FEUDOR: Feature Extraction Using Distinctive Octagonal Regions

Ario Emaminejad and Mike Brookes
Department of Electrical and Electronic Engineering
Imperial College London
{ae102|dmb}@ic.ac.uk

Abstract

This paper introduces a novel feature extraction algorithm called FEUDOR. The features extracted by this method are octagonal homogeneous regions that have different mean square difference compared to their surrounding area. By using integral images we have implemented this algorithm efficiently. We have shown that the repeatability score of FEUDOR under various image transformations is comparable and in some cases better than other existing algorithms.

1 Introduction

Finding correspondences between different images of the same scene is a requirement in many application areas including object recognition, image retrieval, 3D scene reconstruction and object tracking. In recent years research has focused on extracting *features* from each image first, and then by matching these features, correspondences between images is found. By a feature we mean either a point (*Interest Point*) or a patch of pixels (*Region of Interest*) which can be reliably extracted and matched between different images. A good feature is one that is repeatable and distinctive. Repeatable means that the same feature can be extracted from different images of the same scene. Distinctive features are those that will only match closely to corresponding features from another image.

Our aim in this paper is to extract features that would primarily be useful in solving two challenging problems. One is matching two images of the same scene with very different viewing angle, so-called wide-baseline matching. The other problem is matching objects seen by cameras with non-overlapping fields of view.

In solving any of these problems there are several issues that need to be addressed first. For example, a wide view point angle will give rise to geometric distortions. Also, increase in the view point angle, will increase occurrences of self-occlusion. At the same time the background of objects will be very different as the view point changes. Finally the illumination that the object is viewed under can vary between images.

To extract highly repeatable features useful in wide-baseline matching, algorithms have been developed that extract features that are invariant to scale, rotation or affine transformations. However we note that making features invariant to transformations that do not actually occur in an image reduces the distinctiveness of features. In section 2, we will discuss some previously proposed feature extraction algorithms that are invariant to some or all of the transformations mentioned above. As the background of objects will often be completely different when viewed from different angles, we would like our features not to cover any part of the background. Having features based on the neighbourhood of an object's corner within an image is therefore undesirable.

Our main contribution in this paper is a novel feature extraction algorithm (FEUDOR) that extracts homogeneous regions that differ from their surrounding. In extracting such features we make use of the information available in a RGB color image. Our features are invariant to scale change but only approximately to rotation. FEUDOR can be implemented in an efficient manner which will be explained later in 3.3.

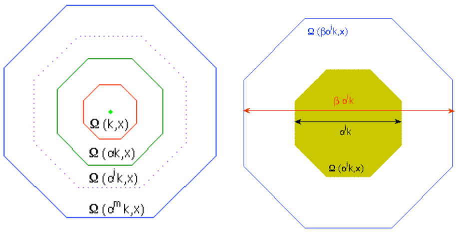
2 Related Work

Many feature detectors begin by identifying interest points in the image and then determine a feature from the neighbourhood of the point. One of the early Interest Point detectors that remains widely used is the Harris corner detector [2]. It detects corners by looking for regions whose local auto-correlation function has high curvature in both x and y direction. Many algorithms have subsequently been developed that have either used Harris corners as a first step in extracting features or have modified the detection method described in [2] .

The Harris measure used for extracting corners is not scale invariant. Mikolajczyk and Schmid [6] introduced a scale-adapted Harris measure that uses the determinant of the Hessian matrix of the image to identify Interest Points. For each Interest Point it then assigns a scale at which the Laplacian is maximised. In [4], Lowe approximated the Laplacian with the Difference of Gaussian function resulting in improved speed. Recently, Bay et al. [1] introduced the *Fast Hessian Detector*; in their method, they rely on the determinant of the Hessian matrix to select both the location and the scale for the Interest Points. They also use integral images to compute box filters, which are used to approximate the Gaussian second order derivative used in calculating the Hessian matrix. By this mean they have substantially improved the speed of extracting features without degrading the performance.

