Global Context Descriptors for SURF and MSER Feature Descriptors

Gail Carmichael
School of Computer Science
Carleton University
Ottawa, ON, Canada, K1S 5B6
gbanaszk@connect.carleton.ca

Robert Laganière
VIVA research lab
SITE, University of Ottawa
Ottawa, ON, Canada, K1N 6N5
laganier@site.uottawa.ca

Prosenjit Bose
School of Computer Science
Carleton University
Ottawa, ON, Canada, K1S 5B6
jit@scs.carleton.ca

Abstract

Global context descriptors are vectors of additional information appended to an existing descriptor, and are computed as a log-polar histogram of nearby curvature values. These have been proposed in the past to make Scale Invariant Feature Transform (SIFT) matching more robust. This additional information improved matching results especially for images with repetitive features. We propose a similar global context descriptor for Speeded Up Robust Features (SURFs) and Maximally Stable Extremal Regions (MSERs). Our experiments show some improvement for SURFs when using the global context, and much improvement for MSER.

1 Introduction

The process of matching two images by finding points of interest (also called feature points) that correspond to one another has been a heavily researched topic in the field of computer vision. There are many applications of this research, from object recognition to the determination of the geometry between two cameras. Most of the focus has been on the case of matching two planar images, such as those captured as photographs. These correspondences can be used for such applications as wide baseline matching, 3D reconstruction, image retrieval, and building panoramas.

There have been many feature detectors and descriptors proposed, including the popular Scale Invariant Feature Transform (SIFT) [9], its successor Speeded Up Robust Features (SURF) [2], and the affine invariant Maximally Stable Extremal Regions (MSER) [10]. While these feature types have been shown to work well in many situations [12, 13, 14], they may benefit from an addition of some additional context in an image with many repeating features. In fact, a global context descriptor has been proposed as an addition to SIFT descriptors [15, 7]. We propose a similar global context descriptor for SURF and MSER, and compare the performance of these feature types for several images.

An explanation of these features types (SIFT, SURF, and MSER) is provided in the background section next, along with information about the existing methods of producing a global context descriptor for SIFTs. A global context descriptor for SURF and MSER in Section 3 is followed by the results of experimentation in Section 4. Conclusions and suggestions for improvement are given in Section 5.

2 Background

The first step in matching features is to reliably locate points or areas of interest. Many detection methods aim to find points or corners in an image using image contours, image intensity, or parametric models [16]. For example, the popular Harris corner detector [6] uses image derivatives to locate intensity changes in two directions, indicating the presence of a corner. Because corners are detected throughout an image with good repeatability, this is one of the most popular detectors. Unfortunately, Harris corners are quite sensitive to changes in image scale, and become less repeatable under such conditions [16].

There has been much work on scale-invariant feature detection [4, 8, 11, 17], the highlight of which is Lowe's [9]

Scale Invariant Feature Transform (SIFT). Coupled with the SIFT descriptor, feature points are translation, rotation, and scale invariant, and are often chosen as the best among other detectors and descriptors [12, 14]. However, like most detector/descriptor combinations, SIFT only works well up to about a 30° change in viewpoint between the two images being matched, with larger changes handled for images of planar surfaces [14].

Bay et al. developed Speeded Up Robust Features [2] to improve the runtime efficiency of SIFT while still obtaining good matching results. The SURF detector is based on the determinant of the Hessian matrix and second order Gaussian derivative approximations. It makes use of box filters to compute the Hessian since the computational cost of evaluating these filters is independent of their size once the integral image has been computed. According to an analysis comparing SIFT and SURF [1], SURF does indeed perform more efficiently than SIFT with a smaller but sufficient number of quality detected points.

