3—1 Nonlinear Refinement of Camera Parameters using an Endoscopic Surgery Robot

Jochen Schmidt, Florian Vogt,* and Heinrich Niemann
Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, 91058 Erlangen, Germany
E-mail: {jschmidt,vogt,niemann}@informatik.uni-erlangen.de
www: http://www5.informatik.uni-erlangen.de

Abstract

We present an approach for nonlinear optimization of the parameters of an endoscopic camera mounted on a surgery robot. The goal is to generate a depth map for each image in order to enhance the quality of medical light fields. The pose information provided by the robot is used as an initialization, where especially the orientation is inaccurate. Refinement of intrinsic and extrinsic camera parameters is performed by minimizing the back-projection error of 3-D points that are reconstructed by triangulation from image features tracked over an image sequence.

Optimization of the camera parameters results in an enhancement of rendering quality in two ways: More accurate parameters lead to better interpolation as well as to better depth maps for approximating the scene geometry.

1 Introduction

In order to introduce computer support into minimal-invasive surgery, methods from image-based rendering – so-called light fields [2, 5] – were used recently for generating realistic models of human organs and for substituting highlights [10]. These medical light fields were generated using a structure from motion approach [4] to compute camera motion and 3-D scene geometry simultaneously. In that approach it is necessary to track feature points over many images of the image sequence assuming a rigid scene. However, when reconstructing a light field from an endoscopic image sequence, this rigidity constraint usually does not hold. The reasons are e. g. respiration and heart beat of the patient. This results in either badly calibrated cameras and wrong depth information or in no reconstruction at all, i. e. no light field.

When using an endoscopic surgery robot like the AESOP 3000 from Computer Motion Inc., it is possible to get information on the camera positions directly from the robot without computing them by structure from motion algorithms. But since the endoscope has to be mounted manually and the camera's orientation cannot be fixed exactly, the camera pose provided by the robot is not accurate. Additional inaccuracies are introduced by the robot motions themselves, which lead to inconsistent camera poses that result – compared to structure from motion approaches – in very high back-projection errors of triangulated 3-D points.

Depth maps for each image can be used for improving light field rendering quality considerably. In our approach those depth maps are computed using a stereo approach for dense disparity maps as described in [7]. The problem that arises here is the robot's accuracy, especially considering the orientation of the camera which is important for rectification and disparity computation using [7]. That is why we apply a nonlinear refinement step using the camera parameters obtained from the robot as an initialization in order to improve the depth maps and thus light field quality.

In the following we are going to describe the methods used for optimization and show the resulting depth maps before and after optimization.

2 Computation of 3-D Points

For nonlinear refinement as described in the next Section, 3-D points and their corresponding image points are required. Since the robot provides only its movement but no 3-D information about the scene, we have to apply a tracking algorithm at the beginning that results in correspondences between image points over many frames. From those correspondences 3-D points can be triangulated. Note that we do not use those correspondences for applying a structure from motion algorithm, since that information is readily available from the robot itself. Here is an overview of the 3-D point computation:

- Undo radial lens distortions, which are actually very dominant in endoscopic images. The intrinsic camera parameters needed for that step were calibrated in a previous step using a calibration pattern, since they do not change during camera movement.
- Detect and track feature points q_{ij} over the image sequence where the camera parameters are to be optimized. For this purpose we apply the tracking method described in [9] with the extension made in [8] for estimating affine transformations between the feature windows. In contrast to the computation of camera parameters using the structure from motion approach only, a small number of features which could be tracked over the frames are sufficient.
- Triangulate 3-D points w_j from the corresponding image points q_{ij} using a least squares method.

