# A Fast and Robust Framework for Semi-Automatic and Automatic Registration of Photographs to 3D Geometry

Ruggero Pintus, Yale University, USA and CRS4, Italy
Enrico Gobbetti, CRS4, Italy

We present a simple, fast and robust complete framework for 2D/3D registration capable to align in a semi-automatic or completely automatic manner a large set of unordered images to a massive point cloud. Our method converts the hard to solve image-to-geometry registration task in a Structure-from-Motion (SfM) plus a 3D/3D alignment problem. We exploit a SfM framework that, starting just from an unordered image collection, computes an estimate of the camera parameters and a sparse 3D geometry deriving from matched image features. We then coarsely register this model to the given 3D geometry by estimating a global scale and absolute orientation using two solutions: a minimal user intervention or a stochastic global point set registration approach. A specialized sparse bundle adjustment (SBA) step, that exploits the correspondence between the sparse geometry and the fine input 3D model, is then used to refine intrinsic and extrinsic parameters of each camera. Output data is suitable for photo blending frameworks to produce seamless colored models. The effectiveness of the method is demonstrated on a series of synthetic and real-world 2D/3D Cultural Heritage datasets.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**] Picture and Image Generation; I.3.7 [**Computer Graphics**] Three-Dimensional Graphics and Realism

General Terms: Cultural Heritage

Additional Key Words and Phrases: image registration, photo blending

**ACM Reference Format:**

Ruggero Pintus, Enrico Gobbetti. 2013. A Fast and Robust Framework for Semi-Automatic and Automatic Registration of Photographs to 3D Geometry. *ACM J. Comput. Cult. Herit.* 0, 0, Article 0 (January 2015), 23 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

## 1. INTRODUCTION

Modern 3D acquisition systems are able to rapidly digitize an object geometry with high accuracy and resolution, producing massive digital models with billions of samples. Such highly detailed models are extremely well suited for Cultural Heritage (CH), where both dense and extensive sampling is required. Just a dense geometry is, however, not enough for all CH needs: additional color information plays a key role in reconstructing a high-quality digital model.

Many approaches exist to obtain object color. Some range scanners also acquire color, but their color resolution and quality are often insufficient for CH purposes. Moreover, some of them lack this capability at all. One possible and automatic solution is a calibrated camera rigidly mounted on the

Address: CRS4, POLARIS Ed. 1, 09010 Pula (CA), Italy; email: firstname.lastname@crs4.it www: http://www.crs4.it/vic/

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2015 ACM 1556-4673/2015/01-ART0 $15.00

DOI:http://dx.doi.org/10.1145/0000000.0000000

scanner. Unfortunately, the different position of the range sensor and image sensor results in possible occlusions, so that the color in some portions of the geometry will probably be missed. Even when these alignment problems can be solved, this simultaneous acquisition approach has too many limitations. For example, lighting conditions need often to be different between 3D scanning and photographic campaigns, and the photographic dataset is often required to be modified at a later time, e.g., to evaluate the effects of restoration. Some modern high resolution scanners do not even provide the possibility to attach additional cameras for color acquisition (e.g., Minolta Range7 [Minolta/Range7 2013]). Moreover, although the presence of a mounted camera makes the coarse alignment straightforward, by simply including those images into the SfM framework presented in Pintus et al. [2011c], it won't work with a general 2D signal. Multi-spectral (MS) acquisitions (e.g., Far Ultra-Violet, Far Infra-Red Thermal images) are very important in Cultural Heritage preservation and restoration, and they could contain very different appearance features, due to the different absorption and transmittance properties of these wavelengths, which are far from the visible spectra. Since these signals are hardly comparable, a multi-view approach that includes both them and RGB data will be more likely to fail. This kind of capture is typically performed using devices that are not available in modern scanners, and in general they cannot be mounted on them. Moreover, the different acquisition times and hardware setups in the 3D and MS acquisitions make these two pipelines difficult to be performed simultaneously.

Recent powerful sensors allow us to effectively measure color with off-the-shelf cameras. Mapping acquired images requires the solution of a 2D/3D registration problems. Several approaches have been proposed that cope with the image to 3D geometry registration problem, ranging to reliable but time-consuming manual approaches to more effective (semi-)automatic techniques, which are, however, typically applicable only to limited object classes (see Sec. 2).

Our main contribution is a practically useful, robust method for simultaneously registering a unordered collection of photographs to a 3D geometry. Our method converts the hard to solve image-to-geometry registration task to a Structure-from-Motion (SfM) plus a 3D/3D alignment problem. We exploit a SfM framework that, starting just from an unordered image collection, computes an estimate of the camera parameters and a sparse 3D geometry deriving from matched image features. We then use this data to compute a coarse 2D/3D registration by aligning the 3D point cloud produced by the SFM to the geometry of the object. Finally, a specialized sparse bundle adjustment (SBA) step, that exploits the correspondence between the sparse geometry and the fine input 3D model, refines intrinsic and extrinsic parameters of each camera.

This work is a significantly extended version of our VAST2011 contribution [Pintus et al. 2011c]. Besides supplying a more thorough exposition, we provide here significant new material. Our main novel contributions are the following:

- a completely automatic solution for estimating a coarse image-to-geometry alignment by using a GPU-based global affine 3D point set stochastic registration approach (Section 6.2); the method complements our previous semi-automatic technique. Besides its comparable quality in registration accuracy, the automatic approach opens the door to the more interesting implementation of unattended alignment and mapping services;
- a robustified and adaptive method, using robust statistics to cope with outliers and adaptive local tolerances based on sample spacing to manage non-uniformly sampled 3D geometry; the method is therefore more general and robust with respect to our previous solution and does not require the hand-tuning of tolerance parameters (Section 7);
- an extensive quantitative and qualitative evaluation of our method on a series of synthetic and real-world 2D/3D CH datasets (Section 8).

