Shape Descriptors for Maximally Stable Extremal Regions

Per-Erik Forssén and David G. Lowe
Department of Computer Science
University of British Columbia
{perfo,lowe}@cs.ubc.ca

Abstract

This paper introduces an affine invariant shape descriptor for maximally stable extremal regions (MSER). Affine invariant feature descriptors are normally computed by sampling the original grey-scale image in an invariant frame defined from each detected feature, but we instead use only the shape of the detected MSER itself. This has the advantage that features can be reliably matched regardless of the appearance of the surroundings of the actual region. The descriptor is computed using the scale invariant feature transform (SIFT), with the resampled MSER binary mask as input. We also show that the original MSER detector can be modified to achieve better scale invariance by detecting MSERs in a scale pyramid. We make extensive comparisons of the proposed feature against a SIFT descriptor computed on grey-scale patches, and also explore the possibility of grouping the shape descriptors into pairs to incorporate more context. While the descriptor does not perform as well on planar scenes, we demonstrate various categories of full 3D scenes where it outperforms the SIFT descriptor computed on grey-scale patches. The shape descriptor is also shown to be more robust to changes in illumination. We show that a system can achieve the best performance under a range of imaging conditions by matching both the texture and shape descriptors.

1. Introduction

Recently there has been much interest in object detection and view matching using local invariant features [2, 7, 8, 9, 10, 11, 12, 14]. Such features allow correspondences to be found in cluttered scenes with significant amounts of occlusion. Computation of a local invariant feature basically consists in first detecting a local image region in an affine covariant manner. The next step is to sample a local patch of the input image in the covariant reference frame and describe its texture. The re-sampling step results in a representation of the local patch that is invariant to view changes. Typically the size of the local patch is significantly larger than the size of the initial region, and thus this approach makes the implicit assumption that the feature neighbourhood is locally planar.

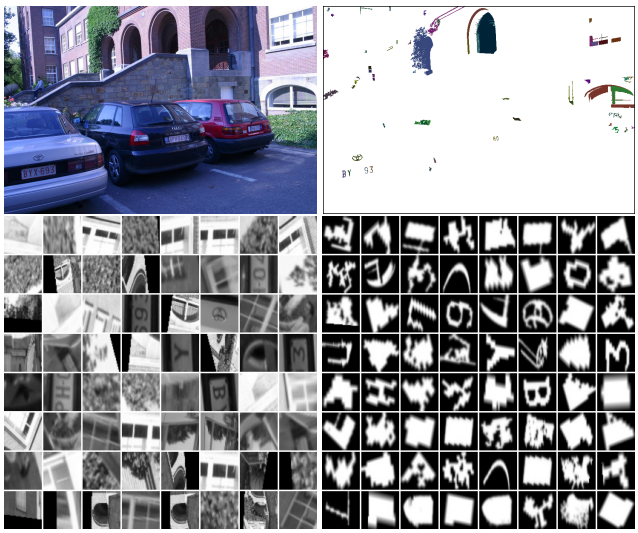

Figure 1. Top left: Input image. Top right: A random selection of 64 MSERs. Bottom left: grey-scale patches in normalised frames. Bottom right: MSERs in normalised frames.

Although successful for many scenes, such *texture patch descriptors* tend not to work so well on full 3D scenes under changing illumination conditions. They also have problems with scenes that contain structures at several different depths, and thus significant amounts of occlusion and changes of background to the detected features.

In this paper we will make use of Maximally Stable Extremal Region (MSER) features [8]. MSERs are regions that are either darker, or brighter than their surroundings, and that are stable across a range of thresholds of the intensity function. MSERs have also been defined on other scalar functions [13], and have been extended to colour [4]. Figure 1 shows an image from the on-line data-set at [17], and a random selection of MSERs detected in the image. The lower left image in the figure shows the affine normalised patches normally used to construct feature descriptors, and

the lower right image shows the MSER shapes themselves after affine normalisation. As can be seen, the detected MSERs exhibit a wide variety of shapes, and this motivates us to use them to construct *shape descriptors* as an alternative to the texture patch descriptors.

1.1. Related work

Another non-texture based descriptor for MSER was recently introduced by Chum and Matas [2]. Their approach relies on *local affine frames* (LAF). LAFs are defined from triplets of points affine-invariantly selected from the MSER contour and its centroid. The descriptor in their approach is based on two LAFs, one LAF is used as a reference frame, and one is used as a descriptor frame. The three points in the descriptor LAF are expressed in the reference frame and subsequently used to construct a descriptor. This descriptor is then used as an index to a hash-table. Their method is used to match planar scenes such as logos with different colours, from different view angles, and with and without transparency, and different spectral bands of satellite images, even though contrast is often inverted between the bands. Although these results are impressive, their method suffers from two serious limitations, which we amend in this paper. Firstly, the method requires that the two LAFs are co-planar (otherwise the descriptor will not be repeatable), which severely limits the applicability of the approach to full 3D scenes (especially scenes of non-man-made objects, where co-planarity is rare). Secondly, the use of only six scalars to describe the LAF pair effectively rules out the use of the descriptor for bag-of-features style object recognition (see *e.g.* [18]), since the frequency of hash-table collisions will become too high when many objects and views need to be stored.

