Robust 3D Reconstruction by Feature Tracking over Large Displacements

ALDO CUMANI and ANTONIO GUIDUCCI
Computer Vision Lab, Systems Engineering Department
Istituto Elettrotecnico Nazionale Galileo Ferraris
Str. delle Cacce, 91
ITALY
{

Abstract: Accurate recovery of geometric structure from an image sequence strongly depends upon two contrasting requirements: numerical conditioning, needing large image disparities, and ease of matching, which needs small ones. This work discusses a solution for an actively controlled observer (a camera on the end-effector of a robot arm) using feature tracking of image features along the camera trajectory. Restricting the scene to objects with straight line edges allows easy assessment of the reconstruction accuracy.

Key-Words: 3D reconstruction – Feature matching – Trifocal tensor – Reconstruction accuracy

1 Introduction

Any approach to recovering scene structure from image sequences must face the problem of *feature matching*. Indeed, the reconstruction results strongly depend upon the balance of two contrasting requirements: good conditioning, requiring large image disparities between corresponding features, and ease/reliability of matching, for which small disparities are better.

In passive vision, the inter-view displacement is either fixed (stereo head), or uncontrolled (passive navigation). This is not the case when the observer can be actively controlled, as e.g. for a camera mounted on the end-effector of a robot arm. A strategy for the latter case can be sketched as follows:

- grab a number of images of the scene with the camera moving along a predefined trajectory;
- extract image features and track them along the trajectory by inter-frame matching;
- estimate the viewing geometry and the scene structure as soon as the accumulated image disparity is deemed sufficient.

In this way, the reconstruction accuracy from large displacements is reconciled with the ease of matching for small ones. Note that the "predefined" trajectory can in fact be modified on-line (e.g. by tuning the size of the displacement) according to the results of processing. Moreover, the current estimate of the viewing geometry can be used to refine matching by feature transfer [1, 15].

Feature tracking, and the use of the estimated viewing geometry for match refinement, have already been suggested in the cited works. The emphasis of this paper is rather on the effective usability of such results in an actual application, with particular regard to the *accuracy* of the reconstruction. To this extent, we restrict ourselves to scenes from a "blocks world", consisting of objects characterised by planar faces with straight-line edges. With this restriction, a natural choice for image features is that of face vertices. Such features are easily and accurately identifiable on the image plane, and the accuracy of the resulting reconstruction can be assessed against a CAD model of the scene.

2 Notation and preliminaries

Homogeneous coordinates are used for both 3D and 2D objects, so the 3D point of coordinates (x, y, z) is $\mathbf{X} = [x\, y\, z\, 1]^\top$ and a 2D point (u, v) is $\mathbf{x} = [u\, v\, 1]^\top$. A 2D line is represented by a vector $\mathbf{l} = [l_1\, l_2\, l_3]^\top$ such that point $\mathbf{x}$ belongs to $\mathbf{l}$ iff $\mathbf{l}^\top \mathbf{x} = 0$.

Single image acquisition is described by a standard pin-hole. If $\mathbf{x} = [x_1, x_2, x_3]^\top$ is the image of the world point $\mathbf{X} = [X_1, X_2, X_3, X_4]^\top$, then

$$\mathbf{x} = \mathsf{P}\mathbf{X} \quad \text{with} \quad \mathsf{P} = \mathsf{A}\,[\mathsf{R} \mid \mathbf{t}] \tag{1}$$

where the factoring of the *projection matrix* P into a rototranslation $\mathsf{R}, \mathbf{t}$ and an *intrinsic matrix*

$$\mathsf{A} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

only holds for Euclidean world/image plane frames.

The above does not take into account *lens distortion*, which is seldom negligible, but can usually be fitted by a simple radial model:

$$\mathbf{x} - \mathbf{x}_{0d} = (1 + k_1 \rho^2 + ...)(\mathbf{x}_d - \mathbf{x}_{0d}) \tag{3}$$

where $\mathbf{x}_d$ are the so called distorted coordinates (i.e. those actually measured on the image), and $\mathbf{x}_{0d} = [u_d, v_d, 1]^\top$ the distortion center (which needs not coincide with the principal point $[u_0, v_0, 1]^\top$). Distortion correction can be incorporated in the feature extraction phase (Sec. 3.1).