Our results demonstrate that this registration pipeline provides high quality results and consistent color mapping all over the surface of 3D models.

## 2. RELATED WORK

Our system extends and combines state-of-the-art results in a number of technological areas. In the following, we only discuss the approaches most closely related to our novel contributions. We refer the reader to the classic survey of Hantak and Lastra [2006] for a general verview of registration results in terms of different information-theoretic metrics. This paper focuses on the registration step of the colorization pipe-line. Coloring the model starting from registration data is an orthogonal problem. Results presented here are based on our color blending pipe-line [Pintus et al. 2011b; 2011a].

### 2.1 2D/3D Registration

*Manual 2D/3D correspondence selection.* The classic photo-mapping approach requires users to manually define correspondences between the image and the 3D model, typically using a point-and-click interface [Dellepiane et al. 2008], which are then refined through error minimization. Since this straightforward approach is tiring and time-consuming, research has focused on reducing or simplifying manual operations, e.g., by assisting the user by showing possible feature matches between photos and rendered model [Borgeat et al. 2009], or reducing the manual effort by exploiting both matches between image and geometry, and correspondences between pixels in different images [Franken et al. 2005]. These manual methods are robust, but easily become hard to apply to large image sets. Our method automatically computes 2D/3D alignment and remains suitable even if the image set size grows (hundreds of photos).

*Automatic 2D/3D feature detection and matching.* Feature-based techniques match 3D features with image features to solve the image-to-geometry problem in a completely automatic framework. This problem is in general very complex, since photographs and geometric models have a very different appearance. For these reasons, methods in this area are limited to some specific models (e.g., architectural models with sharp edges in 3D and high contrast features in 2D). A class of methods [Kaminsky et al. 2009; Stamos and Alien 2001] rely on the presence, in outdoor and indoor scenes from 3D LIDAR scanning, of linear edges in the geometry, and straight lines in images of maps and/or floor plans. Methods exploiting silhouette information to find the camera poses by minimizing the error between the contours of the rendered input 3D model and the object in the input images [Lowe 1991; Brunie et al. 1992; Lensch et al. 2000] typically need to have the whole object visible in each image and a good separation between foreground and background. Other approaches rely on linear or circular 3D features [Stamos et al. 2008], orthogonality constraints [Liu and Stamos 2005], edge intensity [Neugebauer and Klein 1999], clusters of vertical and horizontal lines [Liu et al. 2006], or viewpoint-invariant patches with strong geometric features [Wu et al. 2008]. We do not rely on particular geometrical features defined a priori, such as lines, orthogonal planes, circular features. As a SfM-based approach, the proposed method works in the general case of a moderate presence of any kind of geometrical and/or texture features, which is normally required by any classical SfM algorithms. Our method is more generally applicable, since we do not rely on finding similarities between images and geometry, but only among images, which is a much simpler problem.

*Semi-automatic 2D/3D statistical registration.* Intensity-based registration techniques rely on global measures such as photo- consistency and mutual information [Viola and Wells 1997], avoiding feature extraction. Correlation is maximized between image content and some measure present in the range maps, such as normals [Viola and Wells 1997], intensity of the reflected laser beam [Williams et al. 2004; Hantak and Lastra 2006], reflectance [Ikeuchi et al. 2007], or LIDAR elevation and probability of detection [Mastin et al. 2009], and simulated renderings [Cleju and Saupe 2007; Corsini et al. 2013]. These approaches require, however, a manual camera pose initialization to converge to the right solution, and attributes used for correlation purposes are not always available. Again, our method is more generally applicable, since it does not depend on any additional attribute and does not require manual camera pose initialization.

*Geometric multi-view reconstruction and matching.* Robust multi-view reconstruction techniques derive both a global coarse estimation of camera poses and a sparse point cloud from images, corresponding to triangulated feature points. The alignment of the dense input geometry with the computed sparse point cloud implicitly solves the original 2D/3D registration problem. Zhao et al. [2005] recover relative camera positions and a point cloud from a video sequence using motion stereo. The user has to manually register only two frames with the 3D model to obtain absolute orientation and global scale. Instead of being limited to dense and ordered frame sequences, our method deals with unordered sets of sufficiently overlapping photos. It then refines camera parameters during 3D/3D registration, and does not require user intervention. Although a uniform image sampling might be a good starting point, eventually some complex regions must be acquired with an adaptive number of frames, which depends on the number and the nature of surface occlusions. This will produce in practice an unordered image sequence, with an ordered sequence as one of its possible subsets. Further, adopting unordered frame sequences results in two main advantages: besides the constraints given by the SfM algorithm (i.e., sufficient overlap between images), we do not properly require a dense image sampling, saving the memory footprint of the 2D dataset; we do not even rely on known ordered captures, coping with a more general and more common case in the Cultural Heritage field. Moreover, it will be possible to merge additional photographic datasets acquired in the future with different capture strategies. Similarly to our approach, Li and Low [2009] apply SfM to an image set. However, their refinement step depends on the presence of artificially textured planes in the geometry, obtained by projecting special light patterns. Further, their cost function mixes measures in both world (point-to-plane distances) and pixel (re-projection errors) coordinates, weighting these terms with a heuristic parameter, that heavily depends on the object geometry/extent, and requires manual tuning. Conversely, our energy function contains only squared error measurements in image space and does not require any additional parameter. The most closely related works in this area are the method proposed by Banno and Ikeuchi [2010], and our VAST2011 contribution [Pintus et al. 2011c]. They presented similar semi-automatic techniques that requires some user-defined parameters and a minimal amount of user manual operations. After the initial manual alignment they launch a refinement step based on robust minimization. They differ in the used image dataset and in the error metric employed in the minimization; the former employs spherical stereo images and an object space distance, while the latter uses a more general unordered image sequence and a cost function that relies on a 2D distance proportional to vertex re-projection errors. Pintus et al. [2011c] proposes an optimization procedure in order to refine the image-to-geometry registration based on a multi-view reconstruction and matching, but it is only applicable to uniformly sampled 3D models. Compared with these previous works, here we do not require user-defined parameters, we remove the user intervention with a fully automatic coarse registration, we provide the possibility to deal with non-uniformly sampled models, and we improve the robustness of the registration refinement by modifying the error cost function that drives the minimization. Recently, Corsini et al. [2013] proposed a closely related automatic method, exploiting an extension of the 4 Point Congruent Set (4PCS) algorithm [Aiger et al. 2008] for 3D/3D coarse alignment between the 3D model and the sparse point cloud resulting from SfM, and refining it using mutual information. Their image-to-geometry pipeline is general, robust and its main advantage is that it doesn't require any particular assumption regarding the objects, being capable of dealing with small and big geometry of any topology. Here we propose an orthogonal approach that automatically computes the coarse registration with a stochastic global method and performs refinement with a sparse bundle adjustment framework. In contrast to Corsini et al. [2013], we do not require a heavy preprocessing for cloud densification.