In the experiment section, we will compare our shape descriptor to a texture descriptor. The texture descriptor we use is the SIFT descriptor computed on a grey-scale patch, re-sampled in an affine-covariant reference frame. This descriptor was found to be the best descriptor for MSER in a recent evaluation [11]. MSER with SIFT texture descriptors were also used recently in a large scale object recognition system by Nistér and Stewénius [12]. We are unable to compare our method with the LAF approach [2], since the LAF-point selection is not discussed in their paper.

Recently there has been considerable interest in using shape as a feature for object recognition by matching portions of edge contours extracted from the image [3, 15, 16]. This line of research is complementary to ours, since these papers match short fragments from object contours, whereas we match closed contours for stable regions. Due to the difficulty of identifying stable continuations in edge maps, this previous work has so far used shorter and less distinctive fragments than our region shapes. We rely on the stability of the MSER regions to provide us with a repeatable segmentation of the complete closed contour. However, these previous approaches share with ours the overall goal of extracting shape rather than grey-scale patches in order to reduce the influence of illumination and nearby clutter, so a combination of these approaches could be considered in future work.

2. Multi-Resolution MSER

Although detected MSERs come in many different sizes, they are all detected at a single image resolution. When a scene is blurred or viewed from increasing distances, many details in the image disappear and different region boundaries are formed. This means that the MSER detector could potentially achieve better invariance to scale change by observing the scene at several different resolutions. To see whether this is the case, we have made a simple multi-resolution extension of the MSER detector.

Instead of detecting features only in the input image, we construct a scale pyramid with one octave between scales, and detect MSERs separately at each resolution. After detection, duplicate MSERs are removed by eliminating fine scale MSERs with similar locations and sizes as MSERs detected at the next coarser scale. The location requirement that we use for elimination is that the centroid distance should be smaller than 4 pixels in the finer grid. For two areas, a_1 and a_2, we additionally require that $\text{abs}(a_1 - a_2)/\max(a_1, a_2) < 0.2$. This typically results in the removal of between 7% and 30% of all regions. On all scales except the finest, we require the minor axis of the ellipse to be larger than 5. The scale pyramid is constructed by blurring and sub-sampling with a 6-tap Gaussian kernel with $\sigma = 1.0$ pixels. To deal with border effects, we employ *normalized averaging* [5].

Our experimental results described in section 4.4 confirm that this approach gives considerably improved invariance to scale change and image blur. As a result, we use multi-resolution MSERs in all our other experiments throughout this paper.

3. Descriptor computation

The computation of the shape descriptor and the texture patch descriptor are quite similar. As we go through the algorithm details, we will introduce a number of parameters, which we will later tune in section 3.4.

3.1. Affine normalisation

To compute the patches shown in the lower part of figure 1, we first blur the input image (or for the shape descriptor, each of the binary masks in the top right) with a Gaussian kernel with scale σ_i. The mask centroid $\mathbf{m}$ and the eigenvalue decomposition of the mask covariance matrix $\mathbf{C} = \mathbf{R}\mathbf{D}\mathbf{R}^T$ (with $\det \mathbf{R} > 0$) define a rectifying

transform as:

$$\mathbf{x} = s\mathbf{A}\hat{\mathbf{x}} + \mathbf{m}\,, \text{ for } \mathbf{A} = 2\mathbf{R}\mathbf{D}^{1/2}\,. \quad (1)$$

A point at position $\hat{\mathbf{x}}$ in the patch should now be sampled in the image at position $\mathbf{x}$. The parameter s is a scaling factor that determines how much wider the patch should be compared to the covariance matrix. The sampling at $\hat{\mathbf{x}} \in [-1, 1]^2$, is performed using bilinear interpolation, followed by blurring with another Gaussian kernel with scale σ_p. The number of samples in the patch is determined by a parameter N_s. The amount of blur before sampling is automatically determined from the maximum sample density change, to give $\sigma = 0.5$ pixels after resampling. This results in the expression $\sigma_i = bs/N_s$, where b is the minor axis of the approximating ellipse.