The relation between corresponding points $\mathbf{x}'$ and $\mathbf{x}''$ in two views of the same scene is defined by the epipolar constraint

$$\mathbf{x}'' \mathsf{F} \mathbf{x}' = 0 \tag{4}$$

where the *fundamental matrix* F can be estimated from seven point matches between the two views (robust computational techniques for more point matches are discussed in [14] and [11]). Eq. (4) simplifies the search for correspondences in the two views: as $\mathbf{x}''$ lies on the epipolar line $\mathbf{l}'' = \mathsf{F}\mathbf{x}'$, a bidimensional search reduces to one-dimensional.

In the case of three views of the same scene, an analogous role is played by the *trifocal tensor* T, which allows transfer of corresponding objects (points, lines or combinations of both) among the views. For example, if points $\mathbf{x}'$ and $\mathbf{x}''$ are known, $\mathbf{x}'''$ can be estimated from the trilinearities [10]

$$x'^i (x''^j \varepsilon_{jpr})(x'''^k \varepsilon_{kqs}) T_i^{pq} = \mathbf{0}_{rs} \tag{5}$$

Estimating T needs at least six point correspondences (or nine line correspondences) over three views [1]. With more points available, robust algorithms similar to those devised for F can be used.

From F or T, a set of canonical projection matrices for the two or three camera positions can be computed. However, the amount of scene structure that can be inferred from T or F depends on the available independent information about scene constraints or camera parameters: a full metric (Euclidean) reconstruction needs a *calibrated camera* (i.e. a known A). Although useful information (e.g. point-plane relationships) can be obtained from projective reconstruction only, we shall assume in the sequel that camera parameters are available, as metric reconstruction is usually the ultimate goal.

In all cases, once the camera matrices are available, the scene structure is computed by standard triangulation, possibly taking into account image plane noise. For the two view case, we use the epipolar correction method described in [9, 13].

3 Feature extraction and matching

As the environment considered here is a "blocks world" (objects bounded by planar polygonal faces), *segments* (images of object edges) and *vertices* (images of object corners) are a natural choice for image features. The feature extraction phase consists then of contour line extraction followed by segmentation of the contours into rectilinear strokes, and computation of vertices as intersections of the lines through nearby segments.

3.1 Contour extraction

Contour lines are extracted from the image by applying a second-order differential operator to the Gaussian smoothed image and linking the resulting zero crossing points [7, 2]. This algorithm yields contour lines as lists of image points to sub-pixel precision, which are then corrected for lens distortion using Eq. (3). Each contour point carries information about the behaviour of the luminance in its neighborhood, namely the luminance at the point and an estimate of its total variation across the contour line. These quantities are used to get estimates of the "far" luminances on the two sides of the contour line (i.e. the values of luminance just outside the region of rapid variation which determines the visual contour), needed to better characterise the contour for the subsequent matching phase.

3.2 Segments and vertices

The distortion-corrected contour lines are segmented into rectilinear strokes by a standard algorithm ([12], chap.12), breaking each contour point list C into sublists C_i such that the maximum distance of each point from the line through the first and last points in C_i does not exceed some threshold. For best accuracy, the segment is then represented by the least squares line $\mathbf{l} = [l_0\, l_1\, l_2]^\top$ over the contour points belonging to C_i, with its endpoints $\mathbf{e}_1 = [x_1\, y_1\, 1]^\top$ and $\mathbf{e}_2 = [x_2\, y_2\, 1]^\top$ computed as projections on $\mathbf{l}$ of the first and last points of C_i. Photometric attributes of the segment, namely the "far" luminance values L_R and L_L on either of its sides, are computed as averages of the corresponding features over the points in C_i. A segment S (see Fig. 1) is therefore described by a set of 9 parameters (eight of which independent):

$$S = \{x_1, y_1, x_2, y_2, l_0, l_1, l_2, L_R, L_L\}$$

A *vertex* (or, more precisely, a *face vertex*) can be loosely defined as the image of a physical corner of a polygonal object face. Many "corner detectors" have been proposed in the literature (e.g. [8]). In our framework, however, a more natural and reliable definition of vertex is as intersection of the

lines on which nearby segments lie. Such a definition has the advantages of yielding a good localisation accuracy and of supplementing in a natural way the vertex characterisation with geometric and photometric parameters from the defining segments.