## 2.2 3D/3D registration

The 3D/3D registration step in our method must align the point cloud resulting from SfM with the 3D model by estimating relative scale, translation, and orientation. Point cloud registration has long been studied, and we refer the reader to the survey of Tam et al. [2013] for a very up-to-date coverage of the state-of-the-art. While the Iterative Closest Point (ICP) algorithm [Besl and McKay 1992; Chen and Medioni 1992] constitutes a gold standard for local alignment tasks, i.e., when a rough alignment already exists, efficiently finding an initial pose is still a very active area of research.

Classic techniques, such as the generalized Hough transform [Hecker and Bolle 1994], geometric hashing [Wolfson and Rigoutsos 1997], and pose clustering [George and Stockman 1987] are guaranteed to find the optimal solution (at least in the rigid case), but are limited to very small and noise-free point clouds. Noisy measurements are often handled through robust statistics, such as general maximum likelihood estimation [Granger and Pennec 2002], kernel correlation [Tsin and Kanade 2004], or mixture of Gaussians [Jian and Vemuri 2005]. Instead of using one-to-one correspondences, these approaches work with multiple, weighted correspondences. Although this significantly widens the basin of convergence, the computational cost limits the applicability to very small point clouds (hundreds of samples) [Tamaki et al. 2010].

The most common approaches for point cloud registration rely on local geometric descriptors, such as spin images [Johnson and Hebert 1999], or integral volume descriptors [Gelfand et al. 2005], which are detected in both clouds and then matched. Our problem, is, however, characterized by a strong asymmetry and large presence of outliers, since the point cloud derived from SfM is too sparse to reliably compute local descriptors and the overlap between photo-captured environment and 3D model is widely variable. The 4PCS method [Aiger et al. 2008] achieves robustness by combining a non-local descriptor (four coplanar points) with a generate-and-test RANSAC scheme. The method has been extended to Corsini et al. [2013] for the estimation of different scale between the point clouds to align. This approach, however, requires coarse point cloud densification, as well as a partitioning into planar regions of the two point clouds, which is achieved through variational shape approximation [Cohen-Steiner et al. 2004], and could fail if the SfM point cloud is too sparse.

As an alternative to combinatorial optimization based on feature matching, pose estimation can be attacked by minimizing a cost function with a global optimizer. For small datasets, rigid alignment algorithms have been proposed by using deterministic branch-and-bound methods [Breuel 2003; Olsson et al. 2009], or Lipschitz global optimization [Li and Hartley 2007]. Papazov and Burschka [2011] recently proposed a stochastic global optimization approach for rigid robust point set registration, based on Bilbro and Snyder's tree annealing algorithm [1991]. It is a stochastic sampling method which uses a generalized Binary Space Partitioning (BSP) tree and allows for minimizing nonlinear scalar fields over complex shaped search spaces like the space of rotations. As a result, the method is robust and outlier resistant. In our work, we extend this approach to similarity transforms, and combine the pure stochastic sampling approach with a more domain-specific method that locally solves absolute orientation problems. Outliers are handled by employing a robust and adaptive pruning strategy, and efficiently solving absolute orientation through iteratively re-weighted least square solution. This approach is made possible by harnessing the power of GPUs [Cayton 2010] to rapidly compute correspondences between the two point clouds during optimization.

## 3. TECHNIQUE OVERVIEW

Our technique is outlined in Fig. 1. We take as input a dense 3D model and a set of $n$ photographs. The photographic dataset can cover the complete surface of the 3D object, only a part of it, or a larger area. No constraints are placed on the nature of the input dense geometry; it could be either a triangle mesh or a point cloud, and we don't need particular geometric attributes (e.g., normals or influence radii) or the presence of known geometric features (e.g., lines).

Our 2D/3D calibration is performed in three main stages: SfM, semi-automatic or completely automatic coarse alignment, and fine registration. In the first stage, we apply a SfM algorithm for unordered image collection to self-calibrate images and obtain an initial sparse 3D reconstruction of the part of the model covered by the photographic campaign (Sec. 5). This provides us a sparse 3D model derived from matched image features, all camera poses in a common reference frame, and the intrinsic parameters of each camera.

In the second stage, the SfM model, reconstructed up to an unknown scale-factor, is coarsely aligned in a semi-automatic or automatic manner to the dense input 3D model (Sec. 6). In the semi-automatic pipeline the user manually selects correspondences between a small subset of photos (typically just one) and the detailed model. These matches and the camera parameters are used to solve for the affine transformation that maps the SfM world to the dense model. In the latter case we reach the same result with a completely automatic stochastic global optimization approach, by exploiting the power of modern GPU devices.

In the final stage, a SBA calculates the final registration in a non-rigid deformable manner, constraining the features detected in the images to lie on the fine 3D model, (Sec. 7).

