J Math Imaging Vis (2012) 44:150–167
DOI 10.1007/s10851-011-0317-8

Hessian-Based Affine Adaptation of Salient Local Image Features

Ruan Lakemond · Sridha Sridharan · Clinton Fookes

Published online: 28 September 2011

Abstract Affine covariant local image features are a powerful tool for many applications, including matching and calibrating wide baseline images. Local feature extractors that use a saliency map to locate features require adaptation processes in order to extract affine covariant features. The most effective extractors make use of the second moment matrix (SMM) to iteratively estimate the affine shape of local image regions. This paper shows that the Hessian matrix can be used to estimate local affine shape in a similar fashion to the SMM. The Hessian matrix requires significantly less computation effort than the SMM, allowing more efficient affine adaptation. Experimental results indicate that using the Hessian matrix in conjunction with a feature extractor that selects features in regions with high second order gradients delivers equivalent quality correspondences in less than 17% of the processing time, compared to the same extractor using the SMM.

Keywords Local image features · Affine adaptation · Wide baseline matching · Shape estimation

R. Lakemond (✉) · S. Sridharan · C. Fookes
Image and Video Research Laboratory,
Queensland University of Technology, GPO Box 2434,
2 George St, Brisbane, Queensland 4001, Australia
e-mail: r.lakemond@qut.edu.au

S. Sridharan
e-mail: s.sridharan@qut.edu.au

C. Fookes
e-mail: c.fookes@qut.edu.au

1 Introduction

Local image features are patterns in an image that are defined in limited image areas and are distinguishable from the surrounding image in some way. Such features may be extracted from each view of a scene independently and then matched to find sets of correspondences between views. The correspondences are commonly used for a large variety of tasks, including automatic camera calibration [1], 3D reconstruction [2, 3], mosaicking [4], object recognition and classification [5] and arranging image databases [6]. The various applications have different requirements in terms of feature robustness. In surveillance environments cameras are almost invariably distributed very sparsely (wide baseline). Calibrating cameras for intelligent surveillance environments [7] therefore requires features that are highly robust to view change. A comprehensive review of local image feature extractors may be found in [8].

This paper focuses on derivative-based feature extractors. These extractors make use of adaptation processes to produce features that are robust to affine deformations. The most successful affine adaptation algorithms make use of the second moment matrix (SMM) to estimate local feature shape. The purpose of this paper is to show theoretically and experimentally that the Hessian of image intensities may also be used for estimating local affine shape. The Hessian is simpler to compute than the SMM and may enable a reduction in computation time. It may also be more convenient to compute, for example, when using the determinant of Hessian extractor. Results indicate that a blob detector with Hessian shape adaptation requires on average 17% of the processing time of an equivalent detector with SMM based adaptation, while achieving marginally better repeatability and producing more correspondences. Conditions where the Hessian does not provide a useful shape measure have also been identified and are discussed in this paper.

2 Background and Notation

Among the various types of feature extractors is a class of extractor that locates features in an image by computing a saliency map of the image using partial derivatives. The saliency map shows the regions of the image that have high curvature. Feature points are selected by finding maxima in the saliency map. The partial derivative operators are inherently sensitive to changes in view point and do not produce view covariant features directly. One possible approach to extracting affine covariant features is to expand the image space to a five dimensional affine space by applying various degrees of affine scaling. This is computationally intractable for most applications, as demonstrated by the salient region detector [9] and the performance figures in [10]. Though the method of [9] uses entropy to evaluate saliency, the problem of complexity due to dimensionality applies to partial derivative-based methods. More efficient methods have been developed to adapt salient features to be affine covariant, but this adaptation process remains computationally expensive.

Other feature extractors have been demonstrated that are less computationally demanding, however these do not share the same strengths as the affine adapted saliency based extractors. The MSER extractor [11] for example produces affine covariant features using a process similar to the watershed algorithm. It is less computationally demanding, but it delivers far fewer features than the saliency-based methods [10]. Saliency-based methods have been shown to perform better than MSER when scene objects do not consist of predominantly flat surfaces [12]. Even faster extractors exist, such as SURF [13] and FAST [14]; however, these do not generate affine covariant features and are ineffective in wide baseline scenarios.

Local features may be defined in terms of an affine transformation of $\mathbf{x} \in \mathcal{R}^2$, expressed as,

$$
\begin{aligned}
&H(k,\theta,q,\phi,t_x,t_y,\mathbf{x}) = k\mathbf{R}(\theta)\mathbf{A}(q,\phi)\mathbf{x} + \mathbf{t}(t_x,t_y),\\
&\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},\\
&\mathbf{t}(t_x,t_y) = \begin{bmatrix} t_x & t_y \end{bmatrix}^\top,\\
&\mathbf{A}(q,\phi) = \mathbf{R}(-\phi)\begin{bmatrix} q & 0 \\ 0 & q^{-1} \end{bmatrix}\mathbf{R}(\phi).
\end{aligned}
\tag{1}
$$

These components are referred to as the position, $\mathbf{t}$, scale, k, shape, $\mathbf{A}$ and orientation, $\mathbf{R}$, of a feature. A transformation of the above form may be interpreted as a mapping from a unit circle centred at the origin to an ellipse circumscribing a feature and can be used to normalise a feature.

The saliency map approach can be used to extract the position of features ($\mathbf{t}$) at a given scale, but does not provide a method for extracting regions that are covariant with changes in viewpoint. Example saliency operators include the Harris & Stephens operator [15], the determinant of Hessian operator [16] and the Laplacian of Gaussian and Difference of Gaussians operators [17]. Methods have been developed to adapt saliency features such that the resulting features are affine covariant. Many methods adapt different components or parameters of the affine transformation separately (examples are given in the following discussion). The following subsections discuss the development of adaptation methods for scale, shape and rotation adaptation.

2.1 Scale Adaptation

A saliency map only reveals features that are of the scale or resolution at which the map is computed. In order to find features with a wide range of sizes in an image, a multi-scale analysis of the image is required. This is achieved by defining a scale space for images and applying the saliency operator and maxima extraction over a range of scales. In the feature extraction literature, the most commonly used definition for image scale space is based on the Gaussian smoothing operator. Other definitions are also available. See [18] for a detailed discussion on scale space theory.

The Gaussian scale space operator is defined as,

$$
g(\mathbf{x},\sigma) = \frac{1}{\sigma^2 2\pi}\exp\left(\frac{-\mathbf{x}^\top\mathbf{x}}{2\sigma^2}\right), \tag{2}
$$

where σ is referred to as the scale parameter. The scale space of image $I(\mathbf{x})$ is generated by convolution with this operator, yielding a three-dimensional space denoted $I(\mathbf{x},\sigma)$, where the third dimension corresponds to the scale parameter of the Gaussian operator.

A well established method of producing scale invariant features is to select characteristic scale features. In [19] it is proposed that some combination of gamma normalised derivatives computed in the vicinity of an image structure will assume a local maximum at the scale corresponding to the structure size. Normalised derivatives of a Gaussian scale space image are defined as,

$$
I_{i^n}(\mathbf{x},\sigma) = \sigma^{\gamma n}\frac{\partial^n}{\partial i^n} I(\mathbf{x},\sigma),
$$

where i is any dimension of $I(\mathbf{x})$.

The simplest method of characteristic scale selection is to locate scale-space maxima—points that represent local maxima of the saliency map in the spatial and scale dimensions. Example implementations have been published describing scale-space non-maximum suppression using the Laplacian [19], Difference of Gaussians [20] and determinant of Hessian [13] functions, for example.

The scale space maxima approach was extended in [21] by locating points that are local maxima of the Harris function in the spatial dimensions and maxima of the scale normalised Laplacian in the scale dimension. While functions

such as the Harris and determinant of Hessian provide good spatial localisation, it was found that the Laplacian is more effective for scale selection. An iterative version of this algorithm is proposed in [21], which offers higher accuracy scale selection than simple 3D non-maximum suppression.

In [22, 23] a method is presented for composing a graph (termed the scale-space feature sketch) of multi-scale features such that the graph is a discrete representation of the loci of features in scale-space. Characteristic scale features can be selected by evaluating a scale response function (such as the normalised Laplacian) at each feature and finding local maxima in the graph. This method yields superior results to 3D non-maximum suppression, while enabling greater computational efficiency compared to both 3D non-maximum suppression and iterative scale selection.

2.2 Shape Adaptation

Extending the three parameter scale covariant feature extraction problem to a five parameter affine problem results in a parameter space that is too large to search exhaustively. Affine adaptation is therefore initialised either with a set of multi-scale features or a set of characteristic scale features.

The majority of modern shape estimation methods derive from [24], which uses the second moment matrix (SMM) to iteratively measure local shape. In [24], the Gaussian scale space is extended to affine Gaussian scale space. The affine Gaussian operator is of the form,

$$g(\mathbf{x}, \Sigma) = \frac{1}{2\pi|\Sigma|} e^{-\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}}, \tag{3}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix and $|\Sigma|$ is the determinant of $\boldsymbol{\Sigma}$. Affine scale space is then constructed as, $I(\mathbf{x}, \boldsymbol{\Sigma}) = g(\mathbf{x}, \Sigma) * I(\mathbf{x})$. The second moment matrix computed in affine scale space is defined as,

$$m_2(\mathbf{x}, \boldsymbol{\Sigma}_D, \boldsymbol{\Sigma}_I) = |\boldsymbol{\Sigma}_D| g(\mathbf{x}, \boldsymbol{\Sigma}_I) * \mathbf{D}_A,$$

$$\mathbf{D}_A = \begin{bmatrix} I_x^2(\mathbf{x}, \boldsymbol{\Sigma}_D) & I_x I_y(\mathbf{x}, \boldsymbol{\Sigma}_D) \\ I_x I_y(\mathbf{x}, \boldsymbol{\Sigma}_D) & I_y^2(\mathbf{x}, \boldsymbol{\Sigma}_D) \end{bmatrix},$$

with $\boldsymbol{\Sigma}_D$ and $\boldsymbol{\Sigma}_I$ differing only in scale. The SMM is essentially a local estimate of the covariance matrix of image gradients.

Affine adaptation is performed by iteratively computing the SMM as,

$$\mathbf{M}_i = m_2(\mathbf{x}, k_D \mathbf{M}_{i-1}, k_I \mathbf{M}_{i-1}),$$

where i is the iteration number, k_D is chosen to maximise the value of the Laplacian at $\mathbf{x}$, k_I is chosen so that the minimum eigenvalue of $k_I \mathbf{M}$ remains constant during iterations and $\mathbf{M}_0 = \mathbf{I}$ (the identity matrix). It is shown that if $\mathbf{M}$ is computed as above, then it converges such that, for sufficiently large n,

$$m_2(\mathbf{x}, k_D \mathbf{M}_n, k_I \mathbf{M}_n) \approx \mathbf{M}_n,$$

and that the resulting matrix, $\mathbf{M}$, is covariant under affine transformations of the image. This method effectively adapts the scale and shape components while the feature position is kept fixed at its initial position.