Therefore, given a pair of segments S_1, S_2, the resulting intersection $\mathbf{v} = [v_1\ v_2\ 1]^\top$ is accepted as a vertex position if its image plane distances from the nearest endpoints of S_1 and S_2 are below a given threshold, and if the far luminances of S_1 and S_2 on their sides belonging to the convex angle α formed by the two segments, say L_{c1} and L_{c2}, are *compatible* (i.e. not too different). A vertex V is therefore characterised by 6 parameters (5 independent):

$$V = \{v_1, v_2, \alpha, c, s, L\}$$

where c, s are the direction cosines of the bisecting line of angle α and L the mean luminance over α (average of L_{c1} and L_{c2}).

Note that, when two or more faces incident in a same object corner are simultaneously visible, the vertices from each face form a cluster of nearby vertices, not exactly coincident due to noise (see Fig. 1, where distances have been exaggerated for the sake of clarity). Such nearly coincident vertices do not contribute useful geometric information. Therefore, while every vertex is tracked individually along the sequence, for what concerns geometry/structure estimation the vertices in each group are averaged together and considered as a single point.

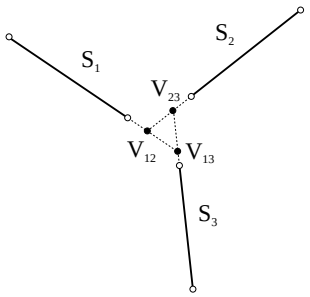

Figure 1: Segments and vertices.

3.3 Matching

Matching requires the definition of a suitable function $\mu_V(V_A, V_B)$ measuring the *similarity* of pairs of candidate corresponding vertices V_A, V_B. This similarity is defined as the product

$$\mu_V \ = \ \mu_{Vd}\, \mu_{V\phi}\, \mu_{V\alpha}\, \mu_{VL} \qquad (6)$$

where the factors

$$\begin{aligned}
\mu_{Vd}(V_A, V_B) &= \frac{d_0^2}{d_0^2+(v_{1A}-v_{1B})^2+(v_{2A}-v_{2B})^2} \\
\mu_{V\phi}(V_A, V_B) &= \tfrac{1}{2}(1 + c_A c_B + s_A s_B) \\
\mu_{V\alpha}(V_A, V_B) &= 1 - \frac{|\alpha_A-\alpha_B|}{\alpha_A+\alpha_B} \\
\mu_{VL}(V_A, V_B) &= 1 - \frac{|L_A-L_B|}{L_A+L_B}
\end{aligned}$$

weigh the image distance of the vertices and their differences in orientation, angular amplitude and luminance. The value of μ_V is always between 0 and 1, with the maximum value attained only if the two vertices are identical and in the same position.

Note that μ_V as defined above is a purely heuristic, hence rather arbitrary, measure; its form has been chosen so as to avoid the introduction of too many arbitrary parameters (just a distance range d_0). With respect to the latter issue, however, it may be noted that we are trying to match very similar features, which allows to safely add some, albeit arbitrary, threshold-like parameters to improve efficiency (e.g. when μ_ϕ is under some threshold value, the other factors need not be evaluated).

4 Tracking/geometry estimation

As said before, the computation of either the fundamental matrix or the trifocal tensor is reliable only if the two or three views used are well displaced and rotated. However, the corresponding images are then very different and the identification of correspondences very difficult. To overcome this difficulty we propose the following approach.

The whole sequence of images taken while moving the camera, is processed. Under the hypothesis that the displacement between two successive frames is small, the vertices in the first image (image I_1) can be tracked along the sequence by matching them over each pair of subsequent frames, according to the above defined μ_V. To this extent, each vertex V_i' in the first frame I' of a pair is associated to the unique V_j'' in the next frame I'', which maximises $\mu_V(V_i', V_j'')$ as defined in Eq. (6), provided that this maximum μ_V is above a specified threshold (otherwise V_i' remains unmatched).

Tracking continues either up to the end of the sequence, or until the number of matches is about to drop under a suitable threshold, greater than six (11 in our tests). If I_3 is the last processed image, an intermediate image I_2 is chosen and the three-view geometry (i.e. T) is estimated by a Least Median Squares (LMedS) algorithm, analogous to the one

described in [14] for robust estimation of F. This estimate is used for match refinement as follows.

