Fundamental Matrix of a Stereo Pair, with A Contrario Elimination of Outliers

Lionel Moisan[1], Pierre Moulon[2], Pascal Monasse[2]

[1] MAP5, Université Paris Descartes (`lionel.moisan@parisdescartes.fr`)
[2] LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, Champs-sur-Marne, France
(`pmoulon@gmail.com`, `monasse@imagine.enpc.fr`)

Abstract

In a stereo image pair, the fundamental matrix encodes the rigidity constraint of the scene. It combines the internal parameters of both cameras (which can be the same) and their relative position and orientation. It associates to image points in one view the so-called epipolar line in the other view, which is the locus of projection of the same 3D point, whose particular position on the straight line is determined by its depth. Reducing the correspondence search to a 1D line instead of the 2D image is a large benefit enabling the computation of the dense 3D scene. The estimation of the matrix depends on at least seven pairs of corresponding points in the images. The algorithm discarding outliers presented here is a variant of the classical RANSAC (RANdom SAmple Consensus) based on a contrario methodology and proposed first by Moisan and Stival in 2004 under the name ORSA. The distinguishing feature of this algorithm compared to other RANSAC variants is that the measure of validity of a set of point pairs is not its sheer number, but a combination of this number and the geometric precision of the points.

Source Code

The open-source code is available at the IPOL the web page of this article[1]. Most of the code is shared with the companion article by the same authors [12] dealing with a contrario estimation of a homography. This emphasizes the generality of the methodology and its simple specialization to different geometric problems.

Keywords: fundamental matrix; epipolar lines; a contrario; RANSAC

1 The Fundamental Matrix

1.1 The Epipolar Constraint

The fundamental matrix of a stereo pair, first discovered by Luong [10], is a simple extension of the essential matrix discovered a decade earlier by Longuet-Higgins [8]. However, its importance exceeds highly the essential matrix, since its usage does not require the knowledge of the internal parameters of the cameras, and the constraints for a valid fundamental matrix are much easier to handle than
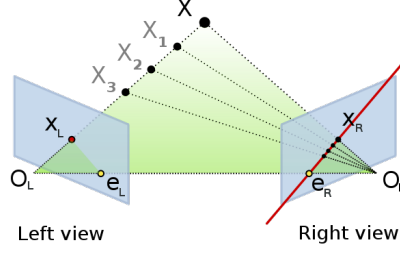
Figure 1: Configuration of two pinhole cameras with a 3-D point X projected at X_L and X_R in the two views.

those of the essential matrix. This matrix encodes all the geometric constraints linking both views and can be computed using only point correspondences between the views.

Figure 1 illustrates the configuration: we have two pinhole cameras, qualified arbitrarily as left and right views, with respective optical centers O_L and O_R. A space point X is viewed at X_L and X_R. The plane $O_L O_R X$ intersects the image planes along two straight lines, called epipolar lines. Notice that whatever the point X, epipolar lines go through the intersections e_L and e_R of the line $O_L O_R$ with the image planes. The mapping from a point X_L or X_R to the corresponding epipolar line is encoded in the essential matrix. Let us take as world coordinate system the left camera frame, of origin O_L, and suppose the right camera frame is related by the translation $T = O_L O_R$ and by the rotation matrix R expressing the orientation of the principal axes of the right camera relative to the left camera. We observe that the three vectors $O_L X_L$, $O_R X_R$ and $O_R O_L$ are coplanar. The vector $O_R X_R$ has coordinates $R X_R$, so that we can write the coplanarity condition of the three vectors as

$$\det \begin{pmatrix} X_L & T & R X_R \end{pmatrix} = 0, \tag{1}$$

which we can rewrite using the matrix $[T]_\times$ of the cross-product[2] with T as

$$X_L^T [T]_\times R X_R = 0, \text{ i.e., } X_L^T E X_R = 0 \text{ with } E = [T]_\times R. \tag{2}$$

E is the essential matrix. In the uncalibrated case, the vectors X_L and X_R (with real-world units) are not directly known, but their pixel coordinates counterparts x_L and x_R are: we have $x_L = K_L X_L$ and $x_R = K_R X_R$ with equality in homogeneous coordinates, that is up to a scalar factor for each. Substituting these expressions in (2) yields

$$x_L^T F x_R = 0, \tag{3}$$

where

$$F = K_L^{-T} E K_R^{-1} = K_L^{-T} [T]_\times R K_R^{-1} \tag{4}$$

is the fundamental matrix. From the formula (4) of F, it should be apparent that its rank is 2, but the reverse is also true: a 3×3 matrix of rank 2 is a fundamental matrix, see Appendix A. The exception is when $T = 0$ (no parallax), in which case $F = 0$ and the constraint $x_L^T F x_R = 0$ is trivially verified whatever x_L and x_R. When $T \neq 0$, we interpret the fundamental equation as:

- x_L is on the epipolar line of equation $F x_R$ (in the left image) associated to x_R;

- x_R is on the epipolar line of equation $F^T x_L$ (in the right image) associated to x_L;

---

[1] http://dx.doi.org/10.5201/ipol.2016.147

[2] If $T = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix}$, we have $[T]_\times = \begin{pmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{pmatrix}$ so that $T \times x = [T]_\times x$ for any 3-vector x.

- from $e_L = K_L T$, homogeneous coordinates of the left epipole, that is, the projection of O_R in the left image, we get $e_L^T F = 0$, showing that any epipolar line $F x_R$ goes through e_L;

- symmetrically, $e_R = K_R R^{-1} T$ represents the right epipole, $F e_R = 0$ and any epipolar line $F^T x_L$ goes through e_R.

Alternative, more algebraic derivations of the formula for F are in Appendix A.

Remark. *Notice that the scalar equation $x_L^T F x_R = 0$ can also be written $x_R^T F' x_L = 0$ with $F' = F^T$, notably in the influential book of Hartley and Zisserman [7], but we prefer our convention where left coordinates x_L go to the left and right coordinates x_R go to the right of F. Given F, the reader must be careful of which convention is used, and consider F^T instead if it does not match his own convention.*

1.2 Estimation of the Fundamental Matrix

The knowledge of F is very important, be it for external calibration (extraction of T and R from (4) knowing the internal calibration K_L and K_R), or for epipolar rectification as a preliminary step to the computation of the disparity map, see for example [14]. In most instances, the computation of F from its formula (4) is not an option, as rotation and translation are usually not known, and even the internal calibration matrices K_L and K_R may be unknown. In that case, we note that (3) can be interpreted as a linear homogeneous constraint on F if a pair of matching points (x_L, x_R) is known. Writing $x_L^i = \begin{pmatrix} x_i & y_i & 1 \end{pmatrix}^T$ and $x_R^i = \begin{pmatrix} x_i' & y_i' & 1 \end{pmatrix}^T$ for the i-th correspondence (x_L^i, x_R^i), (3) becomes

$$A_i^T f = 0 \text{ with } A_i^T = \begin{pmatrix} x_i x_i' & x_i y_i' & x_i & y_i x_i' & y_i y_i' & y_i & x_i' & y_i' & 1 \end{pmatrix}, \tag{5}$$

and f is the vector of the coefficients of F,

$$f = \begin{pmatrix} F_{11} & F_{12} & F_{13} & F_{21} & F_{22} & F_{23} & F_{31} & F_{32} & F_{33} \end{pmatrix}^T. \tag{6}$$

Stacking 8 such equations for so many correspondences (x_L^i, x_R^i), we see that f is in the kernel space of the 8×9 matrix A of rows the A_i^T. Since the trivial solution $f = 0$ is not interesting and the scale of f is indifferent (λf with any $\lambda \in \mathbb{R}$ represents an equivalent fundamental matrix), any non-null element of the kernel of A fits. There is however no guaranty that the resulting F has rank 2. An alternative to this 8-point algorithm is the 7-point algorithm, which enforces this constraint.

In the 7-point algorithm, A is of dimensions 7×9, so it normally should have a kernel of dimension 2. Taking a basis f_1, f_2 of this kernel, we know that f should be written $f = (1 - t) f_1 + t f_2$ with $t \in \mathbb{R}$. Writing

$$\det F = \det \left((1 - t) F_1 + t F_2 \right) = 0 \tag{7}$$

results in a polynomial of degree at most 3 of which t is a real root. Therefore there are at most 3 possible values for t. The correct one cannot be determined directly from the 7 correspondences, but additional ones can be used to check.

The input of our algorithm is a set M of putative pairs of matching points (and the dimensions of the images), the output a fundamental matrix F and the subset M_{in} of M that was taken into account to estimate F. Indeed, starting from a pair of images, we need two preliminary steps before computing F:

1. Detecting a set of keypoints in each image, P_L and P_R;

2. Matching the two lists of keypoints, resulting in a set $M \subset P_L \times P_R$.

Numerous variants exist to achieve the above steps, most ot them based on the famous SIFT algorithm [9, 17], which we use also. However, some putative pairs of matching points issued from the second step may be mismatches and should be discarded from the estimation of F. This paper describes an algorithm which does simultaneously the necessary tasks:

1. Filter the list of matching keypoints $M_{\text{in}} \subset M$;

2. Estimate the fundamental matrix F from M_{in}.

2 A Contrario RANSAC

The 7-point algorithm can be used as minimal solver when incorporated into RANSAC [5] or one of its variants, whose goal is to discriminate inliers and outliers among the n correspondences (x_L^i, x_R^i). At each iteration of the algorithm, a sample of 7 correspondences is drawn and supposed to be composed of inliers. From this sample, up to three fundamental matrices are obtained, and the compatibility of the rest of the correspondences is checked. In this regard, the 7-point algorithm is much preferred to the 8-point algorithm, since the chance of having a wrong correspondence among the sample is significantly lower when its cardinality is 7 instead of 8.