A high quality affine co-variant detector was developed to look for what are called Maximally Stable Extremal Regions, or MSERs [10]. In this case, the features are actually shapes rather than points or corners. This detector can be described in simple terms using its similarity to the watershed algorithm [18] for image intensities. Suppose that the image represents a terrain viewed from above, where black areas are low ground and white areas are high ground. If the terrain is slowly flooded, certain areas will collect water in such a way that the pool does not change shape for some time. These areas are considered to be the most stable and are chosen as features. When combined with certain descriptors, MSERs perform very well when detected on flat surfaces, and have average performance for use with images of 3D objects [14]. They can also work well for changes in illumination between images [5].

After features have been located in an image, some unique way of describing them is required so that features in another image can be compared, and correspondences found. The SIFT framework defines how to describe a feature point in addition to the detection method mentioned above. A scale for each feature is decided during SIFT's difference of Gaussians detection process. The scale determines the local working area around a feature point. The SIFT feature descriptor obtains rotation invariance by detecting one or more prominent orientations from the image gradient (obtained from the image derivative), and then rotating the working area to match. The rotated working area is divided into a 4x4 grid, totalling 16 regions. An 8-bin histogram is filled by the directions of the image gradients found in each region. The counts in these bins for all regions are used to form the SIFT descriptor, which will be a vector of size 128.

The SURF descriptor relies on first order Haar wavelet responses in the x and y directions, differing from the use of gradients in SIFT. This, the authors claim, makes calculating the SURF descriptors more efficient. Like SIFT, SURF also assigns an orientation to each feature point. The typical SURF descriptor is a vector of length 64. Being smaller than the SIFT descriptor by half, fewer comparisons are needed for computing distances between feature descriptors while finding possible matches.

Detected MSER features can be described in a variety of ways. In the original work describing MSERs [10], Matas et al. proposed an affine invariant procedure that uses several multiples of the original MSER as measurement regions, transforming the measurement region so its covariance matrix is diagonalized, and then computing rotational invariants based on complex moments. This method is based on the actual intensity values found in the image, but it is also possible to describe MSERs by their shape alone. Another approach [3] uses local affine frames defined from affine-invariant sets of three points chosen from the MSER contour and centroid. A more recent method given by Forssén and Lowe [5] works with affine-normalized patches that contain either the actual image values, or a binary image representing the MSER shapes. In this case, the SIFT descriptor is used to describe the patches, as it was evaluated as the best choice for describing MSERs [14].

3 Global Context for SURF and MSER

While SURF and MSER features have performed well in practical experiments [1, 5, 13], they may be benefited with the addition of a global context descriptor in the same way SIFT results were improved. To help distinguish repetitive features in an image, two similar frameworks were proposed for augmenting a SIFT descriptor with a global context vector. The first work by Mortensen et al [15] uses the whole image to gather context from all surrounding pixels, while the second by Li and Ma [7] uses a local measurement area for each feature and gathers context only from other feature points within this area. In both cases, a log-polar histogram tallies principal curvature points in the measurement region. The principal curvature $c(x, y)$ is taken as the maximum absolute eigenvalue of the Hessian matrix for the region. We propose computing the global context of a SURF point with a similar technique used for SIFT, and adapt this strategy for use with MSERs.

Essentially the same algorithm that was used for computing the global context descriptor for SIFT points may be used with SURF. The only difference in the implementation suggested here is that the SURF points will not have affine covariant measurement regions, which were used by Li and Ma, but not Mortensen et al. That is, while Li and Ma modified SIFT detection to be affine covariant by using elliptical regions transformed into circles, we will not mod-

Algorithm 1 Compute the global context vector for a single SURF feature located at $\tilde{\mathbf{x}}$.

