SMD: A Locally Stable Monotonic Change Invariant Feature Descriptor

Raj Gupta[1] and Anurag Mittal[1]

Indian Institute of Technology, Madras

Abstract. Extraction and matching of discriminative feature points in images is an important problem in computer vision with applications in image classification, object recognition, mosaicing, automatic 3D reconstruction and stereo. Features are represented and matched via descriptors that must be invariant to small errors in the localization and scale of the extracted feature point, viewpoint changes, and other kinds of changes such as illumination, image compression and blur. While currently used feature descriptors are able to deal with many of such changes, they are not invariant to a generic monotonic change in the intensities, which occurs in many cases. Furthermore, their performance degrades rapidly with many image degradations such as blur and compression where the intensity transformation is non-linear. In this paper, we present a new feature descriptor that obtains invariance to a monotonic change in the intensity of the patch by looking at orders between certain pixels in the patch. An order change between pixels indicates a difference between the patches which is penalized. Summation of such penalties over carefully chosen pixel pairs that are stable to small errors in their localization and are independent of each other leads to a robust measure of change between two features. Promising results were obtained using this approach that show significant improvement over existing methods, especially in the case of illumination change, blur and JPEG compression where the intensity of the points changes from one image to the next.

1 Introduction

Extraction and matching of distinctive feature points in images has been a major focus of research in the Computer Vision community for quite some time. Such an approach has been used in many applications such as mosaicing, classification, object recognition, automatic 3D reconstruction and stereo matching among others. The basic idea is to determine certain feature points in images that have certain properties that allow them to be distinguished from other points, either in the same image or in other images. Then, certain properties of the region around the point are used in order to transform this region to a normalized region that should remain the same under some (affine) transformation of the original patch. Finally, certain features are extracted from these normalized regions which form the feature “descriptor”. These feature descriptors are matched between two feature points to determine the similarity between them.

Many methods have been proposed in the literature for feature point extraction and subsequent affine normalization. Popular methods include the Harris corner detector and its affine normalization [1, 2], the Hessian-affine detector [1], ’Maximally Stable Extremal Regions’(MSER) [3], edge and intensity-extrema based detectors [4, 5] and a ’salient regions’ detector [6]. All of these feature point extractors have different strengths and weaknesses and yield different number of points depending on the image. An evaluation of the performance of these detectors was presented in [7]. MSER was consistently shown to outperform others in repeatability and matching scores using SIFT but produced lesser number of features, while hessian-affine and harris-affine produced more features than other methods while giving good repeatability and matching scores.

Once the features have been detected and normalized, a descriptor is determined for matching. Popular descriptors include SIFT (Scale-Invariant Feature Transform) [8], Shape Context [9], GLOH (Gradient Location Orientation Histogram) [10], SURF (Speeded up Robust Features) [11], PCA-SIFT[12], differential invariants and spin images among others. A performance evaluation of these descritors was presented in [10], where it was shown that the SIFT-based descriptors such as SIFT and GLOH perform the best while Shape Context, which is also a based on histogram of gradients/edges, comes quite close. More recently, Moreels and Perona [13] have reported results for matching in 3D objects where they show the best performance for hessian-affine detector combined with SIFT for viewpoint changes and harris-affine with SIFT and hessian-affine with Shape Context for lighting change and camera focal length change respectively.

The remarkable outperformance of SIFT can possibly be attributed to the fact that since it uses a statistical measure (*histogram*) of the gradients, it is relatively robust to small errors in feature localization and normalization, and small changes in the shape of the feature due to viewpoint or other changes. Furthermore, it normalizes the gradients which yields a method that is invariant to a *linear* change in intensities. However, while this descriptor has these interesting properties, it is not invariant to a non-linear change in intensities which often occurs in practice. This can happen, for instance, due to gamma correction, a non-linear camera response function especially near saturation and low light [1, 21, 6], small specular reflections, different illumination in different parts of an object, and image effects such as blur and image compression.

To deal with such effects, several papers have proposed the use of orders between pixels rather than the intensities themselves [14, 15]. These methods transform the intensity space to an “order” space that captures the order of a pixel with respect to its neighbors and develops a binary pattern from such orders. Statistical matching of histograms of these binary patterns has shown extremely good performance in some applications such as texture classification[15] and face recognition[16]. Although this may be a good monotonic illumination-invariant scheme for textures and faces, the relatively large space of the binary patterns makes it unsuitable for feature point description where the patch size is limited. Also, the intensity information is totally lost in the process and this can make the descriptor susceptible to Gaussian noise. Also, this gives equal weightage to high gradient and low gradient regions, which can be undesirable (SIFT gives weightage proportional to the gradient value).