By successively thresholding an image, Maximally Stable Extremal Region (MSER) [5] extracts regions where the rate of change of area with threshold value is zero. The algorithm proposed by Tuytelaars and Van Gool [8], first finds interest points using Harris corner detector, but then moves along the edges adjacent to each corner, to extract parallelogram shaped regions. Kadir et al [3] introduced a method for extracting salient regions. In their algorithm, Shannon entropy of local image attributes (e.g. intensity or color) is calculated over a range of scales. Regions of interest are selected to maximise the entropy as a function of scale.

A performance evaluation of several algorithms was presented by Mikolajczyk et al. [7]. The repeatability and matching score of each algorithm was calculated under various image transformations, such as view point change and scale change. The overall conclusion of the test was that current algorithms are often complementary to one another and that



(a) (b)

Figure 1: An arbitrary octagon of size k and m concentric octagons centred on the same pixel is shown on the left. For each octagon, our measure M, is evaluated between the octagon $\Omega(\alpha^i k, \mathbf{x})$ and the octagon $\Omega(\beta\alpha^i k, \mathbf{x})$ surrounding it (as shown in the image on the right).

their relative performance was scene-dependent. However, MSER and Hessian Affine detectors were the two algorithms that consistently gave good results in most of the tests.

3 FEUDOR

3.1 Measure

When an object is seen by two cameras with different view points, it is very likely that the background of the object is not the same in different views. A good feature would be one that corresponds to a distinctive region within an object without containing elements of the background. The type of feature that we aim to extract, is a homogeneous region that is different from its surroundings. To characterise a region being different, we introduce a measure that finds the mean square difference between the pixel values within a region and mean pixel values of a region surrounding it. If we consider $\Gamma^{(1)}$ be a region with mean RGB value $\mu^{(1)}$ and $\Gamma^{(2)}$ the region surrounding $\Gamma^{(1)}$, with mean $\mu^{(2)}$, then we introduce our measure, M, as:

$$
\begin{aligned}
M(\Gamma^{(2)};\Gamma^{(1)}) &= \frac{1}{|\Gamma^{(2)}|}\sum_{\mathbf{x}\in\Gamma^{(2)}}\Big((\omega(\mathbf{x})-\mu^{(1)})^T(\omega(\mathbf{x})-\mu^{(1)}\Big) && (1)\\
&= \frac{1}{|\Gamma^{(2)}|}\sum_{\mathbf{x}\in\Gamma^{(2)}}\omega^T(\mathbf{x})\omega(\mathbf{x})-2(\mu^{(2)})^T\mu^{(1)}+(\mu^{(1)})^2 && (2)
\end{aligned}
$$

Where $|\Gamma^{(2)}|$ is the number of pixels in the region $\Gamma^{(2)}$ and $\omega(\mathbf{x})$ is a 3×1 column vector containing the RGB values of any pixel $\mathbf{x}$. Thus $M(\Gamma^{(2)};\Gamma^{(1)})$ gives a measure of how dissimilar $\Gamma^{(2)}$ is from the mean of $\Gamma^{(1)}$. For the particular case $\Gamma^{(2)}=\Gamma^{(1)}$, $M(\Gamma^{(1)};\Gamma^{(1)})$ is the variance of the region $\Gamma^{(1)}$.

To achieve rotational invariance, many algorithms have started by extracting circular regions. However to reduce computational complexity we can approximate a circle with another geometric shape (such as square or octagon). We have opted to use regular octagons as they closely approximates a circle. In the rest of this paper, the "size" of an octagon denotes its diameter.

For each pixel position, $\mathbf{x}$, in an image, we associate an octagon of size k, centred on pixel, $\Omega(k,\mathbf{x})$. We evaluate M between this octagon and an annular region of size βk surrounding it, $\Omega(\beta k,\mathbf{x})$. The choice of β determines how much of the background is taken into account when evaluating M. As we are interested in extracting regions at various scales, we repeat the above steps with m more octagons of size $\alpha^i k$, $1 \le i \le m$, as shown in Figure 1(a). The choice of α determines the sampling density in the scale space. M is evaluated between each of these octagons, $\Omega(\alpha^i k,\mathbf{x})$, and their respective surrounding octagon of size $\beta\alpha^i k$, $\Omega(\beta\alpha^i k,\mathbf{x})$, as in Figure 1(b). Then for any region, $\Omega(\alpha^i k,\mathbf{x})$, and the region surrounding it, $\Omega(\beta\alpha^i k,\mathbf{x})$, we define the ratio, D, as:

