Iterative Multi-View Plane Fitting

Martin Habbecke and Leif Kobbelt

Computer Graphics Group, RWTH Aachen University
Email: {habbecke,kobbelt}@informatik.rwth-aachen.de

Abstract

We present a method for the reconstruction of 3D planes from calibrated 2D images. Given a set of pixels Ω in a reference image, our method computes a plane which best approximates that part of the scene which has been projected to Ω by exploiting additional views. Based on classical image alignment techniques we derive linear matching equations minimally parameterized by the three parameters of an object-space plane. The resulting iterative algorithm is highly robust because it is able to integrate over large image regions due to the correct object-space approximation and hence is not limited to comparing small image patches. Our method can be applied to a pair of stereo images but is also able to take advantage of the additional information provided by an arbitrary number of input images. A thorough experimental validation shows that these properties enable robust convergence especially under the influence of image sensor noise and camera calibration errors.

1 Introduction

The need for robust matching of correspondences between images arises in many vision related areas such as tracking, camera calibration or 3D reconstruction. A weakness of methods like point feature matching and also of most reconstruction systems based on volumetric or explicit surface representations is the computation of reliable correspondence measures between small image areas. Due to the inevitable noise in digital images, difficult illumination conditions or insufficient texture this is an error-prone problem. A second drawback of many traditional reconstruction methods is that they work well for the case of two images, but are not able to take advantage of additional images showing the same part of a scene which is the type of data typically available in many application scenarios.

We present a method to reconstruct 3D planes from calibrated images that is designed to overcome both of these problems. Our method takes advantage of the stabilizing effect induced by integrating over large image areas. It is thus able to robustly perform matching in noisy images and in regions with little or no texture, resulting in the reconstruction of planes that approximate the original scene with high precision. Furthermore, instead of being limited to two images, our method handles as many images as available, improving the quality of our results especially under the influence of image noise and camera calibration errors. We choose planes because of their stronger approximation power compared to points or lines. For that reason planes are often the method of choice to approximate freeform geometry. In addition, planes only have three degrees of freedom and can hence be reconstructed more robustly than primitives with more degrees of freedom like, e.g., quadrics.

Our method is based on two-dimensional projective mappings between image spaces, also known as *homographies*. In general, such a mapping has eight degrees of freedom. However, it is well known that by requiring calibrated input images it is possible to define *plane-induced* homographies with only three parameters coinciding with the parameters of a plane in space. We apply a scene transformation that simplifies the homographies and enables us to formulate the plane fitting problem in an efficient way.

Several ideas of our algorithm are inspired by classical image alignment methods. Although the research field of image alignment and motion estimation has been explored for more than two decades, we believe that the special case of plane reconstruction from calibrated images is still lacking an in-depth analysis. We fill this gap by deriving a Gauss-Newton style matching algorithm tailored to the specific properties of the plane reconstruction problem. Compared to standard non-linear op-

timization methods like Levenberg-Marquardt, the resulting algorithm turns out to reconstruct planes of equal quality but with much less computation effort. We furthermore experimentally validate our claim of increased robustness against image noise and calibration errors when the number of input images is increased and show preliminary results of a 3D reconstruction system based on our plane fitting method.

2 Related Work

Our work is closely related to motion estimation and image alignment, a research field pioneered by Lucas and Kanade [11]. The general idea is, given a region Ω in a reference image I_r, to find a transformation (or *motion*) T such that the transformed image region best matches a comparison image I_c. This is usually formalized as a sum of squared intensity differences between the reference and comparison image.

Since the work of Lucas and Kanade, many researchers have investigated the problem of estimating the parameters of various transformations T between two images. In the context of point feature and region tracking the most extensively used motions are translations and affine transformations [17, 14, 10, 7]. The more general work of Bergen et al. [4] extends the idea to transformations including projective homographies and also describes the possibility to parameterize the motion between pairs of images by three parameters in case the camera calibration is known. However, their formulation does not allow for the integration of an *arbitrary* number of images to estimate the parameters of *one* plane in object-space, which we show to be the key to improved robustness. An area based on homography matching as well is that of image mosaicing for panorama images (e.g., [16, 15]). Due to the different problem domain with a camera rotating about a fixed center, the solutions cannot easily be transferred to plane reconstruction. Baker et al. [3] describe a system capable of reconstructing a scene as a set of textured planes. Hence, their goal is similar to ours in that they reconstruct planes from calibrated images. They do, however, omit details of the actual fitting process and only state that they apply a standard minimization approach. In the more recent work of the same authors [2], Baker et al. provide an extensive analysis and classification of image alignment algorithms which shows that it is advantageous to exploit the properties of specific alignment problems rather than relying on standard minimization techniques. Given a transformation $T(p)$ parameterized by motion parameters p, each step of their alignment framework computes a parameter update Δp. The authors classify image alignment algorithms as either *additive* if the transformation is updated as $T(p + \Delta p)$ or *compositional* if the update is performed as the composition $T(p) \circ T(\Delta p)$. Furthermore, algorithms are classified as either *forward* or *inverse* if the parameter update Δp is computed as transforming the comparison image I_c or reference image I_r, respectively. We stick to these classes and show that in our case a forward additive algorithm is the only valid choice. Another work of Baker et al. [1] evaluating the properties of different homography parameterizations is of interest to our work. They show for the case of two calibrated images that a minimal parameterization with three parameters is the most robust choice in terms of convergence and yields at least as accurate results as all other evaluated parameterizations. This result stresses that our choice of parameterization is the best possible.