The method presented in [25] applies a normalising affine transformation to a local image region, instead of adapting the parameters of affine scale space. The integration scale and differentiation scale of the SMM operator are set proportional to the scale at which the feature is detected. At each iteration of the algorithm, the local image region around the selected feature is transformed using the inverse square root of the SMM computed during the previous iteration (initially $\mathbf{I}$). The SMM is then computed again from the normalised image region using radially symmetric Gaussian kernels and is normalised to have a determinant of 1. This continues until the measured normalised SMM is sufficiently close to the identity matrix. The final shape transformation is the composition of all the normalisation transformations applied during adaptation. The method of [25] is more easily implemented and more efficient than the method of [24], but only adapts the shape component while leaving the scale and position fixed. This method is also applied to both Harris and Determinant of Hessian features in [10].

A more complete algorithm is presented in [21] that updates the integration scale, differentiation scale and feature location at each iteration, before computing the SMM. A measure of local shape eccentricity is defined as the ratio of the smallest eigenvalue, $\lambda_{\min}$ and the largest eigenvalue, $\lambda_{\max}$ of the SMM,

$$Q(\mathbf{M}) = \frac{\lambda_{\min}(\mathbf{M})}{\lambda_{\max}(\mathbf{M})}. \tag{4}$$

Adaptation concludes when Q is sufficiently close to 1 (symmetric). This is a more computationally expensive algorithm than the method of [25]. It is the only algorithm that allows for a change in scale and position as the shape is adapted. In the evaluation presented in [10], the authors chose to use a method most similar to [25], and the method in [21] was not evaluated. The performance of this algorithm is therefore unknown, however the preference for the method in [25] indicates that perhaps the method in [21] does not produce superior results. The authors' own evaluations of a complete iterative scale, position and shape adaptation algorithm show a great increase in computational cost and negligable change in repeatability performance. For this reason, a shape adaptation only framework is used in this paper.

Other methods for measuring the affine shape of each feature include Edge-Based Regions (EBR) and Intensity-Based Regions (IBR) [26–28]. EBR is designed for corner features extracted using the Harris detector and uses the edges emanating from the corner to fit a parallelogram feature. IBR is designed for features selected at the local image intensity extrema. It finds the edges of the feature region by finding the maxima of a cost function evaluated along rays emanating from the feature centre and fits an ellipse to these edge points. The salient region detector [9] evaluates the entropy of a three parameter family of ellipses at each pixel (essentially generating a 5D saliency map) and selects maxima of this map as candidates. The top P regions with highest magnitude of the derivative of the pdf with respect to scale are retained. This method is computationally impractical and performs poorly. The evaluations in [12, 29] show that SMM adapted Hessian features and MSER features most consistently achieve good performance.

2.3 Orientation Selection

The affine covariant extractors found in the literature do not assign an orientation to features and the evaluations [12, 29] do not include orientation in computing feature error. The most popular method for selecting feature orientation is included in the Scale Invariant Feature Transform (SIFT) descriptor [17]. This method selects orientations corresponding to the dominant first order gradient directions in the feature area.

3 Hessian-Based Affine Adaptation

The SMM has to date been the most effective affine shape estimator used in affine adaptation of local image features. This section explores the novel approach of using the Hessian matrix as an affine shape measure for affine adaptation. The Hessian of an image is defined as the matrix of second order partial derivatives of the image intensity with respect to coordinates,

$$\frac{\partial^2 I(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} = \begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial xy} \\ \frac{\partial^2}{\partial xy} & \frac{\partial^2}{\partial y^2} \end{bmatrix} I(\mathbf{x}).$$

The main motivations for using the Hessian matrix, instead of the SMM, is that the Hessian is simpler to implement and requires less computational effort (see Sect. 3.3). The Hessian is already used to compute the Determinant of Hessian saliency operator and the trace of the Hessian matrix is the Laplacian, which is used for scale selection. Using the Hessian for shape adaptation as well can therefore result in an affine covariant feature extractor that employs a single operator at its core. There are, however, some limitations to how the Hessian matrix can be used.

3.1 Symmetric Local Hessian

The following discussion considers continuous signals. Section 3.2 discusses the implications of working with discrete images.

Let the property of second-order symmetry be defined as follows:

Definition 1 A function $f(\mathbf{x})$ is symmetric in terms of the Hessian, around coordinates $\mathbf{x}_i$, if the eigenvalues of the Hessian of $f(\mathbf{x})$, evaluated at $\mathbf{x} = \mathbf{x}_i$, are equal and hence the Hessian (always a symmetric matrix) is a scalar matrix.

Symmetry, by this definition, has several properties:

1. Symmetry is a local property, defined at a particular set of coordinates.
2. Symmetry is shift covariant due to the shift covariance of the Hessian matrix—if $f(\mathbf{x})$ is symmetric around $\mathbf{x}_i$, then $f(\mathbf{x}+\mathbf{d})$ is symmetric around $\mathbf{x}_i - \mathbf{d}$.
3. Symmetry is rotation covariant due to the rotation covariance of the Hessian matrix—if $f(\mathbf{x})$ is symmetric around $\mathbf{x}_i$, then $f(\mathbf{R}\mathbf{x})$ is symmetric around $\mathbf{R}^\top \mathbf{x}_i$.
4. Symmetry is invariant to a scalar transformation of coordinates—if $f(\mathbf{x})$ is symmetric around $\mathbf{x}_i$, then $f(k\mathbf{x})$ is symmetric around $k^{-1}\mathbf{x}_i$. Isotropic scaling of a scalar matrix yields a scalar matrix.
5. Symmetry is ill-defined when the magnitude of the eigenvalues of the Hessian approach zero.

In the following it will be shown that the Hessian is affine covariant, that it can be used to measure the shape of a function at a point and to compute a transformation mapping a function to a Hessian symmetric function. Let function $i(\mathbf{x})$ be an arbitrary function that is symmetric around $\mathbf{x} = \mathbf{x}_i$. Let $f(\mathbf{x})$ be related to $i(\mathbf{x})$ by an affine transformation,

$$f(\mathbf{x}) = i(\mathbf{U}\mathbf{x} + \mathbf{t}),$$

where $\mathbf{U}$ is of the form, $\mathbf{U} = k\mathbf{R}\mathbf{A}$ (components defined in (1)). The Hessian of $f(\mathbf{x})$ is,

$$\begin{aligned} \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top} &= \frac{\partial^2 i(\mathbf{U}\mathbf{x}+\mathbf{t})}{\partial \mathbf{x} \partial \mathbf{x}^\top} \\ &= \mathbf{U}^\top \begin{bmatrix} i_{xx}(\mathbf{U}\mathbf{x}+\mathbf{t}) & i_{yx}(\mathbf{U}\mathbf{x}+\mathbf{t}) \\ i_{xy}(\mathbf{U}\mathbf{x}+\mathbf{t}) & i_{yy}(\mathbf{U}\mathbf{x}+\mathbf{t}) \end{bmatrix} \mathbf{U}. \end{aligned}$$

Evaluating this equation at $\mathbf{x} = \mathbf{x}'_i = \mathbf{U}^{-1}(\mathbf{x}_i - \mathbf{t})$ (the new location of the symmetric point in $i(\mathbf{x})$) gives,

$$\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}'_i} = \mathbf{U}^\top \begin{bmatrix} i_{xx}(\mathbf{x}_i) & i_{yx}(\mathbf{x}_i) \\ i_{xy}(\mathbf{x}_i) & i_{yy}(\mathbf{x}_i) \end{bmatrix} \mathbf{U}.$$

Since $i(\mathbf{x})$ is defined to be symmetric around $\mathbf{x} = \mathbf{x}_i$, the derivative matrix in the above equation is a scalar matrix of unknown scalar a, simplifying the equation to,

$$\begin{aligned}\left.\frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^\top}\right|_{\mathbf{x}=\mathbf{x}_i'} &= a\mathbf{U}^\top\mathbf{U} \\ &= ak^2\mathbf{A}^\top\mathbf{R}(-\theta)\mathbf{R}(\theta)\mathbf{A} \\ &= ak^2\mathbf{A}^2. \end{aligned} \tag{5}$$

The Hessian therefore measures $ak^2\mathbf{A}^2$ and is covariant with $\mathbf{A}$. $\mathbf{A}$ may be computed by normalising the Hessian such that it has determinant one and positive diagonal elements and then taking the square root. The centre point, $\mathbf{x}_i'$, of the function must be known in order to measure its shape. The angle of rotation, θ, is not measured and the scale, k, of the original $\mathbf{U}$ cannot be recovered from the Hessian due to the presence of unknown scale factor, a (an unknown parameter of $i(\mathbf{x})$). Applying the inverse of $\mathbf{A}$ to $f(\mathbf{x})$ gives,

$$\begin{aligned} f\left(\mathbf{A}^{-1}\mathbf{x}\right) &= i\left(\mathbf{U}\mathbf{A}^{-1}\mathbf{x} + \mathbf{t}\right) \\ &= i\left(k\mathbf{R}(\theta)\mathbf{A}\mathbf{A}^{-1}\mathbf{x} + \mathbf{t}\right) \\ &= i\left(k\mathbf{R}(\theta)\mathbf{x} + \mathbf{t}\right), \end{aligned}$$

which is symmetric around $\mathbf{x} = k^{-1}\mathbf{R}^\top(\theta)(\mathbf{x}_i - \mathbf{t})$, because $i(\mathbf{x})$ is symmetric around $\mathbf{x} = \mathbf{x}_i$ and symmetry is scale, rotation and translation covariant.

3.2 Application to Image Data in Scale-Space

The above method of measuring the affine shape of a function using the Hessian effectively fits a second order surface to $f(\mathbf{x})$ at the shape centre point (and ignores the lower order components). Fitting a second order surface to a real image by evaluating the Hessian at a single point is not of practical use. The image features of interest are rarely smooth, second order functions and are contaminated by noise. The Hessian response will be dominated by noise and the finer scale frequency components of the feature. It is necessary to first isolate the feature of interest by removing finer scale information and noise. Furthermore the feature centre point must be known and must be chosen such that the shape is well defined (the eigenvalues of the Hessian must be significantly large). Characteristic scale feature selection in Gaussian scale space (Sect. 2.1) addresses these issues. Scale selection provides the feature size information and a feature detector such as the Determinant of Hessian detector automatically selects features with significant second order curvature. Note a possible confusion of terms: second order symmetry is invariant to scaling of the coordinate system (see the preceding section), but is not invariant to the application of the scale space operator, i.e. convolution with a Gaussian.