$$D(\Omega(\alpha^i k,\mathbf{x})) \quad = \quad \frac{M(\Omega(\beta\alpha^i k,\mathbf{x});\Omega(\alpha^i k,\mathbf{x}))}{M(\Omega(\alpha^i k,\mathbf{x});\Omega(\alpha^i k,\mathbf{x}))} \tag{3}$$

Our extracted features are those regions for which D is maximum in both scale, i, and space, $\mathbf{x}$. In order to determine the scale of each feature more accurately, we perform quadratic interpolation in the scale space, to better estimate the scale at which D peaks. In Figure 3 we have shown an example of how D varies as the scale changes.

3.2 Cleaning-up Features

The procedure described above normally extracts a large number of features which is undesirable. Therefore we apply a selection procedure to eliminate features that are not well localised. We choose an octagon, only if two of its opposite sides lie next to a boundary of a region. In Figure 4(a) we have shown examples of some features that are not well localised. We can see that only one side of the extracted octagons lies next to the boundary of a region. In (b) however we have shown examples of regions that we would like to keep. For each of the extracted octagons, at least two opposite sides lie close to the boundary of a region. To identify any such octagons, we take the following steps:

1. Expand and shrink each side of the octagon, calculating D at each instance (including D at the original size)
2. Based on the three values of D, determine whether a peak exists (if the peak does not lie within the three D values, use quadratic interpolation to determine the position of the peak)
3. Calculate the peak's size ratio (which is defined as the ratio between the size of the octagon at which the value of peak D drops by $\frac{1}{\sqrt{2}}$ and the size of the original extracted octagon)

4. Keep an octagon only if at least two opposite sides of the octagon, achieve a narrow peak (i.e. small size ratio) during steps 1-3

During experimentation we have noticed that image noise can give rise to flat peaks in the variations of D as a function of scale. Therefore we have included the condition of peak size ratio in step 4, above, to eliminate any such peaks. With experimentation we have decided to classify a peak as narrow if its size ratio is less than 2.7.

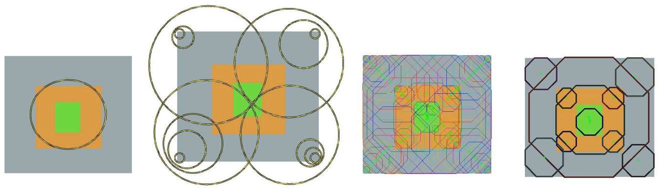


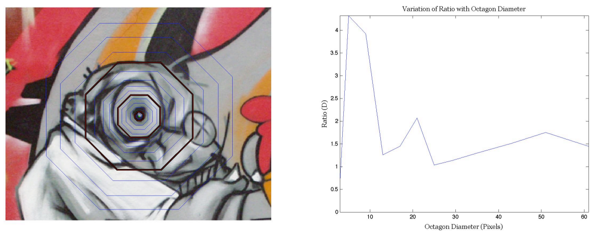
Figure 2: A synthetic image and features extracted by (a) MSER, (b) SURF, (c) FEUDOR. Image (d) shows a subset of FEUDOR regions to highlight the difference between the type of regions extracted by FEUDOR compared to SURF or MSER.

In Figure 2 we have shown a synthetic image which is comprised of three concentric squares of colors green, orange and grey on a white background. We have used this image to highlight the differences between the types of features extracted by various algorithms. In particular we have shown the features extracted by (a) SURF, (b) MSER and (c) FEUDOR.