A second research field related to our work is that of general 3D reconstruction from calibrated images based on planes. Hartley and Zisserman give a thorough overview of the geometrical aspects of this problem in [8]. Favaro et al. [6] solve the structure from motion problem based on approximating nearly planar scene regions with planes. Reconstruction of scene planes is often a part of architectural modeling. For example, the Façade system [5] of Debevec et al. lets the user select corresponding points and lines in a set of uncalibrated images and is then able to compute camera parameters and a 3D model. Werner et al. [18] present a system with the goal to automate Façade. However, it imposes constraints on the placement of the scene planes in that it assumes a ground plane and two main planes of the building, all being perpendicular to each other. An overview of general 3D reconstruction from images – not necessarily based on scene planes – has recently been presented by Seitz et al. [13].

3 Problem Formulation and Notation

The problem we are addressing can be stated as follows. Given is a *reference image* I_r and an arbitrary

number of *comparison images* $I_c, c = 1, \ldots, n$, all images being calibrated. Our goal is to find a plane in 3D space which best approximates that part of the scene which has been projected to a set of pixels Ω in the reference image.

The camera calibration is given in the usual form of 3×4 projection matrices $\mathbf{P} = (\mathbf{M}|\mathbf{m})$ with $\mathbf{M} \in \mathbb{R}^{3\times3}$ and $\mathbf{m} \in \mathbb{R}^3$. A separation into intrinsic and extrinsic calibration is not required. The elements of $\mathbf{m}$ are referred to as m_i. The given images are treated as two-dimensional intensity functions $I(u, v)$, where u, v are pixel coordinates.

Scene planes are denoted by the four parameters $\mathbf{N}^T = (n_0, n_1, n_2, d)$ with $\mathbf{N}^T\mathbf{X} = 0$ for all scene points $\mathbf{X}$ lying on $\mathbf{N}$. During the derivation of the matching equations we utilize homographies mapping image points (u, v) from the reference image to the comparison images. Homographies are denoted by 3×3 matrices defined up to an arbitrary scale factor. We do not make a distinction between image points (u, v) and their homogeneous representation $\mathbf{p} = (u, v, 1)^T$ and we use $I(u, v)$ and $I(\mathbf{p})$ synonymously. We furthermore denote the projection of a point $\mathbf{p}$ with a homography $\mathbf{H}$ by $\mathbf{Hp}$ which implicitly contains the dehomogenization.

4 Projective Matching

In this part our solution to the plane fitting problem is detailed. Section 4.1 revisits a transformation applied to the whole scene to allow for simpler matching equations. Our main contribution, the derivation of these equations, is presented in Section 4.2.

4.1 Scene Transformation

Suppose the projection matrices of the reference camera and an arbitrary comparison image are $\mathbf{P}_r = (\mathbf{M}_r|\mathbf{m}_r)$ and $\mathbf{P}_c = (\mathbf{M}_c|\mathbf{m}_c)$, respectively. The homography induced by the scene plane $\mathbf{N}^T = (n_0, n_1, n_2, d) = (\mathbf{n}^T, d)$, mapping image points from the reference image I_r to the plane $\mathbf{N}$ and then further to the image I_c, is

$$\mathbf{H}_c(\mathbf{N}) = \left(d\mathbf{M}_c - \mathbf{m}_c\mathbf{n}^T\right)\left(d\mathbf{M}_r - \mathbf{m}_r\mathbf{n}^T\right)^{-1}$$

for $d \neq 0$. The global coordinate transformation

$$\mathbf{B} = \left(\begin{array}{ccc|c} & \mathbf{M}_r^{-1} & & -\mathbf{M}_r^{-1}\mathbf{m}_r \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \in \mathbb{R}^{4\times4} \quad (1)$$

simplifies the reference projection matrix $\mathbf{P}_r$ to

$$\mathbf{P}'_r = \mathbf{P}_r\mathbf{B} = (\mathbf{Id}_3|\mathbf{0}).$$

All other projection matrices are transformed analogously with the same matrix $\mathbf{B}$. This transformation triggers two important simplifications. First, now that the optical center of the reference camera always lies in the origin of the world coordinate frame, it is possible to scale the plane vector N such that $d = 1$. Planes which require a zero d would pass through the optical center of the reference camera and the fitting procedure with this camera as reference is not possible anyway. Second, due to $\mathbf{M}'_r$ being the identity and $\mathbf{m}'_r$ being all zero, the plane-induced homography of $\mathbf{N}^T = (n_0, n_1, n_2, 1)$ from I_r to I_c simplifies to

$$\mathbf{H}_c(\mathbf{N}) = \mathbf{H}_c(\mathbf{n}) = \mathbf{M}'_c - \mathbf{m}'_c\mathbf{n}^T. \quad (2)$$