In the ORSA [13] (Optimized RANSAC) variant of RANSAC, the identification of inliers is not based on a fixed distance threshold and the best model on the number of inliers; instead, different thresholds, based on the distribution of errors of remaining correspondences, are tested, and the best model is measured by the Number of False Alarms (NFA), which depends on the threshold *and* the number of errors below that threshold. The NFA is computed based on the *a contrario* principle, introduced by Desolneux et al. [3]. Let us recall the generic formula (1) from the companion article [12]

$$\text{NFA}(\{\epsilon_i : i = 1, \ldots, n\}, k) = N_{\text{outcomes}}(n - N_{\text{sample}}) \binom{n}{k} \binom{k}{N_{\text{sample}}} (\epsilon_k^d \alpha_0)^{k - N_{\text{sample}}}. \tag{8}$$

Here the ϵ_i are the errors of the correspondences with respect to the currently tested model (a fundamental matrix), increasingly ordered, and therefore k is the tested error threshold, with only correspondences exhibiting an error at most k being considered as inliers. Therefore the best model is the one verifying

$$F = \arg \min_F \min_{k = N_{\text{sample}}+1, \ldots, n} (\{\epsilon_i(F)\}, k). \tag{9}$$

For the fundamental matrix estimation, the parameters are:

- $N_{\text{sample}} = 7$ since we use the 7-point algorithm;

- $N_{\text{outcomes}} = 3$ since up to three fundamental matrices are tested by sample;

- the exponent $d = 1$ since the error is uni-dimensional, that is, a point to line distance;

- $\alpha_0 = 2\sqrt{w'^2 + h'^2}/(w'h')$.

Here $w' \times h'$ are the dimensions of the right image. α_0 is an upper bound of the probability of a random point uniformly drawn in the right image area to fall within 1 pixel to a fixed line. The numerator is twice the diagonal length of the image and the denominator its area. Notice that this relies on a background hypothesis $\mathcal{H}_0$ where a random correspondence is a pair of points drawn uniformly and independently in their respective image. A more sophisticated background hypothesis $\mathcal{H}_0'$, adapted to the observed distribution of points, yields sometimes better results [4] but this is out

of the scope of this article. The ORSA algorithm follows faithfully Algorithm 1 in the companion article [12], with the point to line distance defined as

$$\epsilon_F(x_L, x_R) = \frac{\left|x_L^T F x_R\right|}{\sqrt{(x_L^T F e_1)^2 + (x_L^T F e_2)^2}}, \tag{10}$$

where e_1 and e_2 are the first two vectors of the canonical basis of 3-row vectors.

In the traditional RANSAC algorithm [7, Section 4.7.1], the number of iterations is adjusted dynamically with the formula establishing the probability of having a good sample. Suppose there are k inliers. The probability of drawing a good sample of N_{sample} inlier points is $(k/n)^{N_{\text{sample}}}$. We require the probability of having only N_{iter} bad samples to be below some fixed quantity β (for example, $\beta = 1\%$)

$$\left(1 - (k/n)^{N_{\text{sample}}}\right)^{N_{\text{iter}}} \le \beta, \tag{11}$$

so that

$$N_{\text{iter}} \ge \left\lceil \frac{\log \beta}{\log(1 - (k/n)^{N_{\text{sample}}})} \right\rceil. \tag{12}$$

Initially, k is not known, so that N_{iter} is set according to the computation effort we can afford, but each time a model with more supporting inliers is found, N_{iter} is lowered to its minimal value satisfying (12). In comparison, ORSA stops when a meaningful set is found (NFA < 1), at which point it adds a few iterations to optimize the model by trying to find a better sample among the estimated inliers.

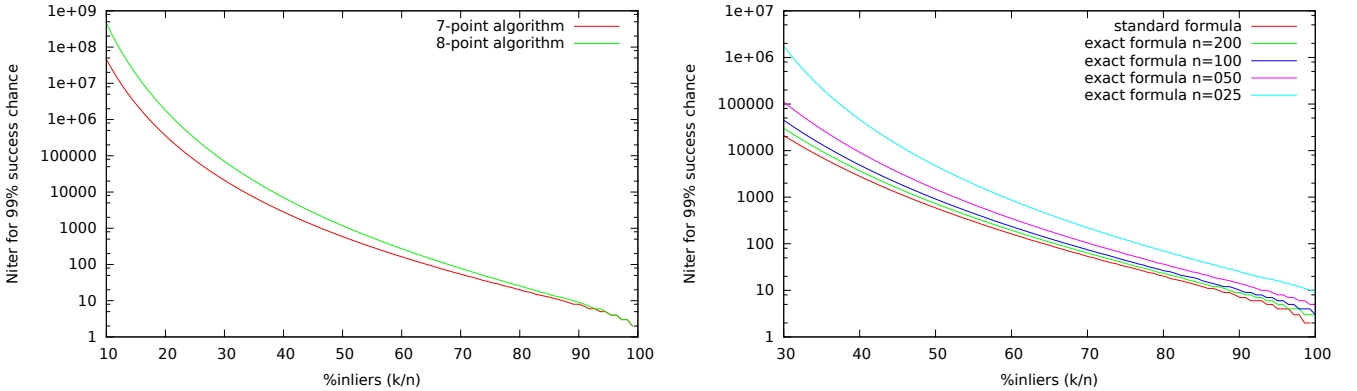

Figure 2: Number of iterations to handle with 99% confidence ($\beta = 1\%$) a given inlier ratio k/n. For a 90% confidence level ($\beta = 10\%$), this number of iterations must be divided by 2. Left: traditional formula (12) with 7- and 8-point algorithm. Right: the exact formula (14) (i.e., excluding repetitions in the samples) for the 7-point algorithm with different values of n.

The advantage of using 7-samples over 8-samples, despite the increased complexity of the 7-point algorithm compared to the 8-point algorithm, is illustrated in Figure 2: A visibly higher outlier ratio can be handled with good confidence (99% chance of success) with the same number of iterations. Notice however that the traditional formula (12), found in the book by Hartley and Zisserman [7] and in numerous publications, is optimistic: it assumes that each element of the sample is drawn independently but that repetitions are allowed. Actually, if a correspondence is repeated in the 7-sample, the equation system is underdetermined and the model cannot be estimated. Therefore only good samples with no repetition should be considered as a good evaluation for the model. With the same sample drawing algorithm, that is, drawing with replacement, the probability of a good sample with no repetition is

$$\frac{(k)_7}{n^7} := \frac{k \cdot (k-1) \cdots (k-6)}{n^7} = (k/n) \cdot (k/n - 1/n) \cdots (k/n - 6/n), \tag{13}$$

and the exact formula updating (12) is

$$N_{\text{iter}} \geq \left\lceil \frac{\log \beta}{\log(1 - ((k)_7/n^7))} \right\rceil . \tag{14}$$

When n is large, this exact formula (14) does not sensibly differ from (12). However, when n is a few tens, Figure 2 shows that N_{iter} must be significantly increased over the traditional formula to handle the same outlier ratio with the same confidence level. The code fixes $N_{\text{iter}} = 10,000$ and we see that it is notably insufficient to handle cases with less than 30% inliers. This is in contrast with the companion article [12] that illustrates successful homography evaluations with as few as 10% inliers.

3 Implementation and Demo

3.1 Implementation

The core implementation of the *a contrario* RANSAC (ORSA) is the same as with the companion article [12], only parts specific to the fundamental matrix are changed. The only change in the generic part is that we found that we could get some error threshold $\epsilon_k = 0$ in (8) for correspondences *not* in the sample set. Such noiseless detection happens only with synthetic data, and it yields NFA = 0, which is optimal, with a null inlier/outlier threshold. This is not sensible for normal data, and we just take a tiny but strictly positive minimal value for ϵ_k.

3.1.1 Normalization

As in the homographic case ([12], Section 4.2), we normalize the point coordinates by premultiplying by the matrix

$$\begin{pmatrix} 1/\sigma_L & 0 & -w_L/(2s_L) \\ 0 & 1/\sigma_L & -h_L/(2s_L) \\ 0 & 0 & 1 \end{pmatrix}, \tag{15}$$

where $\sigma_L = \sqrt{w_L h_L}$ is an average scale of the left image, and similarly for points in the right image with normalization matrix N_R. If we find a fundamental matrix $\tilde{F}$ relating normalized points $(\tilde{x}_L^i, \tilde{x}_R^i) = (N_L x_L^i, N_R x_R^i)$ we have

$$(N_L x_L^i)^T \tilde{F} (N_R x_R^i) = (x_L^i)^T (N_L^T \tilde{F} N_R) x_R^i, \tag{16}$$

so that we identify the denormalization formula $F = N_L^T \tilde{F} N_R$. This normalization is considered crucial [6], and we check numerically that it is so. We take a stereo pair of images of size 640×480 and we normalize corresponding SIFT points by premultiplying with the matrix

$$N = \begin{pmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{17}$$

and look at the condition number of the 8×9 matrix A built by stacking the equations issued from (5) for different samples of 8 correspondences. This condition number κ is the ratio of the highest and smallest singular values of A: $\kappa = \sigma_1/\sigma_8$. Figure 3 shows that we have a best condition number (i.e., lowest value) when $-3 \leq \log(1/\sigma)/\log 10 \leq -2$. We have $\log(1/\sigma_L)/\log 10 = -2.7$ in this case, showing that it is a judicious choice. The scale factor magnitude σ has the most influence on numerical stability; the effect is much lower when the centering values are changed (N_{13} and N_{23}).
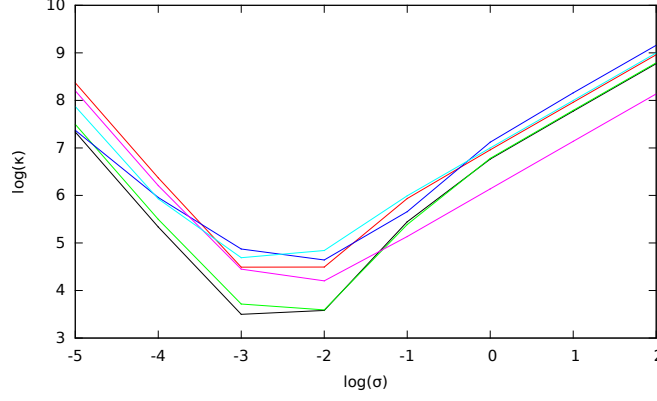
Figure 3: Condition number κ of matrix A in the 8-point algorithm for various samples of 8 point correspondences in images of size 640×480, as a function of the normalization factor σ (x- and y-axes in $\log_{10}$ scale).

