

Automatic Estimation of Epipolar Geometry from Blob Features

Report LiTH-ISY-R-2620

Per-Erik Forssén, Anders Moe

Computer Vision Laboratory, Department of Electrical Engineering

Linköping University, SE-581 83 Linköping, Sweden

August 16, 2004

Abstract

This report describes how blob features can be used for automatic estimation of the fundamental matrix from two perspective projections of a 3D scene. Blobs are perceptually salient, homogeneous, compact image regions. They are represented by their average colour, area, centre of gravity and inertia matrix. Coarse blob correspondences are found by voting using colour and local similarity transform matching on blob pairs. We then do RANSAC sampling of the coarse correspondences, and weight each estimate according to how well the approximating conics and colours of two blobs correspond. The initial voting significantly reduces the number of RANSAC samples required, and the extra information besides position, allows us to reject false matches more accurately than in RANSAC using point features.

1 Introduction

Epipolar geometry is the geometry of two perspective projections of a 3D scene. For a thorough description of epipolar geometry, see [5]. For un-calibrated cameras epipolar geometry is compactly described by the *fundamental matrix* \mathbf{F} . A point $\mathbf{x} = (x_1 \ x_2 \ 1)^T$ in image 1, and the corresponding point $\mathbf{x}' = (x'_1 \ x'_2 \ 1)^T$ in image 2 are related through the fundamental matrix as

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0. \quad (1)$$

If we know \mathbf{F} , we can search for points \mathbf{x}' , corresponding to \mathbf{x} along its *epipolar line* \mathbf{l}

$$\mathbf{x}'^T \mathbf{l} = 0 \quad \text{where} \quad \mathbf{l} = \mathbf{F} \mathbf{x}. \quad (2)$$

Thus \mathbf{F} can be used as a constraint on the correspondences. Once we have a set of correspondences, we can use triangulation to compute a *projective reconstruction* [4].

With more knowledge about the cameras, we could compute distances to objects, and perform a metric scene reconstruction.

Computation of the fundamental matrix is possible using point correspondences alone, provided that not one of the following two *degenerate cases* is present

1. all scene points lie on a plane
2. The camera motion between the two cameras is a pure rotation.

These two situations cannot be told apart from the two projections alone, and since \mathbf{F} is not well defined in situation 2, estimation of \mathbf{F} is not possible [5]. When either of these situations occur, we should use a different model. In this report we use a *homography* \mathbf{H} , that relates corresponding points directly

$$h\mathbf{x}' = \mathbf{H}\mathbf{x} \quad \text{and} \quad g\mathbf{x} = \mathbf{H}^{-1}\mathbf{x}'. \quad (3)$$

1.1 Related work

Most work on estimation of epipolar geometry has involved correspondences between points, lines or conics [5], but recently work has started on using richer features, such as *affine invariant features* [8, 12], curve segments [10, 6] and the *scale invariant feature transform* (SIFT) features [1]. Wide baseline matching is a broad field of research, and we will make no attempt to cover all different approaches here. Instead we direct the reader to the journal paper [10], and the book [5]. We will however briefly describe two approaches similar to ours and outline the differences.

Tuytelaars and Van Gool [12] find affinely invariant regions in two ways, either as parallelogram shaped regions extended from corners and edges, or as elliptical regions found around local maxima in the intensity profile. Each region is then described by its shape (parallelogram or ellipse) and 18 moment invariants computed from *generalised colour moments* in the region.

Obdrzalek and Matas [8] construct *local affine frames* from *maximally stable extremal regions* (MSER) on intensity images. An MSE region is defined as being darker/brighter than all pixels on the region boundary. This allows MSER features to be invariant to monotonic transformations of image intensity, and avoids using an intensity similarity threshold.

The method introduced in this report is similar to the affine invariant region approaches described above in the sense that it uses region based features to estimate the epipolar geometry. We will however make use of the region colour in the detection, and apply recent results on conic matching [10, 6, 9] to estimate the geometry. The features used will be introduced in section 2, and the similarities and differences to the two methods above will be outlined.

1.2 Algorithm overview

The algorithm introduced in this report consists of the following steps:

1. Compute features in the two images to match, see section 2.

2. Find a set of likely correspondences using colour and a local similarity transform voting scheme. The colour constraint is described in section 3. The local similarity transform voting is described in section 4.
3. Find correct correspondence using a global homographic or epipolar constraint. This is done using RANSAC sampling of the correspondences found in step 2. Utilisation of the two geometry constraints for ellipses is described in sections 5, 6, and 7.