For the remainder of this section we assume that all matrices and planes have been transformed by $\mathbf{B}$. Notice that scene points and planes are transformed as $\mathbf{X}' = \mathbf{B}^{-1}\mathbf{X}$ and $\mathbf{N}' = \mathbf{B}^T\mathbf{N}$, respectively. More details on the underlying concepts can be found in [8].

4.2 Derivation of the Matching Equations

To simplify the derivation of the matching equations we consider the case of two images (one reference and one comparison image) first. Hence we can drop the image indices and denote the reference and comparison images by I and J, respectively. The extension to more comparison images is straightforward, as we will see at the end of this section.

The matching process is based on minimizing the well-established sum of squared differences (SSD) of image intensities. The objective function for two images thus is

$$E = \sum_{\mathbf{p}\in\Omega}\left(I(\mathbf{p}) - J(\mathbf{H}(\mathbf{n})\mathbf{p})\right)^2. \quad (3)$$

We solve for the unknown plane $\mathbf{n}$ iteratively and compute a parameter update $\Delta\mathbf{n}$ in each iteration. In iteration $k + 1$ a parameter update $\Delta\mathbf{n}$ is computed, based on the result of the previous step: $\mathbf{n}_{k+1} := \mathbf{n}_k + \Delta\mathbf{n}$. According to the classes developed in [2], this is a *forward additive* approach. As a side note, a compositional formulation (either forward or inverse) as $\mathbf{H}(\mathbf{n}) \circ \mathbf{H}(\Delta\mathbf{n})$ is not possible:

the set of homographies $\mathbf{H}(\mathbf{n})$ does not necessarily include the required identity map due to the above parameterization. An inverse additive approach also cannot be applied since the partial derivatives of the transformation with respect to the motion parameters $\mathbf{n}$ and the image points do not obey the required constraints. Hence, a forward additive approach is the only valid choice in our case. As with all image alignment methods, we need an initial estimate of the plane. How this is obtained will be discussed in Section 4.3. Introducing the iterative formulation in the objective function results in

$$E_{k+1} = \sum_{\mathbf{p}\in\Omega} \left(I(\mathbf{p}) - J(\mathbf{H}(\mathbf{n}_k + \Delta\mathbf{n})\mathbf{p}) \right)^2, \quad (4)$$

which we are going to minimize with respect to $\Delta\mathbf{n}$.

The above problem can be simplified by comparing the reference image I to a transformed image $J^{(k+1)}$ instead of comparing it to the original image $J^{(0)} = J$. The transformation we apply to $J^{(0)}$ in iteration $k+1$ is the homography $\mathbf{H}(\mathbf{n}_k)$ computed in the previous step k. That is, we use the transformed image $J^{(k+1)}(\mathbf{p}) := J^{(0)}(\mathbf{H}(\mathbf{n}_k)\mathbf{p})$. Before substituting the transformed image into the objective function (4), we observe that the homography $\mathbf{H}(\mathbf{n}_k + \Delta\mathbf{n})$ can be written as $\mathbf{H}(\mathbf{n}_k + \Delta\mathbf{n}) = \mathbf{H}(\mathbf{n}_k) - \mathbf{m}\Delta\mathbf{n}^T$ according to (2). With this result we are now able to rewrite the objective function as

$$\begin{aligned} E_{k+1} &= \sum_{\mathbf{p}\in\Omega} \left(I(\mathbf{p}) - J^{(0)}(\mathbf{H}(\mathbf{n}_k + \Delta\mathbf{n})\mathbf{p}) \right)^2 \\ &= \sum_{\mathbf{p}\in\Omega} \left(I(\mathbf{p}) - J^{(k+1)}\Big((\mathbf{Id} - \tilde{\mathbf{m}}\Delta\mathbf{n}^T)\mathbf{p}\Big) \right)^2 \end{aligned} \quad (5)$$

with $\tilde{\mathbf{m}}$ being the last column of J's projection matrix, multiplied by the inverse homography: $\tilde{\mathbf{m}} := \mathbf{H}(\mathbf{n}_k)^{-1}\mathbf{m}$. To improve the legibility of of the derivation we drop the superscript $(k+1)$ from the transformed comparison image and refer to it as J.

