A Finite Element Blob Detector for Robust Features

Dermot Kerr, Sonya Coleman, and Bryan Scotney

[1] School of Computing and Intelligent Systems, University of Ulster, Magee, UK
[2] School of Computing and Information Engineering, University of Ulster, Coleraine, UK
{d.kerr,sa.coleman,bw.scotney}@ulster.ac.uk

Abstract. Traditionally feature extraction is focussed on edge and corner detection, however, more recently points of interest and blob like features have also become prominent in the field of computer vision and are typically used to determine correspondences between two images of the same scene. We present a new approach to a Hessian blob detector, designed within the finite element framework, which is similar to the multi-scale approach applied in the SURF detector. We present performance evaluation that demonstrates the accuracy of our approach in comparison to well known existing algorithms.

Keywords: blob detector, finite element framework.

1 Introduction

Standard corner detectors such as the Harris and Stephens corner detector [4] find corners in an image at one particular scale, but corner points within an image may occur at many natural scales depending on what they represent [7]. To deal with the many natural scales at which features may be present, corner detectors have been developed to work on multiple scales, thereby having the ability to detect all corners. Many of these corner detectors do not only detect actual corner points but also other "interesting points" that may not strictly be recognized as corners [8, 13]. For some particular applications the ability to detect interesting points that are robust to changes within the image is seen as a more desirable characteristic than specifically detection of real corner points. Blobs are interesting features prominent in images. Generally blobs can be thought of as regions in an image that are brighter or darker than the surrounding regions. Blob-type features provide complementary information not obtained from corner detectors [11].

The Laplacian of Gaussian [9] is a popular blob detector, where the image is convolved with a combined Laplacian and Gaussian kernel. However, the main limitation of this detector is that the detector operates at only one particular scale, but features within an image may appear at many different scales. A multi-scale Laplacian of Gaussian detector may be achieved by appropriately adjusting the size of the Gaussian and Laplacian kernels to obtain a set of kernels that are then applied to the image. Thus, a set of features may be detected at multiple scales. However, applying a detector at multiple scales may introduce other issues, as the same feature may be present over a range of scales within the detector's range [10], and by representing the same feature at many scales we increase the difficulty of matching

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part I, LNCS 6978, pp. 504–513, 2011.

the detected features. Hence, a scale invariant approach is more appropriate, where the characteristic scale of the feature is identified. This characteristic scale is the scale at which the feature is most strongly represented, and it is not related to the resolution of the image, but rather the underlying structure of the detected feature [10]. By using an operator to measure the response of the same interest point at different scales, the scale at which the peak response is obtained can be identified. The Hessian-Laplace blob detector [11] is based on an approach analogous to this, where second order Gaussian smoothed image derivatives are used to compute the Hessian matrix. This matrix captures the important properties of the image structure. Using a multi-scale approach where kernel sizes are increased, the trace and the determinant of the Hessian matrix are thresholded and blob features detected.

In this paper we present a finite element based Hessian blob detector (FEH) based on techniques borrowed from our own FESID detector [6] and ideas from the SURF [2] and CenSurE detectors [1]. Performance is evaluated with respect to repeatability and feature matching using the evaluation techniques presented in [10], highlighting improvements when compared with other well known interest point detectors and descriptors.

2 Hessian Blob Detector Design

The finite element hessian blob detector for robust features uses second order derivative operators to detect blob-like features that are robust to various transformations.

For the propose of operator design and development, we consider an image to be represented by a rectangular $m \times n$ array of samples of a continuous function $u(x,y)$ of image intensity on a domain Ω. The most refined level of finite element discretisation of the image domain Ω is based on considering the image as a rectangular array of pixels. Nodes are placed at the pixel centres, and lines joining these form edges of elements in the domain discretisation. The near-circular operators presented in [12] were based on the use of a "virtual mesh", illustrated in Figure 1, consisting of regular triangular elements, which overlays a regular rectangular pixel array.