1. Compute the curvature of the entire image by finding the maximum absolute eigenvalue of the Hessian matrix for each pixel.
2. Compute the circular measurement region with radius R as K times the scale of the SURF feature.
3. For each other SURF feature whose location $\mathbf{x}$ is within R pixels of the SURF feature point location $\tilde{\mathbf{x}}$:
   (a) Compute radial and angular bin for SURF location $\mathbf{x}$.
   (b) Weigh curvature value found in the curvature image at $\mathbf{x}$ with an inverse Gaussian and chosen σ.
   (c) Add weighted curvature value to computed bin.
4. Create vector by flattening radial and angular bins.

ify SURF detection in any way. While this can occasionally be a disadvantage for certain image pairs with large differences in viewing angle, it is a simple addition to the existing SURF detection and description algorithm and requires fewer steps. As will be seen in the next section, experimental results are satisfactory.

3.1 Computing Global Context for SURF

The computation of the SURF global context descriptor is outlined in Algorithm 1. The main idea is to build the global context descriptor using other SURF points that lie within a radius of K times the scale at which the SURF point in question was detected. The curvature is computed once for the entire image and queried as needed by individual SURF points that are included in the global descriptor computations.

In more detail, the first step is to compute the curvature of the entire image using the Hessian matrix:

$$H(x,y) = \begin{pmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{pmatrix} = I(x,y) * \begin{pmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{pmatrix} \quad (1)$$

Then, for each SURF point $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})$ a measurement region is defined as a circle with radius K times the scale at which the point was detected. All SURF points $\mathbf{x}$ that lie within the circular measurement region are found. Each of these nearby SURF points $\mathbf{x} = (x, y)$ in the measurement region are placed in the appropriate angular bin φ and radial bin ρ. There are 12 angular and 5 radial bins, as was proposed in both methods for computing global context for SIFT. The bin indices are computed as

$$\varphi = \left\lfloor \frac{6}{\pi} \left(\arctan\left(\frac{x - \tilde{x}}{y - \tilde{y}} \right) - \alpha \right) \right\rfloor \quad (2)$$

and

$$\rho = \max\left(1, \log_2\left(\frac{r}{r_{max}}\right) + 6\right) \quad (3)$$

where α is the angle at which the SURF point $\tilde{\mathbf{x}}$ was detected and r_{max} is the radius of the measurement region.

The curvature value at $\mathbf{x}$ is weighted with the inverse Gaussian of equation

$$w(x,y) = 1 - e^{-((x-\tilde{x})^2 + (y-\tilde{y})^2)/(2\sigma^2)} \quad (4)$$

and added to the appropriate bins computed as above. When all pixels in range have been added, the bins are flattened into a single vector with dimension 60 and the vector is normalized to have unit length one. This is the global context vector, which can be matched with any preferred method, such as thresholding or nearest-neighbour ratio thresholding. In the experiments, the global context vector is compared using the χ^2 metric, just as the SIFT descriptors are.

3.2 Computing Global Context for MSER

A global context framework was also developed for use with MSERs to improve matching abilities in image pairs with repetitive features. The general idea is the same as finding the global context descriptor for SIFT and SURF feature points in that nearby curvature values are collected in a log-polar histogram for each MSER feature. The two main questions are what the measurement region should be, and how to best collect these curvature values.

The measurement region for the global context descriptor for SURF features is related to the scale at which the SURF point was detected and an elliptical affine region around the feature point. A possible application of this to MSERs, then, would be to use the bounding ellipse of the MSER shape itself as the measurement region. This idea does not work here because of the way MSERs are detected. MSERs are areas with relatively stable intensity values. This means that there often won't be any significant curvature values within the MSER region, and that the bounding ellipse may not contain many curvature values beyond the actual boundaries of the shape. In other words, whatever curvature is available within the bounding ellipse often does not provide much context in terms of surrounding features.

A potentially good measurement region, then, may not be the bounding ellipse of the feature itself, but perhaps some multiple K of it. This way, a certain amount of surrounding context will be measured beyond what is available

in the SIFT patch descriptors. Choosing K involves balancing between having a more distinct descriptor, and remaining robust to image transformations between two views.