The output data (camera parameters) can then be used, together with the $n$ photos and the dense model, to blend the texture data on the geometry to produce a globally coherent colored model.

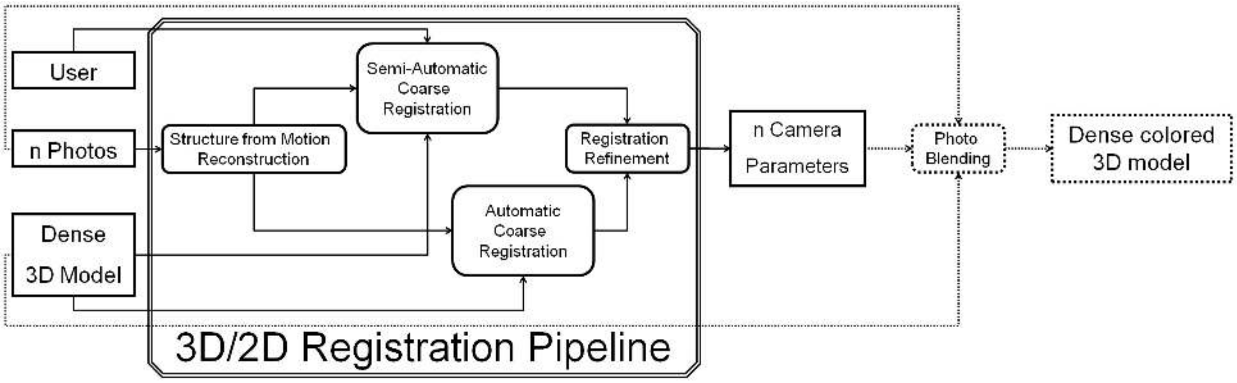

User
n Photos
Dense 3D Model
Structure from Motion Reconstruction
Semi-Automatic Coarse Registration
Automatic Coarse Registration
Registration Refinement
n Camera Parameters
Photo Blending
Dense colored 3D model
3D/2D Registration Pipeline

Fig. 1. **Pipeline.** Given the image set, a SfM algorithm computes a sparse point cloud and related camera poses. In a semi-automatic (minimal user intervention) or automatic (stochastic global optimization) manner we coarsely register the SfM and the input model. The final registration, refined with a specialized SBA, can be used to obtain a globally coherent colored model, blending all registered photos together on the input point cloud.

## 4. PHOTO CAPTURE

Besides avoiding to take images with excessive blur or noise, and under- or over-exposed regions, our pipeline does not impose particular constraints on the image set, since SfM algorithms exist to cope with challenging data, such as images that exhibit large variations in illumination, viewpoint, zoom, resolution, and contain outliers and clutters. For a description of typical SfM capabilities and limitations see the work of Snavely et al. [Snavely et al. 2008]. Further, techniques exist which perform texture blending for producing seamless colored models with such non-ideal color information [Pintus et al. 2011b; 2011a]. For both methods, we only need sufficient overlap among images. A good practice is to have the same feature being visible in, at least, three or four photos.

## 5. STRUCTURE FROM MOTION RECONSTRUCTION

The first step of our pipeline is the self-calibration of the image collection, independently from the dense 3D geometry. This task is performed using a robust SfM algorithm suitable to align unordered large image collections [Snavely et al. 2006]. For each image, the method computes several thousand image features [Lowe 2004] and, then, it matches the features from different images by using approximate nearest neighbors [Arya et al. 1998] and RANSAC [Fischler and Bolles 1981]. Then, a SfM algorithm recovers camera poses and sparse geometry by minimizing a non-linear energy function proportional to the re-projection error of 3D points into original image features. Given $N_C$ photos, the output is a list of $N_C$ estimations of intrinsic (i.e., focal length and distortion coefficients) and extrinsic (i.e., rotation and translation) camera parameters $C = [c_1, c_2, ..., c_{N_C}]$, a list of $N$ triangulated 3D points $\mathbf{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_N\}$, and pixel coordinates $s_{i,j}$ of the projection of a sparse point $\mathbf{q}_j$ in the $i_{th}$ input image (i.e., key point location for that 3D point).

## 6. COARSE ALIGNMENT

After the SfM stage, we have two geometric representations of the scene with different scales, reference frames and resolutions: the dense model point set $\mathbf{P} = \{\mathbf{p}_1, ..., \mathbf{p}_M\}$, provided as input, and the sparse point set $\mathbf{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_N\}$, deriving from SfM, which contains the triangulated image features. In order to position cameras into the reference frame of the detailed input model, we need to find the affine transformation $T$ that determines the scale, rotation and translation, which best aligns $\mathbf{Q}$ with $\mathbf{P}$. Applying the same transformation $T$ to the cameras will then allow us to project acquired colors to the original model.

Here, we present two possible solutions: a practical fast semi-automatic procedure with little human intervention, useful in the conventional 3D acquisition pipeline, and a novel completely automatic registration method, useful, for instance, for the creation of remote 2D/3D registration services.

### 6.1 Semi-automatic coarse alignment

The user has to align one (or a few) image to the fine model by graphically selecting few matches (i.e., typically from 7 to 12) between 3D points in the fine model and image pixels. Using the intrinsic parameters computed by SfM, we can estimate the pose of the selected camera in the reference frame of the fine model by minimizing re-projection error, i.e. the sum of squared distances between the picked image points and the projection of the selected object points. Optionally, the process can be repeated independently for two or three images, chosen so that the mutual distances between camera pairs are as large as possible, minimizing error drift. It should be noted that this procedure assumes that the SfM pipeline is capable to produce a model which is approximately correct and does not contain major geometric errors, especially systematic ones. If this is not the case, e.g., in the presence of large drifts possibly generated by sequential SfM approaches, coarse alignment may fail. In our experience, such failure case occurs very rarely in practice. Moreover, drift-related problems can be mitigated by using more robust multi-stage SfM pipelines [Gherardi et al. 2010; Sinha et al. 2012] or by manually splitting the input image dataset, applying our technique to each obtained subset, and merging the results before refinement.

