Int J Comput Vis (2009) 84: 257–268
DOI 10.1007/s11263-009-0232-2

Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment

Yasutaka Furukawa · Jean Ponce

Received: 23 August 2008 / Accepted: 12 March 2009 / Published online: 21 April 2009

Abstract The advent of high-resolution digital cameras and sophisticated multi-view stereo algorithms offers the promise of unprecedented geometric fidelity in image-based modeling tasks, but it also puts unprecedented demands on camera calibration to fulfill these promises. This paper presents a novel approach to camera calibration where top-down information from rough camera parameter estimates and the output of a multi-view-stereo system on scaled-down input images is used to effectively guide the search for additional image correspondences and significantly improve camera calibration parameters using a standard bundle adjustment algorithm (Lourakis and Argyros 2008). The proposed method has been tested on six real datasets including objects without salient features for which image correspondences cannot be found in a purely bottom-up fashion, and objects with high curvature and thin structures that are lost in visual hull construction even with small errors in camera parameters. Three different methods have been used to qualitatively assess the improvements of the camera parameters. The implementation of the proposed algorithm is publicly available at Furukawa and Ponce (2008b).

Keywords Bundle adjustment · Structure from motion · Multi-view stereo · Image-based modeling · Camera calibration

Y. Furukawa (✉)
Computer Science & Engineering, University of Washington,
Box 352350, Seattle, WA 98195-2350, USA
e-mail: furukawa@cs.washington.edu

J. Ponce
Willow project-team at the Laboratoire d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548,
45, rue D'Ulm, 75230 Paris Cedex 05, France
e-mail: Jean.Ponce@ens.fr

1 Introduction

Modern *multi-view stereovision* (*MVS*) systems are capable of capturing dense and accurate surface models of complex objects from a moderate number of calibrated images: Indeed, a recent study has shown that several algorithms achieve surface coverage of about 95% and depth accuracy of about 0.5 mm for an object 10 cm in diameter observed by 16 low-resolution (640 × 480) cameras. Combined with the emergence of affordable, high-resolution (10 Mpixel and higher) consumer-grade cameras, this technology promises even higher, unprecedented geometric fidelity in image-based modeling tasks, but puts tremendous demands on the calibration procedure used to estimate the intrinsic and extrinsic camera parameters, lens distortion coefficients, etc.

There are two main approaches to the calibration problem: The first one, dubbed *chart-based calibration* (or *CBC*) in the rest of this presentation, assumes that an object with precisely known geometry (the chart) is present in all input images, and computes the camera parameters consistent with a set of correspondences between the features defining the chart and their observed image projections (Bouguet 2008; Tsai 1987). It is often used in conjunction with positioning systems such as a robot arm (Seitz et al. 2006) or a turntable (Hernández Esteban and Schmitt 2004) that can repeat the same motion with high accuracy, so that object and calibration chart pictures can be taken separately but under the same viewing conditions. The second approach to calibration is *structure from motion* (*SFM*), where both the scene shape (structure) and the camera parameters (motion) consistent with a set of correspondences between scene and image features are estimated (Pollefeys et al. 2004; Hartley and Zisserman 2004). In this process, the *intrinsic* camera parameters are often supposed to be known a priori (Nister 2004), or recovered a posteriori through *auto-*

calibration (Triggs 1997; Pollefeys et al. 2004). A final *bundle adjustment* (*BA*) stage is then typically used to fine tune the positions of the scene points and the entire set of camera parameters (including the intrinsic ones and possibly the distortion coefficients) in a single non-linear optimization (Lourakis and Argyros 2008; Triggs et al. 2000). A key ingredient of both approaches to calibration is the *selection of feature correspondences* (*SFC*), procedure that may be manual or (partially or totally) automated, and is often intertwined with the calibration process: In a typical SFM system for example Pollefeys et al. (2004), features may first be found as "interest points" in all input images, before a robust matching technique such as *RANSAC* (Fischler and Bolles 1981) is used to simultaneously estimate a set of consistent feature correspondences *and* camera parameters. Some approaches propose to improve feature correspondences for robust camera calibration (Martinec and Pajdla 2007). However, reliable automated SFC/SFM systems are hard to come by, and they may fail for scenes composed mostly of objects with weak textures (e.g., human faces). In this case, manual feature selection and/or CBC are the only viable alternatives.

Today, despite decades of work and a mature technology, putting together a complete and reliable calibration pipeline thus remains non-trivial procedure requiring much know-how, with various pitfalls and sources of inaccuracy. Automated SFC/SFM methods tend to work well for close-by cameras in controlled environments—though errors tend to accumulate for long-range motions, and they may be ineffective for poorly textured scenes and widely separated input images. CBC systems can be used regardless of scene texture and view separation, but it is difficult to design and build accurate calibration charts with patterns clearly visible from all views. This is particularly true for 3D charts (which are desirable for uniform accuracy over the visible field), but remains a problem even for printed planar grids (the plates the paper is laid on may not be quite flat, laser printers are surprisingly inaccurate, etc.). In addition, the robot arms or turntables used in many experimental setups may not be exactly repetitive. In fact, even a camera attached to a sturdy tripod may be affected during experiments by vibrations from the floor, thermal effects, etc. These seemingly minor factors may not be negligible for modern high-resolution cameras,[1] and they limit the effectiveness of classical chart-based calibration. Of course, sophisticated setups that are less sensitive to these difficulties have been developed by photogrammeters (Uffenkamp 1993), but they typically require special equipment and software that are unfortunately not available in many academic and industrial settings. Our goal, in this article, is to develop a flexible but high-accuracy calibration system that is affordable and accessible to everyone. To this end, a few researchers have proposed using scene information to refine camera calibration parameters: Lavest et al. propose (1998) to compensate for the inaccuracy of a calibration chart by adjusting the 3D position of the markers that make it up, but this requires special markers and software for locating them with sufficient sub-pixel precision. The calibration algorithms proposed in Hernández Esteban et al. (2007) and Wong and Cipolla (2004) exploit silhouette information instead. They work for objects without any texture and are effective in wide-baseline situations, but are limited to circular camera motions.

In this article, we propose a very simple and efficient BA algorithm that does not suffer from these limitations and exploits top-down information provided by a rough surface reconstruction to establish image correspondences. Concretely, given a set of input images, possibly inaccurate camera parameters that may have been obtained by an SFM or CBC system, and some conservative estimate of the corresponding reprojection errors, the input images are first scaled down so these errors become small enough to successfully run a patch-based multi-view stereo algorithm (PMVS; Furukawa and Ponce 2007) that reconstructs a set of oriented points (points plus normals) densely covering the surface of the observed scene, and identifies the images where they are visible. The core component of the approach proposed in this paper is essentially guided-matching procedure in its second stage, where image features are matched across multiple views using the estimated surface geometry and visibility information. Finally, matched features are input to the SBA bundle adjustment software (Lourakis and Argyros 2008) to tighten up camera parameters.[2] Besides improving camera calibration, the proposed method can significantly speed up SFM systems by running the SFM software on scaled-down input images, then using the proposed algorithm on full-resolution images to tighten-up camera calibration. The proposed method has been tested on various real datasets, including objects without salient features for which image correspondences cannot be found in a purely bottom-up fashion, and objects with high-curvature and thin structures that are lost in the construction of visual hulls without our bundle adjustment procedure (Sect. 4). In summary, the contributions of the proposed approach can be described as follows:

[1] For example, the robot arm (Stanford spherical gantry) used in the multi-view stereo evaluation of Seitz et al. (2006) has an accuracy of 0.01° for a 1 m radius sphere observing an object about 15 cm in diameter, which yields approximately $1.0\ [\text{m}] \times 0.01 \times \pi/180 = 0.175\ [\text{mm}]$ errors near an object. Even with the low-resolution 640×480 cameras used in Seitz et al. (2006), where a pixel covers roughly 0.25 mm on the surface of an object, this error corresponds to $0.175/0.25 = 0.7$ pixels, which is not negligible. If one used a high-resolution 4000×3000 camera, the positioning error would increase to $0.7 \times 4000/640 = 4.4$ pixels.