2 Blob features

We will make use of blob features extracted using a clustering pyramid built using robust estimation in local image regions [3]. The current implementation processes 360×288 RGB images at a rate of 1 sec/frame on a Intel P3 CPU at 697 MHz, and produces relatively robust and repeatable features in a wide range of sizes. Each extracted blob is represented by its average colour \mathbf{p}_k , area a_k , centroid \mathbf{m}_k , and inertia matrix \mathbf{I}_k . I.e. each blob is a 4-tuple

$$\mathcal{B}_k = \langle \mathbf{p}_k, a_k, \mathbf{m}_k, \mathbf{I}_k \rangle .$$

Since an inertia matrix is symmetric, it has 3 degrees of freedom, and we have a total of $3+1+2+3 = 9$ degrees of freedom for each blob. For more information about the feature estimation, please refer to [3], and the implementation [11], both of these are available on-line. After estimation, all blobs with approximating ellipses partially outside the image are discarded.

The blob estimation has two main parameters: a colour distance threshold d_{\max} , and a propagation threshold c_{\min} for the clustering pyramid. For the experiments in this report we have used $d_{\max} = 0.16$ (RGB values in interval $[0, 1]$) and $c_{\min} = 0.5$.

The number of blobs in an image depends heavily on image content and resolution. We typically get between 70 and 350 blobs in each image. Figure 1 shows two images where 107 and 120 blobs have been found.



Figure 1: Blob features in an aerial image. Left to right: Input image 1, Input image 1 with detected features (107 blobs), Input image 2 with detected features (120 blobs).

The blob features are related to the MSER features used in [8] in the sense that they are compact and can be nested. Differences are that our features utilise the colour

information in the image, and the regions are not required to be darker/brighter than all neighbours as is the case for MSER regions. Instead we find stable regions using a hierarchical clustering scheme, where pixels that are similar in colour and position are grouped. When the intensity profile looks like a staircase, we can still detect the steps although they have brighter pixels on one side and darker pixels on the other. An advantage with the MSER approach however is that no grouping distance threshold is necessary.

Our approach is also related to the regions used by Tuytelaars and Van Gool [12], but our features are detected using colour instead of intensity. Moreover, we detect features in a wide range of scales, whereas the detection method in [12] uses a single smoothing scale, which in effect limits the sizes of the detected regions. Our colour comparison is less advanced though, since we use a similarity measure on the average RGB values in the region, instead of computing the 18 invariants used in [12].

3 Colour constraint

We will use a voting scheme to find an initial set of correspondences between the two images. A potential correspondence $\mathcal{B}_i \leftrightarrow \mathcal{B}'_j$ can quickly be discarded by making use of the colour parameter of the blobs. Thus we compute the colour distances of all blobs in image 1 and all blobs in image 2, and use them to define a correspondence matrix \mathbf{M} . We set $M_{ij} = 1$ whenever

$$(\mathbf{p}_i - \mathbf{p}'_j)^T \mathbf{W} (\mathbf{p}_i - \mathbf{p}'_j) \leq 1 \quad (4)$$

and $M_{ij} = 0$ otherwise. The matrix \mathbf{W} defines the colour space metric. We have used $\mathbf{W} = \mathbf{T}^T \text{diag}^{-2}[\mathbf{d}]\mathbf{T}$ where

$$\mathbf{T} = \frac{1}{255} \begin{pmatrix} 65.4810 & 128.5530 & 24.9660 \\ -37.7970 & -74.2030 & 112 \\ 112 & -93.7860 & -18.2140 \end{pmatrix} \quad \text{and} \quad \mathbf{d} = \begin{pmatrix} 0.18 \\ 0.05 \\ 0.05 \end{pmatrix}. \quad (5)$$

The matrix \mathbf{T} is the standard mapping from RGB to the YCbCr colour space (as defined in ITU-R BT.601) for RGB values in interval $[0, 1]$. The vector \mathbf{d} thus contains scalings for the Y, Cb, and Cr components respectively. The purpose of this scaling is mainly to reduce the sensitivity to differences in illumination. Typically \mathbf{M} will have a density of about 15%, and thus this simple operation does a good job at reducing the correspondence search space.