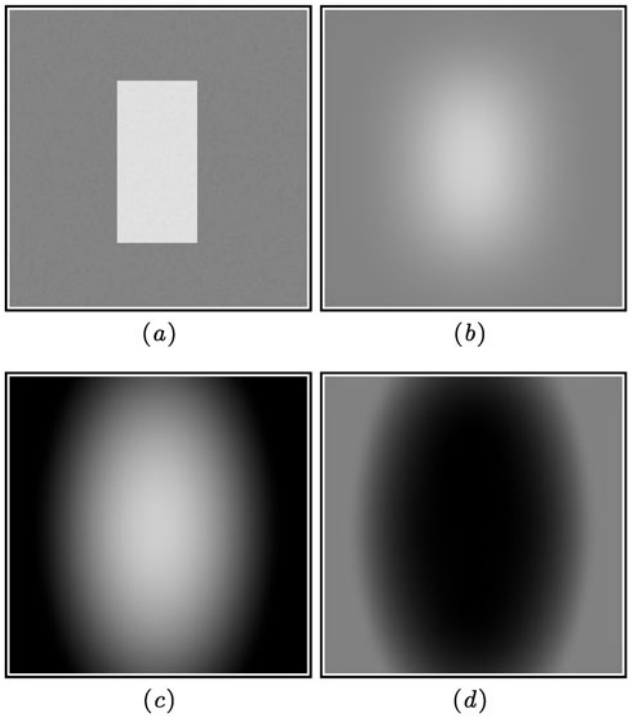

Fig. 1 Second order function approximation of an image feature in Gaussian scale space. (**a**) An example feature. (**b**) Characteristic scale image of (**a**). (**c**) A second order function approximating (**b**). (**d**) The difference between (**b**) and (**c**) (black equals zero). The approximation is a good local fit to the characteristic scale image of the feature

Let a scale-space function, $i(\mathbf{x}, \sigma) = g(\mathbf{x}, \sigma) * i(\mathbf{x})$, be symmetric at coordinates $(\mathbf{0}, \sigma_i)$. Translation, scaling and rotation are omitted from this argument, since it is assumed that the point of interest can be located by means of feature extraction. While seeking to recover the affine shape, it is not practical to model a distorted version of this function as,

$$f_a(\mathbf{x}, \sigma) = i(\mathbf{A}\mathbf{x}, \sigma) = g(\mathbf{A}\mathbf{x}, \sigma) * i(\mathbf{A}\mathbf{x}). \tag{6}$$

because the distorted Gaussian, $g(\mathbf{A}\mathbf{x}, \sigma)$, cannot be produced without knowing $\mathbf{A}$. Instead, a distorted version of $i(\mathbf{x})$ may be observed in scale-space as,

$$f_i(\mathbf{x}, \sigma) = g(\mathbf{x}, \sigma) * i(\mathbf{A}\mathbf{x}). \tag{7}$$

Convolving the function of interest with a Gaussian results in a function that is smoother. A good local approximation of $f_i(\mathbf{x}, \sigma)$ can be obtained by fitting a second order function, as is illustrated in Fig. 1. The second order parameters of the approximation can be obtained using the Hessian and the shape may be normalised as shown in Sect. 3.1. How this approach to shape estimation relates to the true shape of $f_i(\mathbf{x}, \sigma)$ is determined by how the Gaussian convolution affects the shape measurement using the Hessian.

The Hessian can be used to measure the covariance matrix of an affine Gaussian as shown briefly here and in more

detail in [30, 31]. Given a 2D affine Gaussian function with arbitrary gain k,

$$kg(\mathbf{x}, \Sigma) = \frac{k}{2\pi \det(\boldsymbol{\Sigma})} e^{\frac{-\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}}{2}},$$

the covariance matrix may be expressed as,

$$\boldsymbol{\Sigma} = \sigma_\alpha \boldsymbol{\Sigma}' = \sigma_\alpha \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix},$$

with $\det(\boldsymbol{\Sigma}') = 1$. The second order partial derivatives, the Hessian, of this function are,

$$\begin{aligned}
\frac{\partial^2 kg(\mathbf{x}, \boldsymbol{\Sigma})}{\partial x^2} &= kg(\mathbf{x}, \boldsymbol{\Sigma})\Big(-\frac{\sigma_{yy}}{\sigma_\alpha^2} \\
&\quad + \frac{1}{\sigma_\alpha^4}(x^2\sigma_{yy}^2 - 2xy\sigma_{yy}\sigma_{xy} + y^2\sigma_{xy}^2)\Big), \\
\frac{\partial^2 kg(\mathbf{x}, \boldsymbol{\Sigma})}{\partial y^2} &= kg(\mathbf{x}, \boldsymbol{\Sigma})\Big(-\frac{\sigma_{xx}}{\sigma_\alpha^2} \\
&\quad + \frac{1}{\sigma_\alpha^4}(x^2\sigma_{xy}^2 - 2xy\sigma_{xy}\sigma_{xx} + y^2\sigma_{xx}^2)\Big), \\
\frac{\partial^2 kg(\mathbf{x}, \boldsymbol{\Sigma})}{\partial x \partial y} &= kg(\mathbf{x}, \boldsymbol{\Sigma})\Big(\frac{1}{\sigma_\alpha^4}(-x^2\sigma_{xy}\sigma_{yy} \\
&\quad + xy(\sigma_{xx}\sigma_{yy} + \sigma_{xy}^2) - y^2\sigma_{xx}\sigma_{xy}) + \frac{\sigma_{xy}}{\sigma_\alpha^2}\Big).
\end{aligned}$$

Evaluating the Hessian at the origin gives,

$$\begin{bmatrix} \frac{\partial^2}{\partial x^2} & \frac{\partial^2}{\partial x y} \\ \frac{\partial^2}{\partial x y} & \frac{\partial^2}{\partial y^2} \end{bmatrix} kg(\mathbf{x}, \Sigma)\Big|_{\mathbf{x}=\mathbf{0}} = k\sigma_\alpha^{-2} \begin{bmatrix} -\sigma_{yy} & \sigma_{xy} \\ \sigma_{xy} & -\sigma_{xx} \end{bmatrix},$$

which is the negative inverse of the covariance matrix multiplied by an unknown scalar. This derivation is consistent with that in Sect. 3.1, since the Hessian evaluated at the centre of an isotropic Gaussian is a scalar matrix. Note that the scalar may be negative if the image intensity is negated. The sign of the eigenvalues do not have geometrical meaning and the measured Hessian should be normalised to have positive eigenvalues by multiplying the matrix with -1. The eccentricity measure defined in (4) can be redefined to ignore sign:

$$Q(\mathbf{M}) = \frac{|\lambda_0|}{|\lambda_1|}, \tag{8}$$

where $|\lambda_0|$ is the smallest magnitude eigenvalue of 2×2 matrix $\mathbf{M}$ and $|\lambda_1|$ is the largest magnitude eigenvalue.

An approximation to the characteristic scale observation, $f_i(\mathbf{x}, \sigma)$, of function $i(\mathbf{Ax})$ may therefore be computed by setting the covariance matrix of an affine Gaussian to $\Sigma_a = \sigma^2 \mathbf{H_0}^{-1}$, where $\mathbf{H_0}$ is the Hessian evaluated at $\mathbf{x} = \mathbf{0}$ with determinant normalised to 1. Since $f_i(\mathbf{x}, \sigma)$ was produced from $i(\mathbf{Ax})$ by convolving with an isotropic Gaussian, the shape measured using the Hessian is due to $\mathbf{A}$. However, convolution with a Gaussian has a significant effect on the shape measurement, which must be accounted for. The second central moment of $f_i(\mathbf{x}, \sigma)$ is the sum of that of $i(\mathbf{Ax})$ and $g(\mathbf{x}, \sigma)$,

$$\mu_2\big(f_i(\mathbf{x}, \sigma)\big) = \mu_2\big(i(\mathbf{Ax})\big) + \sigma \mathbf{I}.$$

The second central moment (and the covariance matrix, Σ_a) of the affine Gaussian approximation computed using the Hessian is affected in the same way. Comparing the eccentricities of $\mathbf{A}$ and the Hessian of $f_i(\mathbf{x}, \sigma)$, it is clear that,

$$\begin{aligned}
&Q(\Sigma_a) > Q(\mathbf{A}) && \text{for } Q(i(\mathbf{Ax})) < 1, \\
&Q(\Sigma_a) = Q(\mathbf{A}) = 1 && \text{for } \mathbf{A} = \mathbf{I}.
\end{aligned}$$

The combination of the Gaussian operator and the Hessian may therefore be seen as a local estimator of the covariance of the image intensity function (not the intensity distribution) that consistently underestimates the eccentricity of the covariance. The amount by which the eccentricity is underestimated decreases as the true eccentricity approaches symmetry.

These properties lend the problem to an iterative solution. The property that the error in the eccentricity measure decreases as eccentricity decreases guarantees convergence; The property that the eccentricity is consistently underestimated results in stability.

It is assumed that the coordinate frame is translated so that the point of interest is at the coordinate origin and scale, σ, has been selected. The proposed iterative algorithm repeats the following steps at each iteration,

$$f_j(\mathbf{x}) \leftarrow g(\mathbf{x}, \sigma) * i(\mathbf{A}\hat{\mathbf{A}}_j\mathbf{x}), \tag{9}$$

$$\mathbf{H} \leftarrow \frac{\partial^2 f_j}{\partial \mathbf{x} \partial \mathbf{x}^\top}\Big|_{\mathbf{x}=\mathbf{0}}, \tag{10}$$

$$\mathbf{V} \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \mathbf{V}^\top \leftarrow \text{eig}(\mathbf{H}), \tag{11}$$

$$q \leftarrow \frac{|\lambda_0|}{|\lambda_1|}, \tag{12}$$

$$d \leftarrow \sqrt{|\lambda_0 * \lambda_1|}, \tag{13}$$

$$\lambda_1 \leftarrow \left(\frac{|\lambda_0|}{d}\right)^{\frac{1}{4}}, \tag{14}$$

$$\lambda_0 \leftarrow \lambda_1^{-1}, \tag{15}$$

$$\mathbf{A}_u \leftarrow \mathbf{V} \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} \mathbf{V}^\top, \tag{16}$$

$$\hat{\mathbf{A}}_{j+1} \leftarrow \mathbf{A}_u \hat{\mathbf{A}}_j \mathbf{A}_u. \tag{17}$$

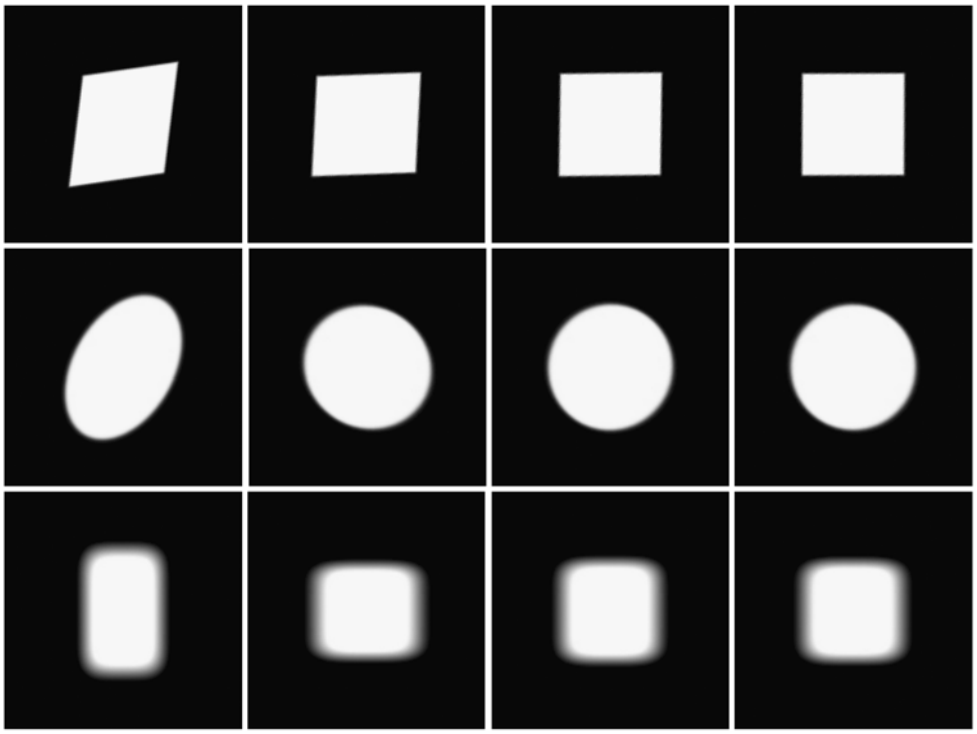

Fig. 2 Four iterations of the Hessian-based affine adaptation process applied to simple shapes. The *left column* displays the original asymmetric shapes. *Each following column* shows the shapes after another iteration of shape measurement and normalisation. These simple shapes approach a symmetric form after only three iterations

Here $i(\mathbf{Ax})$ is the given image function, $\mathbf{A}$ is the unknown affine shape (distortion) of symmetric function $i(\mathbf{x}, \sigma)$ and $\hat{\mathbf{A}}_j$ is the estimated inverse shape or normalisation transformation at iteration j (initially the identity matrix). Equation (9) transforms $i(\mathbf{Ax})$ by $\hat{\mathbf{A}}_j$ and applies the scale space operator. Equation (10) measures the Hessian at the point of interest, which gives an estimate of the inverse shape squared. The Hessian is decomposed into a symmetric eigensystem in (11). The eccentricity of the measurement is computed in (12). The process is completed when q is sufficiently close to 1. Equations (13) to (15) normalise the eigenvalues to be positive, the determinant to be 1 and computes the inverse fourth order root (the inverse square root of the Hessian is the shape estimate and the square root of the resulting matrix is required for equation 17). The update matrix is recomposed in (16). Equation (17) updates the shape estimate by composing it with the measured shape in the form of a quadric transformation.