Using the intrinsic and extrinsic parameters estimated for that small set of cameras, we build a set of correspondences between points in the dense and sparse SfM models. For each feature in the chosen image subset, which already corresponds with a point in the sparse 3D from SfM, we cast a ray to find the corresponding point in the detailed model. At the end of this process, we obtain two sparse point clouds that are subsets of the two 3D geometries with known correspondences. We then find the global scale factor and a rigid alignment of these point-sets (i.e., rotation and translation) by applying a well-known absolute orientation algorithm [Horn 1987]. We implemented it in a robust weighted RANSAC-based framework to remove possible outliers due to the non-complete overlapping among datasets [Chum et al. 2003]. Each point-to-point match in the absolute orientation algorithm is weighted proportionally to the inverse of the pre-computed local density of the fine 3D model. This affine transformation is then applied to the SfM model to approximately register the sparse geometry and all the cameras in the same reference frame of the dense model.

## 6.2 Automatic coarse alignment

Aligning the coarse point cloud $\mathbf{Q}$ with $\mathbf{P}$ requires the solution of a global optimization problem to determine the optimal affine transformation $T$ of the form $T(\mathbf{q}) = sR\mathbf{q} + \mathbf{t}$ for a rotation matrix $R$ and a translation vector $\mathbf{t}$. By parameterizing the transformation with a vector $\mathbf{x}$, and defining a correspondence function $C(\mathbf{q})$ which selects, for each point in $\mathbf{Q}$, the closest (corresponding) point in $\mathbf{P}$, the optimal registration is given by minimizing over $\mathbf{x}$:

$$\mathbf{x} = \underset{\mathbf{x}}{\mathrm{argmin}} E(\mathbf{x}) := \sum_{i}^{N} \epsilon(\|T_{\mathbf{x}}(\mathbf{q}_i) - C(T_{\mathbf{x}}(\mathbf{q}_i))\|) \tag{1}$$

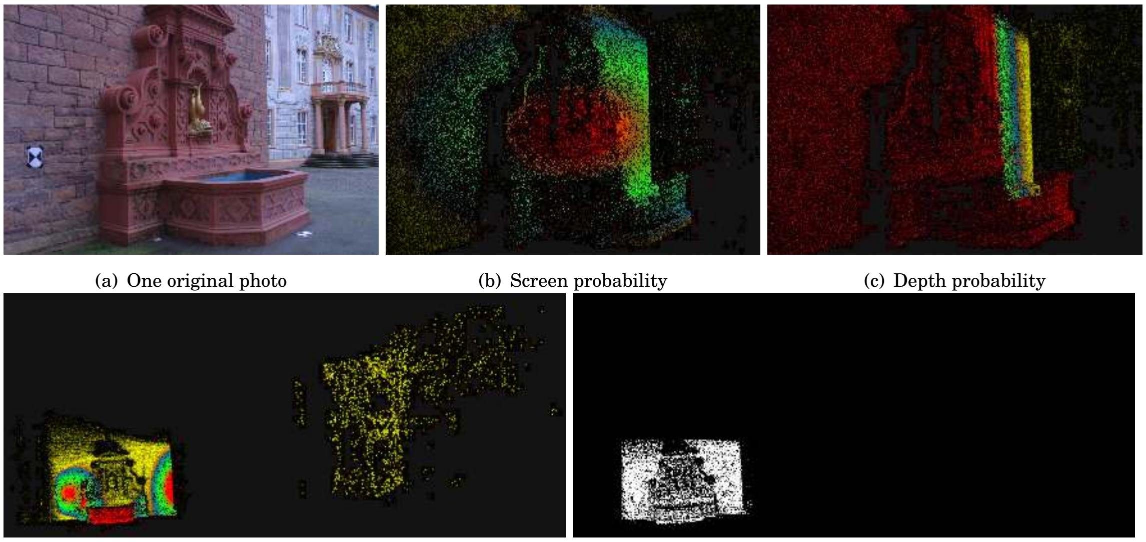
where $\epsilon(d)$ is a robust kernel for ***M-estimation***, i.e., a fitting criterion that is not as vulnerable as least squares to unusual data. For this paper, we use the Huber kernel [Huber and Ronchetti 2009],

$$\epsilon(d) = \begin{cases} \frac{d^2}{2} & d \leq k \\ kd - \frac{k^2}{2} & d > k \end{cases} \tag{2}$$

where $k$ is a tuning constant. We set $k = 1.345\sigma$, where $\sigma$ is the estimated standard deviation of the alignment errors, in order to produce 95% efficiency when the alignment errors are normal, and still offer protection against outliers when no information about outliers is available [Huber and Ronchetti 2009]. We find the solution of Eq. 1 through a problem-aware stochastic search of the parameter space. The value of $\sigma$ is set by default to 20 times the average sample spacing of the dense model.

6.2.1 *Parametrization.* Our parameter vector $x$ has 7 parameters: 1 scalar for the scaling factor $s$, 3 for the translation vector, $t$, and 3 for a the three angles $(\phi, \psi, \theta) \in [0, 2\pi) \times [0, \pi] \times [0, \pi]$ necessary to define a redundant-free rotation space parametrization based on the axis-angle representation of $SO(3)$ (i.e., $\phi$ and $\psi$ for the spherical coordinate representation of the rotation axis, and $\theta$ for the rotation amount around this axis). This rotation parametrization has already been used in other stochastic minimizers in the rotation space [Papazov and Burschka 2011], as it leads to simple techniques for achieving uniform sampling and equal volume bisection of the parameter space. For further details about space parametrization, search space structure and advantages in using this representation, see Section 4.1 in Papazov and Burschka [2011] for a use related to the proposed method, and see Kanatani's book [Kanatani 1990] for a generic treatment of the topic.