3 Nonlinear Refinement

The camera parameters obtained from the robot can be described by a linear mapping using a 3×4 projection ma-

[*] This work was partially funded by the German Research Foundation (DFG) under grant SFB 603/TP B6. Only the authors are responsible for the content.

trix. We assume a perspective camera model. A homogeneous 3-D point $\boldsymbol{w}_j$ is projected onto a homogeneous 2-D point $\boldsymbol{q}_{ij}$ in frame i using the following equation:

$$\boldsymbol{q}_{ij} = \boldsymbol{P}_i \boldsymbol{w}_j = \boldsymbol{K}_i \boldsymbol{R}_i^{\mathrm{T}} (\boldsymbol{I}_3 | -\boldsymbol{t}_i) \boldsymbol{w}_j \quad , \tag{1}$$

where $\boldsymbol{K}_i$ is a 3×3 matrix containing the intrinsic parameters f_{x}, f_{y}, u_0, and v_0, $\boldsymbol{R}_i$ is a rotation matrix whose columns correspond to the axes of the camera coordinate system, $\boldsymbol{t}_i$ is a translation vector giving the position of the camera's optical center, and $\boldsymbol{I}_3$ is the 3×3 identity matrix. The intrinsic parameter matrix $\boldsymbol{K}_i$ as well as lens distortions (which are not modeled by (1)) are obtained by a calibration step before using the robot. Thus each image is already undistorted when the following steps are applied.

An optimal way (in the sense of Maximum-Likelihood estimation) to do nonlinear refinement is optimization of the back-projection error of 3-D points in all images, which is given by:

$$\min_{\boldsymbol{P}_i, \boldsymbol{w}_j} \sum_{j=1}^{n} \sum_{i=1}^{m} \left(\left(x_{ij} - \frac{\boldsymbol{p}_{i1}^{\mathrm{T}} \boldsymbol{w}_j}{\boldsymbol{p}_{i3}^{\mathrm{T}} \boldsymbol{w}_j} \right)^2 + \left(y_{ij} - \frac{\boldsymbol{p}_{i2}^{\mathrm{T}} \boldsymbol{w}_j}{\boldsymbol{p}_{i3}^{\mathrm{T}} \boldsymbol{w}_j} \right)^2 \right) . \tag{2}$$

where m is the number of frames and n the number of 3-D points. The detected image feature points are denoted by (x_{ij}, y_{ij}), $\boldsymbol{p}_{ik}^{\mathrm{T}}$ $(k = 1, 2, 3)$ are the row vectors of the projection matrix $\boldsymbol{P}_i$. Minimization of this function usually is referred to as *bundle-adjustment* [3].

Minimization of (2) is equivalent to minimizing the following expression, which is called *interleaved* bundle-adjustment, which results in a time complexity of $O(nm^3)$ instead of $O(nm^3 + n^2m^2 + n^3m)$ for minimization of (2):

$$\min_{\boldsymbol{P}_i} \sum_{j=1}^{n} \min_{\boldsymbol{w}_j} \sum_{i=1}^{m} \left(\left(x_{ij} - \frac{\boldsymbol{p}_{i1}^{\mathrm{T}} \boldsymbol{w}_j}{\boldsymbol{p}_{i3}^{\mathrm{T}} \boldsymbol{w}_j} \right)^2 + \left(y_{ij} - \frac{\boldsymbol{p}_{i2}^{\mathrm{T}} \boldsymbol{w}_j}{\boldsymbol{p}_{i3}^{\mathrm{T}} \boldsymbol{w}_j} \right)^2 \right) . \tag{3}$$

This is called *interleaved* because in each optimization step of the camera parameters an optimization of all 3-D points is performed, which can be done separately for each point.

For the purpose of nonlinear optimization the Gauss-Newton algorithm with Levenberg-Marquardt extension (see [3] for details) is utilized which computes a new estimate of a parameter vector $\boldsymbol{a}$ (containing the camera parameters) using a local parametrization $\Delta \boldsymbol{a}$ by $\hat{\boldsymbol{a}}_{k+1} = \hat{\boldsymbol{a}}_k + \Delta \boldsymbol{a}$ where

$$\Delta \boldsymbol{a} = -\left(\lambda \boldsymbol{I} + \boldsymbol{J}^{\mathrm{T}} \boldsymbol{J} \right)^{-1} \boldsymbol{J}^{\mathrm{T}} \boldsymbol{\epsilon}(\hat{\boldsymbol{a}}_k) \quad . \tag{4}$$