For each vertex V_i^1 in I_1 the corresponding epipolar line l_2 in I_2 is computed using the fundamental matrix F_{12} extracted from T. For each vertex V_j^2 near l_2 in I_2 a similarity value μ_{12} to the chosen vertex from I_1 is computed as in Eq. (6), but using its orthogonal distance from l_2 in the distance factor μ_{Vd} instead of the Euclidean distance from V_i^1. The trifocal tensor T is then used to transfer the above pair of vertices to I_3 via Eq. (5). Again, a similarity value μ_{23} is computed for each vertex V_k^3 in I_3, using this time its distance from the transferred point. The three vertices in the three images with the best overall similarity factor $\mu_{123} = \mu_{12}\mu_{23}$ are retained as a triple match if the corresponding best μ_{123} is above a predefined threshold, otherwise the corresponding vertex on the first image is left unmatched (perhaps, it is hidden by a nearby object in some frame following the first).

Unless I_3 is the last frame of the sequence (i.e. the disparity between I_1 and I_3 is deemed sufficient for an accurate reconstruction), this procedure is iterated, so obtaining a set of key images $\mathcal{I} = (I_1..I_N)$ such that an estimate of T has been computed on each triple I_1, I_k, I_{k+1} and used to refine the matches over the triple. At last, a global fundamental matrix F is estimated from all the matches available between I_1 and I_N using Kanatani's method [11]; this F is used for scene reconstruction by backprojection, after applying Sturm's epipolar correction [9, 13].

5 Experimental results

The procedures described in Sec. 4 were implemented as a set of C programs, and tested offline on a number of image sequences. We tested both the accuracy of the reconstruction, by comparing the latter with a CAD model of the scene, and its sensitivity to the calibration parameters.

The camera was a Sony XC55 Progressive CCD camera, equipped with a 6 mm lens and with the shutter adjusted to a speed of 1/100 s. Images were acquired via a Matrox Meteor board mounted on a standard Pentium PC, yielding non-interlaced images of 640×480 8-bit pixels. A full calibration of the camera intrinsic parameters (A and the distortion coefficients) was performed using the method described in [4]. Table I summarises the camera parameter values.

The "world" was made up of white-painted wooden blocks of various shapes, placed at known positions over a sheet of dark paper. Two sequences were taken, one with the camera moved manually (HAND sequence), the other with the camera mounted on the end-effector of an industrial robot programmed to follow a smooth trajectory (ROBOT sequence). Several tests were performed on these sequences; in each test, a pair of initial and final views I_1, I_N were chosen for the purpose of 3D reconstruction, and the algorithm selected other two intermediate views so yielding a set $\mathcal{I}$ of four key frames. In the following we report the results of three tests, one for the HAND sequence and two for the ROBOT one.

k_1	u_d	v_d		
3.06e-7	348.0	207.7		
f_u	f_v	s	u_0	v_0
826.2	828.1	0.5	332.8	223.0

Table I: Camera calibration parameters.

Table II summarizes the results for the three tests. In each row, "initial matches" and "refined matches" refer to the number of matched vertices between the first and last image of the corresponding triple, respectively before and after the refinement described in Section 4. The "total" column counts the actual number of matched face vertices, while "distinct" is the number of vertices useful for geometry estimation (note that the number of distinct refined matches is only relevant for the final iterate).

Fig. 2 shows the first and last key frames, and the corresponding segments and vertices, for the second ROBOT tests (other images omitted for reasons of space).

Fig. 3 shows matched vertices between the first and last images of one of the ROBOT subsequences; it is worth noting that it would be rather difficult to get reliable matches as those shown using the first and last images alone.

Four orthographic views of the reconstruction from the same test are shown in Fig. 4. The reconstructed object edges shown in the figure were determined from the links between vertices supplied by the generating segments, as already pointed out in Sec. 3.2. In order to estimate the accuracy of the algorithm, the reconstructed scene was matched against a CAD model of the same, using an ad hoc software that also estimates the rotation, translation and scaling which minimizes the r.m.s. distances between model and reconstructed vertices. The latter value is shown in the column labelled "position" of Table II, while the last column ("length") reports the r.m.s. difference between the lengths of model

sequence	key images			initial matches		refined matches		rec. error [mm]	
				total	distinct	total	distinct	position	length
HAND (220-400)	220	265	310	19	18	41			
	220	310	400	19	16	39	26	1.2	1.0
ROBOT (10-340)	10	90	170	17	14	27			
	10	170	340	21	18	27	21	1.9	0.9
ROBOT (170-495)	170	255	340	25	20	40			
	170	340	495	21	18	38	23	1.2	0.9

Table II: Results for three test sequences (see text for explanation).