SURF and MSER both operate on intensity images. If the synthetic image shown in Figure 2 is converted to gray scale, the difference between the different colors of the squares would not be obvious. For this reason we can see that MSER has extracted only one feature which corresponds to the boundary between two of the inner squares. The features extracted by SURF are concentrated around the corners of the outer square. FEUDOR on the other hand operates on color images. It has therefore extracted features corresponding to individual squares as well as the combination of the squares. A subset of features extracted by FEUDOR is shown in (d) to highlight features that correspond to such regions.

As we mentioned before, for some particular applications such as wide-baseline matching or finding correspondences between images seen by cameras with non-overlapping fields of view, features that do not contain elements of the background are desirable. However we can see for example that most of the features extracted by SURF are concentrated around the corners and they also contain elements of the background. In contrast features extracted by FEUDOR do not contain element of the background at all. This shows that FEUDOR features have the potential to be useful in such applications.

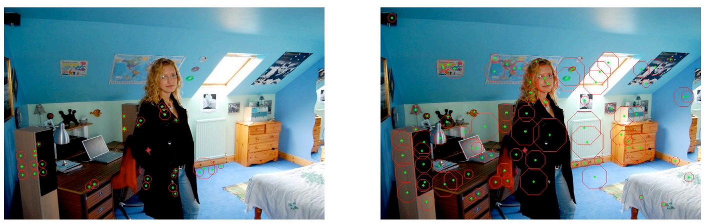
In Figure 3, we have demonstrated an example of a region extracted by FEUDOR in a real image. The blue octagons in the left image represent the concentric octagons centred on a manually selected pixel. Variations of D as the size of the octagons increases has been shown on the right. The red octagons are ones at which D peaks in the scale space.



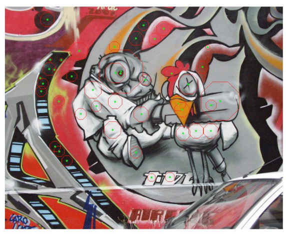
(a) (b)

Figure 3: (a) concentric octagons (blue) centred on a pixel.(b) variation of D as the size of the octagons increases. The red octagons in (a) represent the ones at which the D peaks.

Here we can see an instance at which variations of D as a function of scale has achieved multiple peaks which corresponds to features being extracted at multiple scales. Figure 4(b) and (c) show a selection of features extracted in two different sets of images. We can see that the extracted features are indeed regions that differ from their surroundings.



(a) (b)



(c)

Figure 4: (a) type of regions aimed to be deleted by the procedure introduced in 3.2. (b) and (c) a selection of features extracted on two sets of images.

3.3 Implementation

Use of integral images was introduced by Viola and Jones [9] to evaluate some simple features useful in face detection. Using integral images, sum of the pixel values within

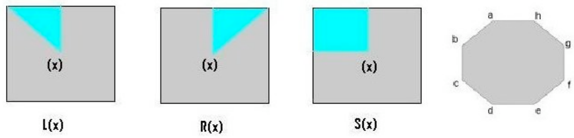


Figure 5: $L(\mathbf{x})$ and $R(\mathbf{x})$ denote the upper left triangle and upper right triangle integral images.

any rectangle, can be calculated with 2 additions and 2 subtractions. Given an image I, with pixel coordinates $\mathbf{u} = [u\,v]^T$, the corresponding integral image, $S(\mathbf{x})$, is defined as:

$$S(\mathbf{x}) = \sum_{x \leq u,, y \leq v} I(\mathbf{u}) \tag{4}$$

To calculate D (eq.3) we need to calculate M for two octagonal regions. For evaluating M (eq.2), mean values and mean squared values of the pixels within each octagonal shaped region has to be computed. By using integral images we have improved the speed of our algorithm substantially. To find the sum of the pixels within an octagonal region, we additionally define $R(\mathbf{x})$, the Right integral image as the sum of the pixels within the upper right triangle and $L(\mathbf{x})$, the Left integral image as the sum of the pixels within the upper left triangle (as shown in Figure 5):

$$R(\mathbf{x}) = \sum_{u \leq x \leq v+u-y} I(\mathbf{u}) \tag{5}$$

$$L(\mathbf{x}) = \sum_{y-v+u \leq x \leq u} I(\mathbf{u}) \tag{6}$$

Then, the sum of the pixels within an octagon with corners denoted by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots, \mathbf{h}$ can be found with only eleven addition/subtraction as follows:

$$\begin{aligned} P_{\mathbf{a},\mathbf{b},\ldots,\mathbf{h}} &= S(\mathbf{a}_2) + R(\mathbf{a}_3) - R(\mathbf{b}_4) - L(\mathbf{c}_2) - S(\mathbf{d}_1) + L(\mathbf{d}_2) + S(\mathbf{e}) & (7) \\ &\quad + R(\mathbf{e}_4) - R(\mathbf{f}_4) - L(\mathbf{g}_3) - S(\mathbf{h}_3) + L(\mathbf{h}_3) & (8) \end{aligned}$$

where subscripts $1, 2, 3, 4$ denote the pixels immediately to the left, upper-left, up and upper-right side of a corner pixel.

To further reduce the computational complexity, we can choose $\beta = \alpha^p,\ 1 \leq p \leq m - p$. Although we have only implemented our algorithm in MATLAB, the extraction of features is still done rather quickly. On an Intel, P4 2GHZ, 2GB RAM, for a 800×600 pixel image, such as the Graffiti scene as shown in Figure 4, around 1300 features has been extracted in under 25 seconds.

4 Performance Comparison

4.1 Evaluation Method

As explained before, one of the criteria for a good feature is that it has to be repeatable. For evaluation of our algorithm in terms of repeatability, we have used a similar methodology to that described by Mikolajczyk in [7]. In his proposed procedure, two regions are considered to correspond if their overlap error as defined below, is less than a certain threshold ε_0 (usually chosen to be 40%):

$$1 - \frac{E_{\phi_a} \cap E_{(H^T \phi_b H)}}{E_{\phi_a} \cup E_{(H^T \phi_b H)}} \quad < \quad \varepsilon_0$$

Where E_ϕ is the elliptical region containing a feature and H is the homography relating the two images to each other.

In [7], before calculating the overlap error, a scale factor is applied to both regions which forces the feature region in the target image to have a fixed size of 30 pixels. This scaling procedure eliminates an unfair advantage that would otherwise be given to feature extraction methods that artificially enlarge the distinctive regions detected in an image. However it also introduces a scale-dependent treatment of region location errors which is undesirable. Since FEUDOR does not enlarge its detected regions, we have chosen to omit the scaling step from the evaluation procedure. We define the "repeatability score" as the ratio between the sum of the feature-to-feature correspondences in the two images and the total number of features located in the portion of the scene that is present in both images.

The test images used during this performance evaluation were the Graffiti (view point change), Wall (view point change) and the Boat (zoom and rotation) scene provided by Mikolajczyk [7]. For our algorithm we chose $\beta = \alpha = \sqrt{2}$. We have deleted features that their peak's size ratio was more than 2.7. We have compared the performance of our algorithm against SURF and MSER. In both cases we have used the default parameter settings given by the author, excluding any parameters used for arbitrarily scaling of the extracted features. For the evaluation with set the region overlap error threshold to %40.

4.2 Results

In Figure 6, we have shown the result of the repeatability score of FEUDOR compared with MSER and SURF. Figure 6(a) and (b) show the result of the repeatability score as the view point angle increases. In (a) our algorithm performs worse than MSER but almost identical to SURF. We can see that as the view point angle goes beyond 40°, the performance of both SURF and FEUDOR degrades rapidly. For our part, one possible explanation could be the fact that in the Graffiti scene, in addition to the view point change, there is an element of camera rotation, which becomes more severe as the view point angle increases. Our algorithm is not designed to be invariant to camera rotation and therefore is not able to cope with this type of transformation. However in (b) where