Mittal and Ramesh[17] proposed an approach that utilizes a combination of intensity and order in order to develop a change measure that is more robust to Gaussian noise and weighs higher gradients more compared to lower gradients while still maintaining invariance to monotonic changes. However, while such a matching technique can be used in some applications such as stereo matching, such an approach cannot handle errors in point localization, scale and shape changes etc. very well that is needed in a feature matching application. In order to localize and normalize the points more accurately so that such an approach can be used, Gupta and Mittal[18] develop a feature point detector that detects feature points at the intersection of two lines. However, although they report quite high performance numbers, the number of features points detected by such an approach is rather low and this approach is not suitable for many applications where such linear structures are not present. Another work that is somewhat related to ours is that of [19] who use the idea of comparing pixel values for some random points around a keypoint in order to drop this keypoint down a randomized tree for recognition. They warp the keypoints in a given image in order to obtain numerous possible patch realizations under viewpoint changes (for wide-baseline point matching) and the problem is posed as a classification problem of a given keypoint in the second image to belong to one of the keypoints in this (first) image.

In this paper, we present a new feature descriptor that obtains invariance to a monotonic change in the intensities while at the same time works with any of the feature detectors used in the literature. We look at orders between certain pixels in the patch and the feature descriptor consists of point pairs. A penalty is awarded if there is an order change for a point pair between the two patches and such penalties for different pairs are summed in order to determine the “difference” between the two features. The point pairs have the property that the points in the pair are relatively stable in their intensity order with respect to both intensity noise and localization error. In order to obtain invariance to Gaussian noise, the points in a pair are chosen such that they have a certain minimum difference between their intensities. On the other hand, robustness to changes in the scale and localization of the feature point is obtained by picking point pairs such that moving the points a certain distance in their neighborhood does not change the order of the intensities of the pair. Furthermore, we allow a point to repeat only a certain number of times in the pairs in order to improve the independence between the different point pairs. Two features are matched by comparing the orders of the pixel pairs. The method was found to be extremely robust and on a standard dataset, it yielded results that are significantly superior to currently used methods. This makes the method highly suitable for many applications.

2 Basic Goals of Our Feature Matching Approach

We have several goals for feature comparison. First, the approach must be invariant to a monotonic change in the intensities. Second, it must be robust to noise in the pixel intensities as well as feature point localization and distortion. Third, the method must be reasonably efficient. Towards these goals, we extract certain point pairs for which the order will be tested across the feature points. Such points must have the following properties:

1. They must have a minimum intensity difference between them. This is needed so that the order between these pixels does not change with some amount of noise in the pixel intensities.

2. The order between these pixels should not change if there is some error in the localization of these points.
3. The different point pairs must not repeat the same points too many times so that the tests for the different point pairs are more or less independent of each other.

Computation of optimal points according to all of the above criteria appears difficult. However, we show that it is possible to do so quite efficiently using the concept of extremal regions and distance transforms. This is discussed next.

3 The Feature Descriptor

3.1 Computation of Extremal Regions

The first step in our algorithm is the computation of extremal regions. Extremal regions are regions that have intensities above or below a given threshold. Given that the points in the point pairs must have a given difference of intensity δ_I between them, we compute extremal regions with two thresholds T_1 and T_2 such that $T_1 - T_2 = \delta_I$:

$$\mathcal{R}^+ = Thresh^+(I, T_1)$$
$$\mathcal{R}^- = Thresh^-(I, T_2)$$

where $Thresh^+(I, T)$ is the set of all points in the Image I that are above a given threshold T and $Thresh^-(I, T)$ is the set of all points I below T.

Such regions are computed over a range of values of T_1 (T_2 is determined automatically as $T_1 - \delta_I$). As pointed out by Matas et.al. [3], the set of all extremal regions can be computed in time $O(n \log \log n)$ where n is the number of pixels in the image, using methods based on the union-find algorithm[20].

As should be obvious, all points in $\mathcal{R}^+$ are greater than all points in $\mathcal{R}^-$ by atleast an intensity difference of δ_I. The next step is to find points in $\mathcal{R}^+$ and $\mathcal{R}^-$ that are as far as possible from the boundaries. This will ensure that the points are stable with respect to localization errors.

3.2 Computation of Point Pairs

Given a pair of extremal regions $\mathcal{R}^+$ and $\mathcal{R}^-$, we wish to compute points that are as far as possible from the boundaries of these regions. This can be done quite efficiently using the distance transform. We use the one based on the Euclidean distance measure[21]. Points that have high distances from the boundaries are selected as possible candidates and points from $\mathcal{R}^+$ are matched with points from $\mathcal{R}^-$. Once a pair is selected for a given region pair, we do not want to select points close to these points. To obtain this behavior, we mark the selected points themselves as boundary points so that the points that are selected next are those that have the largest distance not only from the original boundaries but from the already selected points as well. This procedure is repeated till we cannot obtain points that have a certain minimum "distance" from the boundaries.