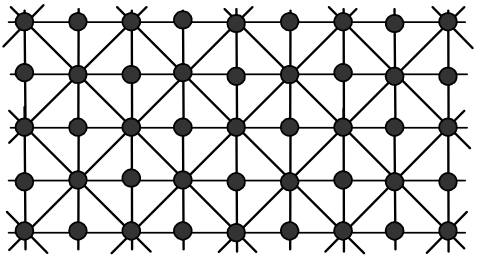


Fig. 1. Virtual mesh of triangular elements

With any node in a mesh, say node *i*, with co-ordinates (x_i, y_i) we associate a piecewise linear basis function $\phi_i(x, y)$ which has the properties

$$\phi_i(x_j, y_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{1}$$

where (x_j, y_j) are the co-ordinates of the nodal point *j*. $\phi_i(x, y)$ is thus a "tent-shaped" function with support restricted to a small neighbourhood centred on node *i* consisting of only those elements that have node *i* as a vertex. For any scale parameter σ, we may define a neighbourhood Ω_i^σ centred on the node *i* at (x_i, y_i) and consisting of a compact subset of elements. If the set of nodes contained in or on the border of Ω_i^σ is denoted as D_i^σ, then we may approximately represent the image *u* over the neighbourhood Ω_i^σ by a function

$$U(x, y) = \sum_{j \in D_i^\sigma} U_j \phi_j(x, y) \tag{2}$$

in which the parameters $\{U_j\}$ are mapped from the sampled image intensity values. The approximate image representation is therefore a simple piecewise linear function on each element in the neighbourhood Ω_i^σ and has the sampled intensity value U_j at node *j*.

To formulate operators involving a weak form of the second order directional derivative in the finite element method, it is required that the image function $u \equiv u(x, y)$ be once differentiable in the sense of belonging to the Hilbert space $H^1(\Omega)$; i.e. the integral $\int_\Omega \left(|\underline{\nabla} u|^2 + u^2 \right) d\Omega$ is finite, where $\underline{\nabla} u$ is the vector $(\partial u / \partial x, \partial u / \partial y)^T$. To obtain a weak form of the second directional derivative, $-\underline{\nabla} \cdot (\mathrm{B} \underline{\nabla} u)$, the respective derivative term is multiplied by a test function $v \in H^1$ and the result is integrated on the domain Ω to give

$$Z(u) = -\int_\Omega \underline{\nabla} \cdot (\mathrm{B} \underline{\nabla} u) v d\Omega \,. \tag{3}$$

Here $\mathrm{B} = \underline{b}\underline{b}^T$ and $\underline{b} = (\cos\theta, \sin\theta)$ is the unit direction vector. Integrated directional derivative operators have been used in [15], though not based on the weak forms introduced above.

Since we are focusing on the development of operators that can explicitly embrace the concept of scale, a finite-dimensional test space $T_\sigma^h \subset H^1$ is employed that explicitly embodies a scale parameter σ. In our design procedure the test space T_σ^h comprises a set of Gaussian basis functions $\psi_i^\sigma(x, y)$, *i*=1,…, *N* of the form

$$\psi_i^\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2}\right)} \tag{4}$$

Each test function $\psi_i^\sigma(x,y)$ is restricted to have support over a neighbourhood Ω_i^σ centred on the node i at (x_i, y_i). The size of the neighbourhood Ω_i^σ to which the support of $\psi_i^\sigma(x,y)$ is restricted is also explicitly related to the scale parameter σ [3]. We construct three operators Dxx_{ij}^σ, Dyy_{ij}^σ, and Dxy_{ij}^σ, representing the second order x- y- and mixed xy- derivatives respectively:

$$Dxx_{ij}^\sigma = \int_{\Omega_i^\sigma} \frac{\partial \phi_j}{\partial x} \frac{\partial \psi_i^\sigma}{\partial x} dxdy,\ i,j = 1,\ldots,N \tag{5}$$

$$Dyy_{ij}^\sigma = \int_{\Omega_i^\sigma} \frac{\partial \phi_j}{\partial y} \frac{\partial \psi_i^\sigma}{\partial y} dxdy,\ i,j = 1,\ldots,N \tag{6}$$

and

$$Dxy_{ij}^\sigma = \int_{\Omega_i^\sigma} \frac{\partial \phi_j}{\partial x} \frac{\partial \psi_i^\sigma}{\partial y} dxdy,\ i,j = 1,\ldots,N \tag{7}$$

The integrals are computed as sums of the element integrals and are computed only over the neighbourhood Ω_i^σ, rather than the entire image domain Ω as ψ_i^σ has support restricted to Ω_i^σ.