3.1.2   7-Point Algorithm

The matrix A is of size 7×9, we use the singular value decomposition (SVD) to find two independent vectors f_1 and f_2 in its kernel. We then have to find the values of t satisfying (7), which we can also write

$$\det(F_1 + t(F_2 - F_1)) = 0. \tag{18}$$

Let us decompose the matrices into their columns

$$F_1 = \begin{pmatrix} c_1 & c_2 & c_3 \end{pmatrix} \qquad\qquad F_2 - F_1 = \begin{pmatrix} d_1 & d_2 & d_3 \end{pmatrix}. \tag{19}$$

Equation (7) amounts to

$$at^3 + bt^2 + ct + d = 0, \tag{20}$$

with

$$a = \det(F_2 - F_1) \qquad b = \det\begin{pmatrix} d_1 & c_2 & c_3 \end{pmatrix} + \det\begin{pmatrix} c_1 & d_2 & c_3 \end{pmatrix} + \det\begin{pmatrix} c_1 & c_2 & d_3 \end{pmatrix}, \tag{21}$$

$$d = \det F_1 \qquad c = \det\begin{pmatrix} d_1 & d_2 & c_3 \end{pmatrix} + \det\begin{pmatrix} d_1 & c_2 & d_3 \end{pmatrix} + \det\begin{pmatrix} c_1 & d_2 & d_3 \end{pmatrix}. \tag{22}$$

The value of d is obtained by letting $t = 0$ and the value of c as the first derivative at 0 of $\det(F_1 + t(F_2 - F_1))$. We used the property, based on multi-linearity and continuity of determinant function,

$$\frac{d}{dt} \det \begin{pmatrix} f(t) & g(t) & h(t) \end{pmatrix} = \det \begin{pmatrix} f'(t) & g(t) & h(t) \end{pmatrix} + \det \begin{pmatrix} f(t) & g'(t) & h(t) \end{pmatrix} + \det \begin{pmatrix} f(t) & g(t) & h'(t) \end{pmatrix}, \tag{23}$$

when f, g and h are derivable functions of t and the prime denotes derivative function. For the values of a and b we notice

$$\det(F_1 + t(F_2 - F_1)) = t^3 \det \left((F_2 - F_1) + \frac{1}{t} F_1 \right), \tag{24}$$

and noting $t' = 1/t$ we just need to take the derivative with respect to t' of the determinant to derive formulas analogous to (22), just inverting the roles of F_1 and $F_2 - F_1$.

To find the roots of the cubic polynomial, we use an algorithm detailed in Appendix B. In this algorithm, the coefficients are first divided by a so as to have a leading coefficient of 1. To diminish the risk of trouble when a could be almost null, we check if $|d| > |a|$. If this is the case, we just

95

swap F_1 and $F_2 - F_1$, a and d, and b and c. Notice that this does not rule out the possibility that $a = d = 0$, or almost. Actually, it is even possible to have $a = b = c = d = 0$, that is, having a plane in the cone of non-invertible matrices. For example, with the following matrices

$$F_1 = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \qquad\qquad F_2 = \frac{1}{2} \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}. \tag{25}$$

We have $\|f_1\| = \|f_2\| = 1$ and $f_1^T f_2 = 0$, showing that $\{f_1, f_2\}$ could be an orthonormal basis of the kernel of A. The last row of any linear combination of F_1 and F_2 is null, so that all determinants are 0. We do not take any additional precautions for such unlikely cases: if infinity or NaN ("not a number") are generated due to division by 0, this leads just to a model with no inlier, which is rejected, and we proceed with the next iteration.

3.1.3   8-Point Algorithm

The 8-point algorithm, often used as minimal problem solver in RANSAC since its implementation is easier than the 7-point alternative[3], can be used with more points to refine the estimation. After the matrix F with lowest NFA in (8) has been found, it is still estimated from a minimal sample of 7 points. To distribute the error more equally among all the inliers, the simplest method is to minimize the algebraic error $\sum_i ((x_L^i)^T F x_R^i)^2$ where the index i spans the inliers. This value is equal to $\|Af\|^2 = f^T A^T A f$ where we incorporate in $k \times 9$ matrix A all the rows built from the k inliers in (5). The solution is the right singular vector f associated to the least singular value of A, that is the 9-th column of V in the SVD of $A = USV^T$. Since this does not constrain F to be of rank 2, we impose it a posteriori, by taking the SVD of $F = USV^T$, and replacing F by

$$US \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T, \tag{26}$$

that is forcing the lowest singular value to 0. This standard procedure, called the Direct Linear Transform (DLT) [7], amounts to projecting the initial F on the manifold of matrices with null determinant, using the Frobenius norm, $F = \sqrt{\operatorname{tr} F^T F}$. Notice that this projection does not give any control on the amount of error it involves. In our implementation, we choose to ignore the result of the 8-point algorithm, and stick to the F found by the 7-point algorithm, if its RMSE of point-to-line distance in the right image gets higher than the maximum error in the 7-point algorithm.

3.2   Online Demo

Figure 4 shows an image pair, and Figure 5 the resulting images with the default parameters. The top image shows in green the endpoints of inliers with only a segment of their epipolar line, to improve readability. In red are the endpoints of outliers. The middle image joins the endpoints of inliers. The bottom image shows the outliers with their corresponding epipolar line according to the estimated fundamental matrix. In order to associate the endpoints to their epipolar line, the orthogonal segment issued from the endpoint to the line is drawn. For example, an erroneous correspondence (a, a') links the third window in the left image to the second window in the right image. This is a small error, and we can check that epipolar lines A and A' go through the correct matching point. More flagrant is the correspondence (b, b') linking a point at the boundary of the

---

[3]The 8-point algorithm involves solving simply a linear system compared to finding the null-space of matrix A and finding cubic polynomial roots.

Figure 4: A pair of stereo images from USAC [16] (`fundmatrix/test1`).

lawn and the tiled floor (left image) to a façade (right image). To see where the correct point should be located, we follow visually the red segment to the right image (incorrect endpoint b'), go down the yellow segment until reaching the orthogonal epipolar line B. This tells that the correct endpoint should be on this line, which looks plausible. Naturally, the same process can be applied to the right endpoint b', though the correct matching point would lie outside the image on line B. Another wrong match (c, c') links points on different roofs. The intersection of lines $\{A', B', C'\}$ and $\{A, B, C\}$ are the epipoles e_L and e_R, here slightly out of the image frame.

The text output of the software program is displayed below:

```
sift:: 1st image: 1332 keypoints
sift:: 2nd image: 1391 keypoints
sift:: matches: 105
Remove 6/105 duplicate matches, keeping 99
  nfa=-108.516 inliers=96 precision=11.7253 im2 (iter=0,sample=14,15,69,85,86,92,97)
  nfa=-121.73 inliers=95 precision=7.71844 im2 (iter=3,sample=85,22,55,64,7,50,34)
  nfa=-162.213 inliers=92 precision=2.05903 im2 (iter=8,sample=7,40,41,79,19,1,92)
  nfa=-174.362 inliers=96 precision=2.13442 im2 (iter=16,sample=16,31,58,28,95,76,60)
Before refinement: Average/max error: 0.775217/2.13442
After refinement: Average/max error: 0.830378/2.73219
F=[ -9.69806e-09 -6.73975e-07 0.000298566;
    7.49243e-07 7.5501e-09 -0.000804739;
    -0.000344953 0.000883091 -0.00859197 ]
```

This tells us that 99 unique SIFT correspondences are found. The model with best NFA is found at iteration 16 with 96 inliers and a precision of $\epsilon_k = 2.1$. The indices of the correspondences in the 7-pair sample are displayed $(16, 31, \ldots)$. Next come the average geometric error in the right image (RMSE) and the maximum error for all inliers

$$\text{RMSE} = \sqrt{\frac{\sum_{i \in I} \epsilon_F(x_L^i, x_R^i)^2}{|I|}} \tag{27}$$

with I the set of inlier indices. Before refinement, the maximum error is naturally $\epsilon_k = 2.1$ pixel; after the 8-point algorithm is applied to all the inliers, the errors slightly increase in this case because:

1. The refinement, that is, the DFT, minimizes the algebraic error $\|Af\|^2$, not the geometric error of (27);
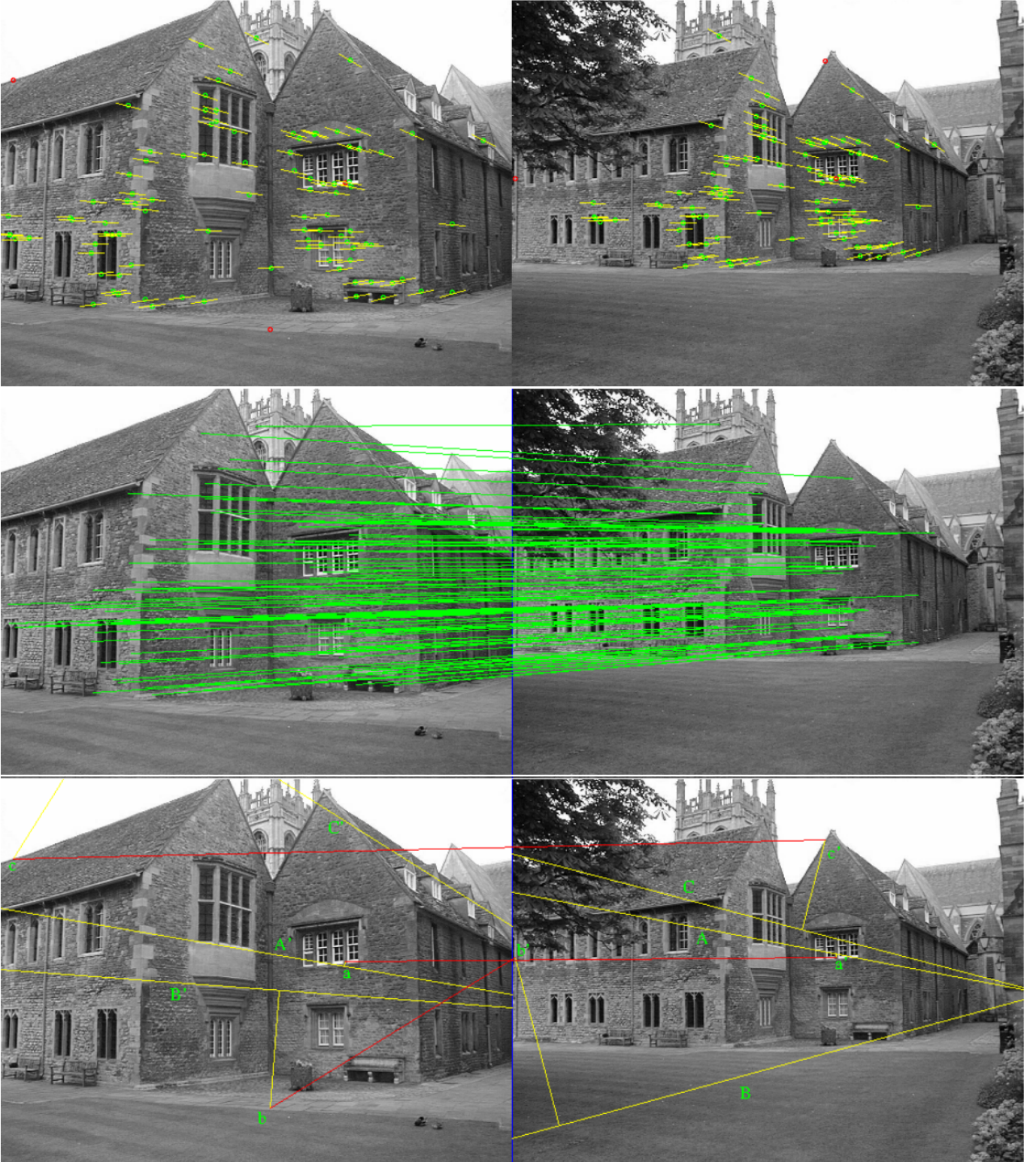
97