6.2.2 *Parameter range estimation.* Given the point clouds $\mathbf{P}$ and $\mathbf{Q}$, it is trivial to determine the bounds for translation and rotation parameters that ensure that at least a minimal overlap between the point clouds exists. Setting bounds for scaling, however, requires some knowledge, e.g., to avoid



(a) One original photo

(b) Screen probability

(c) Depth probability

(d) Overall probability

(e) Pruned point cloud

Fig. 2. **Outlier removal for scaling estimation.** (a) one original photo from the input image set; (b) and (c) are respectively the screen and depth probabilities related to the single image (a); (d) is the per-point overall probability, obtained by accumulating the probabilities from all the input images; (e) the resulting point cloud after the pruning based on the computed probability. We use a color map where high values are depicted in red while yellow values represent low probabilities. Fountain dataset courtesy of Strecha et al. [2008; 2011].

the trivial solution of a null scaling factor and to restrict the search range. We cannot assume that the two point clouds have perfect overlap, since the sparse point cloud $\mathbf{Q}$ computed by the SfM algorithm will certainly contain many points belonging to the object, but also a possibly large number of outliers from the surroundings. These points lead to potential problems in the estimation of the feasible scale range for the stochastic global optimization routine. Thus, in order to automatically determine a good scaling range, we use an heuristic to extract from the coarse point cloud a subset of points belonging to the object of interest. We assume that during the photo capture process, the acquired images will typically have the object of interest as the main photographic subject in the foreground, and the whole environment surrounding it as background. So, for each camera image, we estimate a per point foreground probability value, which represents the probability that a point in $\mathbf{q}_i \in \mathbf{Q}$ appearing in the image belongs to the foreground object. This probability is the mixture of depth-based and screen-based Gaussians. The depth-based value is determined with a Gaussian function centered at z eye coordinate of the nearest 3D sparse point, with a variance proportional to the median of all z eye coordinates for that camera. The screen-space weight is computed instead with a Gaussian function with the peak at the image center, and the variance equal to a quarter of the image width. We then loop for all images, accumulating the probabilities (see Fig. 2). Finally, we sort the points in $\mathbf{Q}$ based on their probabilities and keep the top $50\%$ of them for scale estimation. Once we have this subset, with high probability clean of background objects, we assume that the reduced cloud $\tilde{\mathbf{Q}}$ cannot be much larger/smaller than the model point cloud $\mathbf{P}$. In our implementation, we set as default limits for scaling 0.1-10 times the ratio between the two bounding spheres. Since this approach is an heuristic, it is possible to find situations where this automatic scale initialization fails. A wrong determination of the scaling range would lead to a longer computational time (if range is valid but too big), or, worse, an unsuccessful coarse alignment (if too restrictive). The default limits are very conservative and proved effective for all the test cases used in this paper. It should be noted that it is also easy to override these values in an application by asking the user to coarsely initialize the relative scale.

6.2.3 *Stochastic optimization process.* The optimal affine transform $A$ is found with a method that effectively combines a stochastic exploration of the multidimensional parameter space with a local improvement scheme. Similarly to the purely stochastic method of Papazov and Burschka [2011], we use a generalized BSP to represent our 7-dimensional search space, and exploit it to adaptively add more detail to promising regions in the search space. Each tree node contains the number of times it has been visited, the parameter bounds and the best parameter value and cost function value evaluated in the subtree. At the end of the procedure, the parameter values associated to the root is taken as solution. In contrast to previous work, however, instead of simply constructing the tree by guided random sampling, we exploit the problem structure by improving the solution by solving absolute orientation problems at each new search point and pushing the values to the structure. More specifically, the overall procedure proceeds as follows:

(1) Initialize the root of the tree with the entire search space as bounds. Generate a uniformly sampled parameter value as position, compute the correspondences, evaluate the cost function value, and store its value as current best. Define the maximum number of iterations $T_{max}$.
(2) Select a "promising" leaf according to a probabilistic scheme driven by a cooling schedule. The leaf is identified by taking left/right decision at each node. On iteration $t = 0$, left/right branches are selected with equal probability, while when $t = T_{max}$, the branch corresponding with the lowest cost function value is selected (see Papazov and Burschka [2011] for details).
(3) Bisect the selected leaf and create new children, reassigning the old values to the child that contains the old sampling location; in contrast to previous work, we split nodes at longest edge rather than randomly choosing split planes. The longest edge is identified as the parameter that causes, within the bounds defined by the node, the largest motion of the coarse point cloud.
(4) Generate a new random sampling point x in the leaf that does not contain the old sampled value. Find correspondences $C$ at x between point clouds using a GPU-accelerated method [Cayton 2010]. Evaluate the cost function value $e_x = E(\mathrm{x})$, and store its value as current best in the new node interval. Propagate bottom up the new parameter and function values so that each internal node contains the best parameter location and function value in both children.
(5) Exploit the computed correspondences by finding a new parameter location x and function value $e_x = E(\mathrm{x})$ through the solution of an absolute orientation problem.
(6) Locate the node containing the newly created position through a top-down visit. If x can be separated by bisecting the selected node into equal parts, create children and insert it, otherwise replace the currently present state if the value is worse than $e_x$. Propagate bottom up the new parameter and function values so that each internal node contains the best parameter location and function value in both children.
(7) If the stopping criterion is not met, increment iteration count $t$ and go to step 2, otherwise terminate the algorithm. In this paper, we use as stopping criterion $t = T_{max}$, or the alignment within tolerance of at least 95% of the pruned point cloud (Sec. 6.2.2).

If a successful alignment is not found within the current iteration budget, we re-execute the alignment process by doubling the budget, until a maximum amount of time is exceeded.