Figure 2 shows the Hessian affine adaptation process applied to simple example shapes (left). The shape centre and scale were found automatically using a Determinant of Hessian feature extractor. After only a few iterations the shapes are normalised to be symmetric (right).

3.3 Comparison of the Complexity of the Hessian and Second Moment Operators

Both the Hessian matrix and the Second Moment matrix can be computed from an image by applying a series of filters to the image. The scale space operator and windowed integration operation can be implemented as rotationally symmetric Gaussian filters. These are most efficiently realised as separable recursive filters [32, 33]. The separable recursive filter complexity is linear in the number of image pixels and independent of the Gaussian scale. The differentiation operations can be implemented as 3×3 finite difference kernel filters shown in Fig. 3. Although it is possible to simplify the derivative kernels to 1×3 kernels, this would result in the rotational invariance of the operators being lost. The complexity of the derivative filters is linear in the number of image pixels. The main difference between computing the Hessian matrix and the Second Moment matrix is the arrangement of the filters.

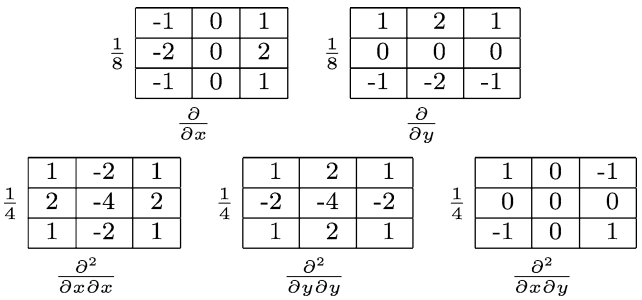

Fig. 3 Image derivative kernels

The Hessian operator consists of the following processing steps:

1. Filter to the feature scale, σ, with a Gaussian filter.
2. Compute second order partial derivatives (three finite difference filters).

The Second Moment operator consists of the following processing steps:

1. Filter to differentiation scale σ_D with a Gaussian filter.
2. Compute the first order partial derivative images, L_x and L_y, using finite difference filters.
3. Multiply the derivative images to produce images L_x^2, L_y^2 and $L_x L_y$.
4. Filter each derivative product image with a Gaussian with scale σ_I to compute windowed integration.

The SMM requires four applications of a Gaussian filter in two stages, where the Hessian only requires one application. The other differences in computational cost are relatively insignificant. The Hessian is therefore expected to require no more than a quarter of the computation cost of the SMM, given similar convergence characteristics.

4 Experimental Evaluation

In this section the Hessian and SMM are compared in terms of their effectiveness and efficiency in affine adaptation of local image features. Results for the MSER feature extractor are included for reference, since MSER is the most attractive alternative affine feature extractor.

4.1 Feature Extraction Framework

A modular affine feature extraction algorithm was implemented with interchangeable modules for computing the saliency map, feature scale response, affine shape adaptation and orientation selection. The algorithm is presented in Algorithm 1. Subroutines are labelled using Roman capital letters. The algorithm follows the common approach of multi-scale feature extraction, followed by scale selection and finally affine adaptation. This framework allows comparing alternative methods for affine shape estimation in a common feature extraction algorithm.

The first stage of the algorithm, in lines 1.3–1.8, extracts a set of multi-scale features by applying the saliency operator, T, to a discrete scale-space pyramid. The subroutine, $f \leftarrow \text{MAXIMA}(I(\mathbf{x},\sigma),\sigma)$, finds local maxima in image $I(\mathbf{x},\sigma)$ and produces feature vector, $f = \{\sigma, x_1, y_1, \sigma, x_2, y_2, \ldots, \sigma, x_n, y_n\}$. The scale response of each feature is computed in lines 1.9–1.12 by evaluating the scale response operator, S, at the coordinates of each feature.

Line 1.13 applies the scale selection method of [22, 23], by means of the subroutine, $f_A \leftarrow \text{SSFSS}(f_m)$. The resulting set of features, f_A, is the subset of f_m that are characteristic scale features.

1.1 $f_A \leftarrow \text{EXTRACT}(I_i(\mathbf{x}), \text{T}, \text{S}, \text{A}, \text{R}, n_a, Q_l, Q_h)$
Input:
$I_i(\mathbf{x})$—An image.
$I_t(\mathbf{x}) \leftarrow \text{T}(I_i(\mathbf{x}))$—The saliency operator.
$s \leftarrow \text{S}\left(I_{\sigma_f}(\mathbf{x}), \mathbf{x}_f, \sigma_f\right)$—The scale response operator.
$\{\mathbf{A}_u, q\} \leftarrow \text{A}(I_n(\mathbf{x}))$—The affine shape estimator, producing the update transformation, $\mathbf{A}_u$, and the eccentricity measure, q, according to equation 8.
$\theta \leftarrow \text{R}(I_n(\mathbf{x}))$—The orientation selector.
n_a—Affine adaptation iteration limit.
Q_l—Affine adaptation eccentricity lower limit.
Q_h—Affine adaptation eccentricity convergence limit.
Output:
$f_A = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_o\}$—A vector of o features with $\mathbf{p} = \{k, \theta, q, \phi, x, y\}$.
1.2 **begin**
1.3 $\sigma_p \leftarrow 2^k$, where k is a linear series with smallest value greater than 1.
1.4 **foreach** $\sigma \in \sigma_p$ **do**
1.5 $I_\sigma(\mathbf{x},\sigma) \leftarrow g(\mathbf{x},\sigma) * I_i(\mathbf{x})$.
1.6 $I_t(\mathbf{x},\sigma) \leftarrow \text{T}(I_\sigma(\mathbf{x},\sigma))$.
1.7 $f_m \leftarrow \{f_m, \text{MAXIMA}(I_t(\mathbf{x},\sigma),\sigma)\}$.
1.8 **end**
1.9 **foreach** $f = \{\sigma_f, x_f, y_f\} \in f_m$ **do**
1.10 $s_f \leftarrow \text{S}\left(I_p\left(\mathbf{x},\sigma_f\right), \mathbf{x}_f, \sigma_f\right)$.
1.11 $f \leftarrow \{\sigma_f, x_f, y_f, s_f\}$.
1.12 **end**
1.13 $f_A \leftarrow \text{SSFSS}(f_m)$.
1.14 **foreach** $f = \{\sigma_f, x_f, y_f, s_f\} \in f_A$ **do**
1.15 $\hat{q} \leftarrow 1$.
1.16 $\hat{\mathbf{A}} \leftarrow \mathbf{I}$.
1.17 $q_u \leftarrow 0.5$.
1.18 $j \leftarrow 0$.
1.19 **while** $(j < n_a) \cdot (q_u < Q_h) \cdot \left(\hat{q} > Q_l\right)$ **do**
1.20 $j \leftarrow j + 1$.
1.21 $I_n(\mathbf{x}) \leftarrow I_i\left(H\left(1, 0, \hat{q}, \hat{\phi}, x_f, y_f, \mathbf{x}\right)\right)$.
1.22 $I_n(\mathbf{x}) \leftarrow g\left(\mathbf{x}, \sigma_f\right) * I_n(\mathbf{x})$.
1.23 $\{\mathbf{A}_u, q_u\} \leftarrow \text{A}(I_n(\mathbf{x}))$.
1.24 $\hat{\mathbf{A}} \leftarrow \mathbf{A}_u \hat{\mathbf{A}} \mathbf{A}_u$.
1.25 $\left\{\hat{q}, \hat{q}^{-1}, \hat{\phi}\right\} \leftarrow \text{EIG}\left(\hat{\mathbf{A}}\right)$.
1.26 **end**
1.27 $I_n(\mathbf{x}) \leftarrow I_i\left(H\left(1, 0, \hat{q}, \hat{\phi}, x_f, y_f, \mathbf{x}\right)\right)$.
1.28 $\theta \leftarrow \text{R}(I_n(\mathbf{x}))$.
1.29 $f \leftarrow \{3\sigma_f, \theta, q_m, \phi_m, x_f, y_f\}$.
1.30 **end**
1.31 **end**

Algorithm 1: Generalised Feature Extraction Algorithm

Affine adaptation is performed in lines 1.14–1.30. Line 1.19 implements the loop termination conditions. The threshold Q_h is the convergence threshold, Q_l is used to reject excessively eccentric features (like those mistakenly detected on straight edges) and the number of iterations is limited to n_a. The inner loop is essentially the algorithm in (9) to (17), with (10) to (16) replaced by the generic affine update function, A. A normalised image is computed in lines 1.21 and 1.22 (H is defined in (1)). The shape is measured and the update transformation computed in line 1.22. The update is applied to the normalisation transformation in line 1.23. $\{e_1, e_2, \phi\} \leftarrow \mathrm{EIG}(\mathbf{M})$ computes the eigenvalues, e_1 and e_2, and the angle, ϕ, of the first eigenvector of 2×2 matrix M. Because $\hat{\mathbf{A}}$ has determinant one, its eigenvalues are reciprocally related.

4.2 Evaluation Method

Four feature extractors were implemented using the framework defined in Sect. 4.1, each using a different combination of estimation functions. These extractors are listed in Table 1. Figure 4 shows the output of the four extractors on corresponding sections of a pair of images. Note that the Hessian and second moment matrices produce different shapes for the same feature, but that the shape is none the less covariant between images. The MSER extractor [11] was also included in the evaluation. All extractors used the orientation selection method (R) and SIFT descriptor described in [20]. The adaptation iteration limit was set to $n_a = 8$, since it was found that a higher value does not significantly improve repeatability, but increased computation time. The eccentricity thresholds were set to $Q_l = 0.97$ and $Q_h = 0.05$.