Figure 5: Output images of the algorithm run on the pair of Figure 4: endpoints of correspondences with partial epipolar lines for inliers (green points), inliers, outliers with estimated epipolar lines.
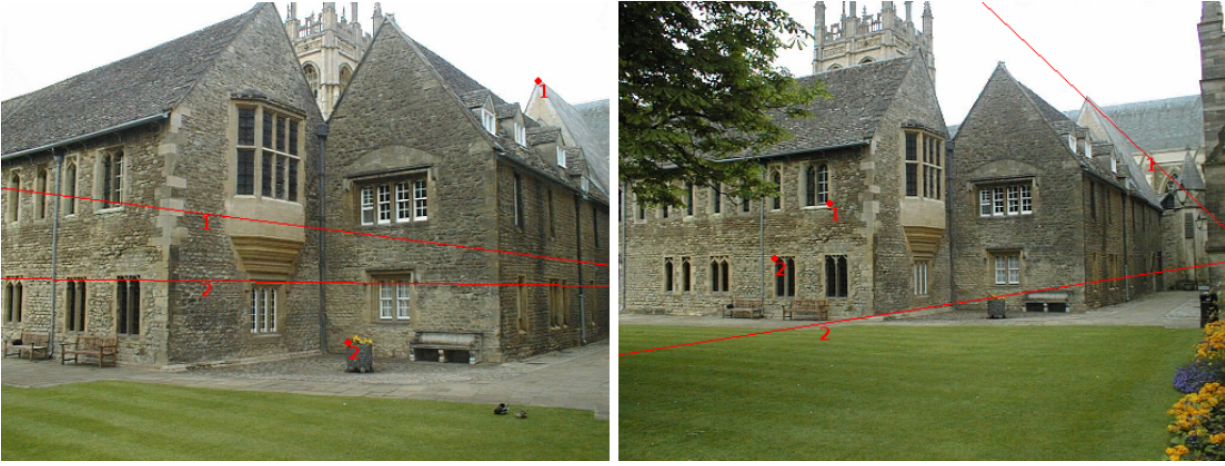
Figure 6: Stereo pair with interactive display of epipolar lines. Line numbers are associated to the same point numbers in the other image (these numbers do not appear in the online demo).

2. The a posteriori projection (26) brings no control over the final error increase.

The resulting F is displayed. The demo also allows clicking in one image and seeing corresponding epipolar lines in the other image, as in Figure 6.

4 Examples

The example datasets[4] are issued from the software package associated to the Universal RANSAC (USAC) [16], which is a RANSAC algorithm incorporating the best of breed of many variants. Still, USAC depends on a fixed inlier/outlier threshold. Our tests show that using the putative correspondences provided by USAC yields good results with our algorithm, except for quasi-degenerate cases (see below), because we do not make any provision for dealing with those.

4.1 The Epipolar Constraint

The epipolar constraint is not a very strong constraint, because it is one-dimensional, in contrast to the case of homography estimation for example, where the constraint on a match is two-dimensional. It is frequent in artificial environments that the translation direction is parallel to aligned structures. This is the case in aerial photography when the flying vehicle follows the streets, so that roof boundaries are aligned to the epipolar direction, or as in Figure 7 in street level imagery, where the camera moves horizontally, one of the main directions of alignment of features. For example, the correspondence numbered 274 matches the wrong window, but is still considered as inlier. Inliers 243 and 262 are plainly wrong but still satisfy the epipolar constraint.

It can even happen that a meaningful matrix F is found between fairly unrelated images. This is the case in Figure 8, where the wrong façades of a same building are matched. This is due to architectural consistency, yielding wrong correspondences due to repeated structures, yet geometrically consistent. Despite the fact that almost all concern coplanar points, which *could* be detected, such errors would still be possible with non-degenerate structures.

---

[4]USAC datasets are composed of image pairs and sets of putative correspondences. Unless otherwise stated, we do not use the provided correspondences but replace them with the output of the SIFT algorithm.

Figure 7: Example of wrong matches still considered as inliers because the epipolar constraint is not very discriminative (pair and correspondences from USAC [16], `fundmatrix/test3`). The top images show all inliers and outliers according to the algorithm. The bottom images show 3 wrong correspondences (243, 262 and 274) that still satisfy the epipolar constraint, and are thus wrongly classified as inliers.

Figure 8: Incorrect estimated model between different façades of the Orangerie de Sceaux, with a threshold of 1.1 pixel.
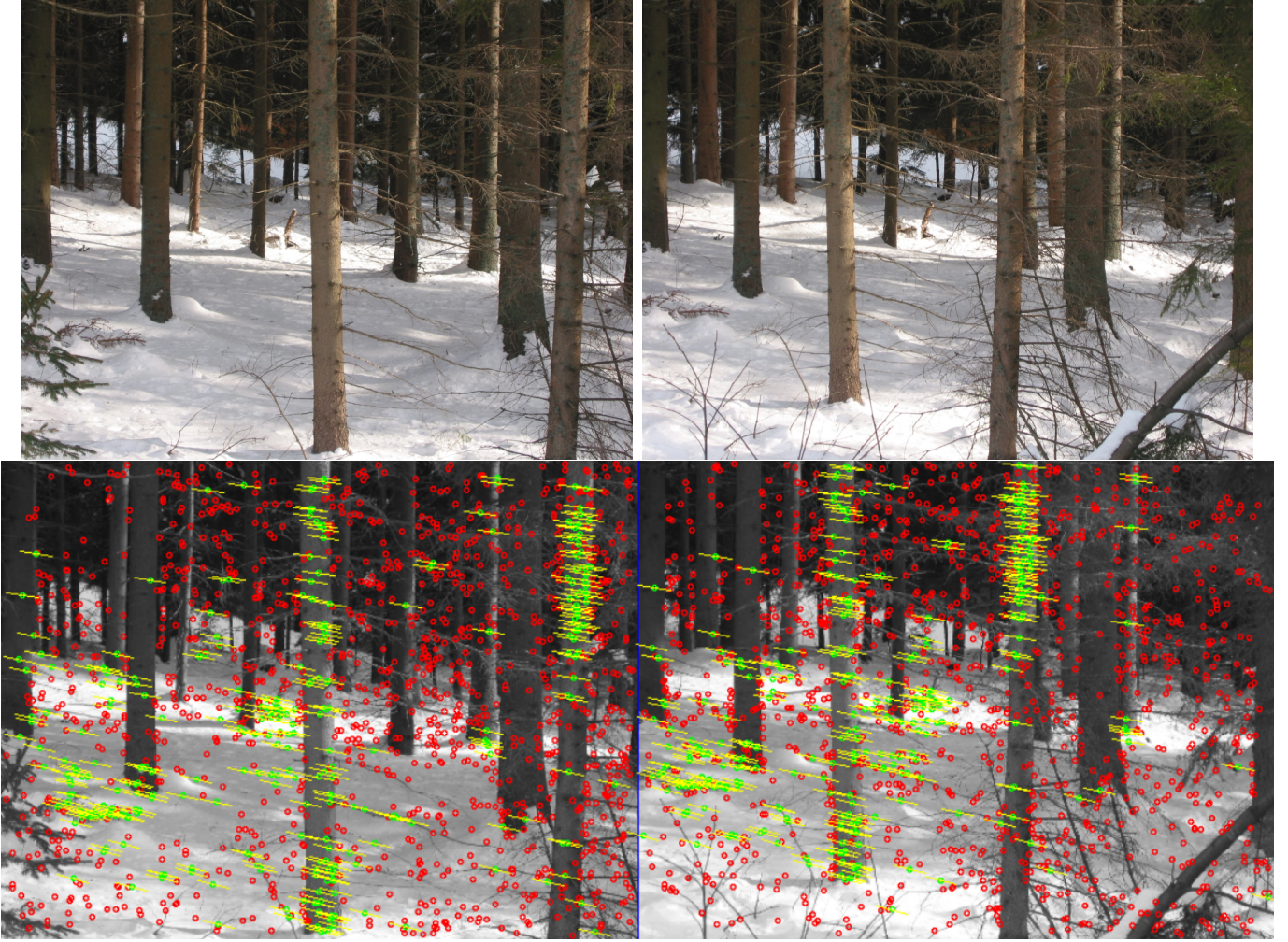
Figure 9: Up: stereo pair provided with the USAC code [16] (`fundmatrix/test4`). Bottom: 301 found inliers and outliers, with threshold of 0.8 pixel, among the provided 1516 putative correspondences.

4.2 The Chirality Constraint

As observed by Chum et al. [1], the chirality constraint (see Appendix A) helps in discarding F models right away, before their inlier support is evaluated. For example, for the `fundmatrix/test4` dataset of the USAC code (see Figure 9, with the provided putative 1516 correspondences), using the chirality constraint allows eliminating 158 of the first 173 estimated models, at which step the first meaningful model is found (NFA < 1). After that, since additional samples are drawn among inliers, the rejection rate decreases a lot: 70% of models estimated later satisfy the chirality constraint.