3.2. SIFT descriptor

The computation of the SIFT descriptor is basically the same as described by Lowe [7]. In order to find reference orientations for the patch, gradients are computed and a histogram of gradient directions, with N_b bins, is formed. The orientation peaks are found as the maxima of a 3-tap quadratic polynomial fit, at local maxima of the histogram. The gradient votes in the histogram are weighted with the gradient magnitude, and a spatial Gaussian weight of σ_h. A reference orientation is formed for each local maximum that is above 80% of the highest peak value. This gives on average 1.6 peaks per patch. After this the gradients are resampled in an orientation normalised frame, *i.e.* $\hat{\mathbf{x}} = \mathbf{R}\mathbf{x}_r$. Then the patch is divided into 4×4 squares, and gradient direction histograms are computed for each of them, with linear interpolation between the spatial locations. Just like in the orientation histogram, the votes are weighted with the gradient magnitude, and a Gaussian with scale σ_d. The number of orientation directions in the histogram is set to 8, giving $4 \times 4 \times 8 = 128$ values for the descriptor. Finally, the descriptor is normalised to unit length.

3.3. Dissimilarity score

We use the $\mathcal{X}^2$-metric to compare individual descriptors. This was found to give a significant improvement over least squares in a recent study [18], while still being much faster than the more sophisticated *Earth-movers distance*. The final matching score is given as the ratio between the best and the second best dissimilarities,

$$r = d_1/d_2\,. \quad (2)$$

This *ratio score* was introduced in [7]. It basically discredits matches for which there are similar alternatives, and thus a higher likelihood of the match being incorrect. In line with [7] we also make sure that the second-best match either has a different size, or originates from a different part of the image.

3.4. Parameter tuning

Mikolajczyk *et al.* [9] recently performed a large scale repeatability test of different features, and provided the dataset, with corresponding ground-truth on-line at [17]. We have made use of the two view-change sequences in this dataset ('Graffiti' and 'Wall') for tuning the parameters of the descriptors. Each sequence contains one frontal view, and five views at $20°$, $30°$, $40°$, $50°$, and $60°$ angles. This gives us a total of 10 view pairs. For these, we simply tried all combinations of all parameters and checked which tentative matches were correct. By ordering the matches according to the ratio score (2) we can compute the *inlier frequency curve*:

$$f(n) = \frac{1}{n}\sum_{k=1}^{n} \texttt{inlier}(k)\,, \quad (3)$$

where $\texttt{inlier}(k)$ is a function that outputs 1 if the k-th tentative correspondence is an inlier, and 0 otherwise. To evaluate a particular choice of parameters we used the score: $s = \sum_{k=1}^{250} f(n)$. That is, the area under the curve $f(n)$ between 1 and 250. In addition to favouring many correct matches, this score also favours parameter choices that put the correct matches first.

The optimal parameters are given in table 1. Of these parameters, N_s and N_b had no distinct peaks, so we simply picked a value in the interval where they gave high scores.

method	N_s	s	σ_p	N_b	σ_h	σ_d
shape	41	1.2	1.2	38	0.5	0.4
texture patch	41	2.5	1.0	38	0.3	0.9

Table 1. Tuned parameter values for the two methods.

3.5. Pair descriptors

A simple way to extend the complexity of the shape descriptor, and thus allow it to match features that occur many times in a set of images, is to select pairs of nearby features, and append their descriptors. We define a measure of spatial feature proximity using the vector between the centroids, $\mathbf{d} = \mathbf{m}_{\text{ref}} - \mathbf{m}_{\text{neighbour}}$. We tried generating pairs as both the K nearest Euclidean and affine normalised distances:

$$r_e^2 = \mathbf{d}^T\mathbf{d}\,, \text{ and } r_a^2 = \mathbf{d}^T\mathbf{C}_{\text{ref}}^{-1}\mathbf{d}\,. \quad (4)$$

but found that the affine normalised distance in general did better. The descriptor of Chum and Matas [2] directly relies on pairs. In their method, pairs are selected by allowing all features with r_a below a threshold, and which are of similar size to the reference frame. We made the choice of picking the K nearest neighbours instead of thresholding, since this adapts to the local feature density, and removes the need for an additional threshold on the size relation. There can

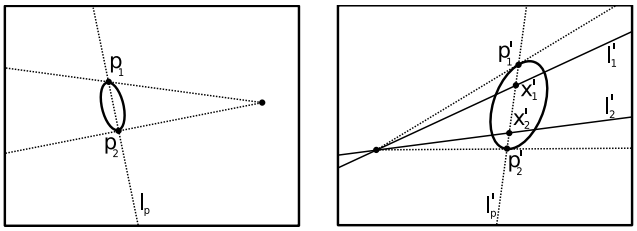

Figure 2. Epipolar geometry. Left: Construction of tangency points $\mathbf{p}_1$ and $\mathbf{p}_2$ from the polar line $\mathbf{l}_p$. Right: Induced epipolar tangents $\mathbf{l}'_1$ and $\mathbf{l}'_2$, and the tangency points in this view. The intersection points $\mathbf{x}'_1$ and $\mathbf{x}'_2$ are used to compute the overlap error.