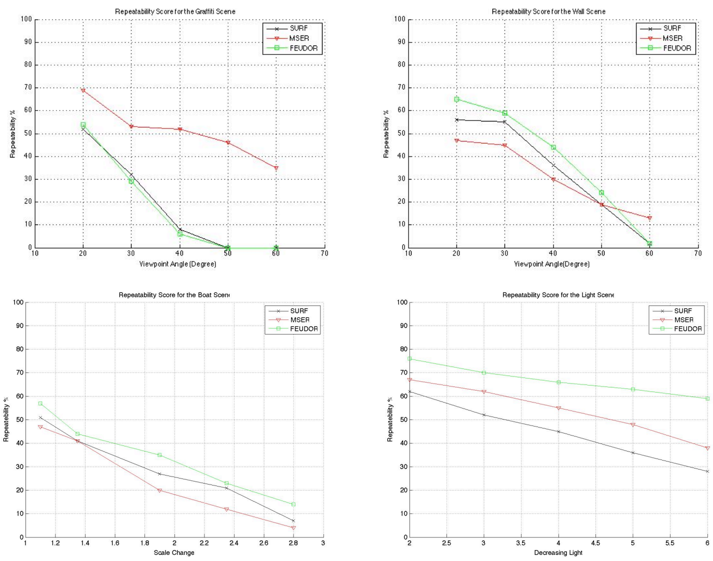


Figure 6: From left to right, top to bottom. Repeatability score under the following transformations: (a) view point change (Graffiti scene) (b) view point change (Wall scene) (c) zoom increasing (Boat Scene) (d) Light decreases (Leuven Scene).

there is no camera rotation as the view point angle increases, we can see that FEUDOR almost always performs better than SURF and MSER.

Figure 6(c) shows the repeatability score as the scale of the image changes. We can see that FEUDOR performs better than both MSER and SURF. However we can see that performance of all three algorithms degrade faster than expected as scale increases. This becomes more evident when our result is compared with those presented in [7] and [1].

This discrepancy is mainly due to the different ways of calculating the repeatability score. If image 1 is the original image used in the scale change test and image 2 is a zoomed out version of image 1, we calculate the repeatability score as the number of correspondences in both images divided by the sum of the overall number of features present in both images. When features present in image 1 are projected to image 2 using the homography, the size of the features will become smaller. This in turn will result in higher overlap errors and therefore lower repeatability score. However in [7], in calculating the repeatability score, only features in image 2 are projected to image 1, not the other way around. In doing so they have avoided this problem. In any case, the result presented in (c) shows that features extracted by FEUDOR are as scale-invariant as those extracted by MSER and SURF.

Finally in (d) we have shown the repeatability score as lighting decreases. Overall FEUDOR has higher repeatability compared to both MSER and SURF under lighting change.

5 Conclusion

In this paper we have introduced a novel scale-invariant feature extraction algorithm called FEUDOR that extracts homogeneous regions that are different from their surroundings. By using integral images our algorithm was implemented efficiently. The average computation time on MATLAB is about 15-25s. We have also shown that performance of FEUDOR under various image transformations, is better than some of the existing algorithms such as SURF and MSER except on occasions where there is an element of camera rotation present. However this was expected as FEUDOR is not designed to be invariant to camera rotations.

By using color images, FEUDOR extracts features that are otherwise indistinguishable when viewed in an intensity image. Moreover unlike many other algorithms, the features extracted by FEUDOR are not concentrated around corners. Most features are within boundaries of a region in the image. So that regions contained in each feature does not contain a significant element of background. This property will make FEUDOR features a good choice for finding correspondences between images viewed by cameras with non-overlapping fields of view and also Wide-baseline matching scenarios.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded-up robust features. In *Proc. European Conf. on Computer Vision*, pages 404–407, 2006.

[2] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proc Alvey Vision Conf*, pages 147–151, 1988.

[3] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. *Proc. European Conf. on Computer Vision*, pages 228–241, 2004.

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. Computer Vision*, 60(2):91–110, 2004.

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

[6] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. European Conf. on Computer Vision*, pages 128–143, 2002.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool. A comparison of affine region detectors. *Intl. J. Computer Vision*, 65(1):43–72, 2005.

[8] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. *Intl. J. Computer Vision*, 59(1):61–85, 2000.

[9] Paul Viola and Michael J. Jones. Robust real-time face detection. *Intl. J. Computer Vision*, 57(2):137–154, 2004.