[2] The spirit of guided-matching is also used in Fua (2000) to match features of objects with weak textures, although approximate geometry of an object must be known in advance and manual feature correspondences are required in their work.

- Better feature localization by taking into account surface geometry estimations.
- Better coverage and dense feature correspondences by exploiting surface geometry and visibility information.
- An ability to handle objects with very weak textures and resolve accumulation errors, two difficult issues for existing SFM and BA algorithms.

The rest of this article is organized as follows. Section 2 presents our imaging model together with some notations, and briefly introduce the MVS algorithm used in the article (Furukawa and Ponce 2008c). Section 3 details the proposed algorithm. Experimental results and discussions are given in Sect. 4. Note that PMVS (Furukawa and Ponce 2008c), SBA (Lourakis and Argyros 2008), and several CBC systems such as Bouguet (2008) are publicly available. Bundled with our software, which is also available online at Furukawa and Ponce (2008b), they make a complete software suite for high-accuracy camera calibration. A preliminary version of this article appeared in Furukawa and Ponce (2008a).

2 Imaging Model and Preliminaries

Our approach to camera calibration accommodates in principle any parametric projection model of the form $p = f(P, C)$, where P denotes both a scene point and its position in some fixed world coordinate system, C denotes both an image and the corresponding vector of camera parameters, and p denotes the projection of P into the image. In practice, our implementation is currently limited to a standard perspective projection model where C records five intrinsic parameters and six extrinsic ones. Distortion is thus supposed to be negligible, or already corrected, for example by software such as DxO Optics Pro (DXO 2008). Standard BA algorithms take the following three data as inputs: a set of n 3D point positions $P_1, P_2, \ldots, P_n$, m camera parameters $C_1, \ldots, C_m$, and the positions of the projections p_{ij} of the points P_i in the images C_j where they are visible (Fig. 1). They optimize both the scene P_i and camera parameters C_j by minimizing, for example, the sum of squared reprojection errors:

$$\sum_{i=1}^{n} \sum_{j \in V_i} (p_{ij} - f(P_i, C_j))^2, \tag{1}$$

where V_i encodes visibility information as the set of indices of images where P_i is visible. Unlike BA algorithms, multi-view stereo algorithms are aimed at recovering scene information alone given fixed camera parameters. In our implementation, we use the PMVS software Furukawa and Ponce (2007, 2008c) that generates a set of *oriented* points P_i, together with the corresponding visibility information V_i. We have chosen PMVS because (1) it is one of the best MVS algorithms to date according to the Middlebury benchmarks (Seitz et al. 2006), (2) our method does not require a 3D mesh model but just a set of oriented points, which is the output of PMVS, and (3) as noted earlier, PMVS is freely available (Furukawa and Ponce 2008c). This is also one of the reasons for choosing the SBA software (Lourakis and Argyros 2008) for bundle adjustment, the others being its flexibility and efficiency.

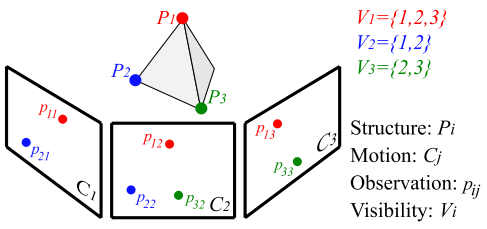

Fig. 1 Notation: Three points P_1, P_2, P_3 are observed by three cameras C_1, C_2, C_3. P_{ij} is the image projection of P_i in C_j. V_i is a set of indexes of cameras in which P_i is visible

3 Algorithm

The overall algorithm is given in Fig. 2. We first use the oriented points P_i $(i = 1, \ldots, n)$ and the corresponding visibility information V_i output by PMVS to form initial image correspondences p_{ij}, then refine these parameters p_{ij} and V_i by simple local image texture comparison in the second step. Given the refined image correspondences, it is possible to rely on SBA to improve the camera parameters. The entire process is repeated a couple of times to tighten up the camera calibration. In this section, we will explain how to initialize and refine feature correspondences.

3.1 Initializing Feature Correspondences

In practice, we have found PMVS to be robust to errors in camera parameters *as long as the image resolution matches the corresponding reprojection errors*—that is, when features to be matched are roughly within two pixels of the corresponding 3D points. Given an initial set of camera parameters, it is usually possible to obtain a conservative estimate of the expected reprojection error E_r by hand (e.g., by visually inspecting a number of epipolar lines) or automatically (e.g., by directly measuring reprojection errors associated with the features matched by a SFM system).[3] Thus,

[3] In practice, reprojection errors reported by a SFM system tend to be small even when camera parameters contain errors due to poor coverage of matched features. Since E_r is just a conservative estimate of reprojection errors, it is advisable to over-approximate the value.

Input: Cameras parameters $\{K_j, R_j, t_j\}$ and expected reprojection error E_r.
Output: Refined cameras parameters $\{K_j, R_j, t_j\}$.

Build image pyramids for all the images.
Compute a level L to run PMVS: $L \leftarrow \max(0, \lfloor \log_2 E_r \rfloor)$.
Repeat four times
- Run PMVS on level L of the pyramids to obtain patches $\{P_i\}$ and their visibility information $\{V_i\}$.
- Initialize feature locations: $p_{ij} \leftarrow F(P_i, \{K_j, R_j, t_j\})$.
- Sub-sample feature correspondences.
- *For* each feature correspondence $\{p_{ij} | j \in V_i\}$
  - Identify a *reference* camera C_{j_0} in V_i with the minimum foreshortening factor.
  - *For* each non-reference feature $p_{ij} (j \in V_i, j \neq j_0)$
    - *For* $L^* \leftarrow L$ down to 0
      - Use level L^* of image pyramids to refine p_{ij}: $p_{ij} \leftarrow \text{argmax}_{p_{ij}} \text{NCC}(q_{ij}, q_{ij_0})$.
  - Filter out features that have moved too much.
- Refine $\{P_i, K_j, R_j, t_j\}$ by a standard BA with $\{p_{ij}\}$.
- Update E_r by the *mean* and *std* of reprojection errors.