After rewriting the function of the comparison image in parametric form with explicit dehomogenization of $(\mathbf{Id} - \tilde{\mathbf{m}}\Delta\mathbf{n}^T)\mathbf{p}$ and $\mathbf{p} = (u, v, 1)$ as

$$\begin{aligned} &J\Big((\mathbf{Id} - \tilde{\mathbf{m}}\Delta\mathbf{n}^T)\mathbf{p}\Big) \\ &\quad = J\left(\frac{u - \tilde{m}_0\Delta\mathbf{n}^T\mathbf{p}}{1 - \tilde{m}_2\Delta\mathbf{n}^T\mathbf{p}}, \frac{v - \tilde{m}_1\Delta\mathbf{n}^T\mathbf{p}}{1 - \tilde{m}_2\Delta\mathbf{n}^T\mathbf{p}} \right), \end{aligned}$$

we are able to derive the first order Taylor expansion:

$$\begin{aligned} &J\Big((\mathbf{Id} - \tilde{\mathbf{m}}\Delta\mathbf{n}^T)\mathbf{p}\Big) \\ &\approx J(\mathbf{p}) + \left(\frac{u - \tilde{m}_0\Delta\mathbf{n}^T\mathbf{p}}{1 - \tilde{m}_2\Delta\mathbf{n}^T\mathbf{p}} - u \right) J_x(\mathbf{p}) \\ &\qquad + \left(\frac{v - \tilde{m}_1\Delta\mathbf{n}^T\mathbf{p}}{1 - \tilde{m}_2\Delta\mathbf{n}^T\mathbf{p}} - v \right) J_y(\mathbf{p}), \end{aligned} \quad (6)$$

where J_x and J_y denote the partial derivatives of the transformed comparison image in the image x- and y-direction.

The above equation is non-linear because of the de-homogenization step. However, it can be linearized by dropping the term $\tilde{m}_2\Delta\mathbf{n}^T\mathbf{p}$ from both denominators. As result the denominators reduce to 1. This linearization is justified by the iterative approach: Dropping the above term is identical to performing the de-homogenization with the denominator of $\mathbf{H}(\mathbf{n}_k)\mathbf{p}$ instead of $\mathbf{H}(\mathbf{n}_k + \Delta\mathbf{n})\mathbf{p}$. When the iteration converges, the computed upgrade $\Delta\mathbf{n}$ rapidly decreases towards the zero-vector and hence the linearized denominator approaches the correct version. The linearized Taylor expansion (6) is

$$\begin{aligned} &J\Big((\mathbf{Id} - \tilde{\mathbf{m}}\Delta\mathbf{n}^T)\mathbf{p}\Big) \\ &\approx J(\mathbf{p}) - \big(\tilde{m}_0\Delta\mathbf{n}^T\mathbf{p}\big)J_x(\mathbf{p}) - \big(\tilde{m}_1\Delta\mathbf{n}^T\mathbf{p}\big)J_y(\mathbf{p}). \end{aligned}$$

Substituting the linearized Taylor expansion into the objective function (5) yields

$$\begin{aligned} E_{k+1} = \sum_{\mathbf{p}\in\Omega} \Big(&I(\mathbf{p}) - J(\mathbf{p}) + \big(\tilde{m}_0\Delta\mathbf{n}^T\mathbf{p}\big)J_x(\mathbf{p}) \\ &+ \big(\tilde{m}_1\Delta\mathbf{n}^T\mathbf{p}\big)J_y(\mathbf{p}) \Big)^2. \end{aligned}$$

We finally set

$$\begin{aligned} F_0 &:= uF_2 \qquad F_1 := vF_2 \\ F_2 &:= \tilde{m}_0 J_x(\mathbf{p}) + \tilde{m}_1 J_y(\mathbf{p}) \\ D &:= I(\mathbf{p}) - J(\mathbf{p}). \end{aligned}$$

and reorder the terms of E_{k+1} with respect to the coordinates of $\Delta\mathbf{n}$ to obtain

$$E_{k+1} = \sum_{\mathbf{p}\in\Omega} \big(\Delta n_0 F_0 + \Delta n_1 F_1 + \Delta n_2 F_2 + D\big)^2.$$

To minimize E_{k+1} we compute the partial derivatives with respect to the coordinates of $\Delta\mathbf{n}$

$$\begin{aligned} \frac{\partial E_{k+1}}{\partial \Delta n_i} = 2 \sum_{\mathbf{p}\in\Omega} F_i \big(&\Delta n_0 F_0 + \Delta n_1 F_1 \\ &+ \Delta n_2 F_2 + D\big), \end{aligned}$$

set them to zero and solve the resulting linear system

$$\mathbf{F}\Delta\mathbf{n} = \mathbf{b} \tag{7}$$

with

$$\mathbf{F} = \sum_{\mathbf{p}\in\Omega} \begin{pmatrix} F_0^2 & F_0F_1 & F_0F_2 \\ F_0F_1 & F_1^2 & F_1F_2 \\ F_0F_2 & F_1F_2 & F_2^2 \end{pmatrix}, \mathbf{b} = -\sum_{\mathbf{p}\in\Omega} \begin{pmatrix} F_0D \\ F_1D \\ F_2D \end{pmatrix}.$$