Within the measurement region of the SIFT or SURF approach for global context, curvature values were collected at each other nearby feature point. The initial instinct is to do the same for MSERs: collect curvature values at the centroids of MSERs that fall within a particular measurement region. As discussed above, however, there is often not much or any curvature within an MSER's bounds, where the centroid usually falls. Even where there are non-zero curvature values, it is clear that this one value does not do a good job of representing the feature as a whole. The location of other features might not be represented in the global context when the centroid does not have a positive curvature value, and additional information distinguishing the features is lost.

To incorporate more information about the features found in the measurement region, one might consider using all of the curvature values within the bounds of each nearby MSER (or a slight expansion of the region to ensure boundary pixels are considered). Each of these values would be individually placed in the appropriate log-polar bins. While this would indeed consider the shapes of MSERs in their entirety, there is a disadvantage. If an MSER with significant size was detected in one image but not the other, its many curvature values would increase the difference between the global context vectors significantly.

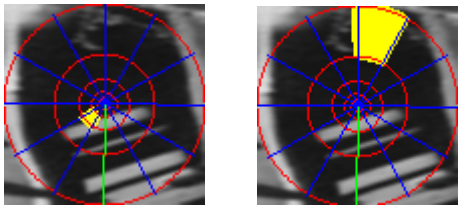
Instead, the approach from earlier work [15] is borrowed. There, all curvature values within the measurement region (which happened to be the entire image in that case) are incorporated into the global context vector. Instead of using the entire image, only pixels within K times the MSER bounding ellipse would be considered. This approach ensures that the number of nearby MSERs need not agree in the two images. Using this idea, the process used to obtain global context descriptors for MSERs is shown in Algorithm 2.

The first step is to normalize the region, transforming the bounding ellipse into a circle. This is accomplished in the same way that shape and texture patches are created by Forssén and Lowe [5]. The global context descriptor can be computed for both shape and texture types of MSER patches. However, in both cases, the pixels from the original image are resampled into the new patch. Thus, the global context descriptor should be the same for a shape and texture patch that are based on the same MSER feature.

Next, the curvature image is found for the patch. The Hessian matrix in equation (1) is computed, and the curvature values set as its maximum absolute eigenvalues. By using the patch version of the MSER to obtain the curvature image, a constant σ may be used. In the experiments that follow, $\sigma = 0.5$.

Algorithm 2 Compute the global context vector for an MSER feature.

1. Normalize the MSER region (shape or texture, same as the original MSER patch) into a square patch of size $2R$ by $2R$. The area included in the patch is K times the bounding ellipse of the MSER region. The patch will be sampled from the original image for both shape and texture MSER features.
2. Compute curvature of new patch by finding the maximum absolute eigenvalue of the Hessian matrix at each pixel.
3. For each pixel within R pixels of the centre of the patch:
   (a) Compute radial and angular bin for pixel.
   (b) Weigh curvature value found at that pixel with an inverse Gaussian and chosen σ.
   (c) Add weighted curvature value to computed bin.
4. Create vector by flattening radial and angular bins.



(a) Angular bin 2, radial bin 3.

(b) Angular bin 7, radial bin 5.

Figure 1: The log-polar graph of a patch shown with a particular bin roughly highlighted.

A log-polar graph is used again here. To achieve rotation invariance, the same feature angle α used when computing the SIFT descriptor for MSER patches will be used to position the first angular bin. In other words, the log-polar graph will be rotated by the same angle. Bin indices are computed as before. Figure 1 shows a patch with the log-polar graph superimposed. The green line indicates where the first angular bin begins, based on the feature's angle calculated using image gradient when computing a SIFT descriptor. In this case, the y-axis points down, so positive angles are in the clockwise direction. Yellow pixels are used to roughly highlight which pixels belong to a particular bin.

4 Experimental Results

We conducted several experiments to compare how well SURFs and MSERs work for certain image types both with

and without global context descriptors. The images used in these experiments were provided online by the Oxford Visual Geometry Group[1] and are used often when comparing the performance of feature detectors and descriptors (e.g. [12, 13]). Three image sets, each with six varying viewpoints, were chosen, as shown in Figure 2. The graffiti and wall image have changes in the viewing angle up to 60°, while the boat image set has changes to the zoom and rotation between the images.