As a side note, we have observed that similar quality in results can be obtained by setting $\mathbf{W} = \alpha \mathbf{C}^{-1}$ where \mathbf{C} is the covariance of the colour of a large set of blobs. This is known as the *Mahalanobis metric*. Using the Mahalanobis metric could be seen as a way towards automatic adaption to new data-sets, since the tuning of the vector \mathbf{d} now is replaced with tuning of a scalar α .

For more advanced approaches to colour matching, see e.g. [7, 12].

4 Initial correspondences

Any pair of points in image 1 can be mapped to any pair in image 2 using a *similarity transform*. In homogeneous coordinates, a similarity transform looks like this

$$\mathbf{x}' = \begin{pmatrix} s\mathbf{R} & t \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{x}. \quad (6)$$

We now generate blob pairs in both images, by joining together spatially adjacent blobs. Each blob gets to form ordered pairs with its three nearest neighbours. Thus, if we had N_1 and N_2 blobs in the two images, we now have $3N_1$ and $3N_2$ blob pairs. We will now try to find correspondences of such blob pairs, i.e. $\langle \mathcal{B}_i, \mathcal{B}_k \rangle \leftrightarrow \langle \mathcal{B}'_j, \mathcal{B}'_l \rangle$. For each such correspondence, we first check if the colours match, using \mathbf{M} (see section 3). This excludes most candidates. For the correspondences that match, we then calculate the similarity mapping (6) from the blob centroids. We then transform both blobs in the pair through the mapping, and compute their shape distance

$$d_{ij}^2 = \frac{\|\mathbf{I}_i - \tilde{\mathbf{I}}'_j\|^2}{\|\mathbf{I}_i\|^2 + \|\tilde{\mathbf{I}}'_j\|^2} \quad \text{where} \quad \tilde{\mathbf{I}}'_j = s^2 \mathbf{R} \mathbf{I}'_j \mathbf{R}^T. \quad (7)$$

Both distances are summed and added in a new correspondence matrix \mathbf{S} according to

$$S_{ij} + e^{-(d_{ij}^2 + d_{kl}^2)/\sigma_s^2} \mapsto S_{ij} \quad (8)$$

$$S_{kl} + e^{-(d_{ij}^2 + d_{kl}^2)/\sigma_s^2} \mapsto S_{kl} \quad (9)$$

where σ_s is a shape distance scaling.

This implements a soft voting scheme, where very few constraints on the image structure have been imposed. A set of potential candidate correspondences $\mathcal{B}_i \leftrightarrow \mathcal{B}'_j$ are now extracted from \mathbf{S} by requiring that the position S_{ij} should be a maximum along both row i and column j . The result of this operation for a pair of aerial images is shown in figure 2. Roughly half of the correspondences are correct in this case.

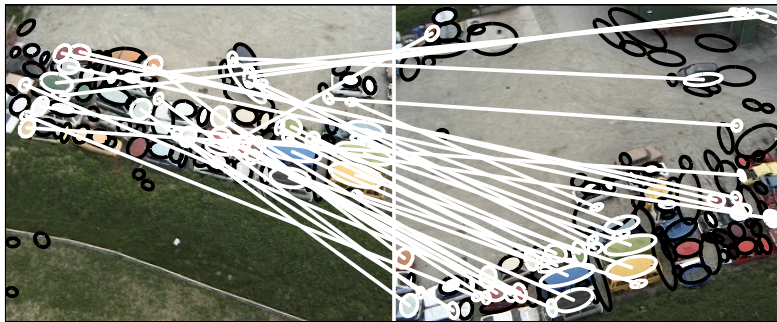


Figure 2: Raw correspondences found using voting. Roughly 50% of the correspondences are correct. Unmatched blobs are painted black.

5 RANSAC estimation of geometry

We now improve the quality of correspondences using outlier rejection with RANSAC, see e.g. [5]. We draw a random subset of the correspondences (4 for homography estimation, and 8 for fundamental matrix estimation), and estimate the applicable mapping (\mathbf{F} or \mathbf{H}) using the centroids only, i.e. $\mathbf{m}_i \leftrightarrow \mathbf{m}'_j$. Note that the centroid is only an affine invariant, not a projective one, and thus this estimation contains a model error. In practise this means that even if the chosen correspondences are in perfect alignment, there will be a slight bias in the estimated mapping (\mathbf{F} or \mathbf{H}). Once we have a set of correspondences, we could however remove this bias with an iterative refinement using correspondences of conics, e.g. using the method in [6]. This has however *not* been tested yet.