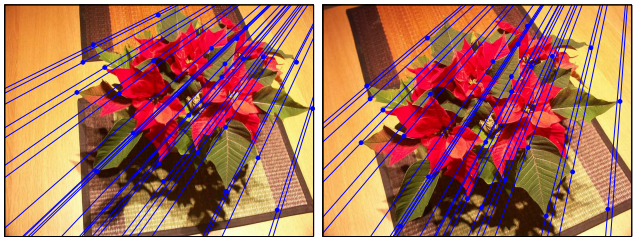

Figure 3. Example of an image pair with known geometry. Images are 800×600. Blue dots are 26 ground truth correspondences selected by hand. Blue lines are epipolar lines for points. The mean absolute epipolar line distance for points is 1.1 pixels.

be several descriptors per feature due to multiple reference orientations, *e.g.* a feature pair with 2 and 3 descriptors will generate 2×3 *pair descriptors*. Note that whenever two pair descriptors have been matched, we have obtained two tentative correspondences.

For pairs, correct individual correspondences often appear in many pairs. We could either give a correspondence a match score based on the best score for a pair descriptor it belongs to, or count the number of pairs that it belongs to, with a dissimilarity below a threshold. We have used the former approach, while Chum and Matas [2] use the latter approach. We made this choice, since our descriptors are more descriptive, and thus a low dissimilarity should be more meaningful in our case. Using the best score also allows matching of features that do not have many matching neighbours.

4. Experiments

4.1. 3D scene correspondence evaluation

Moreels and Perona [11] have previously developed a correspondence evaluation scheme for full 3D scenes. Their scheme geometrically verifies matches between corresponding points in two views of a scene using an additional *auxiliary view*. While this setup guarantees that almost no false matches are let through, it has the disadvantage that only features visible in all three views can be checked. Furthermore, their use of point correspondences implicitly discourages matching of large regions, since these are represented by the region centroid, which is only affine invariant, with an error that increases with the region size.

We instead use a correspondence checking scheme where the auxiliary view is omitted, and the approximating ellipses of the regions are compared using their *epipolar tangents* [6]. The *fundamental matrix* $\mathbf{F}$, relates corresponding points $\mathbf{x}$ and $\mathbf{x}'$ in the two views as $\mathbf{x}^T\mathbf{F}\mathbf{x}' = 0$ [6]. First we extract the *epipoles* of the fundamental matrix $\mathbf{F}$, as the left and right singular vectors. Using these, and the *pole-polar relationship*[6], we then compute the two points $\mathbf{p}_1$, $\mathbf{p}_2$ at which there are epipolar lines in tangency to the ellipse. See figure 2, left. The epipolar lines of these tangency points, $\mathbf{l}'_1 = \mathbf{F}^T\mathbf{p}_1$, $\mathbf{l}'_2 = \mathbf{F}^T\mathbf{p}_2$ are called the *epipolar tangents*. For a perfect correspondence, the tangency points $\mathbf{p}'_1$, $\mathbf{p}'_2$ in the other view, should lie on these epipolar tangents. See figure 2, right. To decide whether to accept a correspondence, we define an *overlap error* along the polar line $\mathbf{l}'_p$, in analogy with the area overlap error defined in [10]. The overlap error is computed from the positions of the tangency points $\mathbf{p}'_1$ and $\mathbf{p}'_2$, and the points $\mathbf{x}'_1$ and $\mathbf{x}'_2$, where the epipolar tangents intersect the polar line. Using their coordinates along the polar line, we can define the overlap error as:

$$\varepsilon = 1 - \frac{\max(0, \min(x_h, p_h) - \max(x_l, p_l))}{\max(x_h, p_h) - \min(x_l, p_l)} \,. \quad (5)$$

Here x_h and x_l are the higher and the lower of the intersection coordinates respectively, and analogously for p_h and p_l. We accept a correspondence whenever the average overlap error in the two views is less than 20%. In principle this risks letting through false matches as correct, but in practise this is quite rare. Since the overlap error is size normalised, we are able to allow more large region correspondences, which tell us more about the scene geometry, without allowing too many false matches between small regions.

Figure 3 shows two views of a scene with 26 hand-picked correspondences indicated as dots. From these correspondences we have computed the fundamental matrix, relating the views. The epipolar lines for the 26 points are also shown in the figure. The average error for all points is 1.1 pixels. This geometry is used as the ground truth.

For the scene in figure 3 we obtain the correspondences shown in figure 4. All these 56 correspondences have been visually verified to be correct.