This method minimizes the mean square error $\boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a residual function that computes in our case the (non-squared) back-projection error between each image feature point (x_{ij}, y_{ij}) and the projection of its corresponding 3-D point $\boldsymbol{w}_j$:

$$\boldsymbol{\epsilon} = \left(x_{11} - \frac{\boldsymbol{p}_{11}^{\mathrm{T}} \boldsymbol{w}_1}{\boldsymbol{p}_{13}^{\mathrm{T}} \boldsymbol{w}_1}, y_{11} - \frac{\boldsymbol{p}_{12}^{\mathrm{T}} \boldsymbol{w}_1}{\boldsymbol{p}_{13}^{\mathrm{T}} \boldsymbol{w}_1}, \ldots, \right. \\ \left. x_{mn} - \frac{\boldsymbol{p}_{m1}^{\mathrm{T}} \boldsymbol{w}_n}{\boldsymbol{p}_{m3}^{\mathrm{T}} \boldsymbol{w}_n}, y_{mn} - \frac{\boldsymbol{p}_{m2}^{\mathrm{T}} \boldsymbol{w}_n}{\boldsymbol{p}_{m3}^{\mathrm{T}} \boldsymbol{w}_n} \right) \quad . \tag{5}$$

$\boldsymbol{J}$ is the Jacobian of $\boldsymbol{\epsilon}$ evaluated at $\hat{\boldsymbol{a}}_k$: $\boldsymbol{J} = \frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{a}}(\hat{\boldsymbol{a}}_k)$. Since the matrix inversion in equation (4) may be numerically instable due to a nearly singular matrix $\boldsymbol{J}^{\mathrm{T}} \boldsymbol{J}$, the factor λ is introduced in the Levenberg-Marquardt algorithm and adapted during each iteration. One Levenberg-Marquardt iteration comprises the following actions: Computation of a parameter update using equation (4) as well as the resulting back-projection error, acceptance of the new parameters if the error is smaller than the error after the last iteration and division of λ by a factor of 10, or rejection of the computed parameters and multiplication of λ by a factor of 10. Since the error may increase during one iteration due to instabilities in matrix inversion, the preceding steps are done until the new parameters yield a smaller error than at the end of the last iteration.

For m frames the parameter vector $\boldsymbol{a}$ contains $3m$ components of the translations $\boldsymbol{t}_i$, $3m$ components parametrizing the rotation matrices $\boldsymbol{R}_i$, and $4m$ components for the intrinsic parameters. The $3n$ coordinates of the 3-D points are optimized separately in each optimization step when using interleaved bundle-adjustment.

A numerically stable parametrization is used for the rotations, each of which has 9 elements but only 3 degrees of freedom (DOF), i. e. either the axis/angle representation or quaternions [6]. When using quaternions for nonlinear optimization it is necessary to consider that a quaternion representing a rotation has 4 elements but only 3 DOF, since it must be normalized to 1. The Levenberg-Marquardt algorithm cannot deal with constraints on the parameters and it must be guaranteed that the norm of a quaternion is always 1 during optimization. In order to accomplish this goal we used the quaternion approach presented in [6] which gives a quaternion parametrization at the operating point using only 3 parameters.

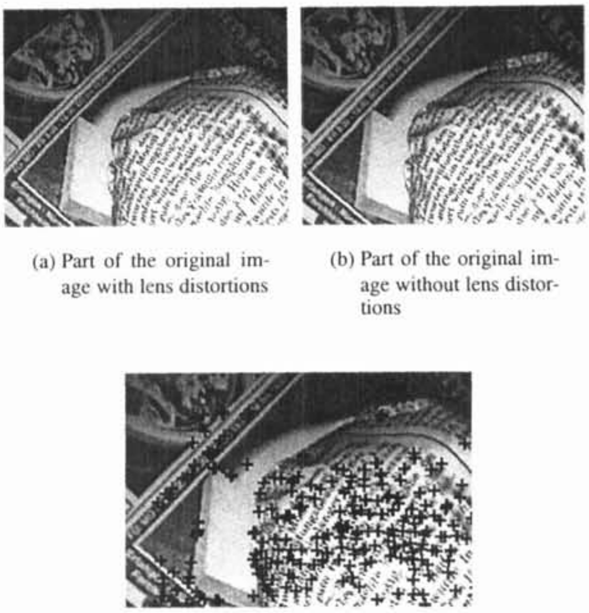
Since we use interleaved bundle-adjustment, time complexity can be reduced further because the Jacobian $\boldsymbol{J}$ is a block-diagonal matrix, resulting in a complexity of $O(nm)$.