For each candidate mapping (\mathbf{F} or \mathbf{H}), we now verify all correspondences that are valid according to \mathbf{M} (see section 3) with respect to projection error of both blob position and shape. The projection error for the homography is derived in section 6, and for the fundamental matrix in section 7.

The decision on how many random samples to draw is often made according to

$$N = \frac{\log(1 - m)}{\log(1 - (1 - \epsilon)^K)} \quad (10)$$

where K is the number of correspondences needed for the estimation of \mathbf{F} or \mathbf{H} , m is the required probability of picking one inlier only sample after N tries (typically set to 0.99), and ϵ is the probability of picking an outlier correspondence [5]. For the example in figure 2, we may set $\epsilon = 0.5$, and $K = 4$, and get $N = 72$.

As noted in [2], there are problems with (10). The obtained N is for instance an underestimate of the actual number of samples needed if the data also has inlier noise.

6 Homography constraint

For a homography it is possible to further constrain which correspondences are allowed as inliers by mapping the ellipse shape through the homography, and rejecting the correspondence if the shape distance, i.e. the quotient in (7), is above a threshold.

We will now derive a homography transformation of an ellipse. Note that even though an ellipse mapped through a homography is a new ellipse, this mapping is merely approximately correct for regions which are not elliptical.

A blob \mathcal{B} in image 1, represented by its centroid \mathbf{m} and inertia \mathbf{I} approximates an image region by a new ellipse shaped region with the outline

$$\mathbf{x}^T \mathbf{C} \mathbf{x} = 0 \quad \text{for} \quad \mathbf{C} = \frac{1}{4} \begin{pmatrix} \mathbf{I}^{-1} & -\mathbf{I}^{-1} \mathbf{m} \\ -\mathbf{m}^T \mathbf{I}^{-1} & \mathbf{m}^T \mathbf{I}^{-1} \mathbf{m} - 4 \end{pmatrix} \quad (11)$$

see [3]. This equation is called the *conic form* of the ellipse. To derive the mapping to image 2, we will express the ellipse in *dual conic* form [5]. The dual conic form defines a conic in terms of its tangent lines $\mathbf{l}^T \mathbf{x} = 0$. For all tangents \mathbf{l} in image 1 we have

$$\mathbf{l}^T \mathbf{C}^* \mathbf{l} = 0, \quad \mathbf{C}^* = \begin{pmatrix} 4\mathbf{I} - \mathbf{m}\mathbf{m}^T & -\mathbf{m} \\ -\mathbf{m}^T & -1 \end{pmatrix} \quad (12)$$

where \mathbf{C}^* is the inverse of \mathbf{C} in (11). A tangent line \mathbf{l} and its corresponding line \mathbf{l}' in image 2 are related according to $\mathbf{l} = \mathbf{H}^T \mathbf{l}'$. This gives us

$$\mathbf{l}'^T \mathbf{H} \mathbf{C}^* \mathbf{H}^T \mathbf{l}' = 0 \quad \text{where we set} \quad \mathbf{H} \mathbf{C}^* \mathbf{H}^T = \begin{pmatrix} \mathbf{B} & \mathbf{d} \\ \mathbf{d}^T & e \end{pmatrix}. \quad (13)$$

We recognise the result of (13) as a new dual conic form

$$-e \mathbf{l}'^T \begin{pmatrix} 4\tilde{\mathbf{I}} - \tilde{\mathbf{m}}\tilde{\mathbf{m}}^T & -\tilde{\mathbf{m}} \\ -\tilde{\mathbf{m}}^T & -1 \end{pmatrix} \mathbf{l}' = 0. \quad (14)$$

This allows us to identify $\tilde{\mathbf{m}}$ and $\tilde{\mathbf{I}}$ as

$$\tilde{\mathbf{m}} = \mathbf{d}/e \quad \text{and} \quad \tilde{\mathbf{I}} = (-\mathbf{B}/e + \tilde{\mathbf{m}}\tilde{\mathbf{m}}^T)/4. \quad (15)$$

Note that since this mapping only involves additions and multiplications it can be implemented very efficiently. The mapping of blob shapes through the homography is illustrated in figure 3 (left).