4.2. 3D scene results

Figure 5 shows how the two descriptors compare on the scene in figure 3. Getting the first few matches right is what matters if algorithms such as RANSAC [6] or PROSAC [1] are to be used for geometry estimation, and thus the left

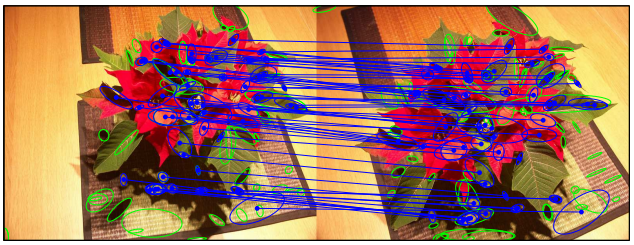

Figure 4. The 56 accepted correspondences are shown in blue. Features from rejected correspondences are shown in green. (Using MSER area threshold 100 pixels, instead of 30, for clarity of presentation.)

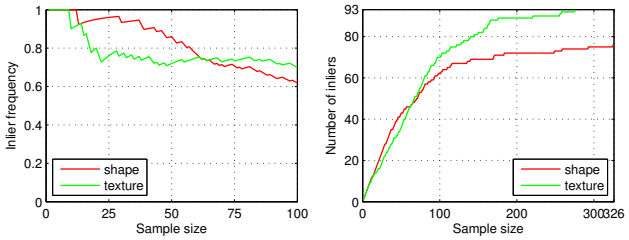

Figure 5. Performance of shape and texture descriptors on the scene in figure 3.

plot shows the inlier frequency curve (3) for the first 100 tentative correspondences. Here the shape patches are better at guessing the first 60 correspondences.

For object recognition techniques such as bag-of-features [18], however, it is the quantity of correct matches that matters, and there the texture patches do better. To illustrate performance for such applications, we have also shown the number of correspondences for different sample sizes in the right plot of figure 5. Note that this graph shows number of inliers for all tentative correspondence set sizes. For this view pair the curve shows only the first 326 correspondences; those that had a ratio score (2) below 0.95.

Figure 6 shows the shapes used to find the first 48 tentative correspondences. It is interesting to compare these with the first 48 patches chosen by the texture patch method, see figure 7. For the shape descriptor, the MSER itself occupies most of the patch, while the texture patch descriptor is dominated by surrounding context. This is in fact even more the case in other published approaches, e.g. [7] mentions a patch scale of 5, and Mikolajczyk *et al*. [9] use 3, where we use 2.5, see table 1.

4.3. Natural 3D scenes

For scenes with many near occlusions, texture patch methods have problems, since they essentially assume that the local neighbourhood is planar. Figure 8 shows such a scene. As can be seen in the graphs, the shape descriptor has both a significantly higher inlier frequency, and finds more correspondences than the texture patch method. We

Figure 6. First 48 tentative correspondences for the shape descriptor. Left: view 1, Right: view 2. False matches are shown in grey. Shapes are ordered top-to-bottom, left-to-right.

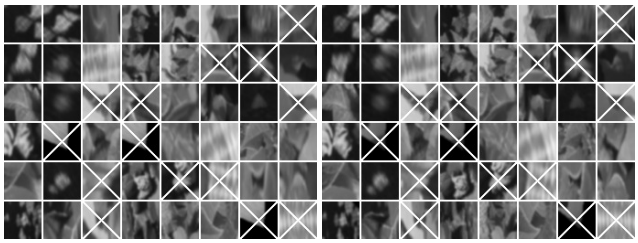

Figure 7. First 48 tentative correspondences for the texture patch method. Left: view 1, Right: view 2. False matches are crossed out. Patches are ordered top-to-bottom, left-to-right.

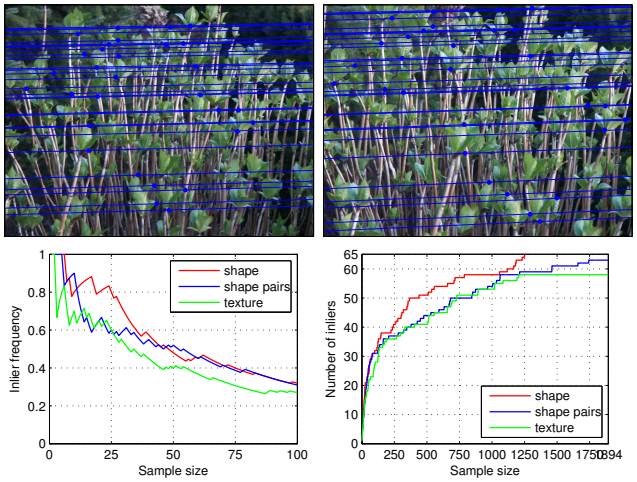

Figure 8. Scene with many near occlusions. For this kind of scene, shape does significantly better than texture.

have also shown the results for the pairs method here. For this particular scene, though, they offer no clear advantage.