4 Experiments

For our experiments we used image sequences of a simplified dummy taken by an endoscope mounted on the AESOP robot. Figure 1 shows the processing steps of one image of a sequence as well as a 3-D plot of triangulated points and camera poses: In Fig. 1(a) one can see a part of an endoscopic image obtained directly by the endoscope. Strong radial distortions are still visible in that image. The image after correction of those distortions is shown in Fig. 1(b). This is the image used for detection and tracking of feature points, which are plotted in Fig. 1(c). These features are tracked over a number of frames and are used afterwards for triangulating 3-D coordinates. In the following we will give results for two sequences, denoted by *Dummy* and *Calib*. *Dummy* is a sequence without any ground truth information, while *Calib* contains images where a calibration pattern was put into the dummy (cf. Fig. 2). The image points used for optimization of the *Calib* scene were extracted by finding the circles of the calibration pattern and are therefore extremely accurate. Table 1 shows the results before and after nonlinear optimization for these sequences. Bundle-adjustment minimizes the back-projection error, i. e. the root mean square error per image point in pixels, which is given in Table 1 and was computed by:

$$\sqrt{\frac{1}{mn} \sum_{j=1}^{n} \sum_{i=1}^{m} \left((x_{ij} - \tilde{x}_{ij})^2 + (y_{ij} - \tilde{y}_{ij})^2 \right)} \quad , \tag{6}$$

where m is the number of frames, n the number of 3-D points, (x_{ij}, y_{ij}) a detected feature point, and $(\tilde{x}_{ij}, \tilde{y}_{ij})$ the

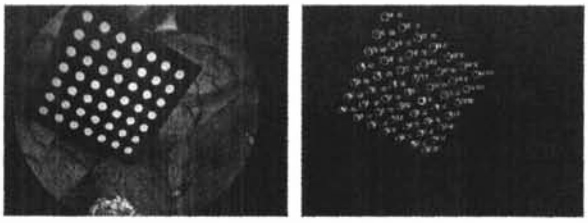


(a) Part of the original image with lens distortions

(b) Part of the original image without lens distortions

(c) Detected feature points

Figure 1: Part of the original image of a simplified dummy taken by an endoscope mounted on AESOP (1(a)), image where radial lens distortions have been corrected (1(b)), and features used for tracking and triangulation (1(c)).



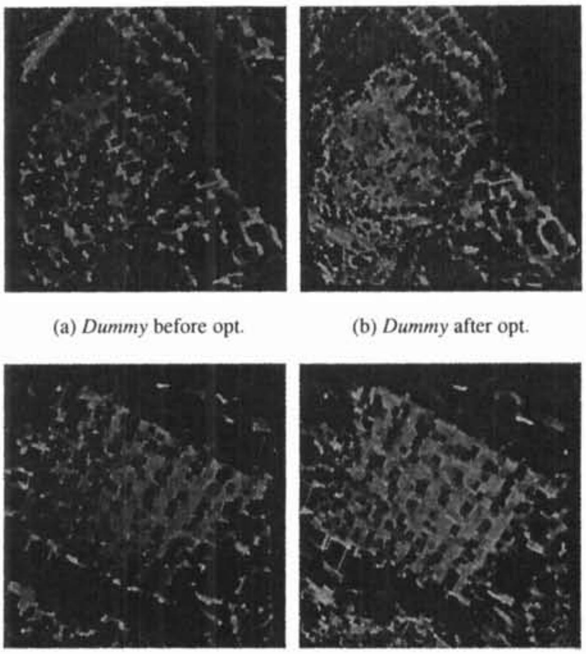
(a) Calibration pattern inside of dummy

(b) Detected circles of pattern

Figure 2: Calibration pattern inside of dummy and extracted circles.