The absence of the chirality constraint can even cause a dramatic failure. This is for example the case for the image pair `fundmatrix/test3` of USAC,[5] see Figure 10. Among the 21 found inliers, only 6 points are involved in the right image. This happens because the SIFT criterion was not applied symmetrically, so that one point x_R in the right image is matched with 16 others in the left image. The minimal sample drawn selects two of these (incompatible) matches and the computed matrix F has x_R as right epipole, $Fx_R = 0$. The other 5 correspondences impose 5 constraints on F that can be accurately satisfied, since the equation system $Fx_R = 0$ leaves 5 degrees of freedom for F: the constraint $\det F = 0$ is already a consequence of $Fx_R = 0$. Therefore,

---

[5]with recalculated SIFT correspondences and a SIFT ratio of 1; the failure does not happen with the correspondences provided by USAC.
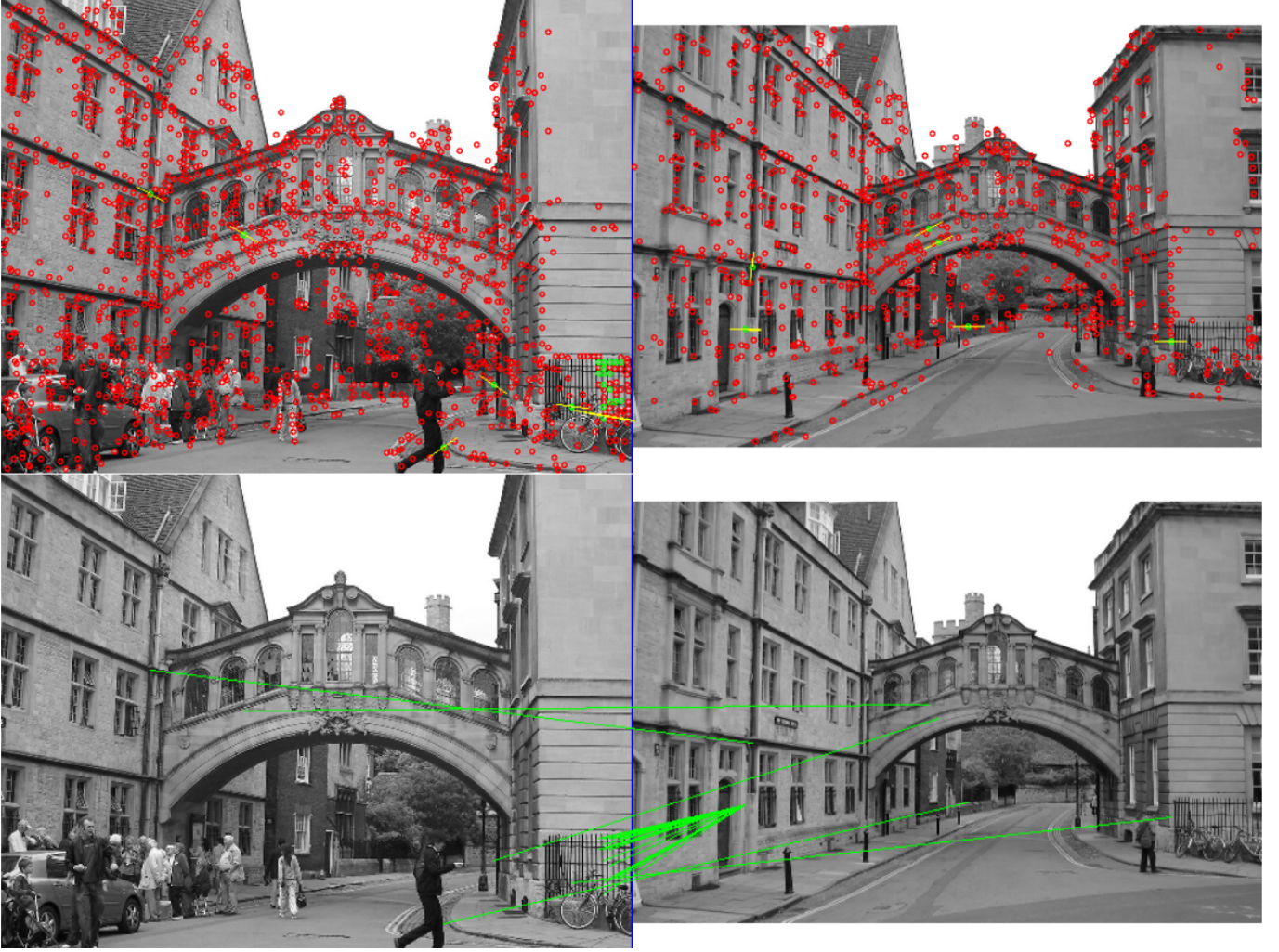
Figure 10: Wrong model found when the chirality check is disabled. A point in the right image is matched with 16 others in the left image, and a model F taking this point as right epipole can accommodate accurately all correspondences shown in green.

the 16 correspondences involving x_R satisfy perfectly the epipolar constraint, and the NFA is quite meaningful as the inlier/outlier threshold can be chosen very low (around 10^{-12} here).

4.3 Degeneracy

The most common degenerate case for the computation of the fundamental matrix is when the observed points are coplanar, as shown in Appendix A. Detecting this configuration during the RANSAC procedure requires an explicit test, as in the DEGENSAC algorithm [2]. This algorithm tries to detect a set of correspondences consistent with a homography H among the inliers and then initiates a sub-procedure of RANSAC with modified sampling: two correspondences are drawn among the outliers of the homography H to estimate F with no ambiguity. The Universal RANSAC (USAC) integrates this test and provides data exhibiting the degenerate behavior, but most RANSAC implementations do not. Our implementation does not provide such an enhancement and may get trapped in a degenerate configuration, see Figure 11.
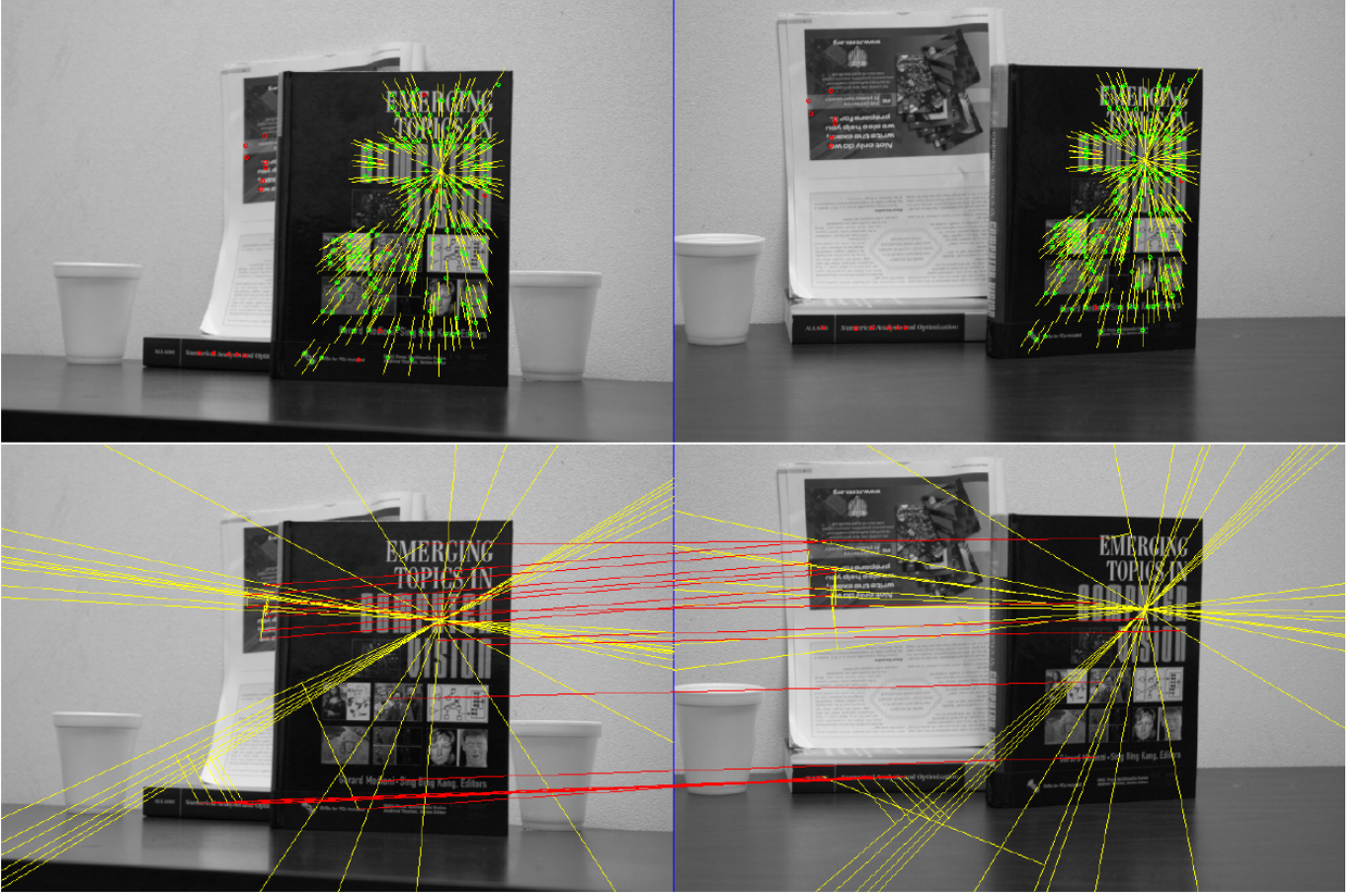
Figure 11: A quasi-degenerate case where our algorithm gets trapped in the degenerate model due to coplanar points (data from USAC [16], `fundmatrix/test8`). The few matches outside the foreground book, although most of them correct, are rejected because they are not consistent with the estimated fundamental matrix, which is any matrix consistent with the dominant plane homography.

