR). Like [10], ambiguity rejection throws out matches if the ratio of the closest match to the second closest match is greater than 0.8. The resulting matches for both NN and AR (after discarding ambiguous matches) are then sorted from best (lowest matching distance) to worst and the best 50, 100, 200, 300, 400, etc. matches are chosen for comparison. A match is correct if it is within 4 pixels of its predicted position.

In Figure 3, SIFT alone correctly matches some of the windows since the reflection of clouds disambiguates the otherwise similar local features. Note that the SIFT scale for both the checkerboard squares in Fig. 1 and the windows in Fig. 3 are large enough to include neighboring squares or windows. Thus, SIFT correctly matches squares on the edge of the checkerboard since the feature neighborhoods extend beyond the edge of the checkerboard; likewise for the windows. Despite this, SIFT+GC still increases the matching rate significantly for these images.

Figure 4 plots the matching rate as a function of the number of matched points for SIFT and SIFT+GC using both NN and AR matching. Matching rates are computed using the artificially transformed images in Figures 1, 3, and 5—four images each for rotation, skew, and both rotation and skew. Note that SIFT+GC has a consistently higher matching rate for a given matching technique and, in many cases, SIFT+GC using NN matching produces a higher matching rate than SIFT alone using AR matching.

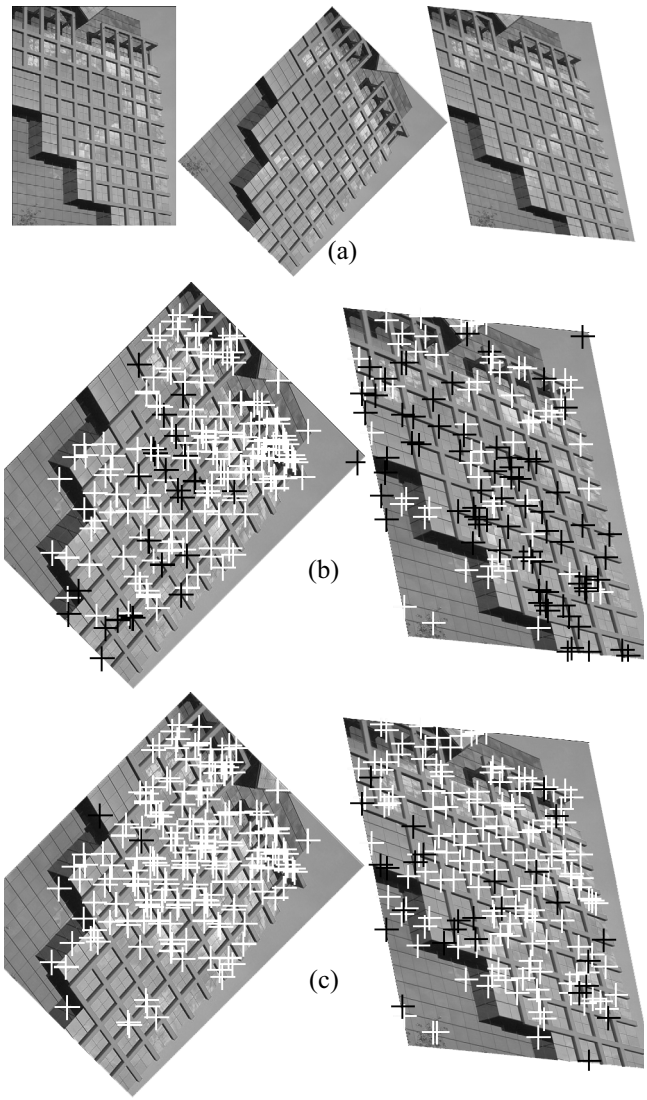

Figure 3: (a) Original and transformed images. Matching results in transformed images using nearest neighbor with (b) SIFT only—rotate: 170/200 correct (85%); skew: 73/200 correct (37%);—and (c) SIFT with global context—rotate: 198/200 correct (99%); skew: 165/200 correct (83%). The corresponding matching points from the original image are not shown.