Fig. 2 Overall algorithm

we first build image pyramids for all the input images, then run PMVS on the level $L = \lceil \log_2 E_r \rceil$ of the pyramids. At this level, images are 2^L times smaller than the originals, with reprojection errors of at most about two pixels. We then project the points P_i output by this program into the images where they are visible to obtain an initial set of image correspondences $p_{ij} = f(P_i, C_j)$, with j in V_i. Depending on the value of L and the choice of the PMVS parameter ζ that controls the density of oriented points it constructs, the number of these points, and thus, the number of feature correspondences may become quite large. Dense reconstruction is not necessary for bundle adjustment, and we sub-sample feature correspondences for efficiency.[4] More concretely, we first divide each image into 10×10 uniform blocks, and randomly select within each block at most ε features. A feature correspondence will be used in the next refinement step if at least one of its associated image features p_{ij} was sampled in the above procedure. In practice, ε is chosen so that the number of feature correspondences becomes ten to twenty percents of the original one after this sampling step. Note that sub-sampling is performed in each block (as opposed to each image) in order to ensure uniformly distributed feature correspondences.

[4]We could increase the value of ζ to obtain a sparser set of patches without sub-sampling, but, as detailed in Furukawa and Ponce (2007), a dense reconstruction is necessary for this algorithm to work well and determine visibility information accurately.

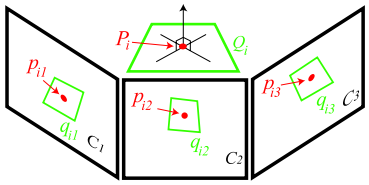

Fig. 3 Given a patch (P_i, Q_i) and the visibility information V_i, we initialize matching images patches (p_{ij}, q_{ij})

3.2 Refining Feature Correspondences

Due to the use of low-resolution images in PMVS and errors in camera parameters, the initial values of p_{ij} are not accurate. Therefore, the second step of the algorithm is to optimize the feature locations p_{ij} by comparing local image textures. Concretely, since we have an estimate of the surface normal at each point P_i, we consider a small 3D rectangular patch Q_i centered at P_i and construct its projection q_{ij} in the set V_i of images where P_i is visible (Fig. 3). We automatically determine the extent of Q_i so its largest projection covers an image area of about $\delta \times \delta$ pixels (we have used $\delta = 7$ throughout our experiments). In practice, as in Furukawa and Ponce (2007), a patch Q_i is represented by a $\delta \times \delta$ grid of 3D points and the local image texture inside q_{ij} is, in turn, represented by a set of pixel colors at their image projections that are computed by a bilinear interpolation method.

Next, our problem is to refine feature locations by matching local image textures q_{ij}. For efficiency, we fix the shapes of the image patches q_{ij} and only allow the positions of their centers to change. This is not a problem because, as explained later, we iterate the whole procedure a couple of times and the shapes of the image patches improve over iterations. Note that this image patch optimization is fundamentally different from 3D patch optimization procedure performed by PMVS in that the optimization is carried out as 2D feature matching and does not enforce epipolar geometry constraints that come from possibly erroneous camera parameters. Let us call the camera with the minimum foreshortening factor with respect to P_i the *reference* camera of P_i, and use j_0 to denote its index. We fix the location p_{ij_0} in the reference camera and optimize every other element $p_{ij}, j \neq j_0$ one by one by maximizing the consistency between q_{ij_0} and q_{ij} in a multi-scale fashion. More concretely, starting from the level L of the image pyramids where PMVS was used, a conjugate gradient method is used to optimize p_{ij} by maximizing the normalized cross correlation between q_{ij_0} and q_{ij}. The process is repeated after convergence at the next lower level. After the optimization is complete at the bottom level, we check whether p_{ij} has not moved too much during the optimization. In particular, if p_{ij} has moved more than E_r pixels from its original

location, it is removed as an outlier and V_i is updated accordingly. Having refined feature correspondences, we then use the SBA bundle adjustment software (Lourakis and Argyros 2008) to update the camera parameters. In practice, we repeat the whole procedure (PMVS, multi-view feature matching, and SBA) four times to tighten up the camera calibration, while E_r is updated to be the mean plus three times the standard deviation of reprojection errors computed in the last step. Note that L is fixed across iterations instead of recomputed from E_r. This is for efficiency, since PMVS runs slowly with a small value of L.

4 Experimental Results and Discussions

4.1 Datasets

The proposed algorithm has been implemented in C++ and tested on six real datasets, with sample input images shown in Fig. 4, and the number of images and their (approximate) resolution listed in Table 1. The *vase* and *step* datasets have been calibrated by a local implementation (Courchay 2007) of a standard automated SFC/SFM/BA suite as described in Hartley and Zisserman (2004). For the *step* dataset, the input images are scaled-down by a factor of five to speed up the execution of the SFM software, but the full-resolution images are used for our refinement algorithm. Our SFM implementation fails on all other datasets except for *predator*, for which 14 out of the 24 images have been calibrated successfully. It is of course possible that a different implementation would have given better results, but we believe that this is rather typical of practical situations when different views are widely separated and/or textures are not prominent, and this is a good setting to exercise our algorithm. The *spiderman* dataset has been calibrated using a planar checkerboard pattern and a turntable with the calibration software from Bouguet (2008), and the same setup has been used to obtain a second set of camera parameters for the *predator* dataset. The *face* dataset was acquired outdoors, without a calibration chart, and textures are too weak for typical automated SFC/SFM algorithms to work. This is a typical case where, in post-production environments for example, feature correspondences would be manually inserted to calibrate cameras. This is what we have actually done for this dataset. The *dino* dataset is part of the Middlebury MVS evaluation project, and it has been carefully calibrated by the authors of Seitz et al. (2006). Nonetheless, this is a very interesting object lacking in salient features and a good example to test our algorithm. Therefore, we have artificially added Gaussian noise to the camera parameters so that reprojection errors become approximately six pixels, yielding a challenging dataset.

Probably due to the use of a rather inaccurate planar calibration board, and a turntable that may not be exactly repetitive, careful visual inspection reveals that *spiderman* and *predator* contain some errors, in particular, for points far

Table 1 The number of images and their approximate resolution (in megapixels) are listed for each dataset. E_r is the expected reprojection error in pixels, L is the level of image pyramids used by PMVS, N_p is the number of patches reconstructed by PMVS, and N_t is the number of patches that have successfully generated feature correspondences after sub-sampling

	# of images	# of pixels	E_r	L	N_p	N_t
vase	21	3 M	12	3	9926	1310
dino	16	0.3 M	7	2	5912	1763
face	13	1.5 M	8	3	7347	1997
spiderman	16	1 M	7	2	3344	840
predator	24	2 M	7	2	12760	3587
step	7	6 M	5	2	106806	9500