Another natural scene is shown in figure 9. In this scene, the shape patches are better than the texture patches at finding the first few correspondences, but over all, the texture patches find more correspondences. In this scene, the advantage of using shape pairs is evident, since the pairs method has both a higher inlier frequency, and in the end finds more correspondences than the texture patch method.

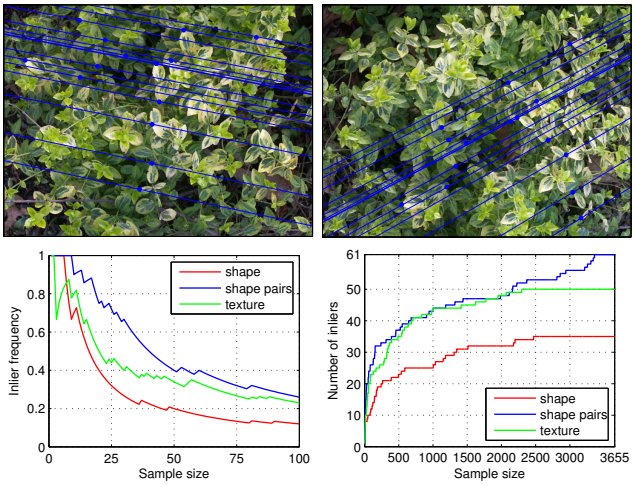

Figure 9. Scene with leaves. For this kind of scene, the use of shape pairs is advantageous.

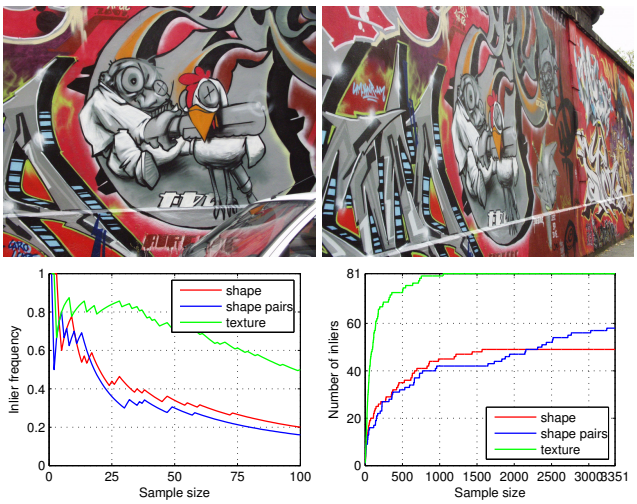

Figure 10. A planar scene with view change. In this case, the texture descriptor has better performance than the shape descriptor.

4.4. Planar and parallax-free scenes

Previously features and descriptors have been extensively evaluated on planar and parallax-free scenes [9, 10]. We have tested the two methods on the scenes available at [17], using a program, also downloaded at [17], to verify correspondences. We have used an overlap error threshold of 50% throughout.

A typical result for planar scenes is shown in figure 10. Sometimes (as in this example) the shape patch method is better at finding the first few correspondences. In general however, the texture patch method finds many more correspondences, typically a factor $1.5 - 2$ times more.

Two of the parallax-free scenes in the test-set also allow us to confirm our suspicion that MSERs benefit from being detected at multiple resolutions. Figure 11 shows a

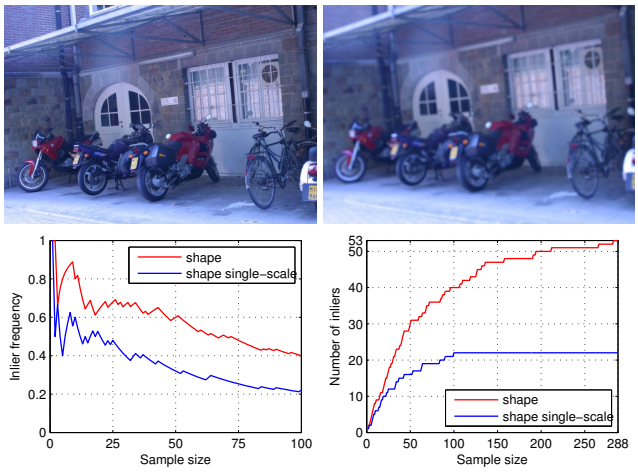

Figure 11. A scene with de-focus blur. The multi-resolution MSER provides better performance than using only the original resolution.

comparison of the single resolution and the multi-resolution MSER detectors, under de-focus blur. As can be seen, the multi-resolution extension gives a consistent improvement. Figure 12 compares single and multi-resolution MSERs on a large scale change. Again, the multi-resolution version does consistently better.