A major advantage of this formulation is the easy integration of additional images. Taking more than one comparison image into account is achieved by extending the objective function (3) as

$$E = \sum_{c=1}^{n} \sum_{\mathbf{p}\in\Omega} \Big(I_0(\mathbf{p}) - I_c(\mathbf{H}_c(\mathbf{n})\mathbf{p}) \Big)^2. \tag{8}$$

The additional summation does not affect the derivation and results in an accumulated linear system which is built by summation over the systems shown in (7), one for each comparison image. Thus, the problem stays minimally parameterized for an arbitrary number of comparison images and the time needed to solve the accumulated system does also not change. The matching scheme can be extended to use RGB color images in a completely analogous way: every color channel is treated as separate graylevel image in the outer sum in (8). Since our method combines two (discrete) integration steps we gain improved robustness against image noise and camera calibration errors. This is shown empirically in Section 5.

The main problem of image matching based on squared differences of color intensities is its well-known sensitivity to illumination changes. If the object surface is not perfectly Lambertian, changing camera perspectives lead to changes in the intensity of the same surface patch. A standard approach to compensate for such lighting changes is to apply individual *photometric normalization* to the image regions. In each iteration, i.e. during each construction of (7), the mean intensities μ_r and μ_c of the image regions $I_r(\Omega)$ and $I_c(\mathbf{H}_c(\Omega))$ are subtracted from the pixel intensities, the resulting intensities are then divided by the standard deviations σ_r and σ_c of the respective image regions:

$$E = \sum_{c=1}^{n} \sum_{\mathbf{p}\in\Omega} \left(\frac{I_r(\mathbf{p}) - \mu_r}{\sigma_r} - \frac{I_c(\mathbf{H}_c(\mathbf{n})\mathbf{p}) - \mu_c}{\sigma_c} \right)^2.$$

This allows the plane fitting scheme to work well even in cases of moderate changes in the lighting conditions.

4.3 Implementation Issues

We approximate the partial derivatives J_x and J_y in the image directions by simple central differences. The transformed comparison images introduced in the previous section are not explicitly computed for performance reasons. Instead, we always compute the transformed positions in the original comparison images and apply bi-linear interpolation to compute intensity values at non-integer positions.

An extension to avoid local minima during the matching process (already described in [4]) is to perform the matching on a Gaussian image pyramid. Starting on a coarse resolution, the computed plane parameters are reused as initialization on the next finer level. In our implementation the plane parameters stay unchanged when switching to a different level. Only the projection matrices have to be adjusted to compensate for the change in image resolution. Due to the Gaussian image pyramids the algorithm does not require additional image smoothing.

To improve numerical stability, the set of pixels Ω is translated such that its center of gravity lies at the origin $(0, 0)$ and it is scaled such that the coordinates of all pixels $\mathbf{p} \in \Omega$ lie in the range $[-1, 1]^2$ (see, e.g., [8]). Notice that for consistency all projection matrices have to be transformed as $\hat{\mathbf{P}} = \mathbf{TP}$ with

$$\mathbf{T} = \begin{pmatrix} w & 0 & -wc_x \\ 0 & w & -wc_y \\ 0 & 0 & 1 \end{pmatrix},$$

where (c_x, c_y) is the center of gravity of Ω and w the correct scale factor. Notice further that this transformation affects the scene transformation $\mathbf{B}$ in (1) as well. However, as the image derivatives are computed based on the original images, again no explicit image scaling is necessary.

We compute the initial scene plane $\mathbf{n}_0$ using a simple heuristic. It is parallel to the reference image and its depth is determined by the intersection of the re-projected image centers of the reference image and a nearby (in terms of angle deviation of these re-projected rays) comparison image. Obviously, if the initial plane deviates too much from the correct solution, the matching algorithm may get trapped in a local minimum. Furthermore, as in all iterative approaches to non-linear optimization problems, there is no guarantee for our matching algorithm to converge. However, according to our observations, when using a moderate damping

of the plane update of 0.75 and initial plane parameters sufficiently close to the correct values, divergence or oscillating configurations did not occur in any of our experiments.

5 Results

The first series of tests analyzes the running time of our algorithm. All tests have been performed on an AMD Athlon 64 3500+ system. Figure 1 shows the average time for one iteration of the matching procedure with increasing pixel sets Ω and increasing number of comparison images. As expected, the algorithm shows close to linear behavior in the number of comparison images since the time needed for the construction of the linear system dominates the time for its solution. We furthermore see that the algorithm is sufficiently fast for real-world applications, even for large sets Ω and large numbers of comparison images.