3 Blob Detection

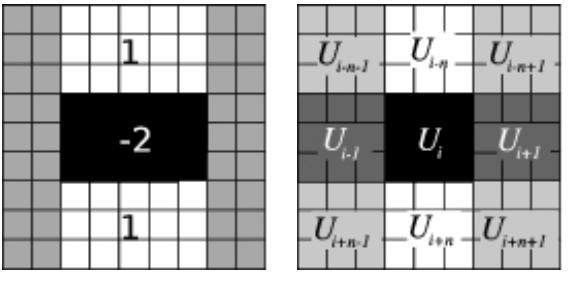
For efficient implementation, we have adopted the use of integral images introduced by Viola and Jones [14]; more recently integral images have been a key aspect of the SURF detector and we have previously successfully used integral images with the FESID detector [6] as integral images provide a means of fast computation using small convolution filters.

If an intensity image is represented by an array of $m \times n$ samples of a continuous function $u(x,y)$ of image intensity on a domain Ω, then the integral image value $I_\Sigma(x)$ at a pixel location $\mathbf{x} = (x,y)$ is the sum of all pixel values in the original image I within a rectangular area formed by the origin of the image and location $\mathbf{x}$:

$$I_\Sigma(\mathbf{x}) = \sum_{i=0}^{i\le x} \sum_{j=0}^{j\le y} I(i,j). \tag{8}$$

The time and number of operations required to compute any rectangular area of the integral image is independent of the size of that region, as four memory reads and three additions are required to compute any region,

Using the same multi-scale approach as the SURF detector we select the first filter size of a 9×9 pixel region. In our approach we partition the 9×9 pixel region slightly differently as illustrated in Figure 2.



(a) (b)

Fig. 2. 9×9 filter partitioning for (a) SURF and (b) FEH detector

Our approach differs from the SURF detector in that we need to compute 9-regions for each operator, rather than the 3 or 4 regions that are computed with the SURF detector. The filter partitioning allows the operator values to be simply mapped to the appropriate 3×3 region on the 9×9 filter. The operator values, Dxx_{ij}^{σ}, Dyy_{ij}^{σ}, and Dxy_{ij}^{σ}, (equations 5-7) mapped to the 9×9 regions are then convolved with the sum of the pixel intensities from each of the areas illustrated in Figure 2(b) to form the Hessian matrix

$$H=\begin{bmatrix} Dxx & Dxy \\ Dxy & Dyy \end{bmatrix} \tag{9}$$

The Hessian matrix captures the important properties of the local image structure by describing how the underlying shape varies [11].

The normalised determinant of the Hessian matrix is computed using the formula

$$\det(H)=Dxx\times Dyy+(0.81\times Dxy)^2 \tag{10}$$

where the constant term 0.81 is determined by the size of the filter, i.e., 9×9. This approximated determinant of the Hessian represents the blob response in the image at that particular location.

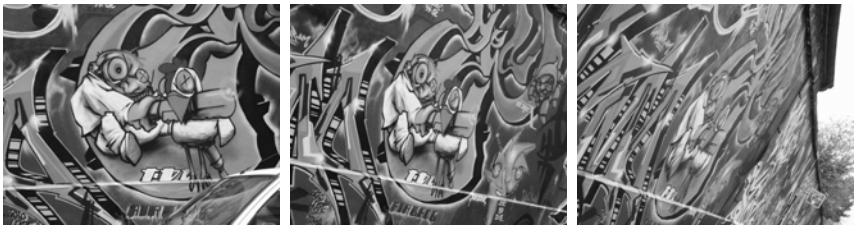
Similarly, blob responses are computed over further scales by increasing the overall size of the filter, but maintaining the 9 regions. For example, within the first octave filter sizes of 9×9, 15×15, 21×21, and 27×27 are used, and each of these filters has 9 individual regions of size 3×3, 5×5, 7×7, and 9×9 respectively. The blob response is computed over a total of 4 octaves that each contain 4 scale ranges. Blob responses that are not maxima or minima in the immediate neighbourhood of the selected blob are rejected by examining a $3\times3\times3$ neighbourhood (in the *x*- and *y*- spatial dimensions,

and the scale dimension) around the selected blob. For the remaining blobs that give responses above a specified threshold, these blobs are interpolated in 3D (two spatial dimensions, and scale) to accurately localise the blob [6].

4 Experimental Results

We present results of comparative performance evaluation for our proposed FEH blob detector and other well-known blob detectors, the SURF detector and the Hessian Laplace detector; the detectors used for comparison are limited to those that are most similar to FEH in terms of operation. A full evaluation of various detectors using the same software and images has been carried out in [11], and the reader is referred to this work for full details.