In figures 13 and 14 we have also shown the correspondences actually found using both single and multi resolution MSERs for these scenes. As can be seen, the multi-resolution algorithm creates new regions by joining non-connected nearby regions with similar colours, such as the panes in the windows of figure 13, and also regions divided by straws of grass and thin wires in figure 14. Blur often causes MSERs to shrink somewhat in size, and this problem is also reduced by the algorithm, since it picks coarse scale MSERs (which are less affected by blur) instead of fine scale versions whenever there is a choice.

4.5. 3D scenes with planar parts

For 3D scenes with planar parts, the texture patch approach has the advantage that it includes the surroundings of the MSER, and can exploit this to tell similar regions apart. For such scenes, the pair descriptors are a distinct improvement over using individual shapes, but since the two descriptors in the pair are normalised separately (*e.g.* all rectangular windows will be normalised to squares), the texture patches still do significantly better. See figure 15 for an example of such a scene. When the illumination conditions are significantly different, the shape, and shape pair descriptors again do better, see figure 16.

4.6. Combining the methods

The texture and shape patch approaches are in general complementary. The shape patches handle large illumi-

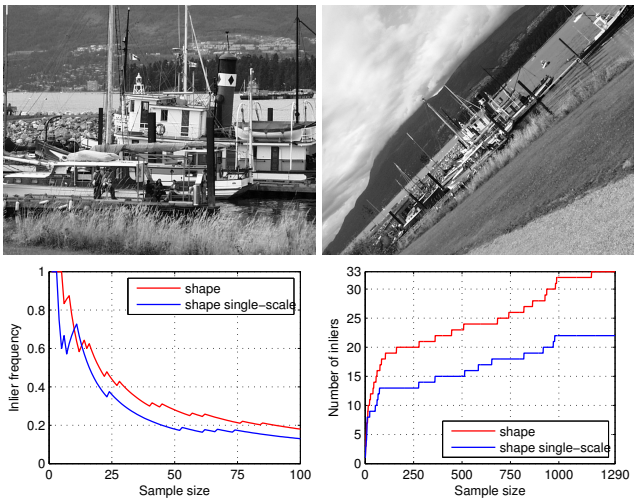

Figure 12. A scene with scale change. Again, the multi-resolution MSER gives better performance.

Figure 13. Correspondences found on de-focus blur scene. Top image shows the 22 correspondences found using single resolution MSERs. Bottom image shows the 53 correspondences found using multi-resolution MSERs.

nation changes and near occlusions better, while texture patches work better for small regions, and locally planar scenes. Thus, it would make sense to combine their results, to obtain a more robust matching system. A simple way to combine the results is to merge-sort the correspondence lists, according to the ratio score (2), and then remove duplicates from the list. Figure 17 shows the result of such a merger. As can be seen, the combination is consistently as good as, or better than, the best individual method (here the shape pairs).

5. Conclusions

We have introduced novel shape descriptors for matching MSER regions that often provide better robustness to illumination change and nearby occlusions than existing methods. When used in a complete vision system, these descriptors can be added to existing ones to extend the system performance under difficult matching conditions.

Our shape descriptor inherits the affine invariance properties of the MSER region detector. It achieves a high degree of invariance to illumination and nearby clutter by binarising the region shape. The SIFT descriptor is used to describe the shape boundary in order to minimise sensitivity to small shape deformations.

The original MSER detector is not fully scale invariant, as it is applied to just a single initial image resolution. We have shown how to improve its scale invariance at low additional cost by computing the MSER regions over a scale pyramid and removing duplicate detections. Our results

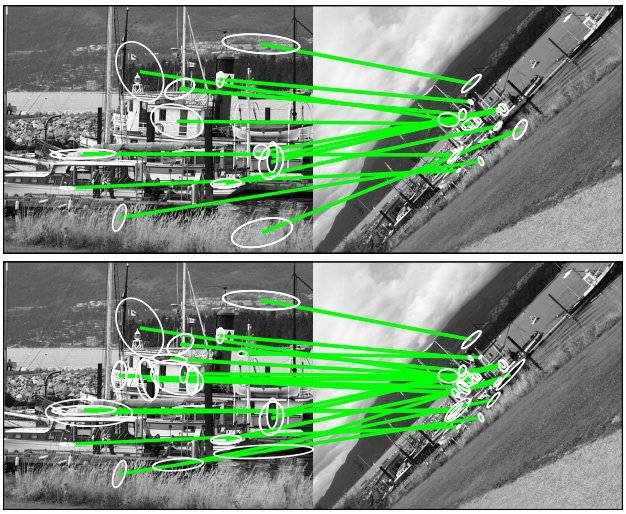

Figure 14. Correspondences found on scale change scene. Top image shows the 22 correspondences found using single resolution MSERs. Bottom image shows the 33 correspondences found using multi-resolution MSERs.