Table 1 Test feature extractor configurations

Label	T	S	A
$\mathrm{He_h}$	Det. Hessian	Det. Hessian	Hessian
$\mathrm{He_s}$	Det. Hessian	Det. Hessian	SMM
$\mathrm{Ha_h}$	Harris	Laplacian	Hessian
$\mathrm{Ha_s}$	Harris	Laplacian	SMM

Two evaluation methods are employed to compare the above extractors. The first is the affine covariant regions performance evaluation and data described in [10]. The second evaluation method is the epipolar geometry computation task described in [34].

4.2.1 *Affine Covariant Regions Performance Evaluation*

The data and software for this evaluation are available from http://www.robots.ox.ac.uk/ vgg/research/affine/. The following metrics are measured:

1. Repeatability.
2. Correspondence count.
3. Match score.
4. Number of correct matches.
5. Efficiency.

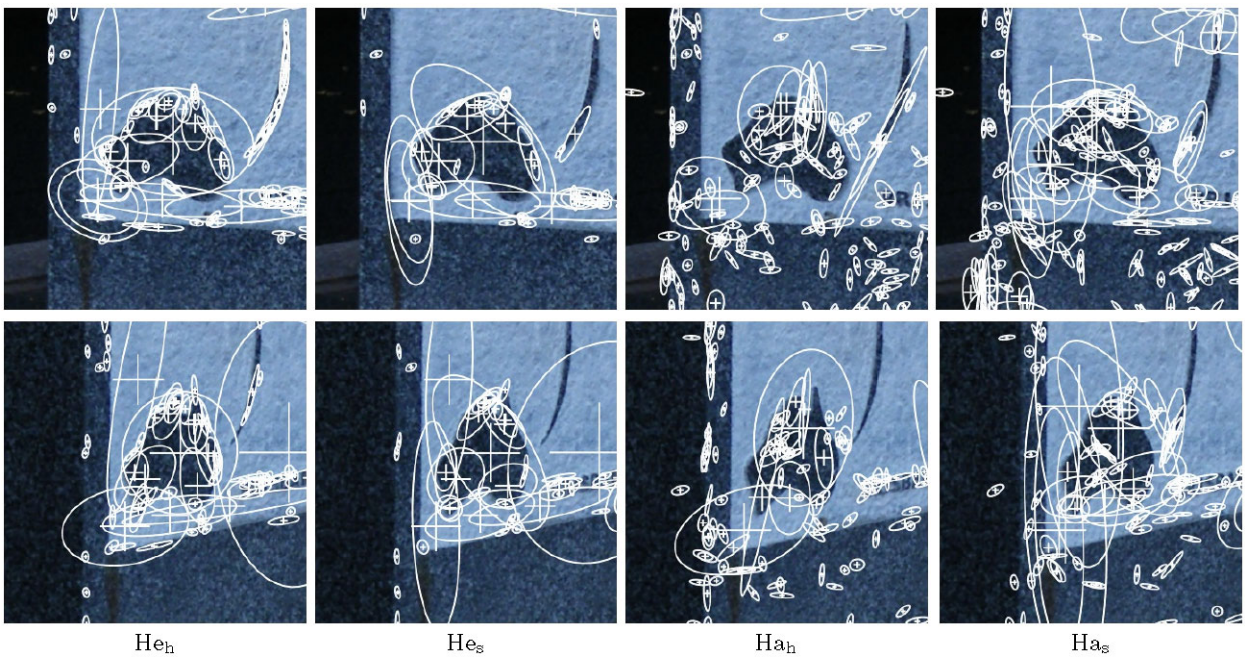

Fig. 4 Image sections showing the output of the evaluated extractors. *Each ellipse* indicates a feature. The feature normalisation transformation maps a *unit circle* to this *ellipse*

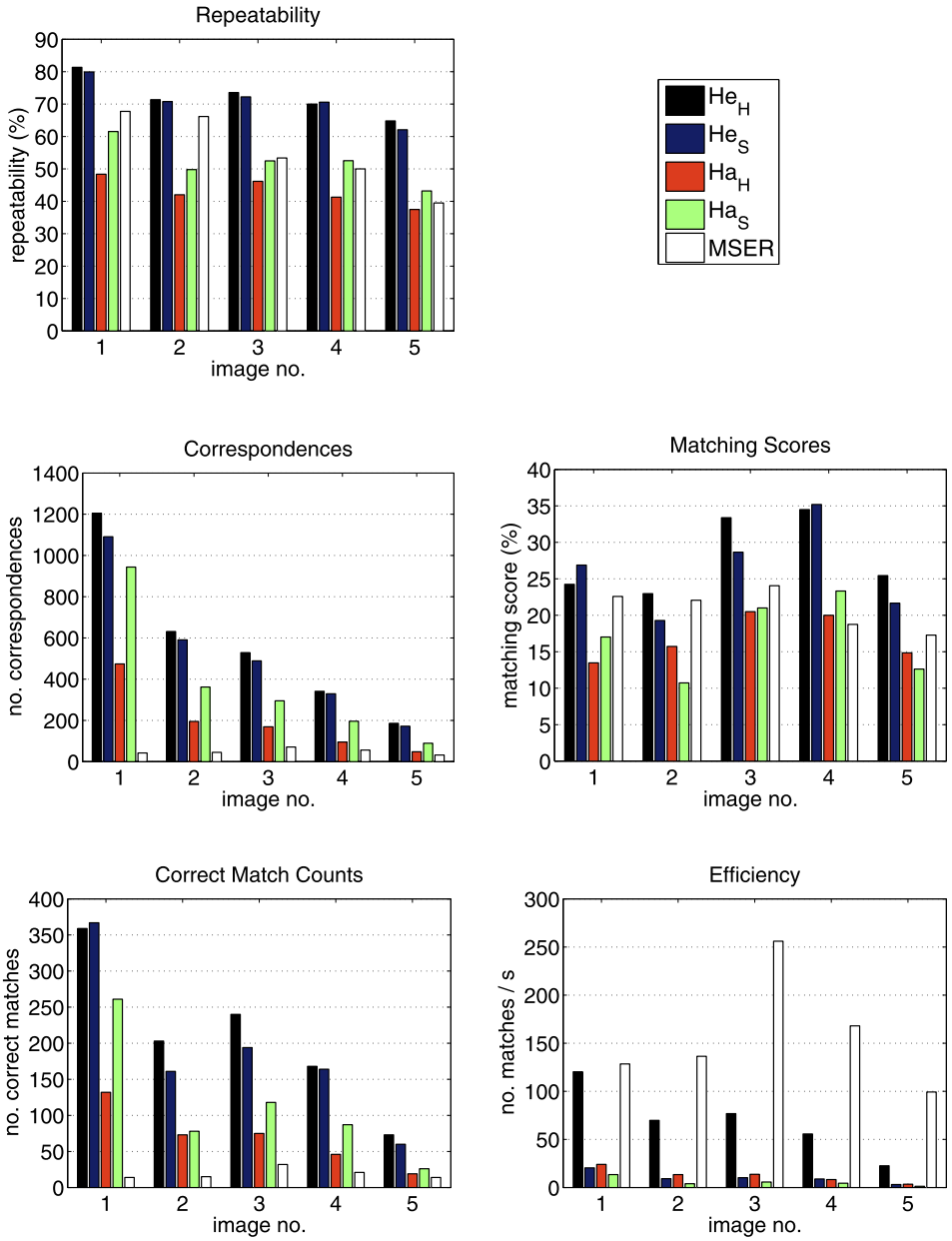

Fig. 5 Affine covariant regions performance evaluation results for dataset *bark* (varying scale)

The repeatability test functions by projecting features detected in a set of test image to a base image using ground truth homographies. If the overlap error between the projected feature and a feature in the base image is below 40%, the feature is considered repeated. The correspondence count is the number of features with overlap error below 40% and the repeatability score is the ratio of correspondences to the total number of features in the common image area. Features are matched using their descriptors and one-to-one nearest neighbour matching. Matches are deemed correct if the overlap error of a given match is below 40%. The matching score is the number of correct matches divided by the minimum number of features in the common part of the two images. The efficiency metric (not included in [10]) is expressed in terms of the rate at which matches are produced, $r = c_i / (t_i + t_0)$, where c_i is the number of correct matches between image 0 and test image i, and t_i and t_0 are the times taken to extract features from image i and 0, respectively.

4.2.2 *Epipolar Geometry Computation Task*

Computing the epipolar geometry is the first stage in many calibration algorithms. It can be used to constrain the search for further correspondences and can be used to generate a reconstruction of the scene and cameras with projective am-

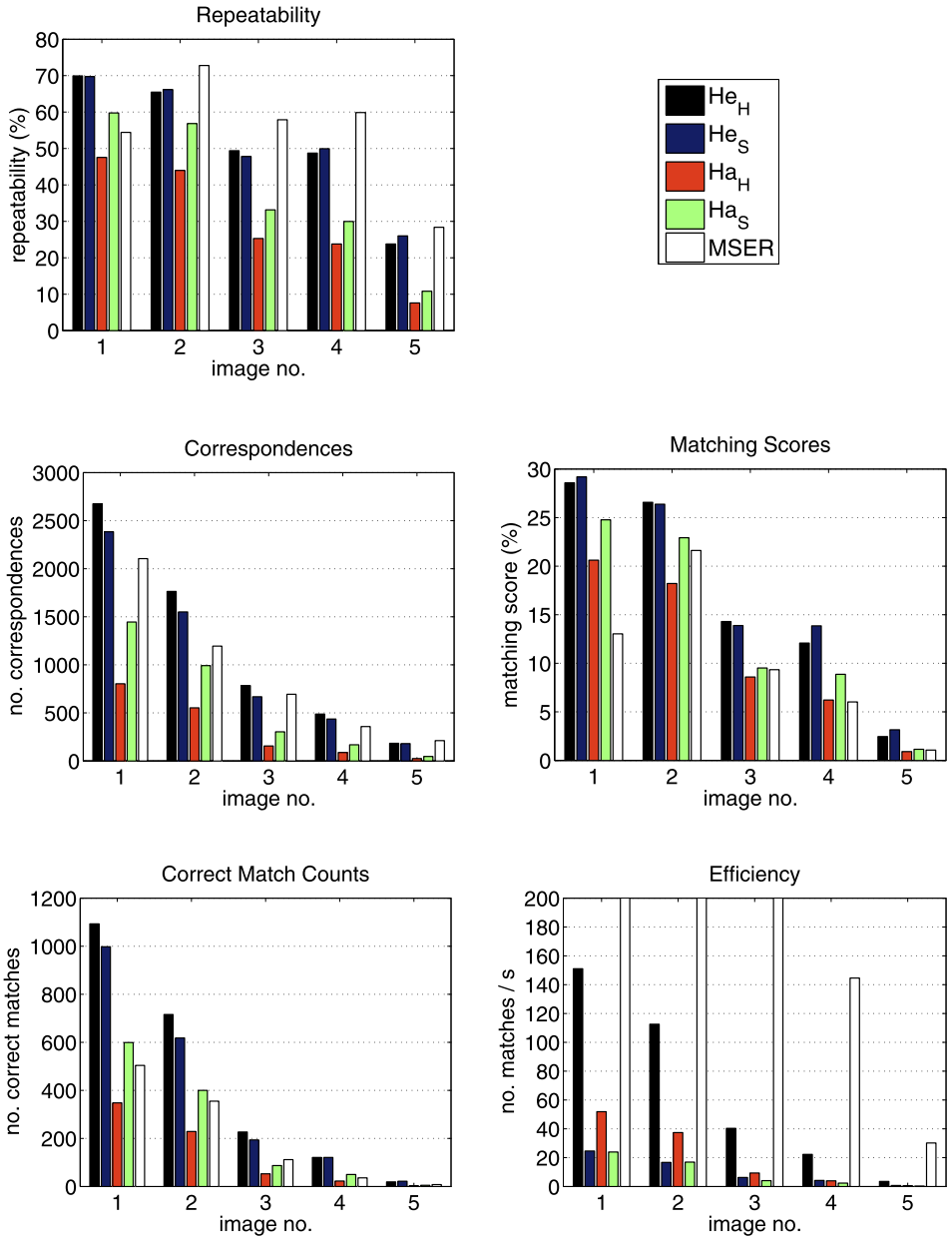

Fig. 6 Affine covariant regions performance evaluation results for dataset *boat* (varying scale)

biguity [1]. The epipolar geometry computation task evaluates how likely it is that a particular correspondence extraction technique will generate an epipolar geometry estimate that is sufficiently accurate to be used for practical applications.