Fig. 4 Sample pictures for the six datasets used in the experiments

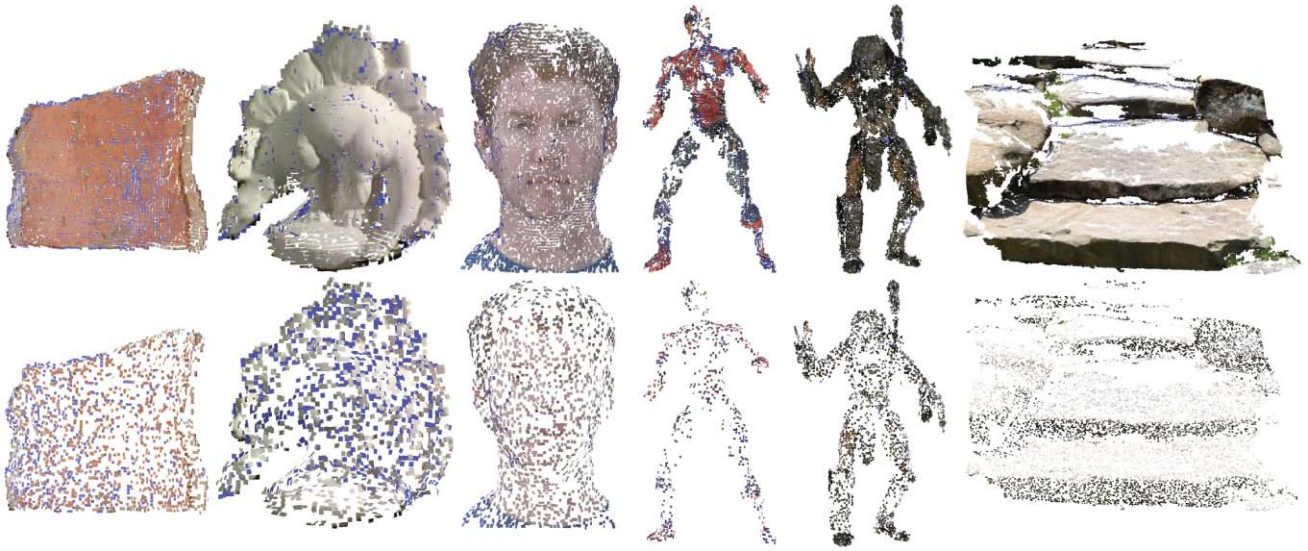

Fig. 5 *Top*: Patches reconstructed by PMVS at level L of the pyramid. *Bottom*: Subsets of these patches that have successfully generated feature correspondences after sub-sampling

away from the turntable where the calibration board was placed. The calibration of *face* is not tight either, because of the sparse manual feature correspondences (at most a few dozens among close-by views) used to calibrate the cameras. The *vase* dataset has relatively small reprojection errors with many close-by images for which SFM algorithms work well, but some images contain large reprojection errors because of the use of a flash and the limited depth of field, and errors do accumulate. The *step* data set does not have these problems, but since scaled-down images are used for the SFC/SFM/BA system, it contains some errors in full resolution images. Note that since silhouette information is used both by the PMVS software and the visual hull computations described in the next section, object silhouettes have been manually extracted using PhotoShop for all datasets except *dino*, where background pixels are close to black, and thresholding followed by morphological operations is sufficient to obtain the silhouettes. Note that the silhouette extraction is not essential for our algorithm, although it helps the system to run and converge more quickly. Furthermore, the use of PMVS is not essential either and this software can be replaced by any other multi-view stereo system.

4.2 Experiments

The two main parameters of PMVS are a correlation window size γ, and a parameter ζ controlling the density of the reconstruction: PMVS tries to reconstruct at least one patch in every $\zeta \times \zeta$ image window. We use $\gamma = 7$ or 9 and $\zeta = 2$ or 4 in all our experiments. Figure 5 shows for each dataset a set of patches reconstructed by PMVS (top row), and its subset that have successfully generated feature correspondences after sub-sampling (bottom row). Table 1 gives some statistics on the matching procedure. E_r denotes a conservative estimate of the expected reprojection errors in pixels, and L denotes the level of image pyramids used by PMVS to reconstruct a set of patches. The number of patches reconstructed by PMVS is denoted by N_p, and the number of patches that successfully generated feature correspondences after sub-sampling is denoted by N_t.

Examples of matched 2D features for each dataset are shown in Fig. 6. The histograms of the numbers of images where features are matched by the proposed algorithm are given in Fig. 7. By taking into account the surface orientation and the visibility information estimated by PMVS, the proposed method has been able to match features in many views taken from quite different angles even when image textures are very weak, and hence, producing strong constraints for the BA step. This is also clear from Fig. 8 that shows histograms for feature correspondences obtained by standard SFC/SFM/BA procedure (Courchay 2007) for the *vase* and *step* datasets,[5] and illustrates the fact that features are matched in fewer images compared to the proposed method.

It is impossible to give a full quantitative evaluation of our results given the lack of ground truth 3D data, because constructing such dataset is difficult and expensive, which is beyond the scope of this paper. We can, however, demonstrate that our camera calibration procedure does its job as far as improving the reprojection errors of the patches associated with the established feature correspondences. Fig-

[5]Histograms are shown only for *vase* and *step* in Fig. 8, because the SFC/SFM/BA software (Courchay 2007) fails on the other datasets due to the problems mentioned earlier.

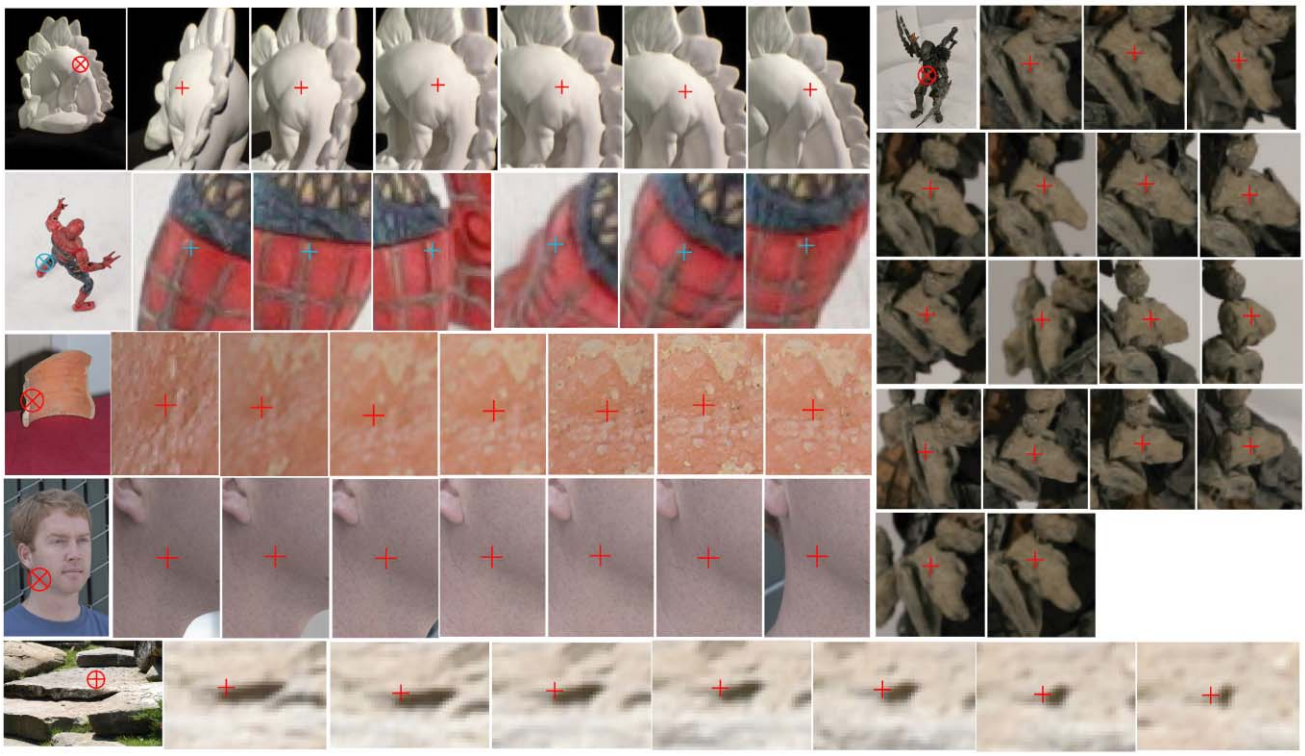

Fig. 6 A set of matching 2D features is shown for each dataset. The proposed method is able to match features in many images even without salient textures due to the use of surface geometry and visibility information estimated by the multi-view stereo algorithm