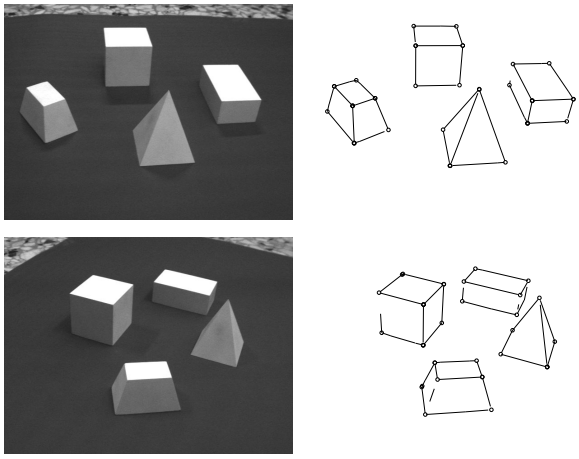

Figure 2: Initial and final key frames of the second ROBOT test, and corresponding segments and vertices.

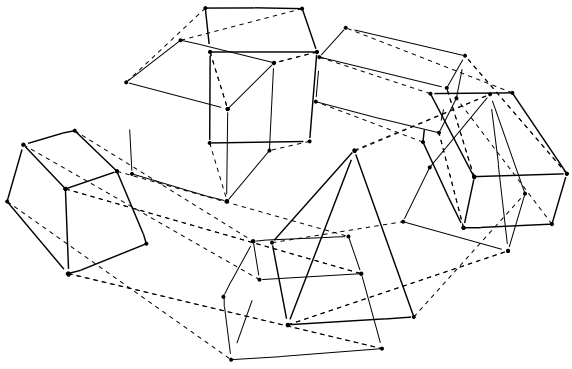

Figure 3: Matched vertices between the first and last images of the second ROBOT test.

and reconstructed object edges.

For what concerns the sensitivity to calibration, we considered the effect on the r.m.s. position error of changing either the camera focal length, or the optic center (u_0, v_0). For reasons of space, we report here some results only for one of the ROBOT tests. Fig. 5(top) shows the position error as a function of focal length, while keeping the optic center fixed to its calibrated position; note that here we have assumed equal focal lengths $f_u = f_v$ and zero skew, which is justified by the values in Table I. Fig. 5(bottom) shows the effect of changing the optic center. These figures, while attesting the goodness of the calibration, show that small errors (of the order of 5-10%) in the calibration parameters do not yield dramatic changes in the reconstruction.

6 Concluding remarks

Reconstructing the view geometry and the scene structure from a series of uncalibrated or partially calibrated views is a problem that has received considerable attention in the last years. Yet, the fundamental task of feature matching, essential to reconstruction, still poses some problems. In this work we have discussed a solution for a restricted case, namely that of an active observer in an environment consisting of objects mostly characterised by straight edges, with particular emphasis on the accuracy of the obtained 3D reconstruction. We have found that tracking image vertices as defined here allows a rather accurate estimation of the relative positions of the corresponding object corners. In view of a full quantitative reconstruction, however, more work is needed for what concerns the determination of scene topology, i.e. linking corners into edges, edges into faces, faces into solids.

Acknowledgments - The authors wish to thank Prof. Basilio Bona of the Politecnico di Torino for providing access to the robot used in some of the tests. This work was partly supported by the Italian Space Agency (ASI) under grant no. I/R/46/00.

References:

[1] P. Beardsley, P. Torr and A. Zisserman, 3D model acquisition from extended image sequences. *Proc. 4th European Conference on Computer Vision*, 1996, pp. 683–695.

[2] A. Cumani, Efficient contour extraction in color images. *Proc. 3rd Asian Conf. on Comp. Vision*, 1998, pp. 582–589.

[3] A. Cumani and A. Guiducci, Robust 3D reconstruction by feature tracking over large displacements. IEN Tech. Rep. 616, 2000.

[4] A. Cumani, Simple and Accurate Camera Calibration. IEN Tech. Rep. 631, 2001.

[5] O. Faugeras and B. Mourrain B, On the geometry and algebra of the point and line correspondences between N images. INRIA Tech. Rep. RR-2665, 1995.