The above mentioned procedure yields a set of points for a given extremal region pair ($\mathcal{R}^+$,$\mathcal{R}^-$) obtained at a certain threshold T_1. This procedure is repeated for different

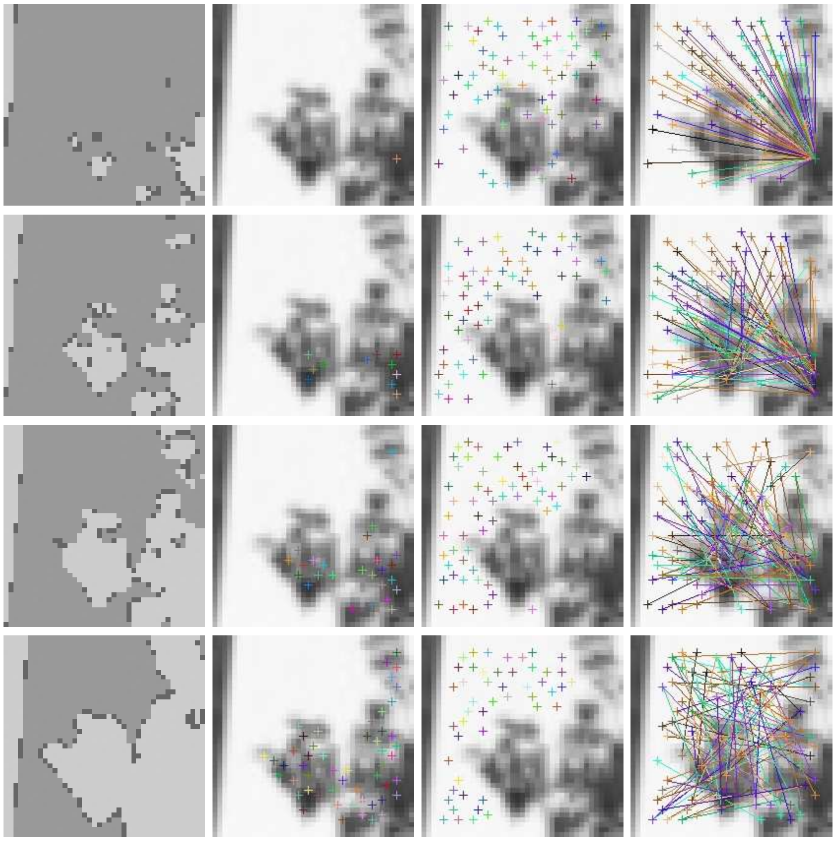


Fig. 1. An Example of pair extraction at different levels. First column shows the extracted extremal regions where the brighter gray is the “Min” extremal region, darker gray is the “Max’ extremal region and black are the boundary points. The second and third columns show the extraction of some points in the min and max regions respectively, super-imposed on the original patch. Finally, the last column shows the pairs formed at each level, again super-imposed on the original patch. These are combined by throwing away pairs having common points in order to obtain the final pairs shown in Fig. 2.

values of T_1 (For the experiments in this paper, we have used 25 such values ranging from 10 to 240 at a gap of 10 each.). Now, given the set of point pairs obtained from all of the above mentioned steps, we select the most stable ones based on the “distance” value associated with the points as per the distance transform. The minimum of the distance values of the two points in a pair is taken as the *stability factor* for that pair. Now, using a greedy approach, we simply select the most stable point-pairs one by one while taking care that any one point in the patch does not have too many close-by points in the already existing point-pairs (For the experiments in this paper, we allow a point to be taken a maximum of 3 times for building the point pairs). This is done

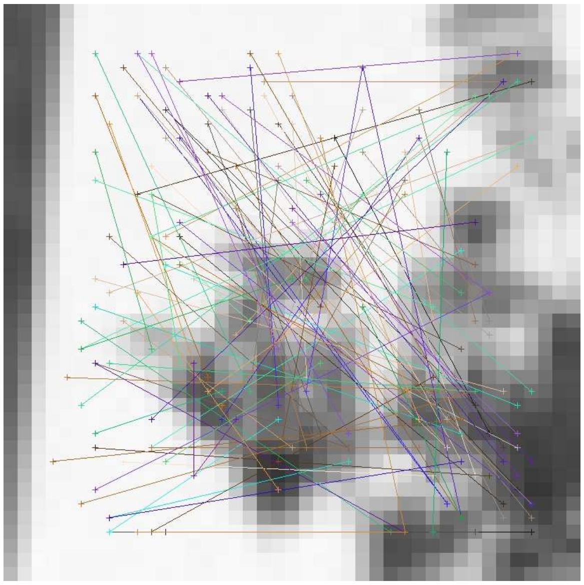


Fig. 2. An Example of pairs extracted in a patch.

to ensure some independence between the different point pairs and to allow the pairs to spread out in the patch so as to obtain discriminability. The above procedure can be done quite efficiently by again using a distance transform image taking the selected points as the model points. The output of this procedure is a set of point pairs along with their stability factors. Features which do not produce a certain minimum number of stable pairs are thrown out as being unreliable for matching. This set of extracted point pairs, thus, forms our feature descriptor.