Finally, Fig. 6 plots the matching rate of SIFT+GC as a function of the relative weighting factor, ω, used in Eqs. (2) and (13) for the images in Figures 1 and 3 as well as the average over all images. As noted earlier, we use a value of $\omega = 0.5$ in all our results.

7. Conclusion and Future Work

This paper presents a technique for combining global context with local SIFT information to produce a feature descriptor that is robust to local appearance ambiguity and non-rigid transformations. Future improvements include making the global context scale invariant by making its size a function of the SIFT feature size and normalizing each bin by the amount of actual image data it contains relative to the bin area—thus ignoring bins that are mostly or completely outside the image. We will also explore another idea where we accumulate descriptors themselves in the shape contexts bins and compare bins by comparing differences between descriptors in each bin. Finally, we will conduct a more comprehensive quantitative study comparing matching rate of SIFT+GC to other techniques using various options for detection, description, and matching under various image transformations.

Acknowledgements

This work was supported by NSF grant 0326052.

References

[1] A. Baumberg, "Reliable feature matching across widely separated views," in *CVPR*, pp.774-781, 2000

[2] S. Belongie, J. Malik and J. Puzicha, "Shape context: A new descriptor for shape matching and object recognition," in *NIPS*, pp. 831-837, 2000.

[3] S. Belongie, J. Malik and J. Puzicha, "Shape matching and object recognition using shape contexts," *PAMI*, 24(4):509-522, 2002.

[4] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching," in *CVPR*, pp. 44-51, 2000.

[5] A. D. J. Cross and E. R. Hancock, "Graph matching with a dual-step EM algorithm," *PAMI*, 20(11):1236-1253, 1998.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *Fourth Alvey Vision Conf.*, pp. 147-151, 1988.

[7] T. M. Koller, G. Gerig, G. Szekely, and D. Dettwiler, "Multiscale detection of curvilinear structures in 2-D and 3-D image data," in *ICCV*, pp. 864-869. 1995

[8] T. Lindeberg, "Feature detection with automatic scale selection," *IJCV*, 30(2):79-116, 1998.

[9] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, pp. 682-688, 1999

[10] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, 60(2):91-110, 2004.

[11] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *ICCV*, pp. 525-531, 2001.

[12] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *ECCV*, vol. I, pp. 128-142, 2002.

[13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *CVPR*, pp.257-264, 2003

[14] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," in *Proc. of the British Machine Vision Conference*, Norwich, U.K, 2003.

[15] P.Pritchett and A.Zisserman, "Wide baseline stereo matching," in *ICCV*, pp. 754-760, 1998.