4.4 A Failure Case

In a few instances, the algorithm can fail for complex reasons. In dataset `fundmatrix/test5` of USAC, using recalculated SIFT points and a SIFT ratio of 1, a meaningful F is found with a high threshold of 41 pixels. It involves 175 inliers among 437 correspondences, see Figure 12. A good proportion of these points lie close to the (wrongly estimated) right epipole, and for these points the constraint to be within 41 pixels of their epipolar line is not a strong constraint. Notice that the distance of points to their epipolar line is clearly not low in the right image, on average 23 pixels; this is dramatic in the left image, where some epipolar lines seem to lie alone in the sky. This is not a contradiction since we use the asymmetric point-to-line distance, in the right image. Using the symmetric error parameter fixes this, but then no meaningful model is found. The "refined" model with all inliers yields in this case a dramatic failure, with an average error of 76 pixel and a maximum of 589! This refinement is nevertheless rejected by our code since the average error of the refinement is above the maximum error based on the initial sample. Several factors can be identified, leading to a wrong estimation:

- The background model assumes points uniformly distributed in the images. In this case, a good portion of the image is the sky, in which no interest point is detected. The use of a more sophisticated background model, learned based on the observed distribution of points, would be helpful [4].
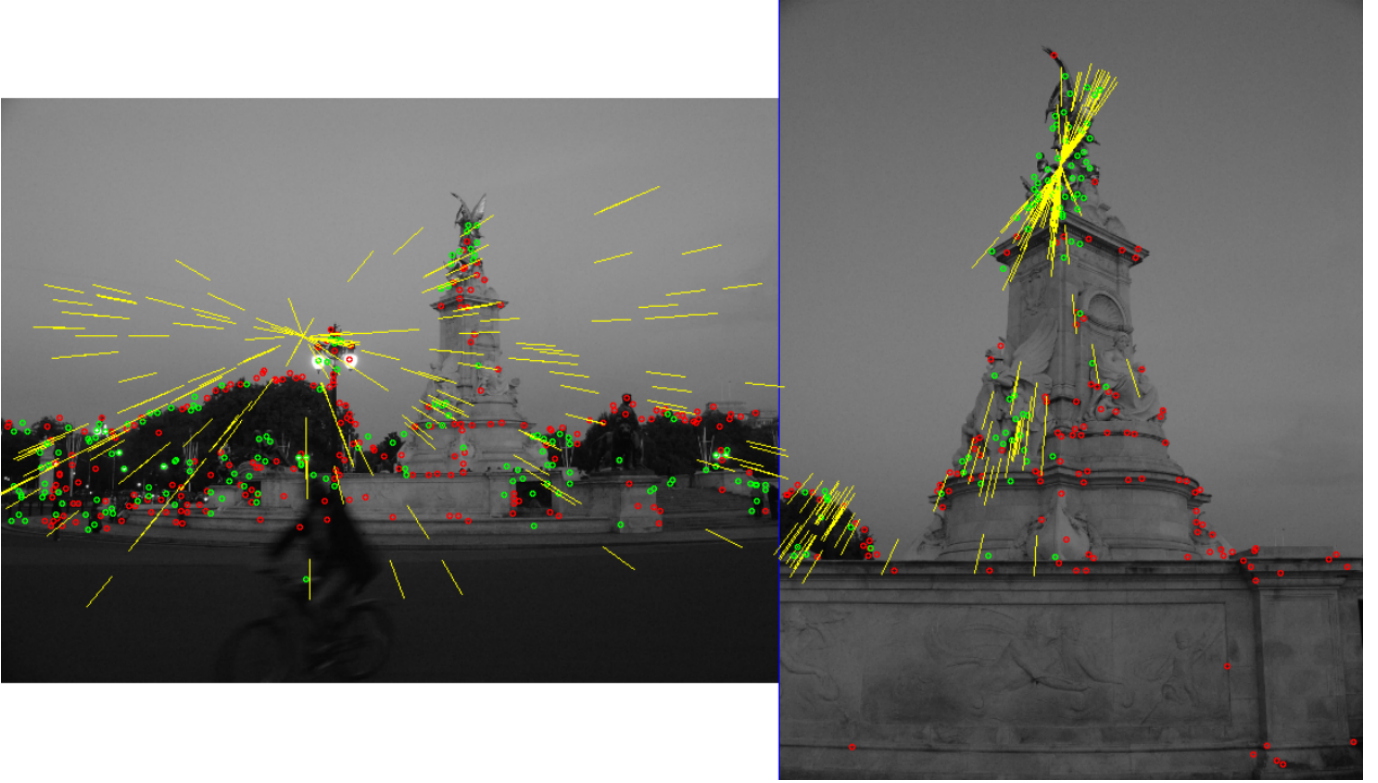
Figure 12: A failure case of the algorithm (data from USAC [16], `fundmatrix/test5`), with an inlier threshold of 41 pixels.

- Many correspondences involve points at the top of the statue, among them some good correspondences but also many erroneous ones. Placing the right epipole close to them and having a not too strong inlier threshold satisfies many of these correspondences. (Even though our chirality test prevents the epipole from being very close to one data point in the sample, the threshold is too low for this case). The distance to the epipolar line is maybe not a very appropriate criterion in this case: since all epipolar lines go through the epipole, this constraint is very weak for points close to the epipole. A better criterion would be the surface spanned between the epipolar line and the line joining the point and the estimated epipole. Notice that such a case can occur when an estimated epipole is inside the image frame.

5   Conclusion

The fundamental matrix estimation has been the subject of an abundant literature, so as the RANSAC algorithm. The most successful ideas were included in the Universal RANSAC, USAC [16], yet the automatic estimation of the inlier/outlier threshold was overlooked. This is a pity as having such an automatic estimation is shown to be beneficial in structure from motion [15]. The openMVG software[6] is open-source and relies heavily on the *a contrario* RANSAC considered in this article. In comparison to USAC, all datasets are handled successfully, except the ones including a quasi-degenerate configuration (dominant plane), which USAC handles with the DEGENSAC algorithm. An adaptation of this algorithm to the *a contrario* methodology will be the subject of another article.

---

[6]http://imagine.enpc.fr/~moulonp/openMVG/

A The Fundamental Matrix

A.1 Two Algebraic Derivations

We present here two more algebraic proofs of the fundamental matrix formula. In the left camera coordinate frame, we can write

$$\lambda x_L = K_L X \qquad\qquad \mu x_R = K_R R^{-1}(X - T), \qquad (28)$$

with $\lambda, \mu > 0$ if the last row of the K matrices is $(0 \quad 0 \quad 1)$, ensuring that image point x is in *front* of the camera iff $x^T e_3 > 0$. Isolating X from the second equation and substituting in the first one, we get

$$\lambda K_L^{-1} x_L = \mu R K_R^{-1} x_R + T. \qquad (29)$$

Taking the cross-product with T of each side yields

$$\lambda [T]_\times K_L^{-1} x_L = \mu [T]_\times R K_R^{-1} x_R, \qquad (30)$$

and the dot product of each term with $K_L^{-1} x_L$ gives the epipolar equation

$$\mu x_L^T K_L^{-T} [T]_\times R K_R^{-1} x_R = 0. \qquad (31)$$

The proof above is the standard one, but another one is interesting as it generalizes to multi-view stereo. It interprets the six equations of (28) as a linear system

$$\begin{pmatrix} K_L & 0_3 & -x_L & 0_3 \\ K_R R^{-1} & -K_R R^{-1} T & 0_3 & -x_R \end{pmatrix} \begin{pmatrix} X \\ 1 \\ \lambda \\ \mu \end{pmatrix} = 0 \qquad (32)$$

and therefore the left-hand side 6×6 matrix must have determinant 0. Multiplying the first column (a 6×3 matrix) by $K_L^{-1} x_L$ and adding it to the third column does not change the determinant

$$\det \begin{pmatrix} K_L & 0_3 & 0_3 & 0_3 \\ K_R R^{-1} & -K_R R^{-1} T & K_R R^{-1} K_L^{-1} x_L & -x_R \end{pmatrix} = 0. \qquad (33)$$

Since K_L has full rank, it amounts to the vanishing 3×3 determinant

$$\det \begin{pmatrix} -K_R R^{-1} T & K_R R^{-1} K_L^{-1} x_L & -x_R \end{pmatrix} = 0. \qquad (34)$$

This also stands when multiplying each column of the matrix by $R K_R^{-1}$ from the left, hence

$$\det \begin{pmatrix} -T & K_L^{-1} x_L & -R K_R^{-1} x_R \end{pmatrix} = 0, \qquad (35)$$

which can be rewritten as (31) by the standard definition of the cross-product. As written above, this method generalizes nicely to more than two views, and rank considerations exhibit the trifocal tensor and show that all multi-linear constraints are combinations of 2- and 3-views constraints [11].

A.2 Chirality

Between (30) and (31), we just used the fact that $\lambda \neq 0 \neq \mu$, but we lost the constraint that they should be positive, as noted by Chum et al. [1]. Notice that this stands in *oriented* projected geometry, that is, image points x_L, x_R in front of the camera have positive third coordinate and K

matrices have positive determinant. We can still recover this information as follows. Multiplying each side of (30) by K_L^{-T} yields

$$\lambda K_L^{-T}[T]_\times K_L^{-1} x_L = \mu K_L^{-T}[T]_\times R K_R^{-1} x_R, \tag{36}$$

and using the identity $[Ax]_\times = \det(A) A^{-T}[x]_\times A^{-1}$ on the left-hand side and (4) on the right-hand side

$$\lambda [K_L T]_\times x_L = \mu \det(K_L) F x_R. \tag{37}$$

Recognizing $K_L T = e_L$ the left epipole, we get

$$e_L \times x_L \overset{+}{\sim} F x_R, \tag{38}$$

with $\overset{+}{\sim}$ meaning equality up to *positive* scale. In practice, e_L and F are computed up to an unknown scale that can be negative, so we cannot use directly the positivity in (38). But once e_L and F are fixed, we can just replace e_L by $-e_L$ if we observe that (38) is not satisfied for a particular pair (x_L, x_R); then all other correspondences should satisfy (38). One issue often overlooked, including in the USAC source code [16], is the possibility that $x_L = e_L$ or $x_R = e_R$, in which case the standard inlier test makes no sense, because there is no epipolar line associated. The right correspondence is naturally (e_L, e_R), and this could be checked, but we find it simpler to just reject such cases. For performance reasons, we only proceed with the chirality test for the correspondences in the minimal sample. We reject the sample if one of the terms of (38) is close to 0, scale of the data being accounted in the test. The chirality test is summed up in Algorithm 1. Notice that to compute the left epipole, an SVD of F could be performed, but it is actually simpler to just take the cross-product of two independent columns of F: assuming they are the first two columns F_1 and F_2, noting $e = F_1 \times F_2$, we have

$$e^T \begin{pmatrix} F_1 & F_2 & \alpha F_1 + \beta F_2 \end{pmatrix} = \begin{pmatrix} e^T F_1 & e^T F_2 & e^T(\alpha F_1 + \beta F_2) \end{pmatrix} = 0. \tag{39}$$

It is possible for one column (but at most one) of F to be null, so we keep the cross-product of two columns with maximal norm, see Algorithm 2.