To rank the correspondences we will use the spatial projection error

$$r_{ij}^2 = \|\mathbf{m}_i - \tilde{\mathbf{m}}'_j\|^2 + \|\tilde{\mathbf{m}}_i - \mathbf{m}'_j\|^2 \quad (16)$$

and the shape distance

$$s_{ij}^2 = \frac{\|\mathbf{I}_i - \tilde{\mathbf{I}}'_j\|^2}{\|\mathbf{I}_i\|^2 + \|\tilde{\mathbf{I}}'_j\|^2} + \frac{\|\tilde{\mathbf{I}}_i - \mathbf{I}'_j\|^2}{\|\tilde{\mathbf{I}}_i\|^2 + \|\mathbf{I}'_j\|^2} \quad (17)$$

to fill in a new correspondence matrix \mathbf{S} according to

$$S_{ij} = e^{-r_{ij}^2/\sigma_r^2} e^{-s_{ij}^2/\sigma_s^2}. \quad (18)$$

A correspondence is considered valid if the correspondence matrix \mathbf{S} has a maximum along both row i and column j , and furthermore is larger than some threshold $s_{\min} = 0.5$. (The scalings are at present set to $\sigma_r = 10$ and $\sigma_s = 0.75$.)

For each random subset of correspondences, we use the number of correspondences as a measure of how good the generated mapping was, and choose the one with most correspondences as the correct one.

As soon as we find a solution with at least 15 correspondences, we leave the RANSAC loop, and instead do a local optimisation¹. Since we have some inlier noise, the local optimisation significantly reduces the number of RANSAC samples (cf. [2]). The final solution is shown in figure 3. Running the algorithm 1000 times gives the average number of RANSAC samples as 40.7 and a 99.8% success rate. By comparing the final (figure 3) and the initial (figure 2) correspondences, we get $\epsilon = 22/39 \approx 0.5641$. Inserting this in (10), gives 125.2 samples for 99% success rate. The reason for the ‘‘better than theoretical’’ result on this image pair is that several outlier contaminated samples give an initial \mathbf{H} that is close enough to the correct solution for the local optimisation to succeed.

¹In the local optimisation we compute a new \mathbf{H} from the inlier correspondences, check for correspondences, compute a new \mathbf{H} using them and so on until \mathbf{H} stops changing.

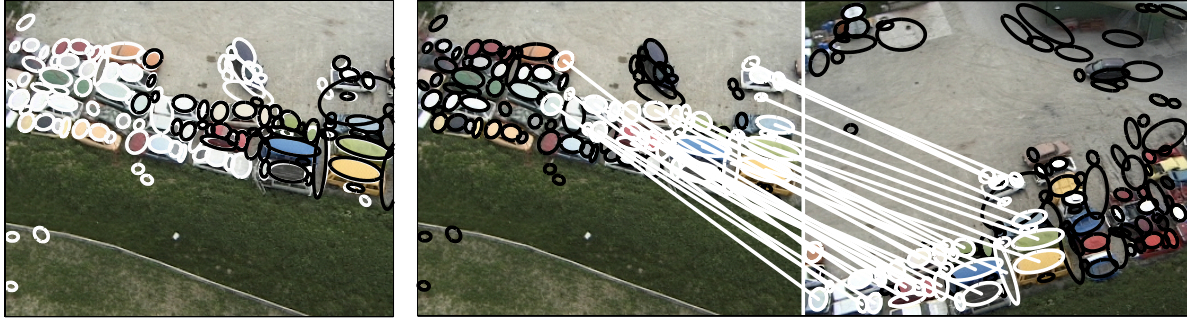


Figure 3: Left: Detected blobs (white), and blobs from image 2 mapped through homography (black). Centre and right: Final 32 correspondences. Unmatched blobs are painted black.

7 Epipolar constraint

We will now describe how the fundamental matrix constraint for points (1)-(2) can be applied to conics. From projective geometry we know that a point in an image corresponds to a point in 3D that lies somewhere on the line defined by the focal point and the point position in the image plane. In analogy with this, an ellipse in an image corresponds to a 3D cone that intersects the image plane in the ellipse, and has its tip passing through the focal point, see figure 4, left. In analogy with [9], we now make the assumption that the object in space is an ellipse. This is convenient, since we know from projective geometry that a projective transformation of a conic will always be another conic.



Figure 4: Epipolar constraints for ellipse. Left: cone in space generated by an ellipse in image 1 and projected into image 2. Centre and Right: The two images.