[6] O. Faugeras, Stratification of three-dimensional vision: projective, affine, and metric representations. *Jour. Opt. Soc. Am. A*, vol. 12, no. 3, 1995, pp. 465–484.

[7] P. Grattoni and A. Guiducci, Contour coding for image description. *Pattern Recognition Letters*, vol. 11, no. 2, 1990, pp. 95–105.

[8] C. Harris and M. Stephens, A combined corner and edge detector. *Proc. 4th Alvey Vision Conference*, 1988, pp. 147–152.

[9] R. I. Hartley and P. Sturm, Triangulation. *Proc. ARPA Image Understanding Workshop*, 1994, pp. 957–966.

[10] R. Hartley and A. R, Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[11] K. Kanatani, Optimal fundamental matrix computation: algorithm and reliability analysis. *Proc. Sixth Symp. Sensing via Imaging Information (SSII2000)*, 2000, pp. 291–296.

[12] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Springer–Verlag, 1982.

[13] P. F. Sturm, Vision 3D non calibrée: contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage. Ph.D. Thesis, Institut National Polytechnique de Grenoble, 1997.

[14] Z. Zhang, Determining the epipolar geometry and its uncertainty: a review. *Int. Jour. Computer Vision*, vol. 27, no. 2, 1998, pp. 161–195.

[15] A. Zisserman, A. W. Fitzgibbon and G. Cross, VHS to VRML: 3D Graphical Models from Video Sequences. *IEEE Int. Conf. on Multimedia and Systems*, 1999, pp. 51–57.

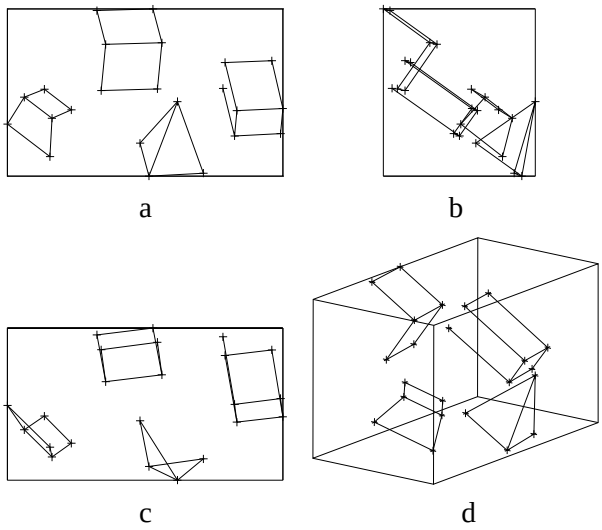

Figure 4: Four orthographic views of the Euclidean reconstruction from the second ROBOT test. a,b,c: along the axes of the camera reference frame (relative to the first image); d: along an intermediate direction. The bounding box (relative to the camera axes) of the reconstructed points is also shown.

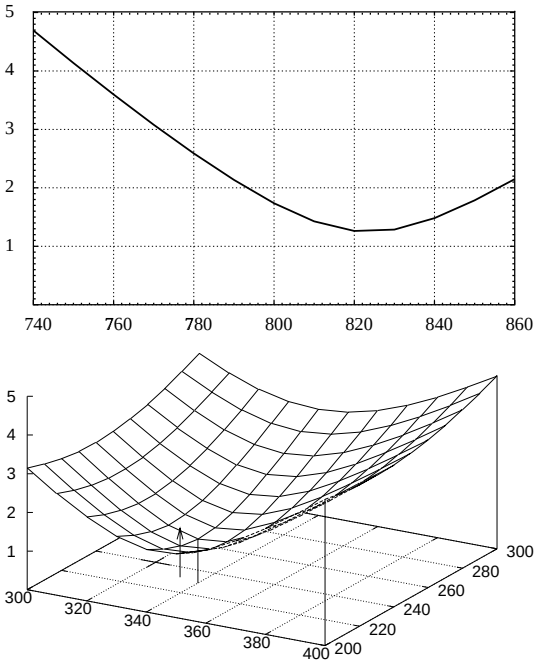

Figure 5: Position error [mm]. top: as a function of focal length in pixels; bottom: as a function of optic center position. The arrow indicates the calibrated position, while the nearby vertical stroke denotes the position of the minimum error.