Algorithm 1: Chirality test of correspondence sample. The threshold 10^{-5} rejects correspondences where one endpoint is close to the epipole.

Input: Sample of correspondences $(x_L^i, x_R^i)_{i=1,\dots,7}$, candidate epipolar matrix F with $\|F\| = 1$
Output: Validity test of F (a boolean)
for $i = 1$ **to** 7 **do**
 $v_L \leftarrow e_L \times x_L^i$ *// Left hand side of* (38)
 $v_R \leftarrow F x_R^i$ *// Right hand side of* (38)
 $d \leftarrow v_L \cdot v_R$ *// Scalar product of both*
 $t \leftarrow \|x_L\|\|x_R\|$
 if $d \leq 10^{-5} t$ **then**
  if $i = 1$ **and** $d < -10^{-5} t$ **then**
   $e_L \leftarrow -e_L$
  else
   return *false*
return *true*

Algorithm 2: Function `leftEpipole`, returning a unit vector representing the left epipole.

Input: 3×3 matrix written in columns $F = \begin{pmatrix} F_1 & F_2 & F_3 \end{pmatrix}$

$\|e_L\| \leftarrow -1$

for *pairs* $(i, j) \in \{(1, 2), (1, 3), (2, 3)\}$ **do**

    if $\|F_i \times F_j\| > \|e_L\|$ **then**

        $e_L \leftarrow F_i \times F_j$

Output: left epipole $e_L / \|e_L\|$

A.3 Rank 2 Constraint

The necessary condition of rank 2 for a fundamental matrix is also sufficient: any 3×3 matrix F of rank 2 is a fundamental matrix. Writing the SVD of such a matrix

$$F = U \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T = U \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T = M \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T, \tag{40}$$

and using QR decomposition of $M^T = QK^{-1}$ with unitary matrix Q and upper triangular matrix K with positive diagonal elements, we get

$$F = K^{-T} Q^T \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^T$$

$$= K^{-T} \left(Q^T \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} Q \right) \left(Q^T \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^T \right). \tag{41}$$

The left parenthesized part is skew-symmetric, which can be written $[T]_\times$, and the right parenthesized part is composed of orthogonal matrices, so it is an orthogonal matrix R. We recognize the fundamental matrix formula with $K_L = K$ and K_R the identity matrix.

A.4 Degeneracy

It is well known that the fundamental matrix cannot be estimated uniquely when only planar points are observed. Indeed, in that case left and right points are related by a homography H such that $x_R = H x_L$. We can then rewrite the epipolar constraint (3) as

$$x_L^T F H x_L = 0, \tag{42}$$

which is always satisfied if FH is skew-symmetric. For a fixed H, the skew-symmetry of FH imposes only 6 linear constraints on F. The condition $\det F = 0$ is already a consequence of these linear constraints, since $\det(FH) = (\det F)(\det H) = 0$ implies that $\det F = 0$, as $\det H \neq 0$. That means there are two undetermined degrees of freedom in F. Another way to see this is to rewrite (5) as

$$A_i^T = \begin{pmatrix} x_i^2 & x_i y_i & x_i & y_i x_i & y_i^2 & y_i & x_i & y_i & 1 \end{pmatrix} \begin{pmatrix} H^T & 0 & 0 \\ 0 & H^T & 0 \\ 0 & 0 & H^T \end{pmatrix}. \tag{43}$$

The 9×9 matrix on the right has full rank, so the rank of A only depends on the dimension of the space spanned by the vectors on the left. The second and fourth columns are the same, so as the third and the seventh, the sixth and the eighth. This shows that in general A has rank 6, except when the points x_L^i are on a conic curve, in which case the rank is at most 5.

B Cubic Polynomial Root Solver

In the 7-point algorithm, we need to find the real roots of a degree 3 polynomial. This section details the solver used and highlights some numerical considerations. We write the polynomial as

$$P(x) = x^3 + ax^2 + bx + c, \tag{44}$$

with a, b, c some real numbers. Obviously, the equation $P(x) = 0$ has between 1 and 3 real solutions, the roots of the polynomial. A classical step is the change of variable $x \to x + a/3$, which eliminates the quadratic term

$$P(x) = (x + a/3)^3 + (b - a^2/3)(x + a/3) + 2(a/3)^3 - ab/3 + c. \tag{45}$$

Writing $a' = a/3$, we get

$$Q(x) := P(x - a') = x^3 + 3px + 2q \tag{46}$$

with

$$p = (b - 3a'^2)/3 \qquad\qquad q = (2a'^3 - a'b + c)/2. \tag{47}$$

The roots of P are deduced from the roots of Q by subtracting a', so we study the polynomial Q. Its derivative Q' is $Q'(x) = 3(x^2 + p)$. If $p > 0$, we have $Q' > 0$, Q is strictly increasing and has a single root. If $p = 0$, we have the single root $x = (-2q)^{1/3}$, with multiplicity 3 if $q = 0$. If $p < 0$, Q has a local maximum at $x_- = -\sqrt{-p}$ and a local minimum at $x_+ = \sqrt{-p}$. We get 3 real roots (counting multiplicity) if and only if $Q(x_-) \geq 0 \geq Q(x_+)$. Since Q is decreasing in $[x_-, x_+]$, the latter condition amounts to $Q(x_-)Q(x_+) \leq 0$. We have

$$Q(x_\pm) = 2\left(q \mp (-p)^{3/2}\right), \tag{48}$$

hence

$$Q(x_-)Q(x_+) = 4d \text{ with } d = q^2 + p^3. \tag{49}$$

B.1 Cardano's Formula (Case $d > 0$)

Notice that the case $p > 0$ discussed above is included in the case $d > 0$, so as the case $p = 0 \neq q$. The method credited to Gerolamo Cardano decomposes the root x as a sum $x = u + v$. This yields

$$u^3 + v^3 + 3(uv + p)(u + v) + 2q = 0. \tag{50}$$

A sufficient condition is to have the two equations

$$u^3 + v^3 = -2q \qquad\qquad uv = -p. \tag{51}$$

Taking the cube of the latter equation, we find that u^3 and v^3 are roots of the polynomial $x^2 + 2qx - p^3$, whose discriminant is $4d \geq 0$. This amounts to writing the root x of Q as

$$x = (-q + \sqrt{d})^{1/3} + (-q - \sqrt{d})^{1/3}. \tag{52}$$

B.2   Viète's Method (Case $d \leq 0$)

The method of François Viète relies on the trigonometric identity

$$\cos 3\theta = 4\cos^3 \theta - 3\cos\theta. \tag{53}$$

Writing the roots x of Q as $x = k\cos\theta$ with $k < 0$, we get

$$k^3 \cos^3 \theta + 3pk\cos\theta + 2q = 0. \tag{54}$$

Comparing with (53), an appropriate choice of k satisfies $k^3 = -4pk$ and therefore $k = -2\sqrt{-p}$, which makes sense since $p^3 \leq d \leq 0$. This yields $(-p)^{3/2}\cos 3\theta = q$ and the roots of Q are

$$x_i = -2\sqrt{-p}\cos\frac{\cos^{-1}\left(q/(-p)^{3/2}\right) + 2i\pi}{3} \quad (i = 0, 1, 2). \tag{55}$$

B.3   Numerical Considerations

The distinction between cases $d > 0$ and $d \leq 0$ is numerically fuzzy when d is barely positive. In this case, we give preference to the 3-root case ($d \leq 0$) in order to avoid missing a possible double root. That means we assimilate the case where d is slightly positive with $d \leq 0$. In view of (47), we consider that each term of the sum may involve an error factor of $(1 + \epsilon)$, hence we define the tolerance factors

$$\tau_p = \max(|b|, 3a'^2) \qquad\qquad \tau_q = \max(2|a'|^3, |a'b|, |c|), \tag{56}$$

and consider that p and q can be respectively $p + \epsilon\tau_p/3$ and $q + \epsilon\tau_q/2$, with ϵ the smallest floating-point increment to unity, i.e., $1 + \epsilon > 1$ (generally $\epsilon = 2^{-23}$ for single precision floating point). Ignoring higher order terms in ϵ, we have

$$(p + \epsilon\tau_p/3)^3 \sim p^3 + \epsilon p^2 \tau_p \qquad\qquad (q + \epsilon\tau_q/2)^2 \sim q^2 + \epsilon q\tau_q, \tag{57}$$

and we take as tolerance threshold for d the value

$$\tau_d = \epsilon \max(p^2 \tau_p, |q|\tau_q). \tag{58}$$

We apply Cardano's formula (52) only when $d > \tau_d$.

Equation (55) only makes sense when $p < 0$ and $d \leq 0$. Since we go into that situation whenever $d \leq \tau_d$, we must take precautions in case d is slightly positive. First, if p is close to 0, q must also be close to 0, and we have the value 0 as triple root. If p is not close to 0, we still explicitly check if $q' = q/(-p)^{3/2}$ is between -1 and 1 because numerical approximations may fail this test. We replace the $\cos^{-1}$ by π if $q' \leq -1$ and by 0 if $q' \geq 1$. To discriminate the case $p \sim 0$ (triple root) and $p < 0$ (three roots), we compare p with $-\epsilon\tau_p/3$. Notice that no root in (55) can exceed $2\sqrt{|p|}$, and we consider that $p \geq -\epsilon\tau_p/3$ entails an acceptable error bound of $2\sqrt{\epsilon\tau_p/3}$ compared to the triple root 0.

Also worth considering, one of the terms $-q \pm \sqrt{d}$ of (52) is subject to dramatic cancellation if p is small compared to q. In any case, the term $t_1 = (|q| + \sqrt{d})^{1/3}$ is safe to compute and the term $t_2 = (|q| - \sqrt{d})^{1/3}$ is better evaluated as $t_2 = -p/t_1$. Now we observe that if $q \leq 0$, $x = t_1 + t_2$ and if $q > 0$, $x = -(t_1 + t_2)$.