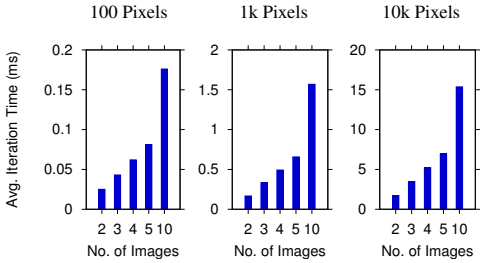

Figure 1: Average computation time for one iteration of our algorithm with sets Ω of 100 (left), 1.000 (center) and 10.000 (right) pixels. Five tests have been performed for each set, one for 2, 3, 4, 5 and 10 images (including the reference image). All times are given in milliseconds.

As pointed out by [2] there are several different possibilities to minimize the objective function (3). One of their results is that the Gauss-Newton and Levenberg-Marquardt methods perform equally well. We therefore have compared our method against a Levenberg-Marquardt minimization of (3) using the MINPACK library [12]. The required Jacobian of the objective function can easily be computed after the scene transformation of Section 4.1. The results are in line with [2]: both methods yield reconstructions of comparable quality. However, the implementation of our linearized algorithm was faster in all our tests, usually by a factor of about two to three.

The second test examines the influence of different numbers of comparison images on the quality of the reconstruction under the influence of calibration errors. A set of 10 synthetic small-baseline images of a textured 3D plane has been generated. One run of the algorithm then consisted of the following steps: A subset of the images was chosen and the corresponding synthetic cameras have been corrupted by noise. For a realistic setup, we projected a set of 3D points into the synthetic images, added Gaussian noise to the resulting image positions and re-computed the projection matrices from the now distorted 3D-2D correspondences using standard techniques [8]. The resulting matrices were then used in our plane reconstruction framework. To minimize statistical artifacts, each measurement has been repeated 50 times with newly selected images and newly generated camera noise. We performed this experiment with different numbers of comparison images and different levels of noise, the results are shown in Figure 2, left.

Analogously we measured the behavior of our algorithm with respect to Gaussian image noise. Again, 10 synthetic images of a plane were taken. The images were now distorted by Gaussian image noise of varying intensity ($\sigma^2 = 0.025$, $\sigma^2 = 0.0125$ and $\sigma^2 = 0.005$ with image intensities between 0 and 1). The results of this test are shown in Figure 2, right. The results of both tests indicate that the algorithm benefits from more images up to about 8 both in terms of mean error and standard deviation. From that point on more images do not seem to improve the result significantly. However, it becomes clear that the results of a simple stereo setup can always be improved considerably by taking more images into account.

Figure 4 shows an approximation of a Chinese statue by a set of 57 planar polygons from five calibrated input images (of resolution 1024x768). Each plane has been fitted by manually selecting the corresponding 2D image polygon in the reference image and then passing it to the automatic matching procedure. All image polygons are shown in the center image of the top row. Each of the polygons has been fitted independently, i.e., without any coupling like smoothness constraints. The accumulated time spent in the matching routine for all 57 polygons has been 1.49 seconds. The matching procedure has been performed using the multiresolution

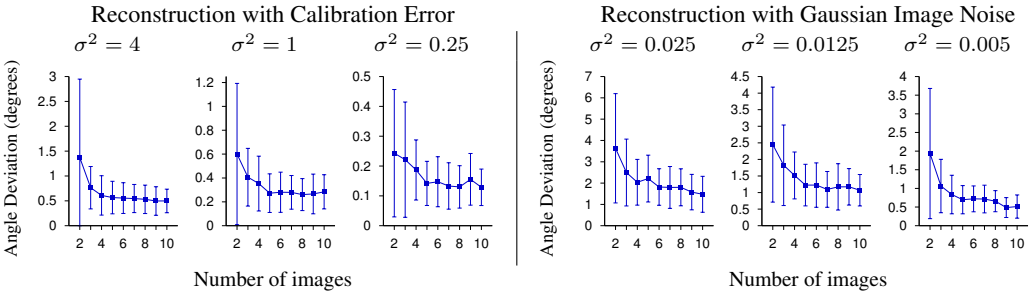

Figure 2: Measurement of reconstruction quality under the influence of calibration error and image noise. Synthetic images (of resolution 685×660) and synthetic cameras have been distorted by various intensities of camera calibration error (left) and different levels of Gaussian image noise (right). The plots show the mean (square dots) and standard deviation (error bars) of the angle between the normals of the reconstructed plane and the ground truth. Notice the different scales of the y-axes.