6.2.4 *Local refinement through iteratively reweighted least squares.* Our procedure, similarly to other alignment procedures, alternates between finding correspondences given a parameter value, evaluating the error using those correspondences, and using the result to move to other more promising parameter values. Since the most costly solution is finding correspondences, we accelerate it through an approximate GPU-accelerated method [Cayton 2010], and exploit them not just for error evaluation but also for error minimization. Given the correspondences, we obtain the minimum of $E$ using a iteratively reweighted least-squares (IRLS) [Holland and Welsch 1977] modification of Horn's method of absolute orientation with Euclidean distances [Horn 1987]. This is achieved by calculating weights $w_i = \epsilon'(r_i)/r_i$, where $r_i$ is the current residual and $\epsilon'(d)$ is the derivative of $\epsilon(d)$ (see eq.2), by solving the weighted least squares problem using the analytical absolute orientation method of Horn, and by re-solving again until convergence. Since this process is included as a refinement step in a higher level global refinement method, we just iterate for a fixed number of times (four for the results in this paper).

## 7. FINE ALIGNMENT

After the coarse alignment stage, we have a good initial configuration of the camera poses and their intrinsic parameters, obtained by applying the transformation $T$ found by our global registration method to the cameras attached to the coarse model. However, these stages do not fully exploit the amount of accurate geometrical data present in the dense model. In fact, SfM reconstruction is completely independent from the fine 3D and it produces a list of camera parameters and 3D points consistent only in the domain of images. To improve the current registration, we should link this result with the dense geometry, putting some constraints on the sparse 3D points; more precisely, it is desirable that the SfM geometry fits as much as possible the fine model. To obtain a fully consistent model, we should formulate our fine alignment in a non-rigid manner, jointly moving the sparse points towards the dense 3D and accordingly tuning the parameters of each camera independently. We thus need to refine a camera model consisting in intrinsic parameters (i.e., focal length, principal point and the first two radial distortion coefficients) and extrinsic parameters (i.e., the rotation-translation map). Similarly to our previous work [Pintus et al. 2011c], we obtain this fine registration with an optimization process. Our approach improves over previous work by employing a more robust optimization method that also works with variable sampling rates, without the need for user-prescribed global tolerances.

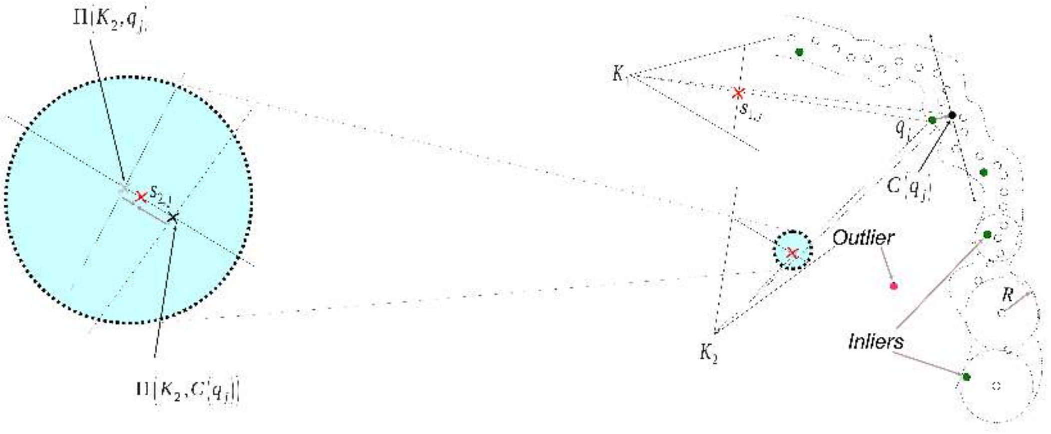


Fig. 3. **Fine registration.** A coarse registration between the original (white dots) and the SfM geometry (gray dots) is given. The fine registration jointly tunes camera parameters and sparse point positions $\mathbf{q}_j$ to make the SfM geometry fit as much as possible the fine model; it minimizes the error between image key points $\mathbf{s}_k$ and the re-projections of correspondences $C(\mathbf{q})$ in the dense model. An outlier removal strategy is employed, based on the local density of the reference 3D model.

Since the method starts with an already coarse aligned sparse point cloud, before applying fine minimization we prune the sparse point cloud by removing clear outliers. The pruning process is based on how much they are close to the dense model, i.e., we find a locally adaptive "closeness" threshold that defines a volume containing inliers. This value is computed locally, and is conservatively proportional to a small multiple of the inverse of the fine geometry local density. We show the limits of the inlier region in Fig. 3 as dotted lines.

In order to find the optimal camera parameters, we perform this task only on inlier points, and we minimize a cost function that aims at jointly reducing the difference between detected image features point and the 2D projection of the corresponding sparse point (as in classic bundle adjustment), and the distance between the 3D sparse point and the given 3D model surface. However, the 3D distance depends on object coordinates and it is not directly comparable with an image domain term. Since a scalar that weights these two errors is hard to globally estimate, we convert, as shown in Fig. 3, the 3D distance measure to an image-space distance through a projection process. Thus, for each sparse 3D point $\mathbf{q}_j$ (green dots in Fig. 3), we compute correspondence $C(\mathbf{q}_j)$ (black dots) in the dense model $\mathbf{P}$ (white dots) and we find optimal camera parameters $K$ and 3D points $\mathbf{Q}$, by minimizing the following cost function:

$$E(K, \mathbf{Q}) = \sum_{j=1}^{N_q} \sum_{i=1}^{N_C} v_{ij} w\left(\mathbf{q}_j\right) \left( \left\| \Pi\left(K_i, \mathbf{q}_j\right) - \mathbf{s}_{i,j} \right\|^2 + \left\| \Pi\left(K_i, C\left(\mathbf{q}_j\right)\right) - \mathbf{s}_{i,j} \right\|^2 \right) \tag{3}$$