Fig. 1 shows the above process on an image patch, where the different rows show the computation of the extremal regions and point pairs at different thresholds. These are combined in the end in order to obtain the final point pairs shown in Fig. 2.

It is not hard to see that our algorithm yields a reasonably good set of pairs that is also optimal in a certain sense. Suppose there exists a point pair such that a spatial perturbation of the points in the pair *and* some point-wise noise in the intensity values of the pixels does not lead to an order change. Suppose the maximum value of change in the intensity of each pixel is $\varDelta I$ and the spatial perturbation possible is δx. Then, if our thresholding levels were continuous and δ_I approximately two times $\varDelta I$, then this point pair will be observed by our method with a stability factor of atleast $\varDelta x$ when the thresholds are set such that the lower threshold value is slightly higher than the lower intensity point and the higher threshold value is just below the higher intensity point. Our algorithm simply tries to pick point pairs from such set that are the most stable and more or less independent, while capturing as much of the patch structure as possible.

4 Matching

Given the feature descriptor, we now describe how one may match any two features. For each of the features, we have a set of points pairs along with their stability factors $\{(p_i^1, p_i^2, s_i), i = 1 \ldots n\}$. For these point pairs, we test if the order of the pixels has changed in the other patch. Then, we simply calculate a weighted sum of the order flips, giving each point pair a weight that depends on its stability factor. Since higher stability points are very important for stable matching and should be given a higher weightage (recall that the stability factor is the minimum distance that any of the two points has to move in order to possibly have an order flip), we propose using the square of the stability factor (s^2) as the weight for a pair that has stability factor s. The pairs from both the feature points are combined in order to obtain the final weighted matching score:

$$M = \frac{\Sigma_{i=1}^{n} s_i^2 \operatorname{sgn}(I_o(p_i^1) - I_o(p_i^2))}{\Sigma_{i=1}^{n} s_i^2} \tag{1}$$

where $I_o(p)$ is the intensity of point p in the patch “other” than the one in which the point pair was computed (i.e. if the pair was computed in the first patch, then I_o is the intensity in the second patch and if the pair was computed in the second patch, then I_o is the intensity in the first patch). sgn is the $sign$ function:

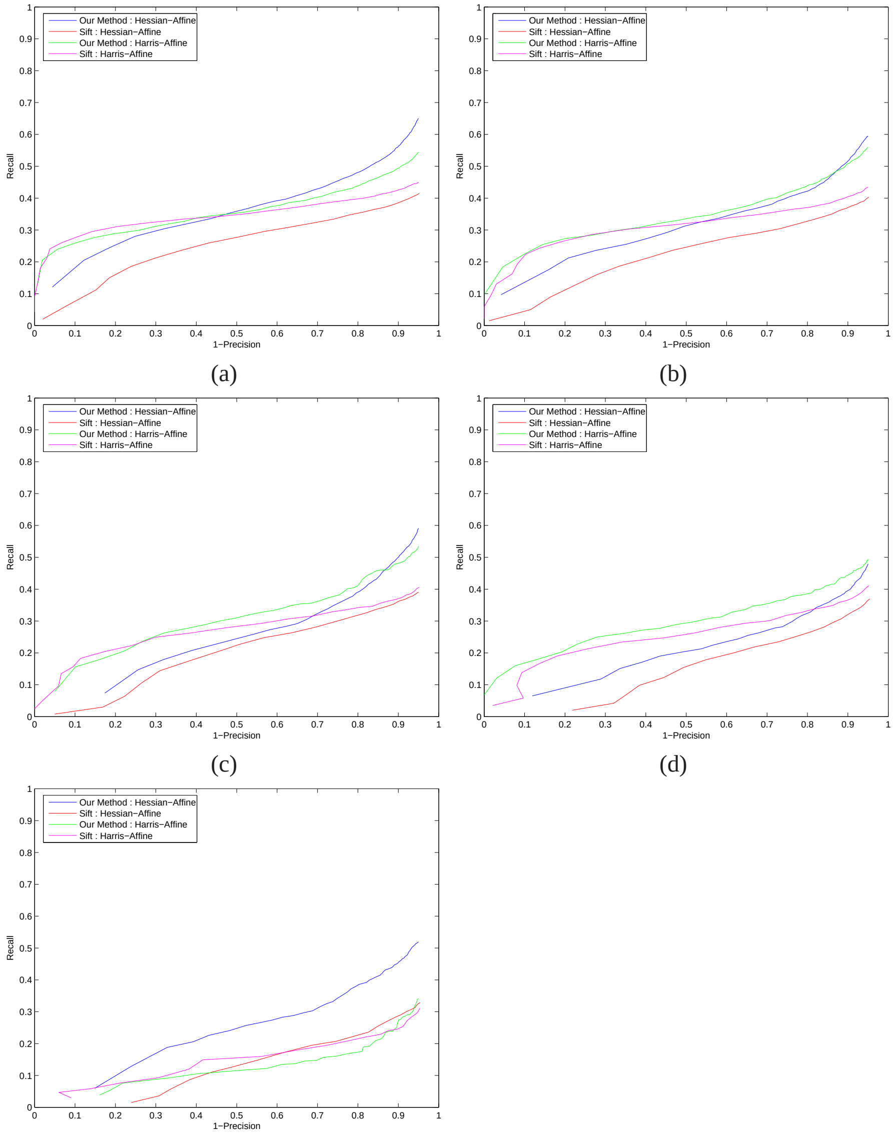
$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{2}$$