approach, where the lowest level of the Gaussian image pyramids is determined automatically such that at least 50 pixels per polygon are left on the lowest resolution. Furthermore, the lowest level is limited to be at most three since lower levels did not yield better results in our experiments due to the strong image blurring. Convergence of the matching iteration can easily be determined by monitoring the length of the update vector $\Delta\mathbf{n}$. Thus, our implementation of the matching algorithm does not depend on any parameter to be set by the user. In this example, the fitting usually converged after roughly 15 iterations. Notice that concave areas do not pose a problem for our matching strategy and all concave regions have been reconstructed correctly. The center image of the lower row in Figure 4 contains the reconstructed planes rendered together with a ground truth geometry obtained with an implementation of [9]. The rendering clearly shows that even in cases where the original geometry is moderately curved, the matching procedure still results in a correct approximation in the least squares sense.

The next example in Figure 3 shows the application of our method to five calibrated images of an outdoor scene without any control on the lighting conditions. The procedure has been the same as for the Chinese statue: The image polygons have been defined manually and each polygon has been passed to the automatic reconstruction algorithm separately. Due to the smaller image resolution (720x576), the accumulated matching time has only been 1.13 seconds. The resulting planes demonstrate that our algorithm works well even under difficult conditions like low image contrast and little object texture.

Figure 3: 60 planes fitted to an outdoor statue. Top: one of the five input images, with and without the manually defined image polygons overlaid. Bottom: rendering of the reconstructed 3D planes.

Finally, Figure 5 shows a reconstruction from a subset (15 images) of the *DinoRing* dataset which is part of the mutli-view stereo evaluation by Seitz et al. [13]. The reconstruction has been obtained with the prototype of a fully automatic system that handles the selection of image regions Ω and the selection of suitable comparison images in addition to the actual plane fitting. The system fits disc-shaped planes which are adaptively resized to project to image regions Ω of roughly 500 pixels. Each of the

discs has been reconstructed independently using 4 comparison images. For the illustration in Figure 5 the size of the discs has been reduced to 60% (center) and 20% (right) of the original radius, respectively. The *Dino* object is particularly difficult for reconstruction methods relying on small, possibly image aligned patches due to its uniformly colored smooth surface. In contrast, our algorithm is able to faithfully fit planes to image regions with very few texture and deeply concave parts of the geometry.

6 Conclusion and Future Work

We have presented a method for the reconstruction of 3D planes from calibrated images with two important properties: It is robust against image noise and camera calibration errors by integrating over large image regions and it is able to take advantage of additional information in the form of more input images. It is furthermore fully automatic and does not depend on any parameters to be set by the user. We have shown that the resulting implementation is fast and applicable to real-world settings.

There is, however, still some room for further improvement: The currently used objective function based on the SSD of intensity values with photometric normalization works well for moderate lighting changes but is known to have problems with stronger changes like specular reflections. A possible direction for future work is the integration of a voting-based approach which computes planes in space for subsets of images and then removes images with specular reflections by detecting outlying planes.

The second main area of future work is the extension of our prototype system to a full-scale automatic reconstruction method. The main problems here are to find image regions suitable for the plane reconstruction process (i.e., regions belonging to approximately planar parts of the scene) and to find a set of comparison cameras all having an unoccluded view on the reconstructed plane. A ground truth evaluation of our *Dino* reconstruction is also part of the future work.

References

[1] S. Baker, A. Datta, and T. Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Robotics Institute, Carnegie Mellon University, 2006.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. Journal of Computer Vision*, 56(3):221–255, 2004.

[3] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. of IEEE CVPR*, pages 434–441, 1998.

[4] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. of ECCV*, pages 237–252, 1992.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Proc. of SIGGRAPH*, pages 11–20, 1996.

[6] P. Favaro, H. Jin, and S. Soatto. A semi-direct approach to structure from motion. In *Proc. of Int. Conference on Image Analysis and Processing*, pages 250–255, 2001.

[7] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI*, 20(10):1025–1039, 1998.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2003.

[9] A. Hornung and L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *Proc. of CVPR*, volume 1, pages 503–510, 2006.

[10] H. Jin, P. Favaro, and S. Soatto. Real-Time feature tracking and outlier rejection with changes in illumination. In *Proc. of ICCV*, pages 684–689, 2001.

[11] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of Int. Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[12] Jorge J. Moré, Burton S. Garbow, and Kenneth E. Hillstrom. User guide for MINPACK-1. Technical Report ANL-80-74, Argonne National Laboratory, Argonne, IL, USA, August 1980.

[13] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. of CVPR*, volume 1, pages 519–526, 2006.

[14] J. Shi and C. Tomasi. Good features to track. In *Proc. of CVPR*, pages 593–600, 1994.

[15] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *Int. Journal of Computer Vision*, 16(1):63–84, 2000.

[16] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision*, pages 44–53, 1994.