Test data was acquired using a pair of digital cameras arranged to view a scene from widely separated views. Each set consists of images taken with the cameras in fixed position. The camera positions were varied between sets. The contents of the scene were altered for each pair of test images. Images were captured at high resolution (4.1 and 10 million pixel cameras were used) to compute the ground truth geometry. Test images were generated by scaling the original images to 640 × 480 pixel resolution. Scale factors were recorded for relating the test image geometry back to ground truth geometry. Scaling images to a low resolution removes a large proportion of the features from the images, ensuring that the task of computing the geometry of these test images is challenging.

The ground truth data consists of a large set of accurate point correspondences for each dataset. The error in a given estimate of the epipolar geometry is measured by computing the error of the set of ground truth correspondences when compared against the estimated geometry. The ground truth data was generated automatically from the high resolution images in the dataset and then mapped to the coordinates of

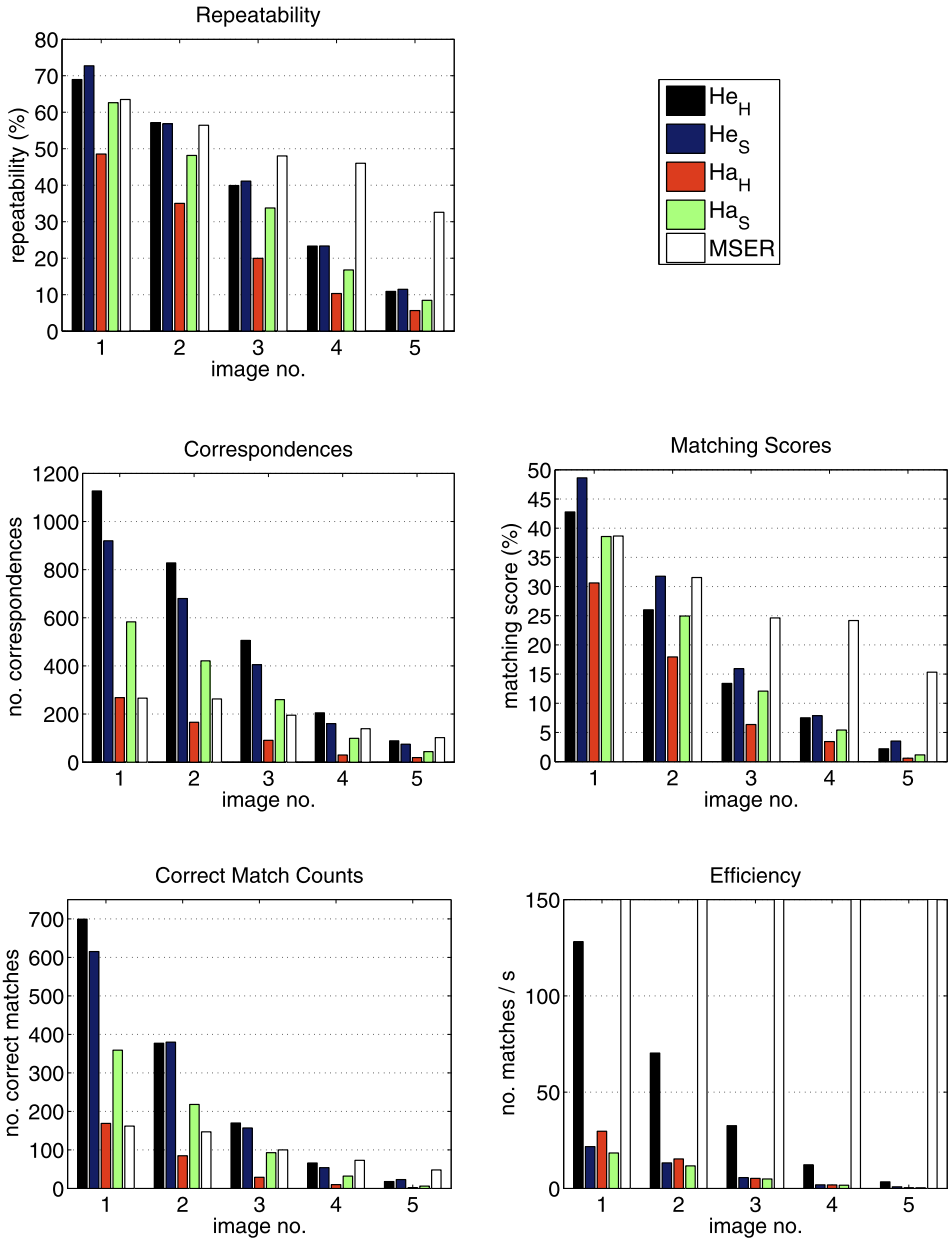

Fig. 7 Affine covariant regions performance evaluation results for dataset *graf* (varying view angle)

the test images. The following procedure was used to generate the ground truth data for each dataset:

1. Extract and match features across each high resolution image pair using the MSER feature extractor. The correspondences for all the image pairs in a dataset are collected into one large set of correspondences.
2. Compute an initial estimate of the epipolar geometry using RANSAC [35] and the normalised eight point algorithm [1, 36].
3. Match features again, this time using the initial geometry estimate to constrain matching so that all correspondences are inliers.
4. Apply the correspondence extraction algorithms defined in [34] to expand the inlier correspondences to a large number of dense, highly accurate correspondences and few outliers.
5. Compute a more accurate estimate of the epipolar geometry using RANSAC. The dense correspondences collected from the entire dataset are used.
6. The set of 10000 inlier correspondences with the lowest reprojection error are scaled to match the resolution of the test images and are kept as ground truth data.

The test procedure is designed to measure the success rates of different correspondence extraction systems when

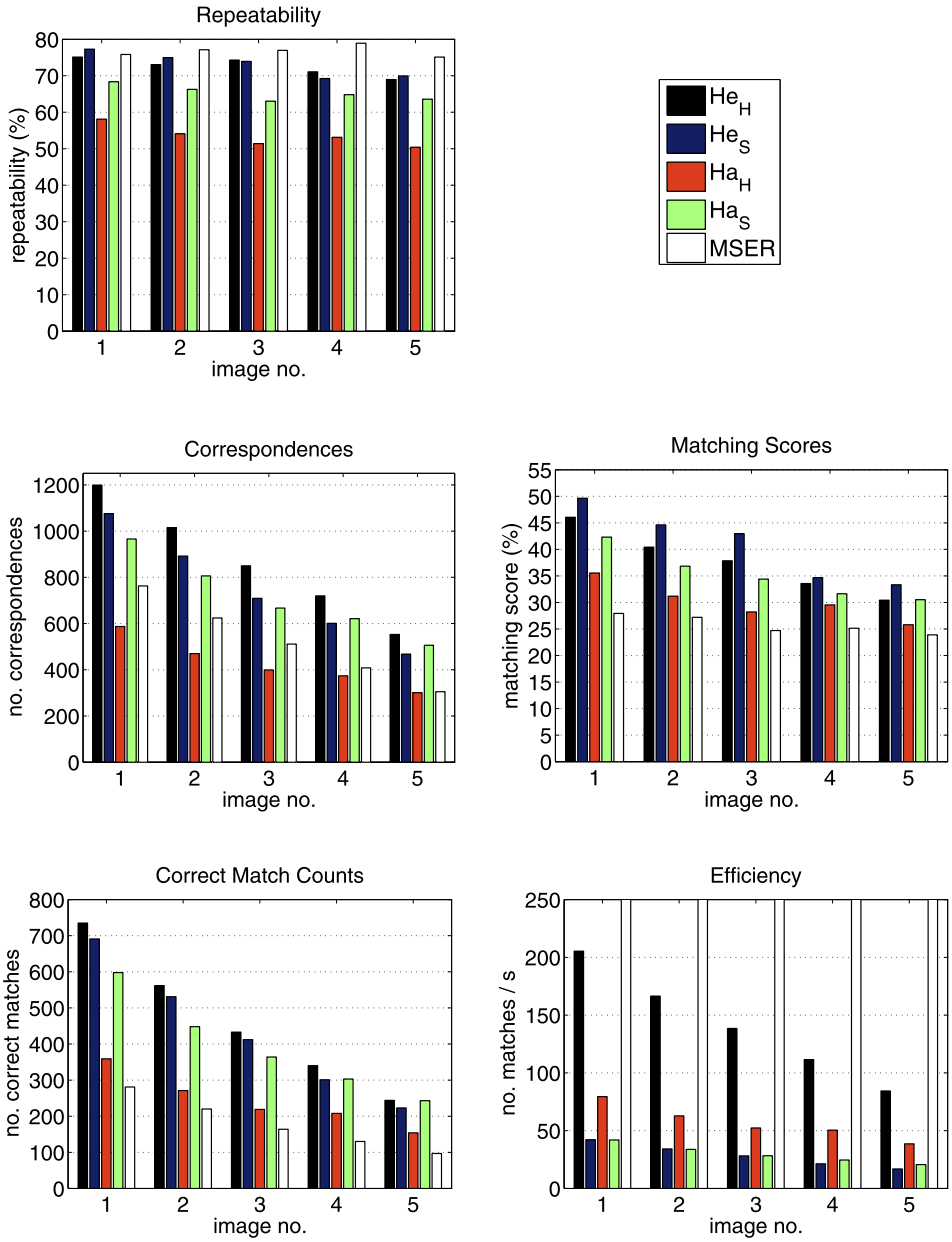

Fig. 8 Affine covariant regions performance evaluation results for dataset *leuven* (varying lighting)

applied to the task of estimating the epipolar geometry. Each test trial proceeds as follows:

1. Extract correspondences between one pair of images from the dataset using a particular extraction system.
2. Estimate the epipolar geometry of the image pair from the correspondences, using RANSAC and the normalised eight point method.
3. Compute the error of the epipolar geometry estimate as the Sampson distance [1] and the ground truth correspondences.
4. Compare the error to three thresholds, $t_1 = 4$, $t_2 = 16$, $t_3 = 64$ pixels squared. If the error is below a threshold, the model is considered sufficiently accurate for that threshold category and the trial is considered a success. The three thresholds are used to represent the precision requirements of three hypothetical users of the geometry estimate.