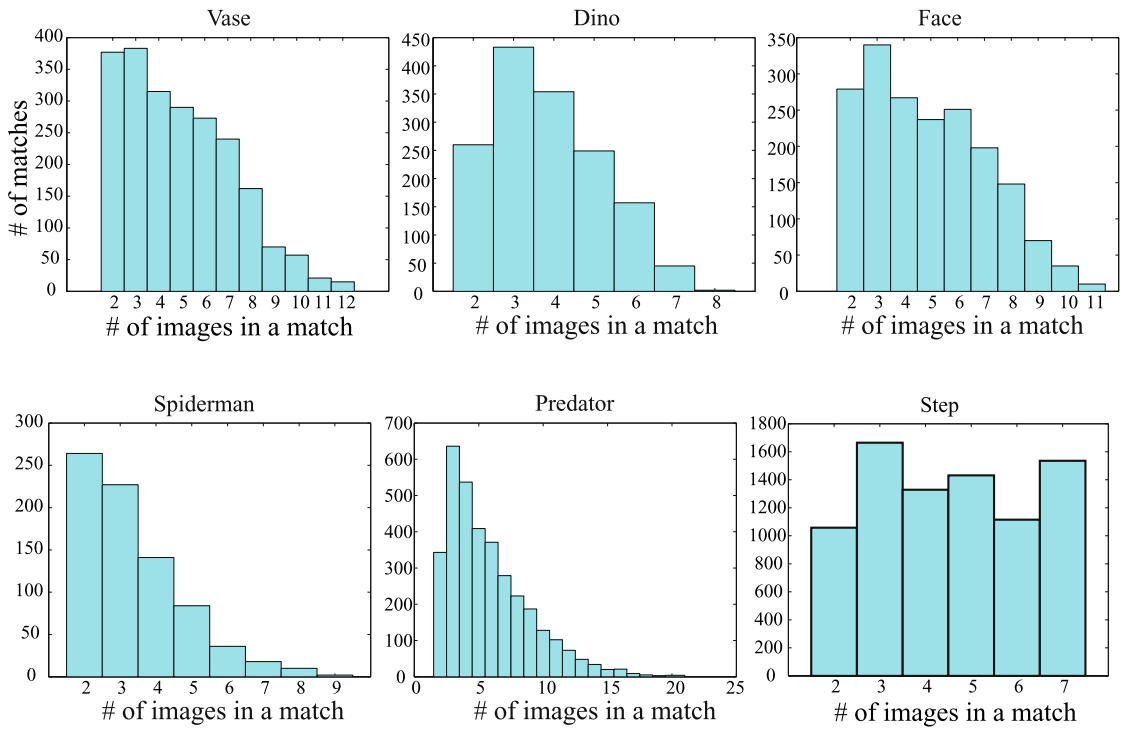

Fig. 7 Histograms of the number of images in which features are matched by the proposed algorithm

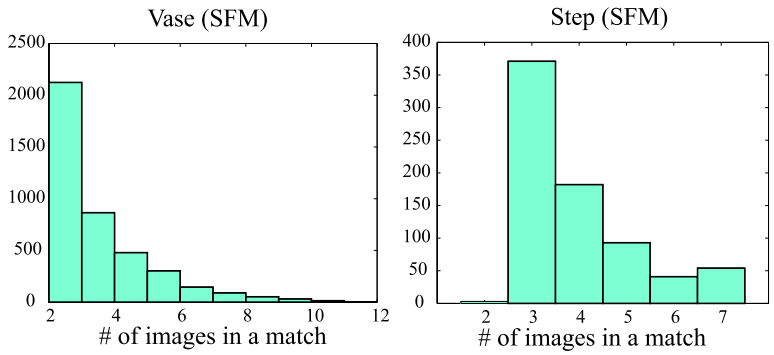

Fig. 8 Histograms of the number of images in which features are matched with a standard SFC/SfM/BA software (Courchay 2007) for *vase* and *step* datasets. In comparison to the proposed algorithm whose results are presented in Fig. 7, features are matched in fewer images

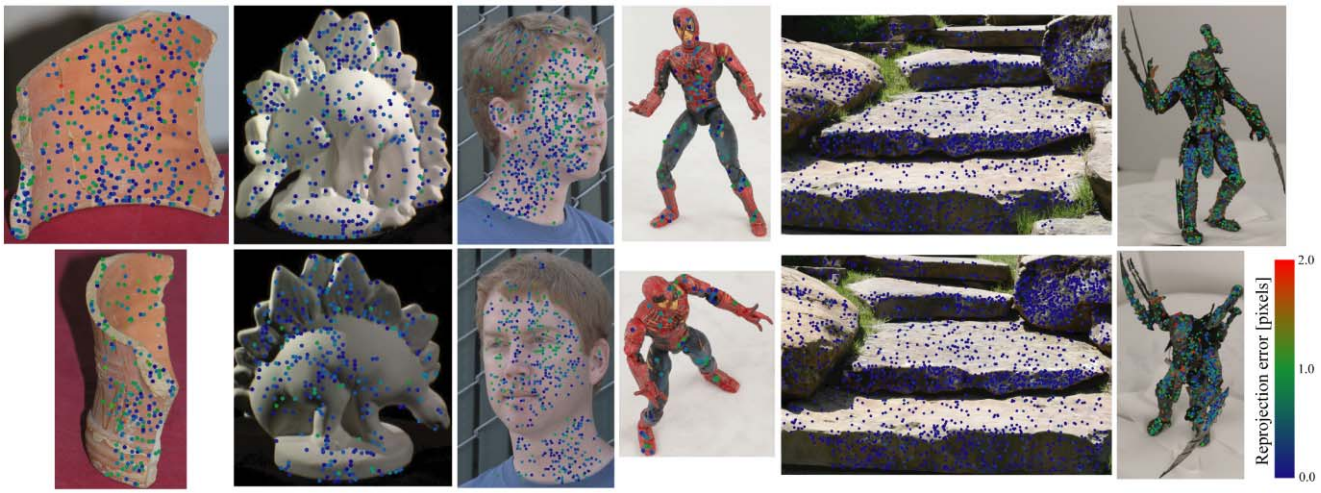

Fig. 9 Matched image features are shown for each data set. The colors represent the associated reprojection errors computed after the last bundle adjustment step. See text for more details

ure 9 shows matched images features for each dataset, while their colors represent the amounts of the associated final reprojection errors: Red, green, and blue corresponds to two, one, and zero pixels, respectively. Figure 10 shows the mean and standard deviation of these reprojection errors at each iteration of our algorithm for every dataset. The bottom-left graph shows the number of 2D features matched and used to refine camera parameters for the six iterations. The mean reprojection error decreases from 2–3 pixels before refinement to about 0.25 to 0.5 pixels for most datasets. As described earlier, the process is repeated for four iterations in practice to obtain the final camera parameters, as the two extra iterations in Fig. 10 show a decrease in error but do not seem to affect the quality of our reconstructions much. Note that the following assessment is performed after the fourth iteration of our algorithm.