We assume in Eq. 1 that the pair (p_i^1, p_i^2, s_i) is stored such that the first point has higher intensity than the second in the original patch in which this pair was computed. We also note here that as opposed to many other methods where only the feature descriptors are matched directly, we also use the underlying image patches for comparison.

5 Results and Experiments

We tested our algorithm on the standard dataset from the evaluation papers of Mikolajczyk and Schmid [10] and Mikolajczyk, Tuytelaars et. al. [7]. In [10], different feature descriptors are evaluated for changes in scale, rotation, viewpoint (affine transformation), blur, jpeg compression and illumination. GLOH and SIFT performed better than others and almost similar to each other, while Shape Context came quite close. Most of the experiments were done using the Harris-affine and Hessian-affine feature detector since these detectors give the most number of points. It was also stated that the relative performance of different descriptors remains almost the same for different feature detectors.

We follow the approach of this paper in order to evaluate our descriptor by plotting the recall vs. 1 - precision. This is similar to an *ROC* curve. Recall is the number of the correct matches found divided by the total number of correct matches present while precision is the number of correct matches found divided by the total number of matches found by the descriptor. The curve is obtained by varying the threshold for each



(a)

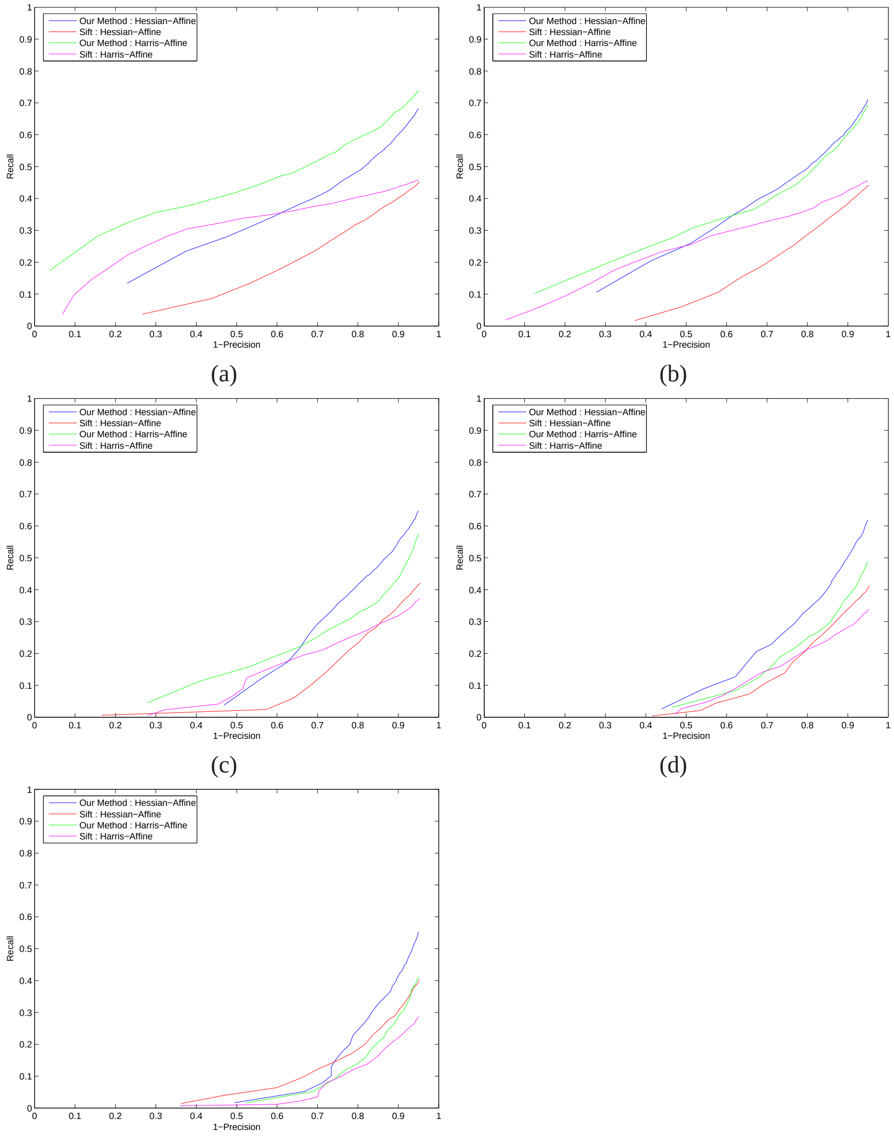
(b)