One hundred trials are run for each extractor and image pair combination and the average success rate is computed to compensate for the variability in the RANSAC method. Results are presented in terms of the average success rate for each dataset and for all datasets combined. Only the two datasets from [34] with the simplest geometry (nos. 3 and 4) were used for this test, because the other datasets are so challenging that they do not give useful results for the set of extractors tested here.

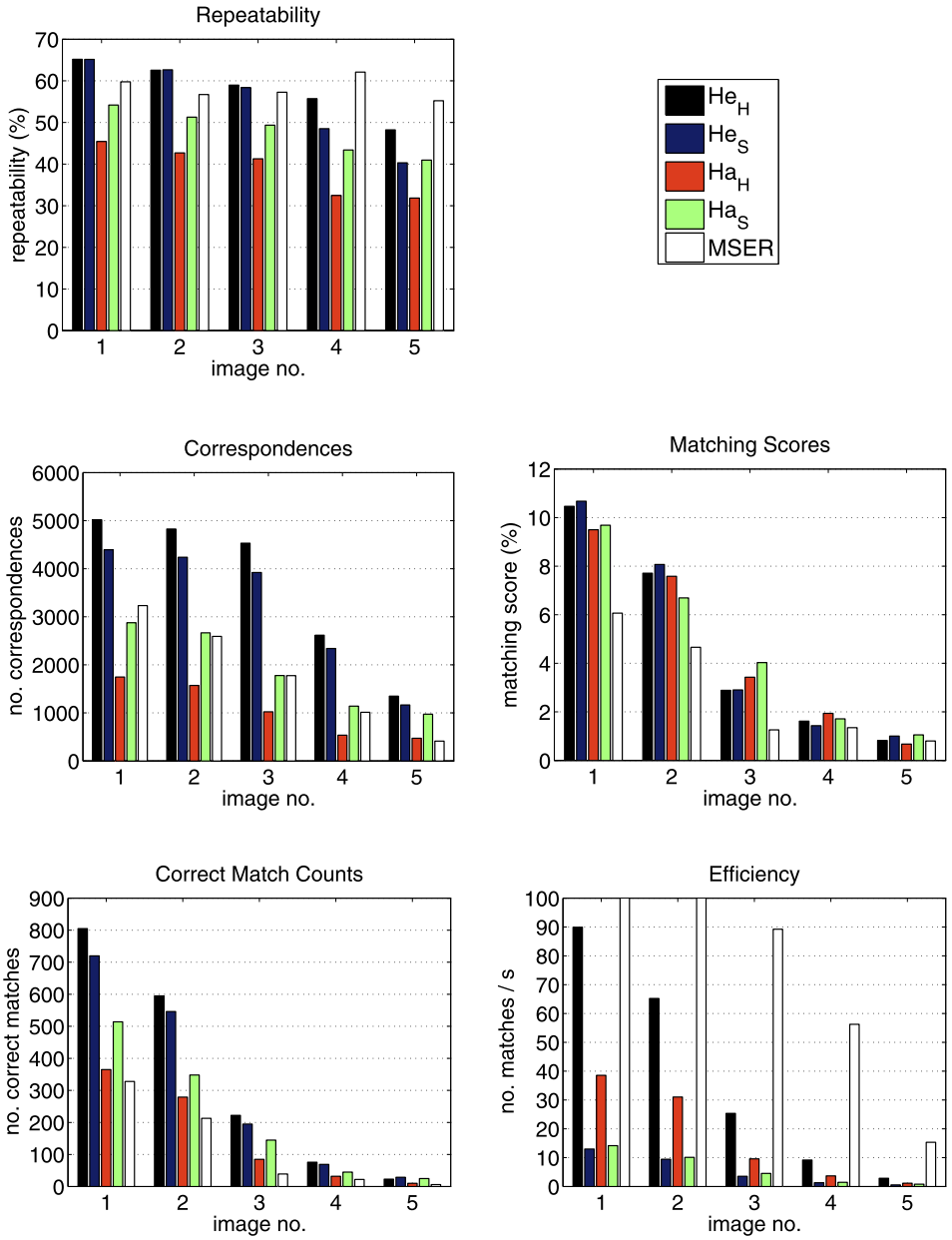

Fig. 9 Affine covariant regions performance evaluation results for dataset *trees* (varying focus or blur)

4.3 Results

Results of the affine covariant regions performance evaluation are presented in Figs. 5 to 10. Note that the efficiency plots have been scaled to highlight the relative performance of the gradient-based feature extractors that use affine adaptation. The efficiency of the MSER extractor is often out of scale as a result. Previous evaluations have shown that the MSER extractor can be more efficient than gradient-based extractors by as much as an order of magnitude.

A pair-wise comparison of the performance of the Hessian and SMM as shape estimators is presented in Table 2 (using the determinant of Hessian extractors) and Table 3 (using Harris extractors). Each table lists four columns of data. The first two list the mean scores for each extractor; The third lists the mean of the ratio between paired scores; The last column lists the results of a paired T-test.

Table 2 Mean repeatability results of $\mathrm{He_H}$, $\mathrm{He_S}$, the mean ratio and the Paired T-test result

	$\mathrm{He_H}$	$\mathrm{He_S}$	$\mathrm{He_H/He_S}$	p
Repeatability (%)	58.67	57.99	1.01	0.0680
Correspondences	1711.93	1564.13	1.12	0.0000
Matching score (%)	20.90	21.94	0.95	0.0295
Correct matches	492.67	467.80	1.05	0.0002
Efficiency (n/s)	95.93	16.34	5.92	0.0000

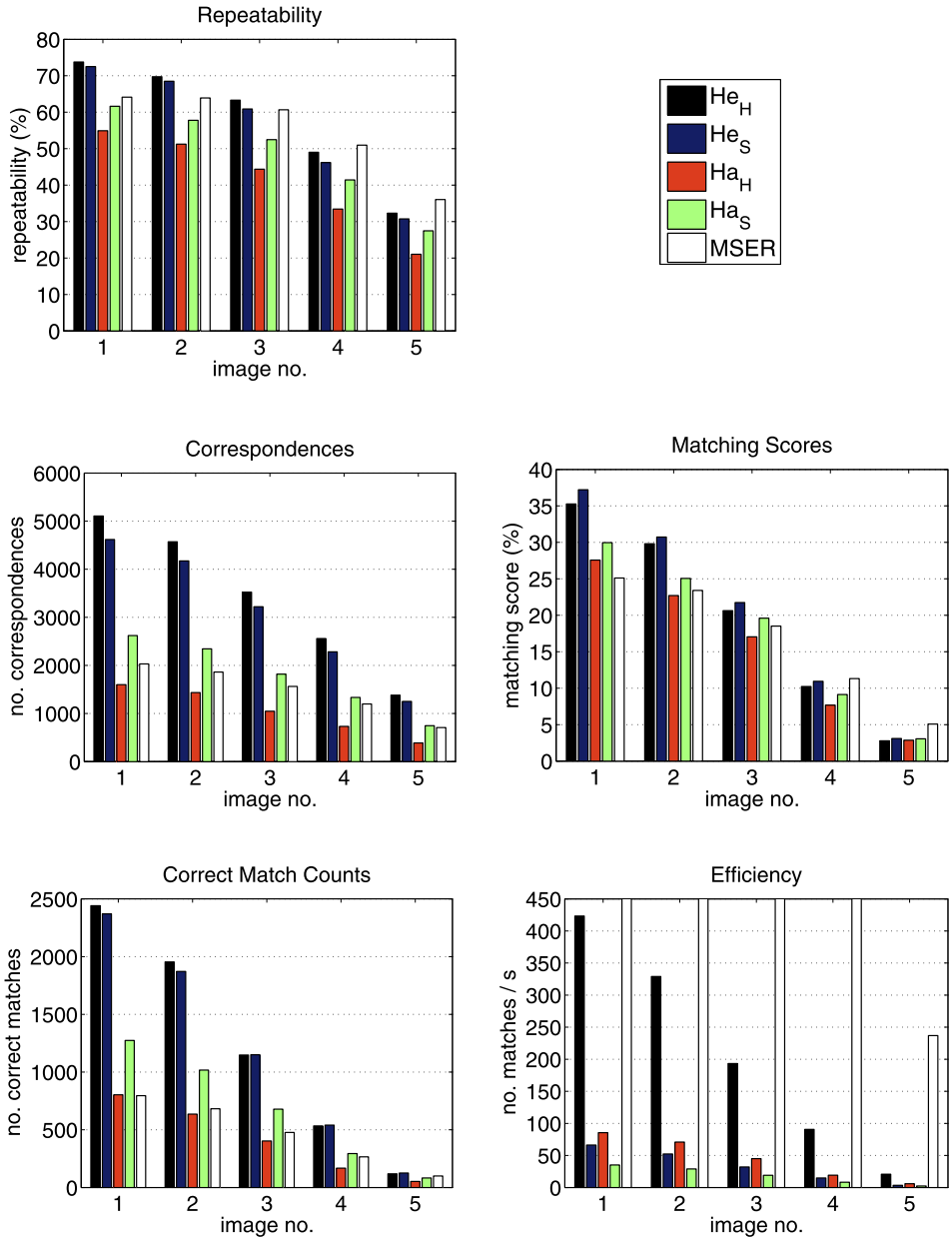

Fig. 10 Affine covariant regions performance evaluation results for dataset *wall* (varying view angle)

Table 3 Mean repeatability results of $\mathrm{Ha_H}$, $\mathrm{Ha_S}$, the mean ratio and the Paired T-test result

	$\mathrm{Ha_H}$	$\mathrm{Ha_S}$	$\mathrm{Ha_H/Ha_S}$	p
Repeatability (%)	38.30	47.53	0.79	0.0000
Correspondences	528.83	937.13	0.53	0.0000
Matching score (%)	14.98	17.33	0.87	0.0001
Correct matches	178.03	293.30	0.58	0.0000
Efficiency (n/s)	27.10	12.93	2.08	0.0000

Table 4 lists the results for the epipolar geometry computation task with the error threshold set to $t = 16$ pixels squared. The table lists the average number of successful epipolar geometry computation trials for each extractor in each dataset, as well as the average success rate over all datasets. Figure 11 shows the average success rate over all datasets at each of the tree error thresholds $t = 4$, $t = 16$ and $t = 64$.