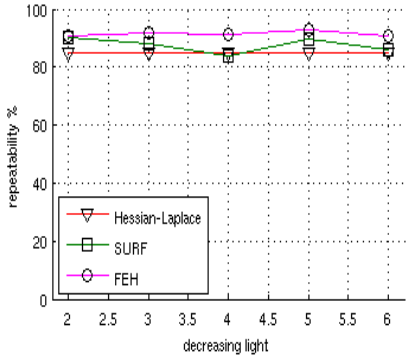
Evaluation of FEH was performed using the set of test images and software provided from the collaborative work between Katholieke Universiteit Leuven, Inria Rhone-Alpes, Visual Geometry Group and the Center for Machine Perception (available for download at [16]). Using the repeatability metric we explicitly compare the geometrical stability of detected points of interest between different images of a scene under different viewing conditions. The test image set consists of real structured and textured images of various scenes, with different geometric and photometric transformations such as viewpoint change, image blur, illumination change, scale and rotation and image compression. For the detectors presented here we describe a circular region with a diameter that is 3× the detected scale of the point of interest, similar to the approach in [10, 11]. The overlap of the circular regions corresponding to an interest point pair in a set of images is measured based on the ratio of intersection and union of the circular regions. Thus, where the error in pixel location is less than 1.5 pixels and the overlap error is below 60%, similar to the evaluation of the SURF detector [2], the points of interest are deemed to correspond. Example images are shown in Figure 3.



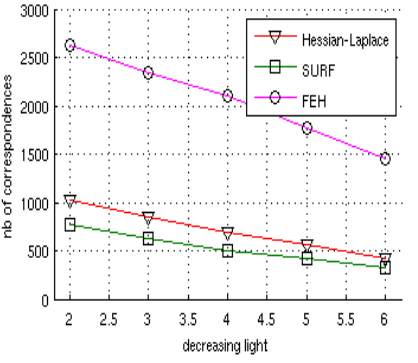
(a) Viewpoint angle 0° (b) Viewpoint angle 30° (c) Viewpoint angle 60°

Fig. 3. Viewpoint change - example image sequence [5]

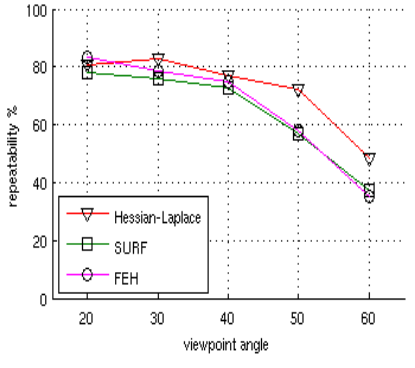
In Figure 4(a) and 4(b) we present comparative evaluation of the detectors using the illumination change scene. The repeatability rate for the FEH detector is consistently better than either SURF or the Hessian-Laplace operators, and generally the number of correspondences is also greater. Figures 4(c) – 4(f) present the evaluation results using the viewpoint change scenes. In Figures 4(c) and 4(d), using the structured viewpoint change, the Hessian-Laplace operator has the best percentage repeatability, and the proposed FEH operator performs similarly to the SURF operator; the FEH has a slightly higher number of correspondences than the Hessian-Laplace operator.



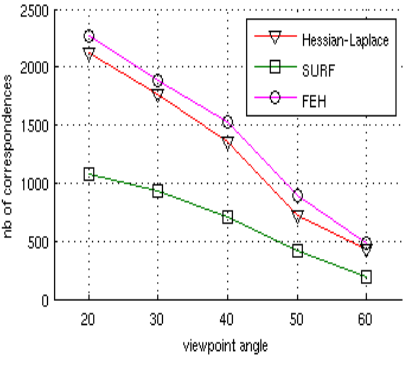
(a) Repeatability score



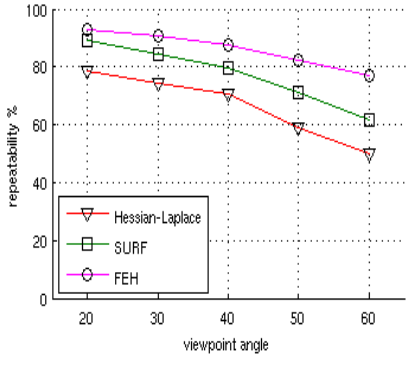
(b) # Corresponding region



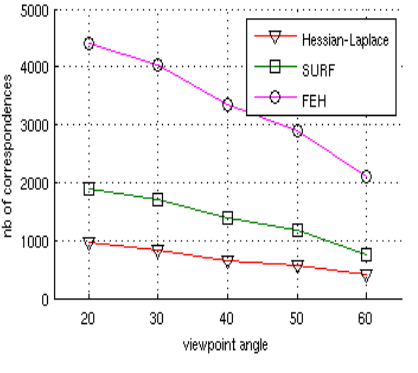
(c) Repeatability score (structured)