(c)

(d)

(e)

Fig. 3. Comparison of our matching approach with SIFT for different amounts of illumination change (leuven) using the Harris-Affine and Hessian-Affine detectors. The amount of lighting change increases from (a) to (e) with (e) having lighting change of approximately 60%. The number of correct correspondences existing in the images were: 3902 and 2205 for Hessian-Affine and Harris-Affine respectively for (a), 2952 and 1511 for (b), 2453 and 994 for (c), 1777 and 668 for (d) and 1197 and 299 for (e). The images in this standard dataset may be obtained from http://www.robots.ox.ac.uk/ vgg/research/affine.



(a)

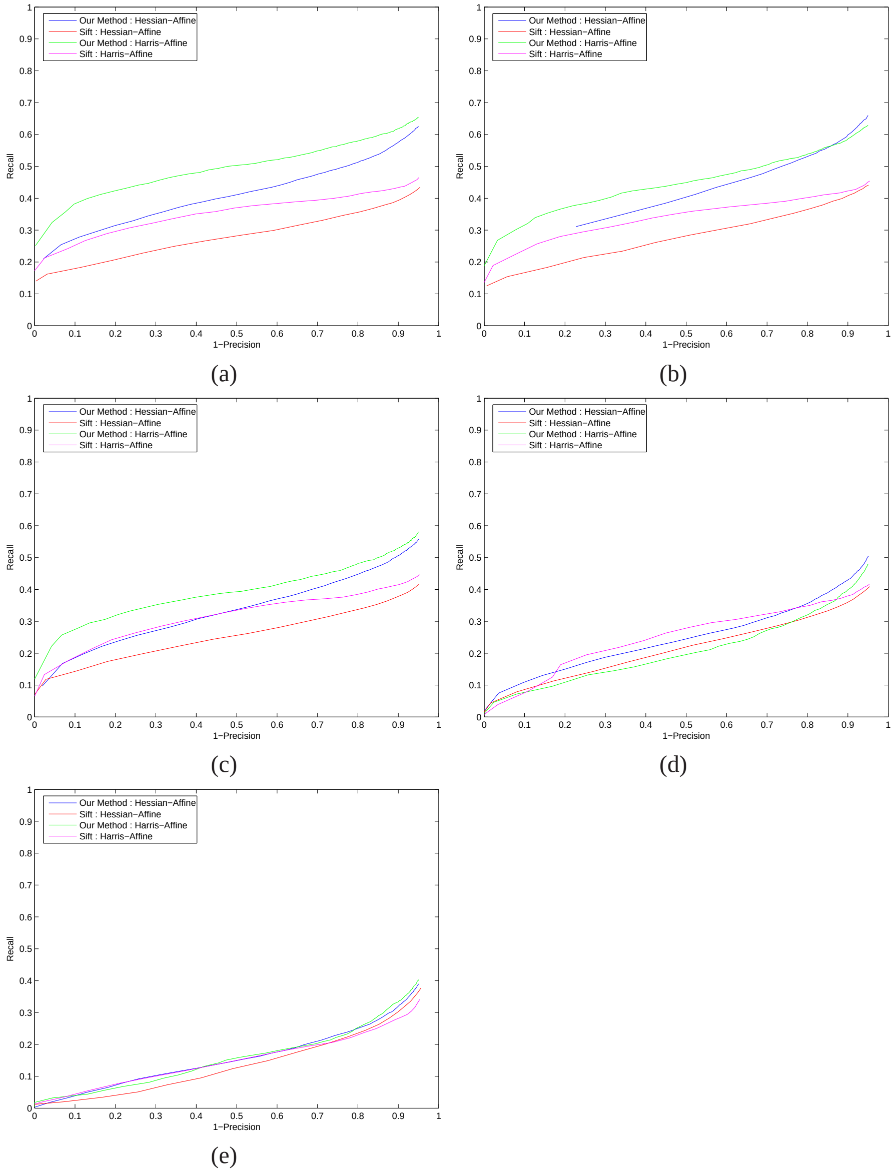
(b)

(c)

(d)

(e)

Fig. 4. Comparison of the performance of our method with SIFT for different amounts of image blur (Bikes) using the Harris-Affine and Hessian-Affine detectors. The amount of blurring increases from a values of 2 to 6 in plots (a) to (e). The number of correct correspondences existing in the images were: 4150 and 2372 for Hessian-Affine and Harris-Affine respectively for (a), 4127 and 2253 for (b), 3147 and 1827 for (c), 2548 and 1478 for (d) and 1640 and 888 for (e).