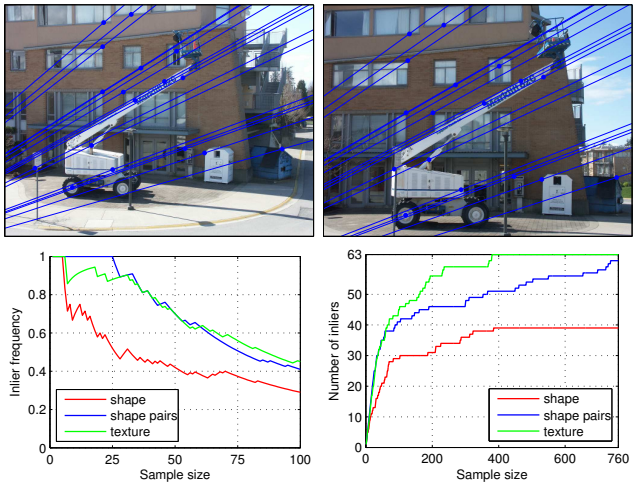

Figure 15. Crane scene, with and without pairs.

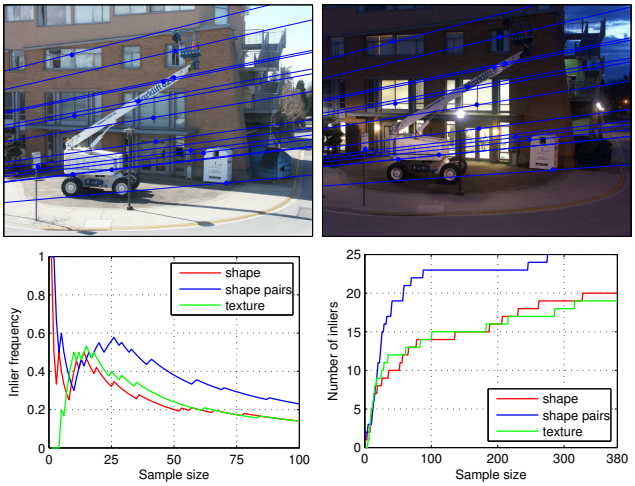

Figure 16. Crane scene, with and without pairs. Day and night.

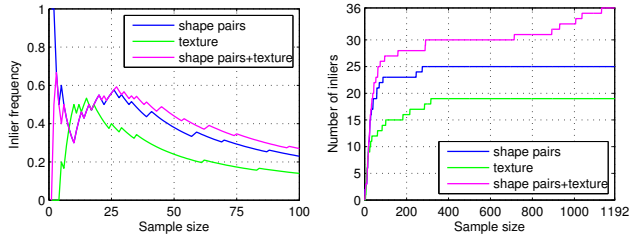

Figure 17. Combination of shape pair and texture patch features for the day and night pair in figure 16.

show that this improves matching performance over large scale changes and for blurred images, which would be useful for any application of MSERs.

Our experiments have shown the value of shape descriptors for matching specific objects, but shape has also been widely recognised as being particularly important for characterising object classes. Therefore, an important area for future work will be to test our features in the context of generic object class recognition. We also intend to test the performance of our features for matching objects to large databases, in which case our pair matching method may be expected to further increase in importance. Another area we intend to examine in future research is to exploit the *Maximally Stable Colour Region* extension of MSER [4] for improving the identification of stable regions.

Acknowledgements

This work was supported by the Swedish Research Council through a grant for the project *Active Exploration of Surroundings and Effectors for Vision Based Robots*.

References

[1] O. Chum and J. Matas. Matching with PROSAC - progressive sample consensus. In *Computer Vision and Pattern Recognition (CVPR)*, pages 220–226, June 2005. 4

[2] O. Chum and J. Matas. Geometric hashing with local affine frames. In *Computer Vision and Pattern Recognition (CVPR)*, pages 879–884, June 2006. 1, 2, 3, 4

[3] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. In *INRIA Technical Report*, Grenoble, September 2006. 2

[4] P.-E. Forssén. Maximally stable colour regions for recognition and matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007. IEEE Computer Society. 1, 8

[5] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. 2

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 4

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 3, 5

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *13th BMVC*, pages 384–393, September 2002. 1

[9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005. 1, 3, 5, 6

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005. 1, 4, 6

[11] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *IJCV*, 73(3):263–284, July 2007. 1, 2, 4

[12] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR'06*, pages 2161–2168, 2006. 1, 2

[13] S. Obdrzálek. *Object Recognition Using Local Affine Frames*. PhD thesis, Czech Technical University, 2007. 1

[14] S. Obdrzálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *13th BMVC*, pages 113–122, September 2002. 1

[15] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR'06*, pages 3–10, 2006. 2

[16] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV'03*, 2005. 2

[17] Web-site. `http://www.robots.ox.ac.uk/~vgg/research/affine/`, 2005. 1, 3, 6

[18] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, June 2007. 2, 3, 5