(d) # Corresponding region (sturctured)

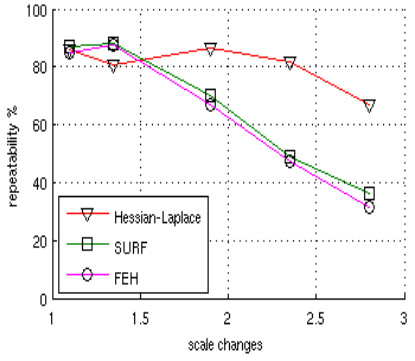


(e) Repeatability score (textured)

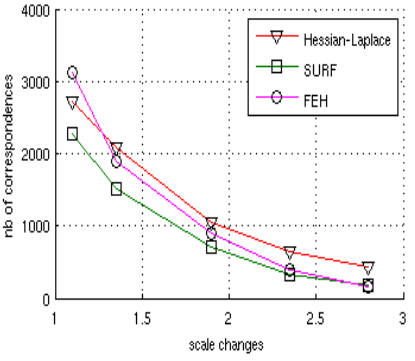


(f) # Corresponding region (textured)

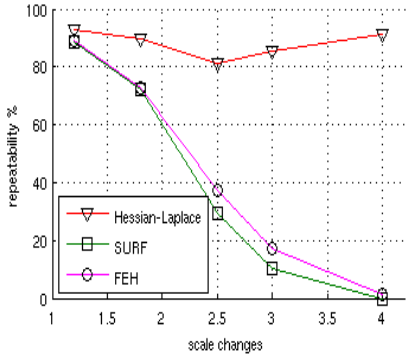
Fig. 4. Repeatability score and number of corresponding regions for image sequences containing illumination change (a)-(b); viewpoint change (c)-(f)



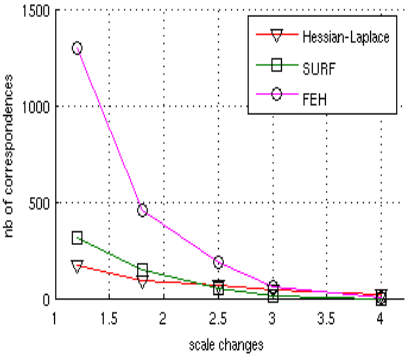
(a) Repeatability score (structured)



(b) # Corresponding region (sturctured)



(c) Repeatability score (textured)



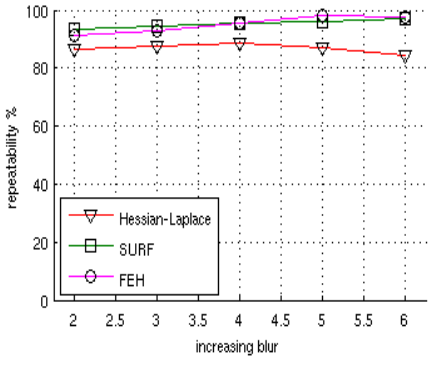
(d) # Corresponding region (textured)

Fig. 5. Repeatability score and number of corresponding regions for image sequences containing scale change

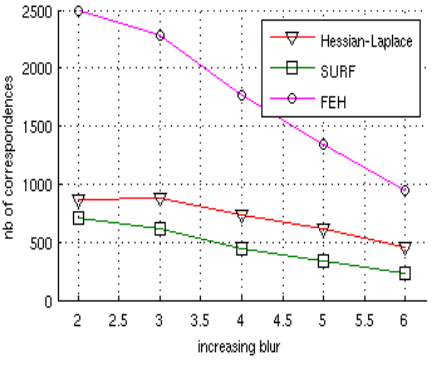
However, in Figure 4(e), using the textured viewpoint change, the proposed FEH operator has the best percentage repeatability, with the Hessian-Laplace operator having the poorest performance; in Figure 4(f), we see that FEH has a significantly larger number of correspondences compared to either the SURF or Hessian-Laplace operators, which possibly leads to the improved repeatability rate.