(a)

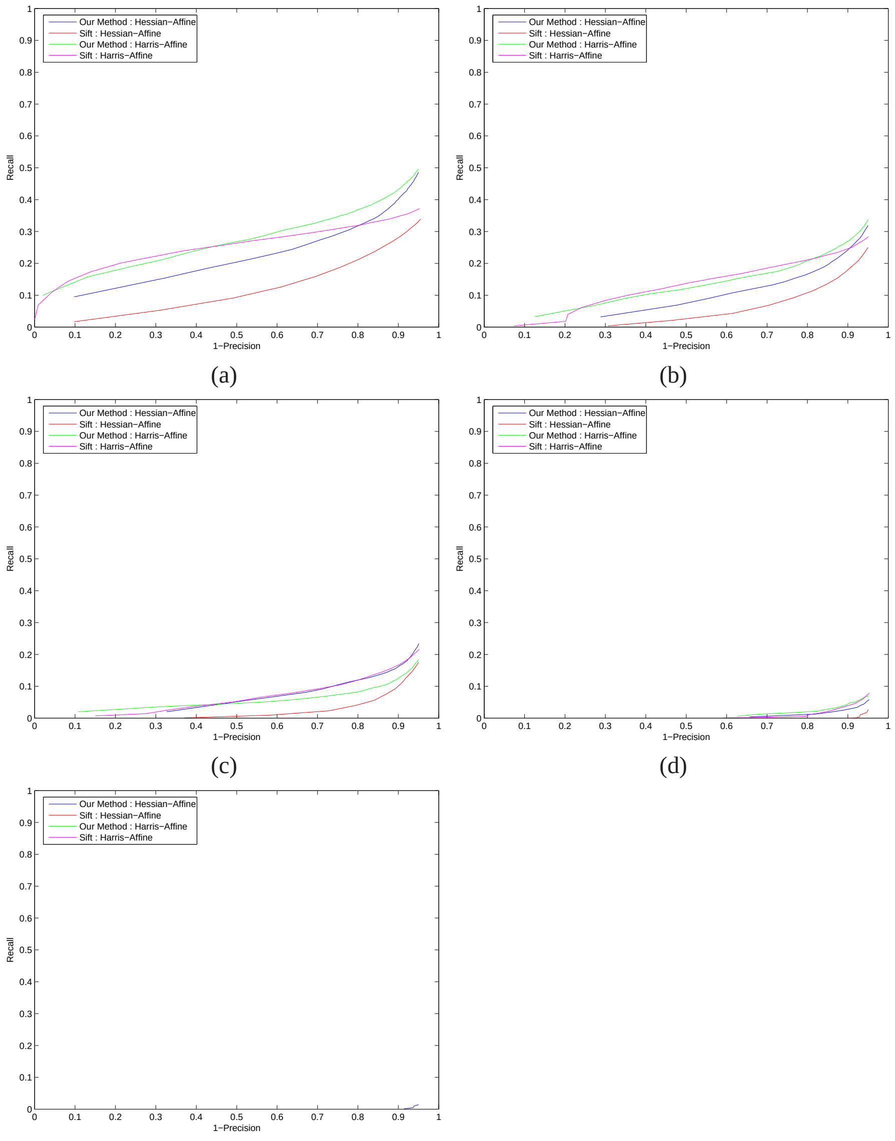
(b)

(c)

(d)

(e)

Fig. 5. Comparison of the performance of our method with SIFT for different amounts of JPEG compression (ubc) using the Harris-Affine and Hessian-Affine detectors. The compression ratios for the different plots are (a) 60 %, (b) 80 %, (c) 90 %, (d) 95, % and (e) 98 %. The number of correct correspondences existing in the images were: 10876 and 4053 for Hessian-Affine and Harris-Affine respectively for (a), 10751 and 3939 for (b), 10492 and 3548 for (c), 9243 and 2699 for (d) and 8043 and 2565 for (e).



(a) (b)

(c) (d)

(e)

Fig. 6. Comparison of the performance of our method with SIFT for different amounts of affine/viewpoint changes (graffiti) using the Harris-Affine and Hessian-Affine detectors. The viewpoint change angle for the different plots are approximately (a) 20^o, (b) 30^o, (c) 40^o , (d) 50^o, and (e) 60^o. The number of correct correspondences existing in the images were: 504 and 228 for Hessian-Affine and Harris-Affine respectively for (a), 497 and 214 for (b), 398 and 191 for (c), 219 and 107 for (d) and 135 and 73 for (e).

descriptor. As in [10], we show results using only the Harris-affine and Hessian-affine feature detectors. Since we throw out some features as being unreliable for matching, the set of feature points that we use is a subset of the total feature points extracted. The reduction in feature points is approximately 50% for all the images combined, although it varies for different images. All comparisons are done using this reduced set of feature points[1]. Also, since the performance of different histogram-based approaches has been shown to be quite similar to each other by several authors, we only compare our method with SIFT in this paper.