The code used in the experiments is a mixture of our own implementations and those available freely online. The detection and description of SURF interest points is packaged as a Matlab mex interface[2], while the MSER region detector is from the VLFeat library[3]. Evaluation code from the Oxford group is used to determine how many SURF or MSER regions can be considered matches; in the case of SURF, the code is modified to work without applying an affine transformation to the regions between images.

Once MSER regions have been detected, we use our own implementation of Forssén and Lowe's [5] approach to describing them. The bounding ellipse of the region is transformed to a circle of a particular size, and the image contents are resampled into a small patch of size 41×41. Two patches are made for each region: one that uses the original image contents, called the texture patch, and one that uses the region's shape mask, called the shape patch. A SIFT descriptor is built for each patch.

The global descriptors for both SURF and MSER are computed using the methods presented in the previous section. The multiple of region size for MSERs is $K = 4$ and for SURF is $K = 10$. Several values were tested, and these were deemed to provide the best balance between distinctiveness and robustness.

The matching results presented next are obtained using nearest neighbour ratio thresholding with a threshold of 0.8. The χ^2 norm is used to find the distance between two MSER SIFT descriptors as suggested by Forssén and Lowe, while the SURF descriptors use the standard Euclidean norm. Shape patches are compared only to other shape patches, and texture patches are compared only with texture patches, but the resulting matches from both types of patches are considered together. When using a global descriptor as well, the nearest neighbours are first found based only on the SURF or MSER descriptors. The results are then filtered by thresholding the χ^2 norm between the global descriptors using a threshold of 0.05 for MSER and 1.6 for SURF (remembering that these descriptors are computed differently). Match correctness is evaluated by applying the homography associated with the image pair to the feature in the first image, and then thresholding the perpindicular distance to the match in the second image.

Matching results are shown in Figures 4-6 with a legend for all the graphs appearing in Figure 3. Each graph shows results for SURF and MSER both with and without global context descriptors. The first graph, recall, shows how many matches found were correct out of the total number of possible correct correspondences. In the case of MSERs, the total number of regions is taken to be twice the number of MSERs that match to account for the fact that there are two patches for each region – one shape patch, and one texture patch – and thus two possible matches. The second graph, 1-precision, is given by

$$1 - precision = \frac{\text{\#false matches}}{\text{\#correct matches + \#false matches}}$$

where a lower number is better. The last graph depicts what percentage of all matches were found to be correct.

The results for the graffiti images are in Figure 4. The recall for most images for all types of matching is relatively low, with most values being under 30%. The recall for results using the global descriptor is lower than those that don't use it. This is to be expected, since the global descriptor is acting as a threshold on the same results that don't use global context. The 1-precision is almost the same for SURF features both with and without global context, as is the recall. These values differ greatly for MSERs, suggesting that the global context really helps distinguish MSER features. It is worth noting that while these experiments use the shape and texture MSER patches all together, one might choose one or the other to suit the application at hand and potentially obtain better results even without the global context. The percentage of correct matches in the third graph shows the value of using MSERs with global context for their performance at more extreme viewing angles, provided the number of correct matches could be improved.

The wall image results in Figure 5 also show that SURF matching it not improved greatly with the use of global context, but that MSER matching is. Interestingly, SURF performs similarly well as MSER, even for the more extreme changes in viewing angle, though SURF has much better recall for the smaller changes. The features seem to be less sensitive to changes in viewing angle for these highly textured images than in the graffiti images, which had more distinct shapes.

Finally, the boat image results in Figure 6 show a much larger gap between the recall of SURF with and without global context. Otherwise, the results are similar to those of the wall image, where the percentage of correct matches is high and the precision good for almost all feature types for all images.