4.4 Discussion

In terms of repeatability, matching score and the number of correct matches, the two determinant of Hessian extractors differ by 5% or less on average. Though the difference is statistically significant, it is practically small. The extractor using the Hessian to estimate affine shape ($\mathrm{He_H}$) produces approximately 10% more correspondences on average than

Table 4 Results for the epipolar geometry computation task. The average number of successful epipolar geometry computation trials with error threshold $t = 16$ are listed

Set	n_f	Ha_S	Ha_H	He_S	He_H	MSER
5	46	16.14	5.26	10.09	14.61	12.31
6	41	12.52	4.38	12.1	18.76	16.89
total	87	28.66	9.64	22.19	33.37	29.30
%		32.94	11.08	25.51	38.36	33.56

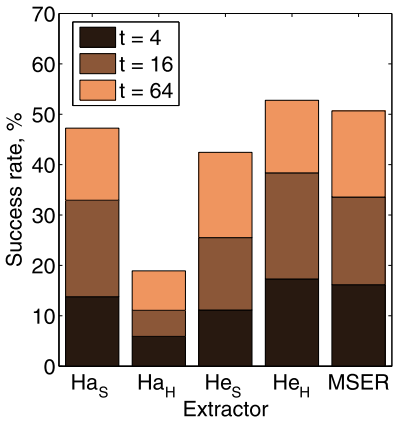

Fig. 11 Average success rates for the epipolar geometry computation task

the extractor using the SMM (He_S). This is likely due to the fact that the SMM is not necessarily well defined wherever the Hessian is well defined, so that the SMM-based adaptation of Hessian features fails to converge more often than Hessian-based adaptation of Hessian features. In terms of efficiency, He_H consistently outperforms He_S by a factor as high as 6.6 times and 5.9 times on average. The improvement in efficiency is higher than what is predicted by the operator complexity alone, and indicate that Hessian based adaptation converges more quickly than SMM based adaptation. The difference in performance between the two extractors is consistent across all types of tests. In the epipolar geometry estimation task, correspondences generated using He_H are successfully used to compute the epipolar geometry in approximately one and a half as many cases as correspondences generated using He_S. This indicates that the Hessian based shape adaptation is more likely to produce useful results in practical problems than the SMM based method.

These two Harris-based extractors produced significantly different results. The extractor using the Hessian to estimate affine shape (Ha_H) consistently achieved lower repeatability scores (19% reduction), matching scores (14% reduction) and produced approximately half the number of correspondences and correct matches on average. Despite lagging in the other metrics, the Ha_H extractor is on average twice as efficient at producing correct matches as the Ha_S extractor. In the epipolar geometry estimation task, using Ha_H produces useful results in only a third as many cases as Ha_S.

This poor performance of Hessian-based affine adaptation applied to Harris features may be attributed to the fact that the Harris operator is based on first order gradients—it does not guarantee that the second order gradients at the feature location are sufficiently strong for the Hessian based shape measure to be well defined (see property 5 in Section 3.1). The determinant of Hessian extractor, on the other hand, only selects points where the second order gradients (and hence the determinant of the Hessian) are large. The results indicate that the Hessian matrix is very effective and efficient in affine adaptation of determinant of Hessian features, but is much less effective for extractors that do not ensure high second order gradients.

The combination of the Hessian shape measure and Determinant of Hessian extractor produces a extractor superior to other gradient-based extractors, especially in terms of efficiency. The MSER extractor is still significantly more efficient than the Hessian-based extractor and is the better choice for very wide baseline matching problems. The Hessian-based extractor is a good choice when large numbers of correspondences are required.

5 Conclusion

In this paper it is shown theoretically and experimentally that the Hessian matrix can be used to estimate the affine shape of local image features using an iterative approach, similar to how the second moment matrix is commonly used. The Hessian is much more efficient in terms of processing time, compared to the SMM, and can lead to marginally improved feature quality and correspondence counts in combination with a suitable feature extractor.

A determinant of Hessian extractor that makes use of the Hessian matrix to estimate feature shape requires on average 5.9 times less processing time than the same extractor using the SMM, while exhibiting the equal or marginally superior performance in terms of repeatability, matching score and the number of correspondences and correct matches. The reduction in computation time is primarily attributed to the fact that fewer filter stages are required to compute the Hessian matrix. The Hessian matrix based method also leads to useful epipolar geometry estimates in 1.5 times the number of cases, compared to the SMM based method.

Using the Hessian for affine adaptation in combination with the Harris corner extractor resulted in a 19% reduction in repeatability, 47% fewer correspondences, 14% fewer correct matches and 22% fewer useful epipolar geometry estimates compared to the SMM. Despite the reduction in

correspondence counts, using the Hessian matrix still produced correspondences approximately twice as efficiently as the SMM.

It was found that the Hessian matrix provides is a useful and efficient method for measuring affine shape as long as it is applied to regions where high second order gradients are found, such as Determinant of Hessian features. Where second order gradients are not significantly strong, the Hessian cannot measure a well defined shape. The Harris corner detector, for example, selects points with high first order gradients, but does not ensure high second order gradients. Using the Hessian shape measure with this detector yields poor results.

Acknowledgements This project was supported by the Australian Research Council Linkage Grant no. LP0990135.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2003)
2. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. Comput. Vis. **59**(3), 207–232 (2004)
3. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
4. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. Int. J. Comput. Vis. **74**(1), 59–73 (2007)
5. Carbonetto, P., Dorkó, G., Schmid, C., Kück, H., de Freitas, N.: Learning to recognize objects with little supervision. Int. J. Comput. Vis. **77**(1), 219–237 (2008)
6. Snavely, N., Seitz, S., Szeliski, R.: Modeling the world from Internet photo collections. Int. J. Comput. Vis. **80**(2), 189–210 (2008)
7. Fookes, C., Denman, S., Lakemond, R., Ryan, D., Sridharan, S., Piccardi, M.: Semi-supervised intelligent surveillance system for secure environments. In: IEEE International Symposium on Industrial Electronics (ISIE), pp. 2815–2820 (2010)
8. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Found. Trends Comput. Graph. Vis. **3**(3), 177–280 (2008)
9. Kadir, T., Zisserman, A., Brady, J.M.: An affine invariant salient region detector. In: Proc. European Conference on Computer Vision, pp. 228–241 (2004)
10. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. Int. J. Comput. Vis. **65**(1-2), 43–72 (2005)
11. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. British Machine Vision Conference, vol. 1, pp. 384–393 (2002)
12. Moreels, P., Perona, P.: Evaluation of features detectors and descriptors based on 3d objects. Int. J. Comput. Vis. **73**(3), 263–284 (2007)
13. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proc. European Conference on Computer Vision, pp. 404–417 (2006)
14. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Proc. European Conference on Computer Vision. Lecture Notes in Computer Science, vol. 3951, pp. 430–443. Springer, Berlin (2006)
15. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. Alvey Vision Conference, pp. 189–192 (1988)
16. Beaudet, P.R.: Rotationally invariant image operators. In: Proc. International Joint Conference on Pattern Recognition, Kyoto, Japan, pp. 579–583 (1978)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
18. Lindeberg, T.: Scale-Space Theory in Computer Vision, 1st edn. Kluwer Academic, Boston (1994)
19. Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vis. **30**(2), 77–116 (1998)
20. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. International Conference on Computer Vision, Corfu, Greece, pp. 1150–1157 (1999)
21. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comput. Vis. **60**, 63–86 (2004)
22. Lakemond, R., McKinnon, D.N.R., Fookes, C., Sridharan, S.: A feature clustering algorithm for scale-space analysis of image structures. In: Proc. International Conference on Signal Processing and Communication Systems, pp. 186–192 (2007)
23. Lakemond, R.: Multiple camera management using wide baseline matching. Thesis (2010)
24. Lindeberg, T., Gårding, J.: Shape-adapted smoothing in estimation of 3-d depth cues from affine distortions of local 2-d brightness structure. In: Proc. European Conference on Computer Vision. Lecture Notes in Computer Science, vol. 800, pp. 389–400. Springer, Berlin (1994)
25. Baumberg, A.: Reliable feature matching across widely separated views. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 774–781 (2000)
26. Tuytelaars, T., Van Gool, L., D'haene, L., Koch, R.: Matching of affinely invariant regions for visual servoing. In: Proc. IEEE International Conference on Robotics and Automation, vol. 2, pp. 1601–1606 (1999)
27. Tuytelaars, T., Van Gool, L.: Wide baseline stereo matching based on local, affinely invariant regions. In: Proc. British Machine Vision Conference, Bristol, pp. 412–425 (2000)
28. van Gool, L., Tuytelaars, T., Turina, A.: Local features for image retrieval. In: Veltkamp, R.C., Burkhardt, H., Kriegel, H.-P. (eds.) State-Of-The-Art in Content-Based Image and Video Retrieval, pp. 21–41. Kluwer Academic, Dordrecht (2001)
29. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
30. Lakemond, R., Fookes, C., Sridharan, S.: Affine adaptation of local image features using the Hessian matrix. In: Proc. IEEE conference on Advanced Video and Signal Based Surveillance, Genoa, Italy, pp. 496–501 (2009)
31. Yuen, K.: Bayesian Methods for Structural Dynamics and Civil Engineering. Wiley, Singapore (2010)
32. Deriche, R.: Recursively implementing the Gaussian and its derivatives. Technical report, INRIA (April 1993)
33. Young, I.T., van Vliet, L.J.: Recursive implementation of the Gaussian filter. Signal Process. **44**(2), 139–151 (1995)
34. Lakemond, R., Fookes, C., Sridharan, S.: Dense correspondence extraction in difficult uncalibrated scenarios. In: Proc. Digital Image Computing: Techniques and Applications, Melbourne, Australia, pp. 53–60. IEEE Computer Society Conference Publishing Services, New York (2009)
35. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
36. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. In: Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, p. 61. Morgan Kaufmann, San Francisco (1987)

Ruan Lakemond is a research associate with the Image and Video Technologies group within the Faculty of Built Environment and Engineering and the Information Security Institute at QUT. He completed an Electrical and Computer Engineering degree in the Dean's Scholars program at QUT. His doctoral research focussed on image registration in difficult, wide baseline scenarios. His research interests include wide baseline matching, automatic calibration of surveillance camera networks, 3D reconstruction from surveillance video and analysis of thermal surveillance imagery.

Sridha Sridharan has a BSc (Electrical Engineering) degree and obtained an MSc (Communication Engineering) degree from the University of Manchester Institute of Science and Technology (UMIST), UK and a PhD degree in the area of Signal Processing from University of New South Wales, Australia. He is a Senior Member of the Institute of Electrical and Electronic Engineers—IEEE (USA). He is currently with the Queensland University of Technology (QUT) where he is a full Professor in the School of Engineering Systems. Professor Sridharan is the Deputy Director of the Information Security Institute and the Leader of the Research Program in Speech, Audio, Image and Video Technologies at QUT. He is the recipient of the QUT Outstanding Contributions to Research award in 2006, the Vice Chancellors award for Excellence in 2009 and the Outstanding Supervision of Higher Degree Students award in 2011. He has published over 300 papers consisting of publications in journals and in refereed international conferences in the areas of Image and Speech technologies during the period 1990–2011. During this period he has also graduated 25 PhD students as their Principal Supervisor in the areas of Image and Speech technologies. Prof. Sridharan has also received a number of research grants from various funding bodies including Commonwealth competitive funding schemes such as the Australian Research Council (ARC) and the National Security Science and Technology (NSST) unit. Several of his research outcomes have been commercialised.

Clinton Fookes is an Associate Professor with the Speech, Audio, Image and Video Technologies group within the Faculty of Built Environment and Engineering and the Information Security Institute within the Queensland University of Technology in Brisbane, Australia. He holds a BEng (Aerospace/Avionics) and a PhD in the field of computer vision. Clinton actively researches in the fields of computer vision and pattern recognition including biometrics, intelligent surveillance, airport security and operations. Clinton has attracted over $7.69M of cash funding from external competitive sources and has published over 90 internationally peer-reviewed articles. He is a member of professional organisations including the IEEE and the APRS. He is an Australian Institute of Policy and Science Young Tall Poppy and is an Australian Museum Eureka Prize winner.