We have 3 degrees of freedom in choosing the ellipse in space, since any 3D ellipse that is defined by the intersection of the ellipse cone and an arbitrary plane is projected to the ellipse in the first image. A general ellipse in 2D has 5 d.o.f. and thus the cone defined by the ellipse in the first image should give us two constraints on a corresponding ellipse in the second image. These constraints can be found by requiring that the ellipse in the other image has the projected cone outline as tangents. An ellipse is uniquely defined by 5 tangent lines, and thus the two cone outline lines, or *epipolar tangents* [10] are our two constraints. The epipolar tangents for an ellipse and a given \mathbf{F} in an image pair are

shown in figure 4 (centre and right).

7.1 Epipolar tangents

To construct the epipolar tangents for an ellipse, we need the epipoles. They are obtained (using SVD) as the left and right null-vectors of \mathbf{F} [5], i.e.

$$\mathbf{e}^T \mathbf{F} = \mathbf{0} \quad \text{and} \quad \mathbf{F} \mathbf{e}' = \mathbf{0}. \quad (19)$$

We will now derive the epipolar tangents in image 2 defined by an ellipse in image 1. The construction of tangents in image 1 is made in a similar way. First we find the *polar line* in the left image, using the *pole-polar relationship* [5]. The polar line is a line which intersects the ellipse in two points. In these two points, the ellipse has tangents which go through the epipole \mathbf{e} . To construct the polar line \mathbf{l} , we use the conic form (11) of the ellipse \mathbf{C}

$$\mathbf{l} = \mathbf{C} \mathbf{e}. \quad (20)$$

We now write points on this line in parameter form

$$\mathbf{x} = \alpha \mathbf{t}_1 + \mathbf{t}_2 \quad \text{where} \quad \mathbf{t}_1 = \begin{pmatrix} l_2 \\ -l_1 \\ 0 \end{pmatrix}, \quad \mathbf{t}_2 = \begin{pmatrix} 0 \\ -l_3 \\ l_2 \end{pmatrix} \quad \text{and} \quad \alpha \in \mathbb{R}. \quad (21)$$

Plugging this into (11) gives us a quadratic polynomial in α , and inserting the two solutions in (21) gives us the two tangent points, \mathbf{x}_1 and \mathbf{x}_2 . Finally we obtain the two epipolar tangents as the epipolar lines for these two points

$$\mathbf{l}_1 = \mathbf{F}^T \mathbf{x}_1 \quad \text{and} \quad \mathbf{l}_2 = \mathbf{F}^T \mathbf{x}_2. \quad (22)$$

7.2 Tangent distances

Since we are using ellipses generated from measurements in real images, we can not expect the epipolar tangents to be exact tangents to a corresponding ellipse. Thus we will now define a measure of how close an epipolar tangent is to being a tangent to an ellipse.

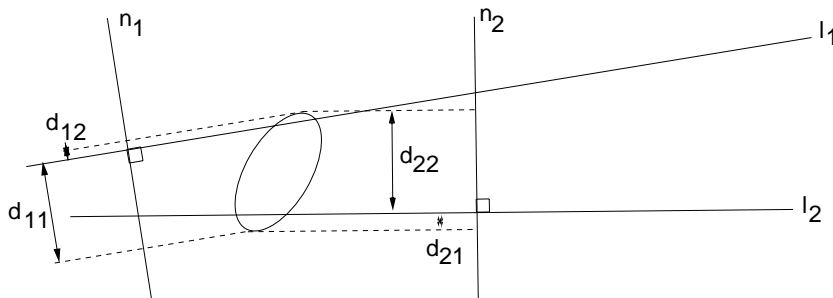


Figure 5: Tangent distances

The points on the ellipse are parametrised according to

$$\mathbf{x} = \mathbf{R}^T \mathbf{D}^{-1/2} \begin{pmatrix} \cos t \\ \sin t \end{pmatrix} + \mathbf{m} \quad t \in [0, 2\pi[\quad (23)$$

where \mathbf{R} and \mathbf{D} are the eigenvalue factorisation of \mathbf{I} , i.e. $\mathbf{I} = \mathbf{RDR}^T$ [3]. If we project the points onto the line normal, i.e. left multiply by $\mathbf{1}$, we get a set of positions along the normal

$$d(t) = (l_1 \ l_2) \mathbf{R}^T \begin{pmatrix} d_1 \cos t \\ d_2 \sin t \end{pmatrix} + (l_1 \ l_2) \mathbf{m} + l_3. \quad (24)$$