An additional set of examples is shown in Figure 7 and Figure 8. The images were taken in Ottawa, Canada. The

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/

[2] http://www.maths.lth.se/matematiklth/personal/petter/surfmex.php

[3] http://www.vlfeat.org/

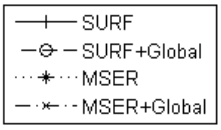


Figure 3: Legend for Figures 4-6.

first image in each pair of Figure 7 is a regular photograph, while the second image is one face of a panoramic image of the area. These images are difficult to match with high precision because of reptitive features (windows) and large variations in scale and blur. The results in Figure 8 were obtained using the same matching techniques used for the above results, but the correct matches were hand counted as the images are not related by a homography. The percentage of correct matches and the total number of matches are plotted against each other in a graph where the ideal location is the top right. For these images, the percentage of correct matches is noticeably improved for both SURF and MSER using global context, while the number of correct matches (related to recall) is again lower for both cases.

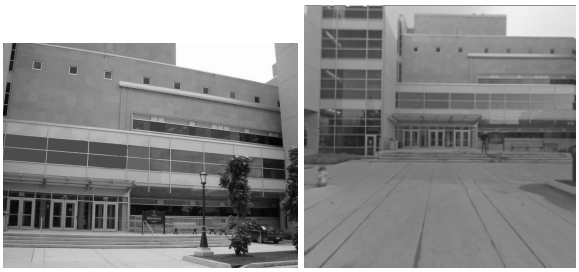
In summary, both feature types with global context have relatively stable precision and percentage of correct matches for viewpoint changes in the case of a textured and repetitive image set as well as for zoom and rotation, but not as stable for an image set with more clearly defined shapes. In all cases, recall goes down as the image pairs become more difficult to match. The percentage of correct matches for SURF is not always greatly improved with the use of the global context descriptor, even in the case of a more repetitive image. However, MSER matching results improve greatly. The recall (and thus the actual number of correct matches) is lower for MSERs, but it may be possible to improve this by, for example, using only shape or texture patches. Further experimentation for a particular image type may help find more appropriate region multipliers and thresholds for the global context descriptor, allowing for more matches at the possible cost of lowering the percentage of correct matches.

5 Conclusion and Future Work

Previous work has established a global context descriptor for SIFT features, used to improve matching results for situations where the SIFT descriptor alone did not prove to be distinctive enough. Based on this, we proposed a similar global context for SURFs and MSERs. While the SURF global context was essentially the same as that used for SIFT, constructing the MSER global context required more consideration.

In the experiments performed here, matching performance for SURF was almost the same with and without the

(a)



(b)

(c)

Figure 7: Additional test images taken in Ottawa.

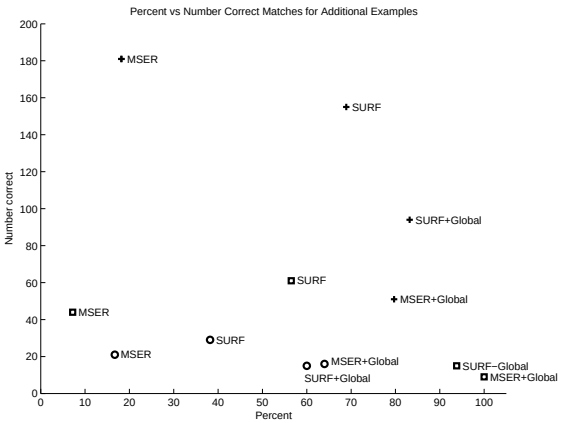
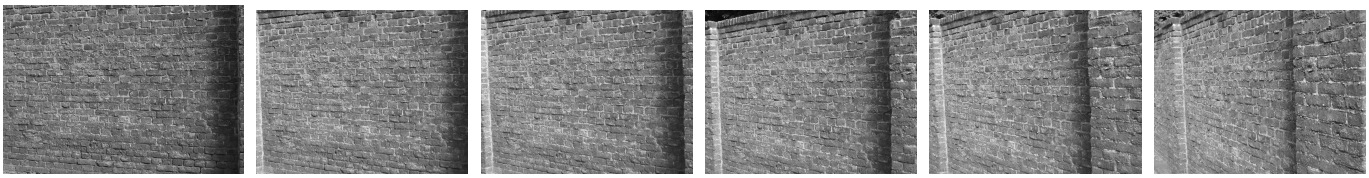


Figure 8: Matching results for images in Figure 7a (+), Figure 7b (o), and Figure 7c (□).