Figure 5 presents results for the three detectors on image sequences containing scale change. In these images sequences, the Hessian-Laplace operator outperforms the other two operators with respect to repeatability score. However, Figures 5(a) and 5(c) illustrate that the performance of the proposed FEH operator is similar to the SURF operator; the performance of FEH is slightly poorer than SURF in Figure 5(a) and slightly better than SURF in Figure 5(c). Figures 5(b) and 5(d) again illustrate that the number of correspondences is high for the FEH operator. Figure 6 presents results for the three detectors on image sequences containing blur change, and here we see that the FEH and SURF operators perform similarly and better than the Hessian-Laplace operator. In addition to the performance evaluation, we include

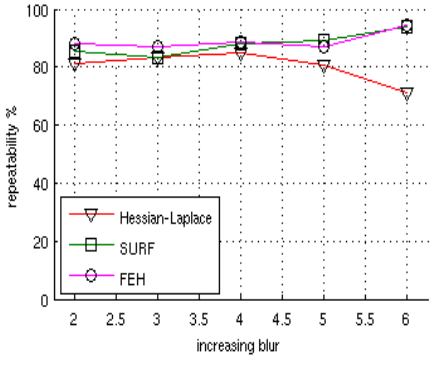
run-time for the three algorithms in Table 1 where it can be seen that FEH is faster than the Hessian-Laplace approach.



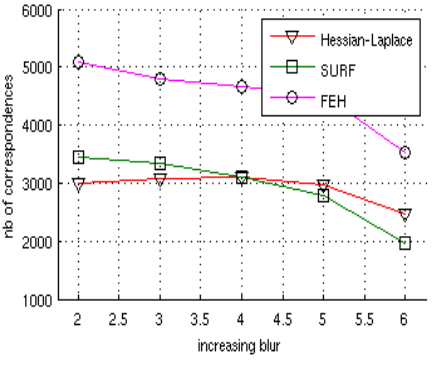
(a) Repeatability score (structured)



(b) # Corresponding region (sturctured)



(c) Repeatability score (textured)



(d) # Corresponding region (textured)

Fig. 6. Repeatability score and number of corresponding regions for image sequences containing blur change

Table 1. Algorithmic run-times

Blob Detector	Run-time(secs)
Hessian-Laplace	0.871
SURF (including SURF-E descriptors	0.636
FEH (including SURF-E descriptors)	0.809

5 Conclusions and Future Work

The results of our comparative performance evaluation indicate that the FEH detector performs better than the SURF detector on most sequences. In some sequences such as the structured viewpoint change the Hessian-Laplace detector performs better than the FEH detector. This is most likely due to the fact that the Hessian-Laplace detector uses large Gaussian derivatives to compute the Hessian matrix rather than the cruder approximations used in the SURF and FEH detectors; however this also makes the Hessian-Laplace operator relatively computationally expensive.

References

[1] Agrawal, M., Konolige, K., Blas, M.R.: CenSurE: Center surround extremas for realtime feature detection and matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)

[2] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). CVIU 110, 346–359 (2008)

[3] Davies, E.R.: Design of Optimal Gaussian Operators in Small Neighbourhoods. Image and Vision Computing 5(3), 199–205 (1987)

[4] Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proc. 4th Alvey Vision Conf., pp. 147–151 (1988)

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

[6] Kerr, D., Coleman, S., Scotney, B.: FESID: Finite element scale invariant detector. In: Foggia, P., Sansone, C., Vento, M. (eds.) ICIAP 2009. LNCS, vol. 5716, pp. 72–81. Springer, Heidelberg (2009)

[7] Lindeberg, T.: Scale-Space Theory in Computer Vision. Kluwer, Academic Publisher/Springer, Dordrecht, Netherlands (1994)

[8] Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. ICCV, pp. 1150–1157 (1999)

[9] Marr, D., Hildreth, E.: Theory of Edge Detection. Proceedings of the Royal Society of London, 215–217 (1980)

[10] Mikolajczyk, K., Schmid, C.: Scale & Affine Invariant Interest Point Detectors. IJCV 60, 63–86 (2004)

[11] 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, 43–72 (2005)

[12] Scotney, B.W., Coleman, S.A.: Improving Angular Error via Systematically Designed Near-Circular Gaussian-based Feature Extraction Operators. Pattern Recognition 40(5), 1451–1465

[13] Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision 3(3), 177–280 (2008)

[14] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, pp. 511–518 (2001)

[15] Zuniga, O.A., Haralick, R.M.: Integrated Directional Derivative Gradient Operator. IEEE Trans. on Systems, Man, and Cybernetics SMC-17(3), 508–517 (1987)

[16] Mikolajczyk, K.: Affine Covariant Features (July 15, 2007), http://www.robots.ox.ac.uk/~vgg/research/affine/ (March 14, 2011)