B.4 Numerical Tests

The unit test `cubicRoots_test.cpp` chooses random roots in the range $[-25, 25]$. We then apply the formulas (59) below to build P and look at the errors of the evaluated roots. We test all cases: single real root, three distinct roots, one double root (and one distinct single root) and a triple root. Noting z_i the (possibly complex) roots of P, we have

$$a = -(z_1 + z_2 + z_3) \qquad b = z_1 z_2 + z_2 z_3 + z_3 z_1 \qquad c = -z_1 z_2 z_3. \qquad (59)$$

A root x with multiplicity one is considered a function of the coefficients a, b and c in the implicit equation

$$x(a, b, c)^3 + ax(a, b, c)^2 + bx(a, b, c) + c = 0, \qquad (60)$$

and we can compute the gradient of this function

$$\nabla_{a,b,c} x = -\frac{1}{3x^2 + 2ax + b} \begin{pmatrix} x^2 & x & 1 \end{pmatrix}^T. \qquad (61)$$

A sensible tolerance around x for the estimated root r is a solution to (60) with a, b and c perturbed by amounts up to $3 \times 25\epsilon$, $6 \times 25^2 \epsilon$ and $3 \times 25^3 \epsilon$. These amounts are obtained from (59) by considering that each term z_i can vary up to a factor 25ϵ (25 is the maximum value of any z_i) and ignoring higher order terms in ϵ. Therefore, we accept r as a good estimation of x if

$$|x - r| \leq 25\epsilon(3x^2 + 6 \times 25x + 3 \times 25^2)/|3x^2 + 2ax + b|. \qquad (62)$$

This analysis is not valid near a double or a triple root since then $3x^2 + 2ax + b = 0$. In this case we just check that any root satisfies $|P(r)| \leq 0.01$, even though this does not ensure that r is close to the double root. Notice that the double root case is tricky since a small perturbation may tip the balance between one and three solutions. For this reason, we do not consider it a mistake if the double root is missed by the algorithm, still we require the third root (with multiplicity one) to be found.

We show in Figure 13 the cumulated histograms of error distribution over 10^5 simulations of each situation. The error scale goes geometrically after 10^{-5} to better visualize larger errors. We compare our implementation to the GNU Scientific Library (GSL) solver[7]. We notice that the single and three roots situations yield fairly equivalent precision[8], but that we get much better treatment of double and triple roots. The main difference is due to the use of $\tau_p = \tau_q = \tau_d = 0$ by the GSL. In the case of a double root, our algorithm detects it in 96.6% of cases (51.8% for GSL), and in 0.5% of cases (4.2% for GSL) do we misinterpret a triple root as three single roots. Overall, we have only 5 instances of the test (62) failing, all with a value of x close to 0.

Acknowledgment

---

[7]https://www.gnu.org/software/gsl/, version 2.1, file `poly/solve_cubic.c`, modified to compute in single precision, as in our solver.

[8]with a small advantage for GSL in the three single roots case, due to one intermediate value computed in double precision.
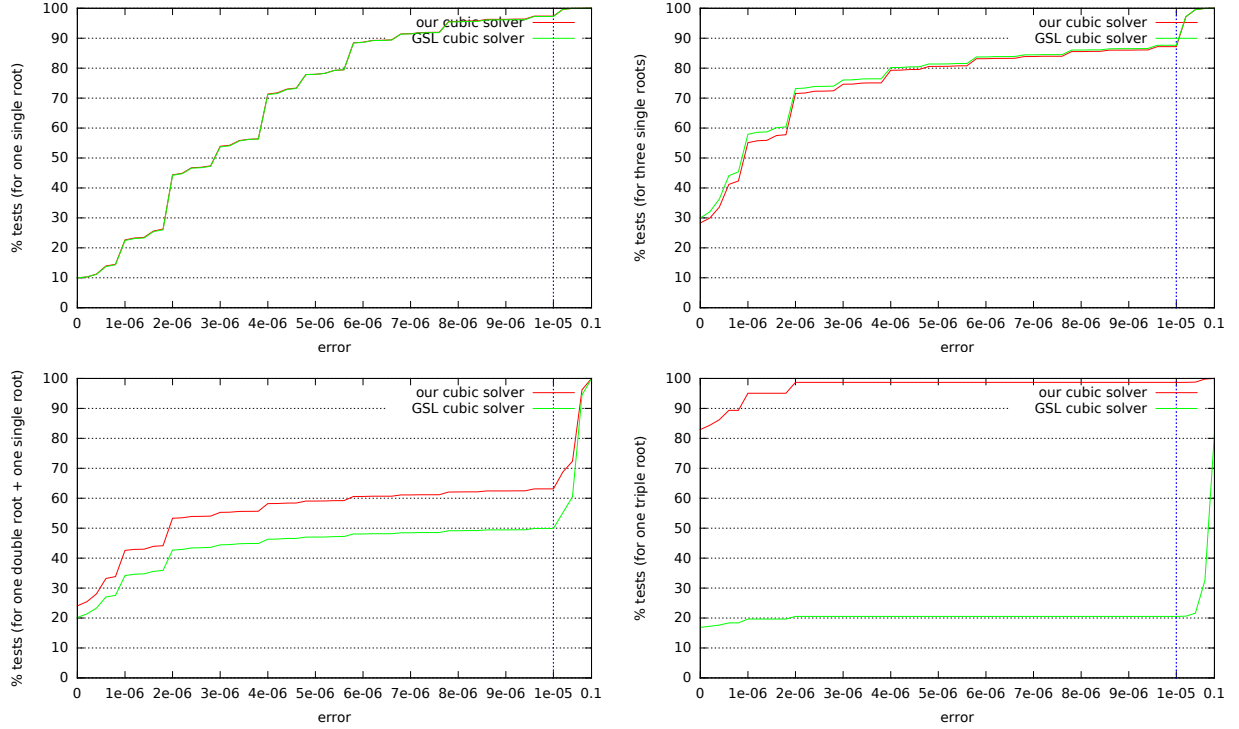
Figure 13: Cumulated histograms of error distributions (single floating point precision) with our cubic polynomial root solver in comparison to the one of the GNU Scientific Library (GSL). The horizontal progression of error is geometrical after `1.0e-5`. Both perform equally with a single root, the GSL solver is a bit more accurate when three roots are present, while our implementation is significantly more accurate with singular cases including a double or a triple root.

Image Credits

All images by the authors (license CC-BY-SA) except:

©Arne Nordmann (CC-BY-SA)[9]

 USAC [16][10]

References

[1] O. Chum, T. Werner, and J. Matas, *Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint*, in Proceedings of the International Conference on Pattern Recognition (ICPR), IEEE, 2004, pp. 112–115. http://dx.doi.org/10.1109/ICPR.2004.1334020.

[2] ——, *Two-view geometry estimation unaffected by a dominant plane*, in Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2005, pp. 772–779. http://dx.doi.org/10.1109/CVPR.2005.354.

---

[9] https://commons.wikimedia.org/wiki/File:Epipolar_geometry.svg
[10] http://www.cs.unc.edu/~rraguram/usac/USAC-1.0.zip

[3] A. Desolneux, L. Moisan, and J.-M. Morel, *From gestalt theory to image analysis: a probabilistic approach*, vol. 34, Springer Science & Business Media, 2007. ISBN 978-0-387-72635-9, http://dx.doi.org/10.1007/978-0-387-74378-3.

[4] F. Espuny, P. Monasse, and L. Moisan, *A new a contrario approach for the robust determination of the fundamental matrix*, in Image and Video Technology–PSIVT 2013 Workshops, vol. 8334 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2014, pp. 181–192. http://dx.doi.org/10.1007/978-3-642-53926-8_17.

[5] M.A. Fischler and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, 24 (1981), pp. 381–395. http://dx.doi.org/10.1145/358669.358692.

[6] R.I. Hartley, *In defense of the eight-point algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 19 (1997), pp. 580–593. http://dx.doi.org/10.1109/34.601246.

[7] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2nd ed., 2004. ISBN 978-0521540513.

[8] H.C. Longuet-Higgins, *A computer algorithm for reconstructing a scene from two projections*, Nature, 293 (1981), pp. 133–135. http://dx.doi.org/10.1038/293133a0.

[9] D.G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision (IJCV), 60 (2004), pp. 91–110. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[10] Q.-T. Luong, *Matrice fondamentale et calibration visuelle sur l'environnement. Vers une plus grande autonomie des systèmes robotiques*, PhD thesis, Université Paris Sud-Paris XI, 1992. https://tel.archives-ouvertes.fr/tel-00549134/.

[11] Y. Ma, K. Huang, R. Vidal, J. Košecká, and S. Sastry, *Rank conditions on the multiple-view matrix*, International Journal of Computer Vision (IJCV), 59 (2004), pp. 115–137. http://dx.doi.org/10.1023/B:VISI.0000022286.53224.3d.

[12] L. Moisan, P. Moulon, and P. Monasse, *Automatic homographic registration of a pair of images, with a contrario elimination of outliers*, Image Processing On Line (IPOL), 2 (2012), pp. 56–73. http://dx.doi.org/10.5201/ipol.2012.mmm-oh.

[13] L. Moisan and B. Stival, *A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix*, International Journal of Computer Vision (IJCV), 57 (2004), pp. 201–218. http://dx.doi.org/10.1023/B:VISI.0000013094.38752.54.

[14] P. Monasse, *Quasi-euclidean epipolar rectification*, Image Processing On Line (IPOL), 1 (2011). http://dx.doi.org/10.5201/ipol.2011.m_qer.

[15] P. Moulon, P. Monasse, and R. Marlet, *Adaptive structure from motion with a contrario model estimation*, in Proceedings of the 2012 Asian Conference on Computer Vision (ACCV), vol. 7727 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 257–270. http://dx.doi.org/10.1007/978-3-642-37447-0_20.

[16] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. Frahm, *USAC: A universal framework for random sample consensus*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 35 (2013), pp. 2022–2038. http://dx.doi.org/10.1109/TPAMI.2012.257.

[17] I. Rey Otero and M. Delbracio, *Anatomy of the SIFT method*, Image Processing On Line (IPOL), 4 (2014), pp. 370–396. http://dx.doi.org/10.5201/ipol.2014.82.