This is an expression of the form

$$d(t) = a_1 \cos t + a_2 \sin t + a_3 \quad (25)$$

which has the two extrema

$$d_{11} = \sqrt{a_1^2 + a_2^2} + a_3 \quad \text{and} \quad d_{12} = -\sqrt{a_1^2 + a_2^2} + a_3. \quad (26)$$

We repeat this for the second line, and combine the distances into two pairs $\langle d_{12}, d_{21} \rangle$ and $\langle d_{11}, d_{22} \rangle$. Finally we compute the epipolar distance as the minimum of the pair sums

$$r = \min(|d_{12}| + |d_{21}|, |d_{11}| + |d_{22}|). \quad (27)$$

We use these to fill in a new correspondence matrix \mathbf{S} according to

$$S_{ij} = e^{-(r_{ij} + r_{ji})^2 / \sigma_r^2}. \quad (28)$$

A correspondence is considered valid if the correspondence matrix \mathbf{S} has a maximum along both row i and column j , and furthermore is larger than some threshold $s_{\min} = 0.5$. (The scaling is at present set to $\sigma_r = 5$.)

The count of valid correspondences is then used as a way to rank the \mathbf{F} matrices obtained from RANSAC. The correspondences obtained in this way for the image pair in figure 4 (centre and right) are shown in figure 6. A total of 35 correspondences were found, and out of these 5 were visually judged as false matches.

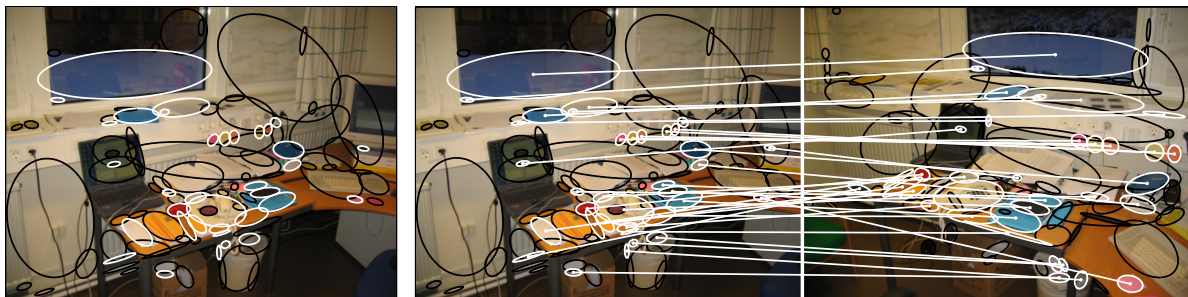


Figure 6: Final 35 correspondences. Left: Unused blobs (black) and used blobs (white) in image 1. Centre and Right: The found correspondences (white), out of these, 5 are wrong. Unmatched blobs are painted black.

8 Concluding remarks

This report has demonstrated the how blob features can be used for direct estimation of geometry. The number of RANSAC samples required is significantly lower than in estimation using point features (c.f. e.g. [2]). Note that although there are still some false matches in the final set of correspondences (see figure 6), the obtained fundamental matrix correctly describes the geometry situation. False matches could probably be rejected by requiring that correspondences are consistent across more than two views.

References

- [1] Matthew Brown and David Lowe. Invariant features from interest point groups. In *13th BMVC*, pages 253–262, September 2002.
- [2] Ondrej Chum, Jirí Matas, and Josef Kittler. Locally optimized ransac. In *Proceedings of DAGM*, pages 236–243, 2003. LNCS 2781.
- [3] Per-Erik Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, March 2004.
- [4] Richard Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.
- [5] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [6] Fredrik Kahl. *Geometry and Critical Configurations of Multiple Views*. PhD thesis, Lund University, September 2001.
- [7] Jirí Matas, Radek Marík, and Josef Kittler. Illumination invariant colour recognition. In *BMVC'94*, pages 469–479, 1994.
- [8] Stepán Obdržálek and Jirí Matas. Object recognition using local affine frames on distinguished regions. In *13th BMVC*, pages 113–122, September 2002.
- [9] Long Quan. Conic reconstruction and correspondence from two views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):151–160, February 1996.
- [10] Cordelia Schmid and Andrew Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 4(30):194–234, 2000.
- [11] Blob source code download.
<http://www.isy.liu.se/~perfo/software/>.
- [12] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC2000*, Bristol, September 2000. Invited Paper.