We have used a couple of different methods to qualitatively assess the accuracy of the estimated camera parameters. First, epipolar geometry has been used to check the consistency between pairs of images (Fig. 11). More concretely, for a pair of images, we draw pairs of epipolar lines in different colors to see if corresponding epipolar lines of the same color pass through the same feature points in the two images. Several images in the *vase* dataset contained large errors before refinement (approximately six pixels in some places) because of the limited depth of field and an exposure difference due to the use of a flash. The *spiderman* and *predator* datasets also contain very large errors, up to seven (or possibly more) pixels for points far from the ground plane where the calibration chart is located. In each case, the proposed method has been able to refine camera parameters to sub-pixel level precision. Inconsistencies in the *dino* dataset introduced by the added noise have also been corrected by our method despite its weak texture.

Next, we have tested the ability of our algorithm to recover camera parameters that are highly consistent across widely separated views. We use the *spiderman* and *predator* datasets in this experiment (Fig. 12) since parts of these objects are as thin as a few pixels in many images. Recovering such intricate structures normally requires exploiting silhouette information in the form of a visual hull (Baumgart 1974) or a hybrid model combining silhouette and texture information (Furukawa and Ponce 2006; Hernández Esteban and Schmitt 2004; Sinha and Pollefeys 2005;

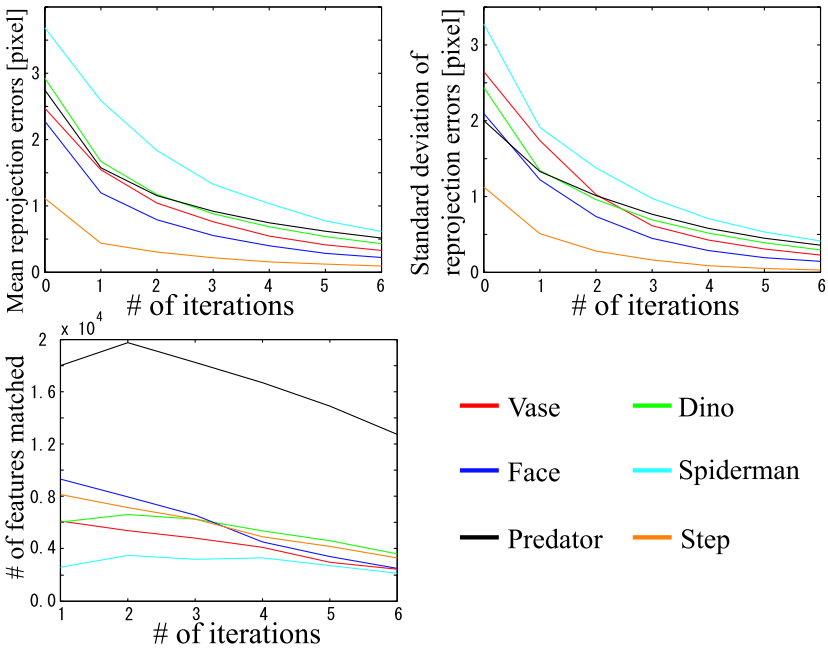

Fig. 10 The mean and standard deviation of reprojection errors in pixel for each dataset at each iteration. The *bottom-left* graph shows the total number of matched 2D features per iteration

Tran and Davis 2006). In turn, this requires a high degree of geometric consistency over the cameras, and provides a good testing ground for our algorithm. We have used the EPVH software of Franco and Boyer (2003) to construct polyedral visual hulls in our experiments, and Fig. 12 shows that thin, intricate details such as the fingers of *spiderman* and the blades of *predator* are successfully recovered with refined camera parameters, and completely lost otherwise.

For *dino* and *face*, we have used PMVS to reconstruct a set of patches that are then converted into a 3D mesh model using the method described in Kazhdan et al. (2006) (Fig. 12, bottom right). The large artifacts at the neck and the chin of the shaded *face* reconstruction before refinement are mainly side effects of the use of visual hull constraints in PMVS (patches are not reconstructed outside the visual hull, Furukawa and Ponce 2007), exacerbated by the fact that the meshing method of Kazhdan et al. (2006) extrapolates the surface in areas where data is not present. Ignoring these artifacts, the difference in quality between the reconstructions before and after refinement is still obvious in Fig. 12, near the fins of the dinosaur, or the nose and mouth of the face for example. In general, however, the accumulation of errors due to geometric inconsistencies among widely separated cameras is not always visually recognizable in 3D models reconstructed by multi-view stereo, because detailed local reconstructions can be obtained from a set of close cameras, and wide-baseline inconsistencies turn out as low-frequency errors. In order to assess the effectiveness of our algorithm in handling this issue, we pick a pair of widely separated cameras C_1 and C_2, map a texture from one camera C_1 onto the reconstructed model, render it as seen from C_2, and compare the rendered model with the input image associated with C_2. The two images should look the same (besides exposure differences) when the camera parameters and the 3D model are accurate. Figure 13 illustrates this on the *vase* and *face* datasets: Mesh models obtained again by combining PMVS (Furukawa and Ponce 2007) and the surface extraction algorithm of Kazhdan et al. (2006) are shown for both the initial and refined camera parameters. Although the reconstructed *vase* models do not look very different, the amount of *drifting* between rendered and input images is approximately six pixels for initial camera parameters. Similarly, for the *face* model, the reconstructed surfaces at the left cheek just beside the nose look detailed and similar to each other, while the rendered image is off by approximately six pixels as well. In both cases, the error decreases to sub-pixel levels after refinement. Note that reducing low-frequency errors may not necessarily improve the appearance of 3D models, but is essential in obtaining *accuracy* in applications where the actual model geometry, and not just their overall appearance, is important (e.g., engineering data analysis or high-fidelity surface modeling in the game and movie industries).

Finally, the running time in minutes per iteration of the three steps (PMVS, feature matching, bundle adjustment) of the proposed algorithm on a Dual Xeon 3.2 GHz PC is given in Table 2. As shown by the table, the proposed algorithm is efficient and takes at most a few minutes per iteration to refine camera parameters. Note that the running time of typical CBC systems is also in an order of a few minutes for these data sets. SFC/SFM/BA systems are more computationally expensive, and in particular, a local implementation (Matlab) of a standard SFC/SFM/BA software takes several hours to calibrate the *step* data set with full resolution images. As explained before, the proposed approach reduces