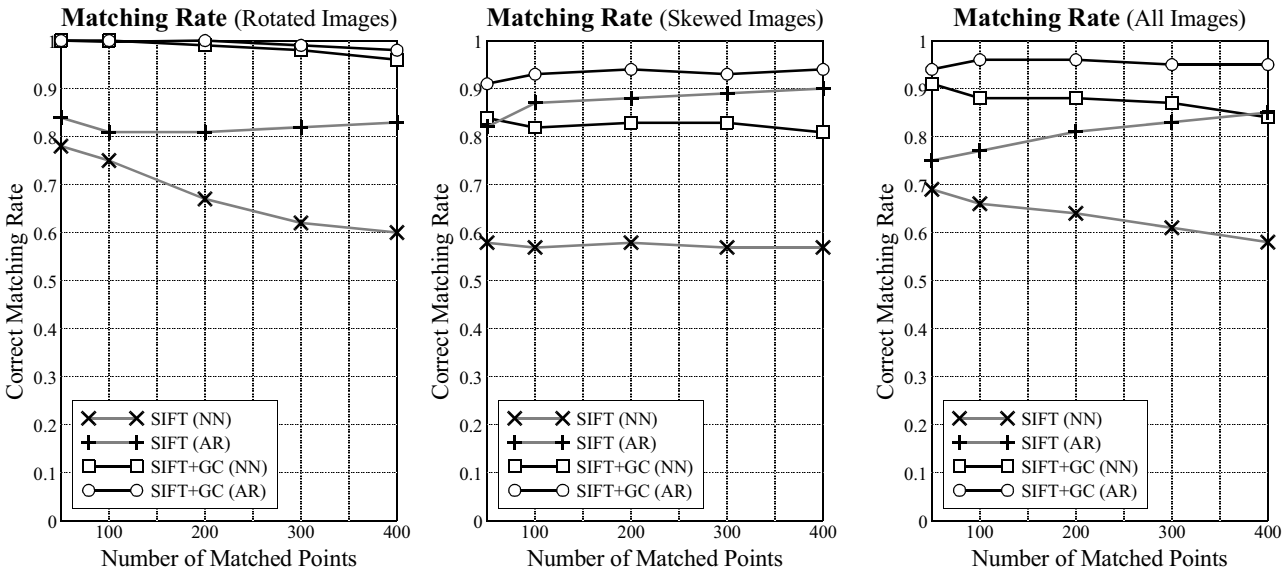

Figure 4: Matching rate as a function of matched points for the (left) rotated images (see Fig. 3), (middle) skewed images, and (right) all images (including images with both rotation and skew). Matching rate is computed for SIFT alone and SIFT with global context (SIFT+GC) using both nearest neighbor matching (NN) and ambiguity rejection (AR).

[16] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *PAMI*, 19(5):530-534, May 1997.

[17] C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and evaluating interest points," in *ICCV*, pp.230-235, 1998.

[18] C. Steger, "An unbiased detector of curvilinear structures," *PAMI*, 20(3):113-125, 1998.

[19] H. Tagare, D. O'Shea, A. A. Rangarajan, "Geometric criterion for shape based non-rigid correspondence," in *ICCV*, pp. 434-439, 1995.

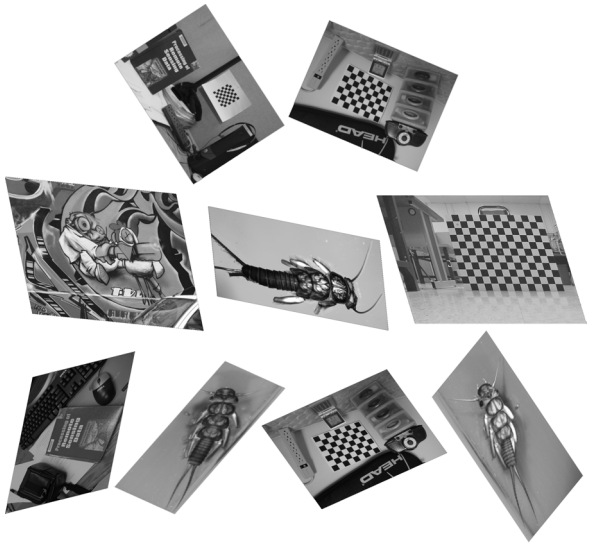

Figure 5: Images used to compute matching rates shown in Fig. 4. Rotated images (top) also include Figures 1.b and 3.a(center) and the skewed images (middle) also includes Fig. 3.a(right). The bottom row of images exhibit both skew and rotation.

[20] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, pp. 87-119, 1995.

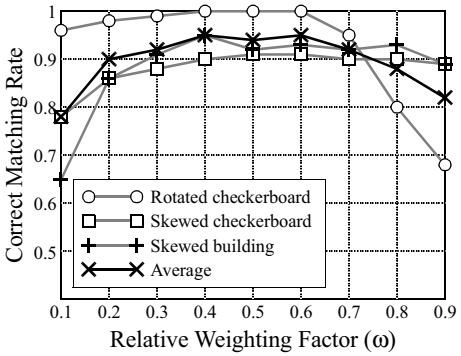

Figure 6: Correct matching rate for 200 matching points as a function of the relative weighting factor (ω) as used in Eqs. (2) and (13).