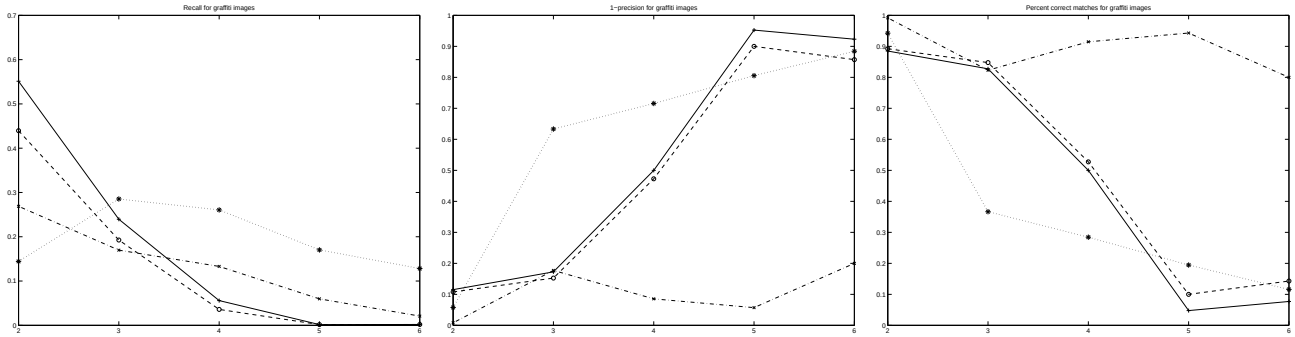
(a) Graffiti test images.



(b) Wall test images.

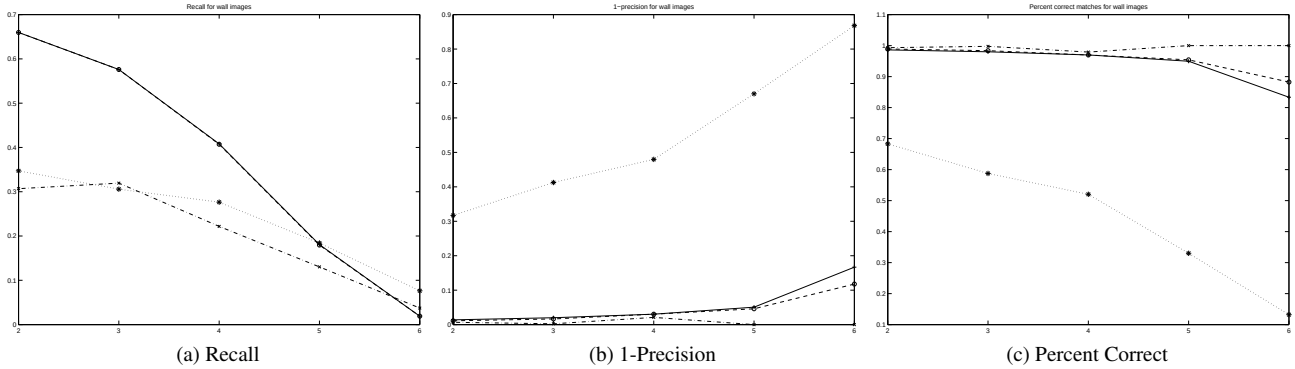
(c) Boat test images.

Figure 2: Test images.