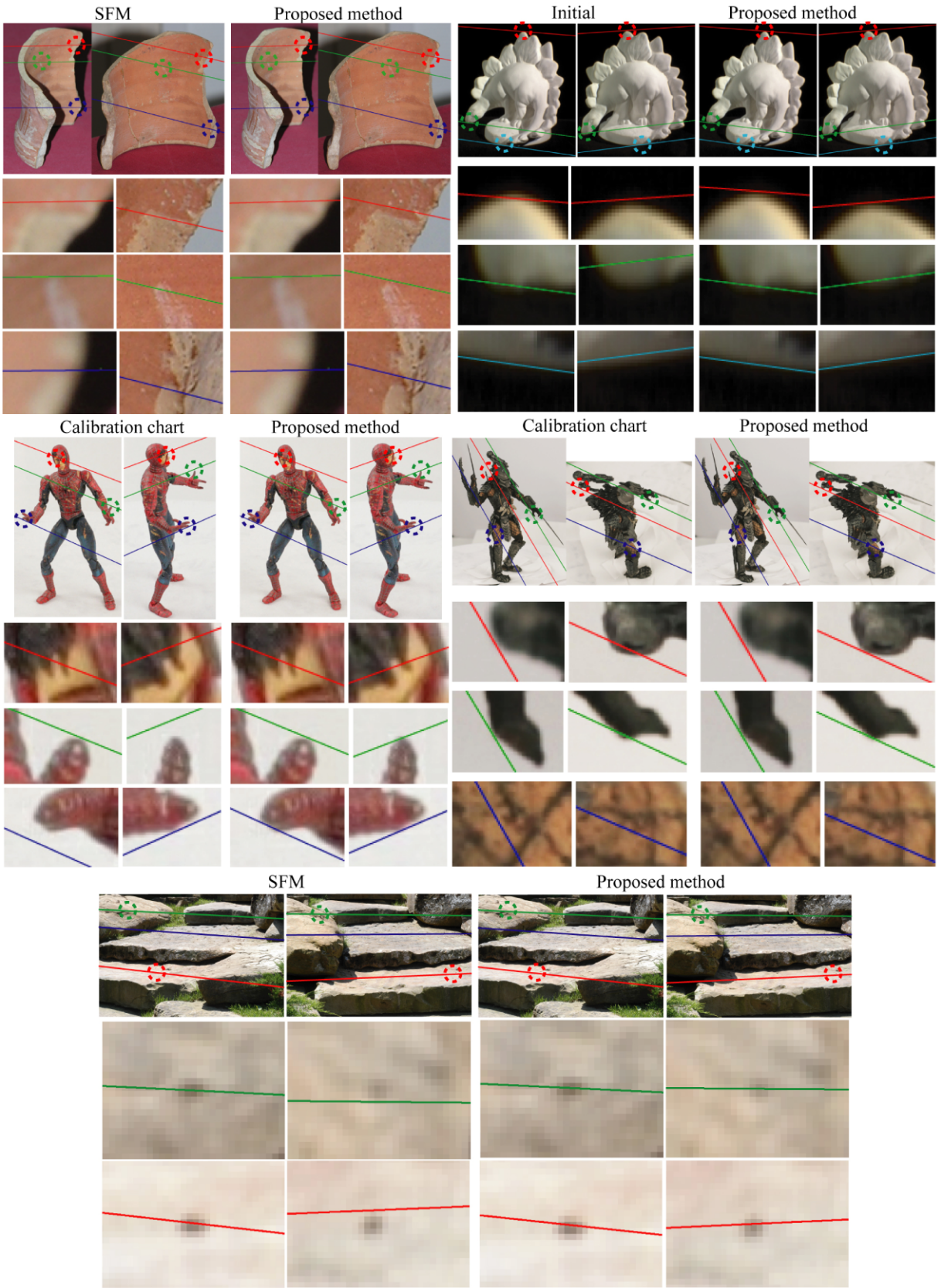

Fig. 11 Epipolar lines are used to assess the improvements in camera parameters. A pair of epipolar lines of the same color must pass through the same feature points

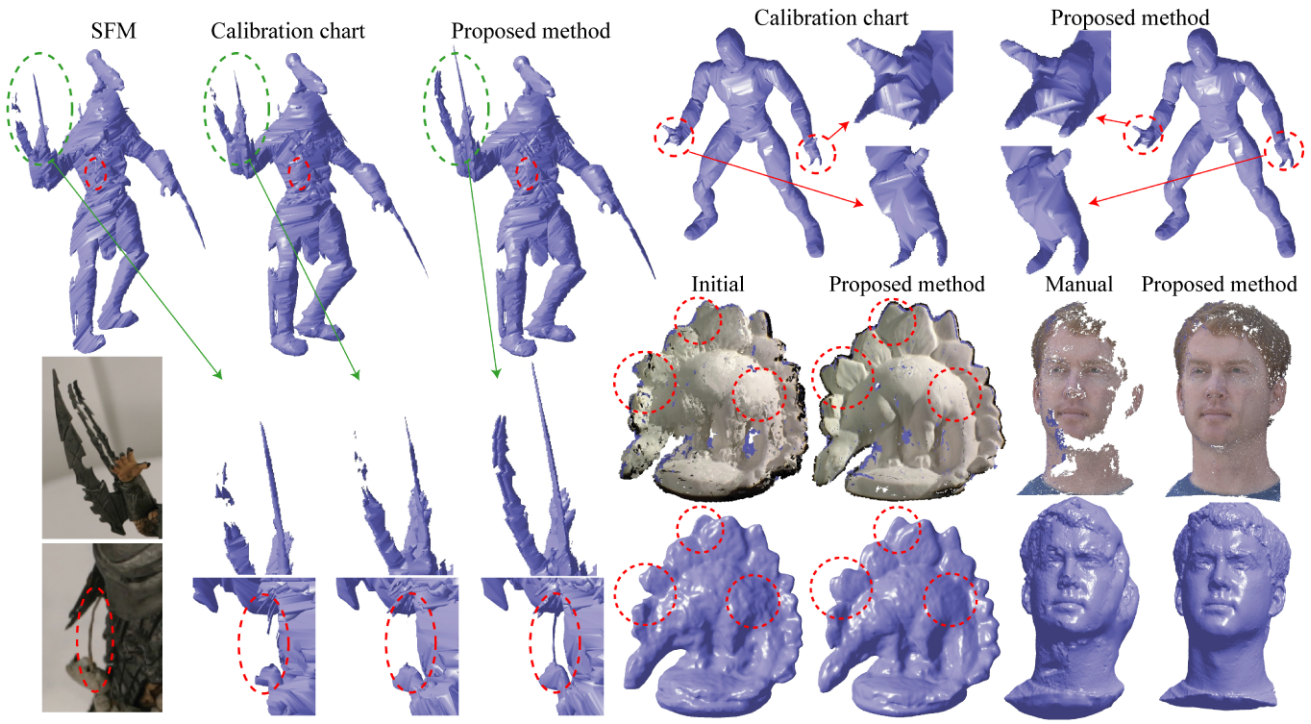

Fig. 12 Visual hull models are used to assess the accuracy of camera parameters for *spiderman* and *predator*. Intricate structures are reconstructed only from the camera parameters refined by the proposed method. For *dino* and *face*, a set of patches reconstructed by PMVS and a 3D mesh model extracted from these patches are used for the assessment. See text for more details