Parameter	*Dummy*		*Calib*	
No. of iterations	100	5	100	5
No. of 3-D points	105	105	49	49
No. of frames	39	39	55	55
Error before opt.	21.4	21.4	17.9	17.9
Error after opt.	1.63	9.01	1.64	17.1

Table 1: Data of image sequences used for the experiments. Shown are the back-projection errors for the two sequences after 5 and after 100 iterations.



(a) *Dummy* before opt.

(b) *Dummy* after opt.

(c) *Calib* before opt.

(d) *Calib* after opt.

Figure 3: Depth maps of images before and after nonlinear optimization for the two sequences.

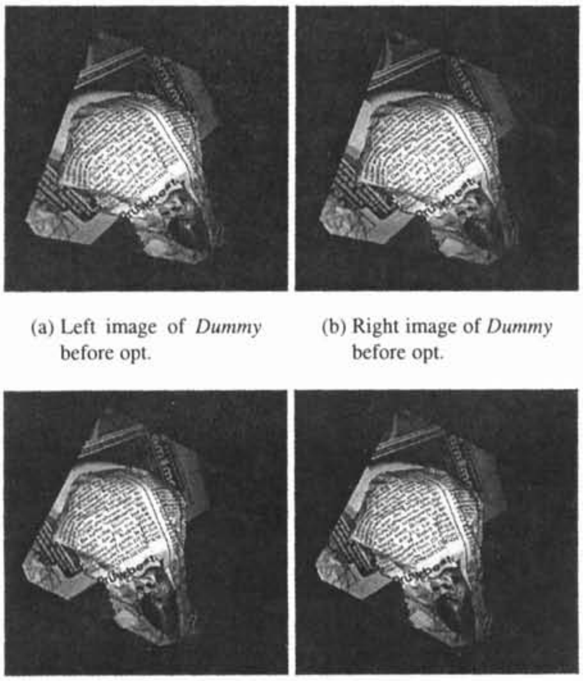
back-projection of a reconstructed 3-D point $\boldsymbol{w}_j$ into frame i. Since the goal of the optimization is computing enhanced light fields, we also evaluated the results by computing disparity maps of rectified image pairs before and after nonlinear optimization. The maps before optimization shown in Fig. 3 on the left-hand side were created using the camera projection matrices obtained from the robot, the maps on the right were computed using the optimized camera parameters. The main problem when computing a depth map from non-optimized camera matrices is that the rectification result is wrong, which means that the left-right correspondences cannot be established correctly. Rectified images before and after optimization are shown in Fig. 4. The nonlinear optimization step however results in consistent camera movements and orientations. 3-D plots of scene points and camera poses before and after nonlinear optimization using 100 iterations are shown in Fig. 5 for the *Dummy* sequence. As can be seen in Table 1, the back-projection error before optimization is very high, which is a hint that the camera parameters obtained from the robot are not very consistent and thus not accurate, since the 3-D points were triangulated using the robot data. If that data were consistent, the triangulation would result in much better fitting 3-D points. During optimization the back-projection error decreases considerably, however.

Figure 6 shows light fields generated using the *Dummy* sequence with depth information before (6(a)) and after (6(b)) nonlinear optimization.

5 Conclusion

In this paper we presented an approach for nonlinear optimization of the parameters of an endoscopic camera mounted on a surgery robot. The goal was to generate a depth map for each image in order to enhance the quality of medical light fields. We showed how to use the robot's pose information as an initialization for interleaved bundle-adjustment by detection and tracking of point features giving 2-D correspondences that were used for triangulation of 3-D scene points. After minimization of the back-projection error we obtained camera parameters that are more accurate and consistent. This is especially important for computation of depth maps using a real-time stereo approach that exploits information about the cameras for rectifying image pairs. Inaccurate camera parameters would result in bad depth maps and thus low-quality light fields.