Fig.s 3-6 show the results of our algorithm on the different images of the dataset provided and used by [10] and [7]. In Fig. 3, we show comparative results as a function of illumination changes, while Figs. 4 and 5 show the results as a function of image blur and compression. Finally, in Fig. 6, we show the performance as a function of affine transformation. In all of these plots, we have shown the results using both Harris-affine and Hessian-affine.

As can be seen from the plots, we improve substantially compared to SIFT on images which had a substantial transformation in the intensity values of the pixels. This happens not only under illumination changes but also when there is an image blur or compression. Since SIFT can only do a linear correction in the intensities, our outperformance on such images is not surprising. Even though blur and image compression do not exactly follow a monotonic change transformation model, they do not deviate far from such a model. Furthermore, since the points pixed for testing by our method are *stable* and hence lie away from points whose blurring or compression effects could cause an order flip, such image degradations do not affect us substantially. Our performance under viewpoint changes was slightly better than SIFT for Hessian-affine and almost the same as SIFT for Harris-affine points and we do not seem to have any particular advantage in this case.

It was also observed that the performance gap between our method and SIFT was larger for Hessian-affine points as compared to Harris-affine points. This could be due to the higher localization accuracy of Hessian-affine as noted by several previous authors. This helps our method more compared to SIFT as histogram-based approaches can handle more localization error as compared to our approach. On the other hand, if the feature point is localized relatively well, then we are able to get more discriminative structural information than just a statistical measure of the gradients in different regions. This, along with our invariance to a monotonic change in intensities, probably explains the superior performance of our approach compared to SIFT when the patches are better localized. A drawback of our approach, on the other hand, is that we are not able to obtain many stable points on patches that have very high frequency gradient changes. On such kind of patches, histogram-based methods would probably give better results.

6 Conclusion

We have presented an approach for feature description and matching that is invariant to a monotonic change in intensities while being robust to Gaussian noise and errors

[1] The degradation in performance of the method without using this feature reduction is around 10% for our method and around 5% for SIFT

in feature localization and normalization. The method can be implemented quite efficiently since it depends on the fast algorithms that are available for extremal region extraction and distance transform computation. We obtained a significant improvement in performance compared to existing techniques such as SIFT on a standard dataset, especially in cases where there is a substantial transformation in the pixel intensities. Thus, the method holds great promise for many applications of feature matching such as image classification, object recognition, mosaicing and 3D reconstruction.

References

1. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: ECCV. (2002) I: 128 ff.
2. Schaffalitzky, F., Zisserman, A.: Viewpoint invariant texture matching and wide baseline stereo. In: ICCV. (2001) II: 636–643
3. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. Image and Vision Computing **22** (2004) 761–767
4. Tuytelaars, T., Gool, L.J.V.: Content-based image retrieval based on local affinely invariant regions. In: VISUAL. (1999) 493–500
5. Tuytelaars, T., Van Gool, L.: Wide baseline stereo matching based on local, affinely invariant regions. In: BMVC. (2000)
6. Kadir, T., Zisserman, A., Brady, M.: An affine invariant salient region detector. In: ECCV. (2004) Vol I: 228–241
7. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV **65** (2005) 43–72
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
9. Mori, G., Belongie, S., Malik, J.: Efficient shape matching using shape contexts. PAMI **27** (2005) 1832–1837
10. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI **27** (2005) 1615–1630
11. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: ECCV. (2006) I: 404–417
12. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: CVPR. (2004) II: 506–513
13. Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. IJCV **73** (2007) 263–284
14. Zabih, R., Woodfill, J.: Non-parametric local transforms fo computing visual correspondence. In: ECCV. (1994) 151–158
15. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI **24** (2002) 971–987
16. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. PAMI **28** (2006) 2037–2041
17. Mittal, A., Ramesh, V.: An intensity-augmented ordinal measure for visual correspondence. In: CVPR. (2006) I: 849–856
18. Gupta, R., Mittal, A.: Illumination and affine- invariant point matching using an ordinal approach. In: ICCV. (2007) 1–8
19. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. PAMI **28** (2006) 1465–1479
20. T. H. Cormen, C. E. Leiserson, R.L.R., C.Stein: Introduction to Algorithms. MIT press (2005)

21. Ricardo Fabbri, Luciano da Fontoura Costa, J.C.T., Bruno, O.M.: 2d euclidean distance transform algorithms: A comparative survey. ACM Computer Survey **40** (2008)