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

[18] T. Werner and A. Zisserman. New techniques for automated architecture reconstruction from photographs. In *Proc. of ECCV*, pages 541–555, 2002.

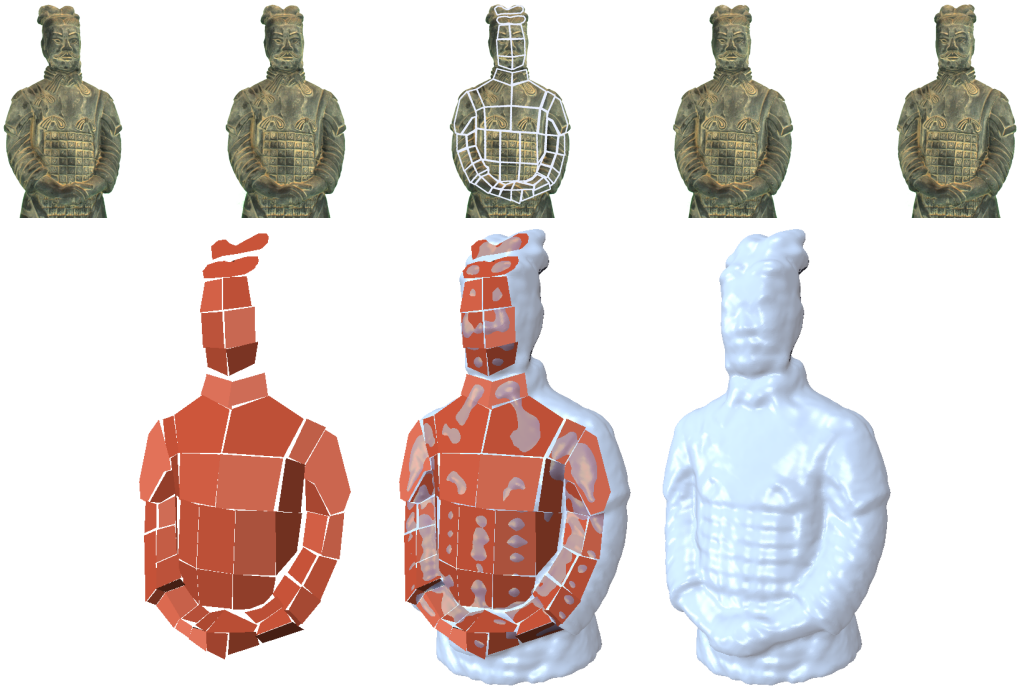

Figure 4: Approximation of the front side of a Chinese statue using 57 planar polygons. The upper row shows the five small-baseline input images used during the matching process. The lower row shows the set of reconstructed planes (left), the same planes rendered together with the ground truth geometry of the statue (center) and the ground truth geometry alone for reference (right).

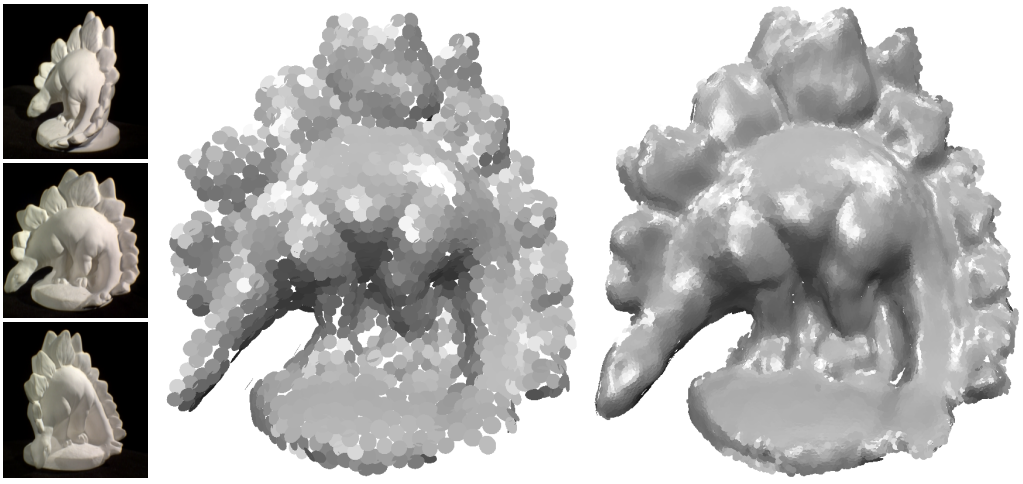

Figure 5: Reconstruction from 15 images of the *DinoRing* dataset which is part of the Middlebury Multi-View Stereo Evaluation [13]. Left: first, middle and last input image. Center: coarse reconstruction using 1100 disc-shaped planes. The discs have been reduced to 60% of their original radius for rendering. Right: fine reconstruction using 14k discs which have been reduced to 20% of their original size.