Although the back-projection error is decreased by our technique, the resulting depth maps do not *always* look better than before the optimization. Sometimes they look quite the same as the original ones, extremely seldom they look



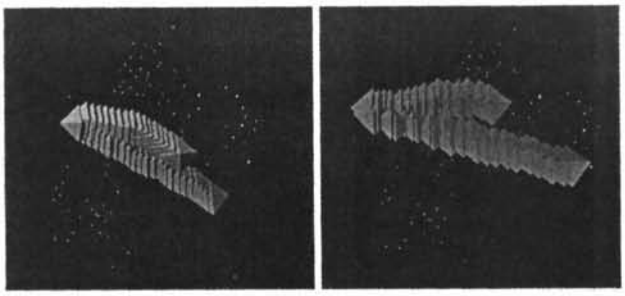
(a) Left image of *Dummy* before opt.

(b) Right image of *Dummy* before opt.

(c) Left image of *Dummy* after opt.

(d) Right image of *Dummy* after opt.

Figure 4: Left and right rectified image pairs before and after nonlinear optimization for the *Dummy* sequence.



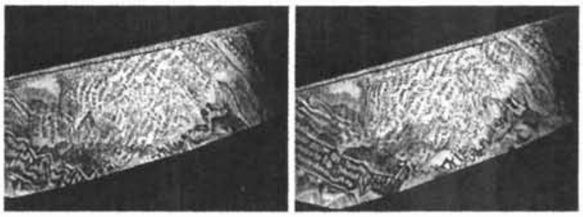
(a) *Dummy* before opt.

(b) *Dummy* after opt.

Figure 5: 3-D points and camera poses (shown as pyramids)) before and after nonlinear optimization (100 iterations) for the *Dummy* sequence.

worse.

Certainly there exist many ways to improve our approach further, e. g. by detection and removal of outliers during tracking image feature points using RANSAC [1]. The fundamental matrices that could be used for RANSAC may be computed from the robot projection matrices $\boldsymbol{P}_i$ before optimization. Another idea would be to optimize only a subset of the camera parameters, e. g. only the 6 pose parameters or the rotation, since rotation is especially important for the rectification step.



(a) *Dummy* with depth information before opt.

(b) *Dummy* with depth information after opt.

Figure 6: Light field of *Dummy* sequence with depth information before (a) and after (b) nonlinear optimization.

References

[1] 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*, 24:381–385, 1981.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In H. Rushmeier, editor, *SIGGRAPH '96 Conference Proceedings*, Annual Conference Series, pages 43–54. ACM SIGGRAPH, Addison Wesley, August 1996.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, Cambridge, 2000.

[4] R. Koch, B. Heigl, M. Pollefeys, L. Van Gool, and H. Niemann. A geometric approach to lightfield calibration. In F. Solina and A. Leonardis, editors, *Computer Analysis of Images and Patterns (CAIP)*, number 1689, pages 596–603, Heidelberg, 1999. Springer.

[5] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 31–42, August 1996.

[6] J. Schmidt and H. Niemann. Using Quaternions for Parametrizing 3–D Rotations in Unconstrained Nonlinear Optimization. In T. Ertl, B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Vision, Modeling, and Visualization 2001*, pages 399–406, Stuttgart, Germany, November 2001. AKA/IOS Press, Berlin, Amsterdam.

[7] J. Schmidt, S. Vogt, and H. Niemann. Dense Disparity Maps in Real-Time with an Application to Augmented Reality. In *IEEE Workshop on Applications of Computer Vision*, Orlando, FL USA, December 2002. to appear.

[8] J. Shi and C. Tomasi. Good features to track. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Seattle, WA, June 1994. IEEE Computer Society Press.

[9] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[10] F. Vogt, D. Paulus, and H. Niemann. Highlight Substitution in Light Fields. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 637–640, Rochester, USA, September 2002. IEEE Computer Society Press.