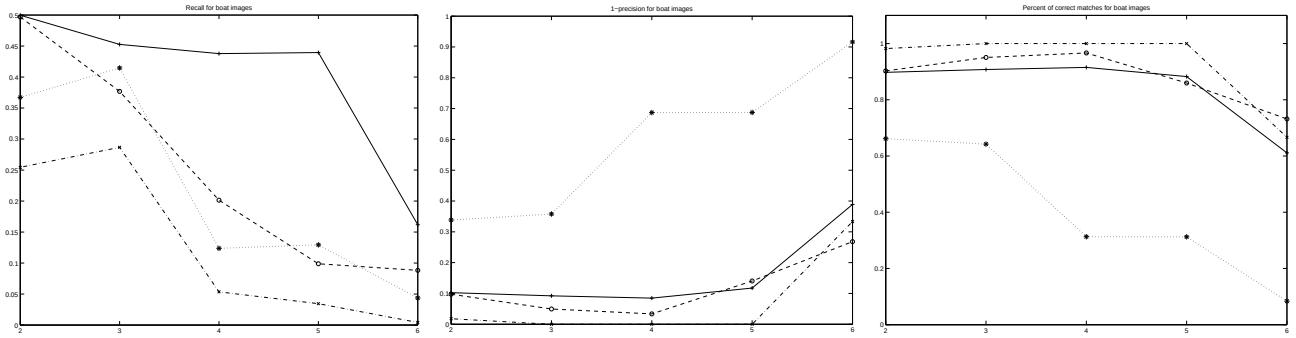
(a) Recall (b) 1-Precision (c) Percent Correct

Figure 4: Matching results for graffiti images.



(a) Recall (b) 1-Precision (c) Percent Correct

Figure 5: Matching results for wall images.



(a) Recall (b) 1-Precision (c) Percent Correct

Figure 6: Matching results for boat images.

global context. This suggests that SURF descriptors are actually fairly distinct without any additional context, at least for the image types tested here. On the other hand, results for MSER were much improved with the use of the global context. The trade-off is that the recall in this case was lower. However, it may be possible to improve this, perhaps even with simple improvements over which shape and texture patches are kept as matches, or making the choice between shape or texture patches for certain image types.

In the future, it would be worth examining the SURF global context descriptor further to see if there are cases of it improving matching performance more significantly. There are ways of improving the global context for MSER that are worth exploring as well, including enforcing a minimum measurement area to avoid too small a surrounding region for smaller MSERs. Finally, additional image types should be tested to see where global context is most useful.

References

[1] J. Bauer, N. Sünderhauf, and P. Protzel. Comparing several implementations of two recently published feature detectors. In *Proc. of the International Conference on Intelligent and Autonomous Systems*, 2007.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[3] O. Chum and J. Matas. Geometric hashing with local affine frames. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 879–884, Washington, DC, USA, 2006. IEEE Computer Society.

[4] A. C. Crowley, James L. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6, Issue:2:156–170, 1984.

[5] P.-E. Forssén and D. G. Lowe. Shape descriptors for maximally stable extremal regions. In *International Conference on Computer Vision (ICCV)*, October 2007.

[6] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[7] C. Li and L. Ma. A new framework for feature descriptor based on SIFT. *Pattern Recogn. Lett.*, 30(5):544–557, 2009.

[8] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21:224–270, 1994.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *In British Machine Vision Conference*, pages 384–393, 2002.

[11] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *In Proc. ICCV*, pages 525–531, 2001.

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65:2005, 2005.

[14] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. 73(3):263–284, July 2007.

[15] E. Mortensen, H. Deng, and L. Shapiro. A SIFT descriptor with global context. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 184–190 vol. 1, June 2005.

[16] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors, 2000.

[17] A. Shokoufandeh, I. Marsic, and S. J. Dickinson. View-based object recognition using saliency maps, 1998.

[18] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6):583–598, Jun 1991.