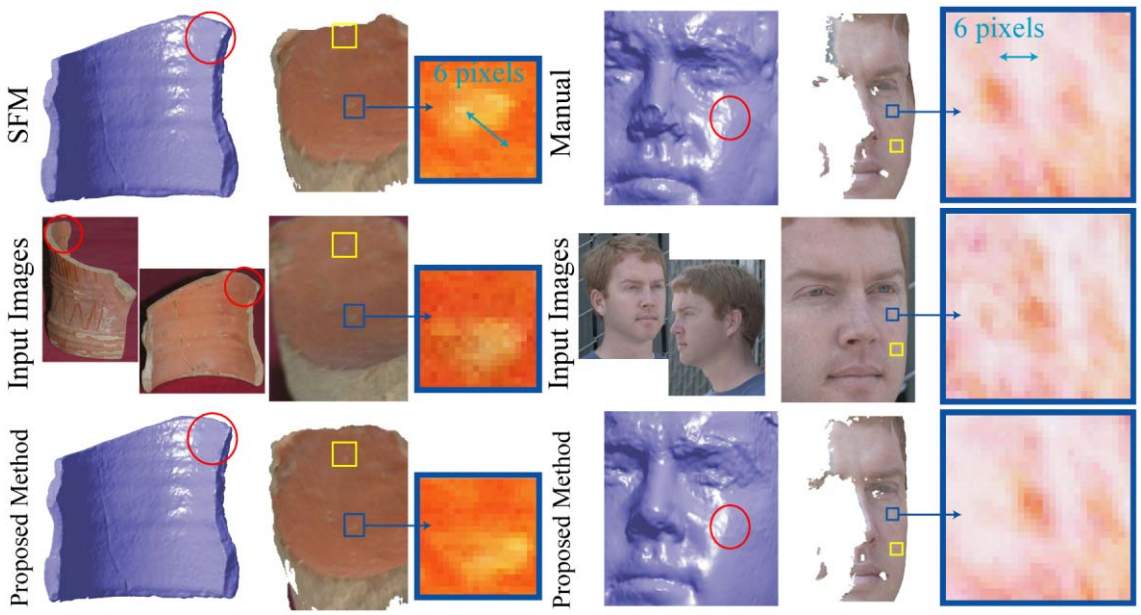

Fig. 13 Inconsistencies in widely separated cameras (accumulation errors) are often not recognizable from 3D mesh models reconstructed by a MVS algorithm. For further assessments, we pick a pair of separated cameras shown in the *middle row*, texture-map the surface from the *right* image, render it to the *left*, and compare the rendered model with the *left image*. The rendered and the input images look the same only if camera parameters and the reconstructed model are accurate. The *top* and the *bottom rows* show rendered images and the reconstructed 3D mesh model before and after the refinement, respectively. The amount of errors with the initial camera parameters (calibrated by SFM for *vase* and manual feature correspondences for *face*) is roughly six pixels for both datasets, which are very large

Table 2 Running time in minutes of the three steps of the proposed algorithm for the first iteration

	vase	*dino*	*face*	*spiderman*	*predator*	*step*
PMVS	1.9	0.40	0.65	0.34	1.9	4.0
Match	1.1	0.66	0.96	0.24	1.6	0.39
BA	0.17	0.13	0.17	0.03	0.38	1.2

such computational expenses for the step data set by first running a SFC/SFM/BA system with scaled-down input images, which takes only a few minutes, then using the proposed method to tighten up camera calibration.

5 Conclusion

We have proposed a novel approach for camera calibration where top-down information from rough camera parameter estimates and the output of a multi-view stereo system on scaled-down input images is used to effectively establish feature correspondences. By taking into account the surface orientation and the visibility information estimated by a multi-view stereo system, the proposed method has been able to match features in many views taken from quite different angles even when image textures are very weak. We have performed three different ways to qualitatively assess the accuracy of refined camera calibration, which shows that the proposed method has successfully reduced calibration errors significantly. Future work will focus on the analysis of remaining errors and influences of various factors that have been ignored in the current framework, such as the second order effects in the camera projection model (distortions) or surface reflectance properties that are assumed to be Lambertian. The implementation of the proposed algorithm is publicly available at Furukawa and Ponce (2008b).

Acknowledgements This paper was supported in part by the National Science Foundation under grant IIS-0535152, the INRIA associated team Thetys, and the Agence Nationale de la Recherch under grants Hfimbr and Triangles. We thank S. Sullivan, A. Suter, and Industrial Light and Magic for the face data set and support of this work. We also thank Jerome Courchay for the SfM software, and Jean-Baptiste Houal for the vase dataset.

References

Baumgart, B. (1974). *Geometric modeling for computer vision*. Ph.D. thesis, Stanford University.

Bouguet, J. Y. (2008). *Camera calibration toolbox for matlab*. http://www.vision.caltech.edu/bouguetj/calib_doc.

Courchay, J. (2007). *Auto-calibration á partir d'une séquence d'images* (MVA internship report).

DXO (2008). *DxO Labs*. DxO Optics Pro (http://www.dxo.com).

Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM 24*(6).

Franco, J. B., & Boyer, E. (2003). Exact polyhedral visual hulls. In *BMVC*.

Fua, P. (2000). Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, *38*(2), 153–171. doi:10.1023/A:1008105802790.

Furukawa, Y., & Ponce, J. (2006). Carved visual hulls for image-based modeling. In *ECCV* (pp. 564–577).

Furukawa, Y., & Ponce, J. (2007). Accurate, dense, and robust multi-view stereopsis. In *CVPR*.

Furukawa, Y., & Ponce, J. (2008a). Accurate camera calibration from multi-view stereo and bundle adjustment. In *CVPR*.

Furukawa, Y., & Ponce, J. (2008b). *PBA*. http://www.cs.washington.edu/homes/furukawa/research/pba.

Furukawa, Y., & Ponce, J. (2008c). *PMVS*. http://www.cs.washington.edu/homes/furukawa/research/pmvs.

Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Hernández Esteban, C., & Schmitt, F. (2004). Silhouette and stereo fusion for 3D object modeling. *CVIU 96*(3).

Hernández Esteban, C., Schmitt, F., & Cipolla, R. (2007). Silhouette coherence for camera calibration under circular motion. *PAMI 29*.

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. In *Symp. Geom. Proc.*

Lavest, J. M., Viala, M., & Dhome, M. (1998). Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In *ECCV*.

Lourakis, M., & Argyros, A. (2008). SBA: *A generic sparse bundle adjustment C/C++ package based on the Levenberg-Marquardt algorithm*. http://www.ics.forth.gr/~lourakis/sba/.

Martinec, D., & Pajdla, T. (2007). Robust rotation and translation estimation in multiview reconstruction. In *CVPR* (pp. 1–8).

Nister, D. (2004). An efficient solution to the five-point relative pose problem. *PAMI 26*(6).

Pollefeys, M., Gool, L. V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., & Koch, R. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*, *59*(3), 207–232. doi:10.1023/B:VISI.0000025798.50602.3a.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*.

Sinha, S., & Pollefeys, M. (2005). Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *ICCV*.

Tran, S., & Davis, L. (2006). 3d surface reconstruction using graph cuts with surface constraints. In *ECCV*.

Triggs, B., McLauchlan, P., Hartley, R., & Fitzgibbon, A. (2000). Bundle adjustment—A modern synthesis. In W. Triggs, A. Zisserman, & R. Szeliski (Eds.). *Vision algorithms: theory and practice* (pp. 298–375). Berlin: Springer.

Triggs, W. (1997). Auto-calibration and the absolute quadric. In *CVPR*.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras. *Robotics and Automation* 3(4).

Uffenkamp, V. (1993). State of the art of high precision industrial photogrammetry. In *Third International Workshop on Accelerator Alignment*, Annecy, France.

Wong, K. K., & Cipolla, R. (2004). Reconstruction of sculpture from its profiles with unknown camera positions. *IEEE Transactions on Image Processing*.