where the term $v_{ij}$ is a visibility factor, that is equal to $1$ if the point $\mathbf{q}_j$ is visible in the image $i$, otherwise is $0$, $w\left(\mathbf{q}_j\right)$ is a per-point weight, $\mathrm{s}_{i,j}$ is the key point image coordinate (see Sec. 5) and $\Pi\left(K, \mathbf{q}\right)$ is a function that projects a 3D point $\mathbf{q}$ into an image given the camera parameters $K$. The cost function has two error terms. The first term, which was not included in our previous work [Pintus et al. 2011c], aims at reducing the squared difference in pixel coordinates between the image feature point and the 2D projection of the corresponding sparse point. The second term takes into account the distance between the sparse and dense point clouds by converting the object space distance in an image space measurement. For each sparse point $\mathbf{q}$, we compute the correspondence on the original model ($C(\mathbf{q})$), and estimating a re-projection error in the image domain. This error strictly depends on the fine geometry, forcing the configuration of the cameras to be consistent with it. As for coarse registration, correspondences are computed by finding a fixed number k of nearest neighbors in the dense cloud (k=8 in this paper), and projecting the point $\mathbf{q}$ on a plane fit by PCA. Compared with the cost function in the previous approach [Pintus et al. 2011c], here the formulation is more robust, since it considers the errors between the image features and both the triangulated points and the points in the original dense model. The big circle in Fig. 3 highlights how, by moving the point $\mathbf{q}_j$ and/or the camera $K_2$, we modify the distance between $\Pi\left(K_2, \mathbf{q}_j\right)$ (and $\Pi\left(K_2, C\left(\mathbf{p}_j\right)\right)$)) and $\mathrm{s}_{2,j}$, as shown by the arrows. This is done jointly and in a non-rigid manner on all sparse points and all cameras. The refinement is formulated as a weighted non-linear least squares minimization problem on the 3D structure and viewing parameters. The technique presented in Pintus et al. [2011c] made the assumption of a uniform distribution of vertices in the dense model. Here, conversely, in order to deal with non-uniformly sampled geometries, we use the inverse of the local density estimation as a per-point weight $w\left(\mathbf{q}_j\right)$ for the errors. Thus, we assign a high uncertainty to points aligned to low density regions of the reference geometry. Since the coarse alignment produces a good initial estimate of the camera parameters, we can perform a local minimization to find the final solution using the Levenberg-Marquardt algorithm [Nocedal and Wright 2006], which, thanks to its effective damping strategy, converges quickly from a wide range of initial guesses.

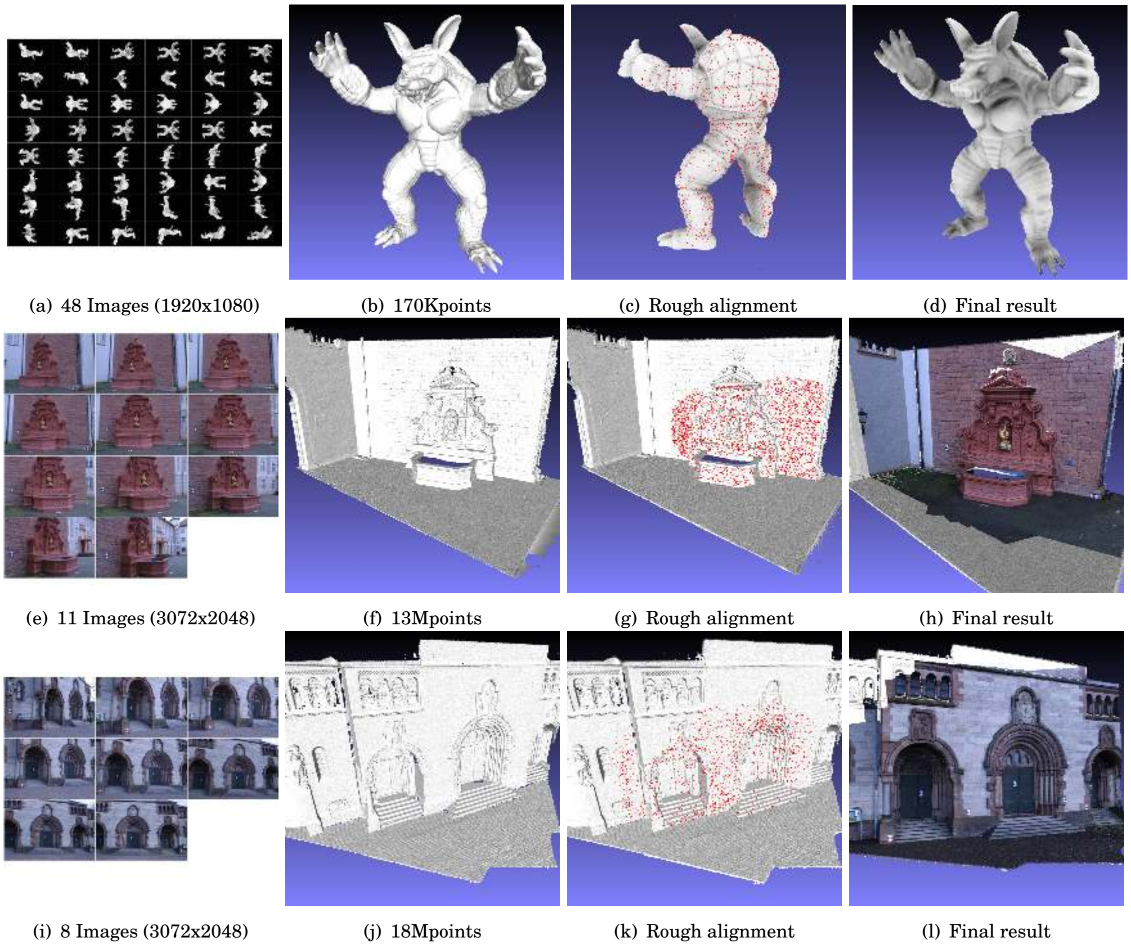
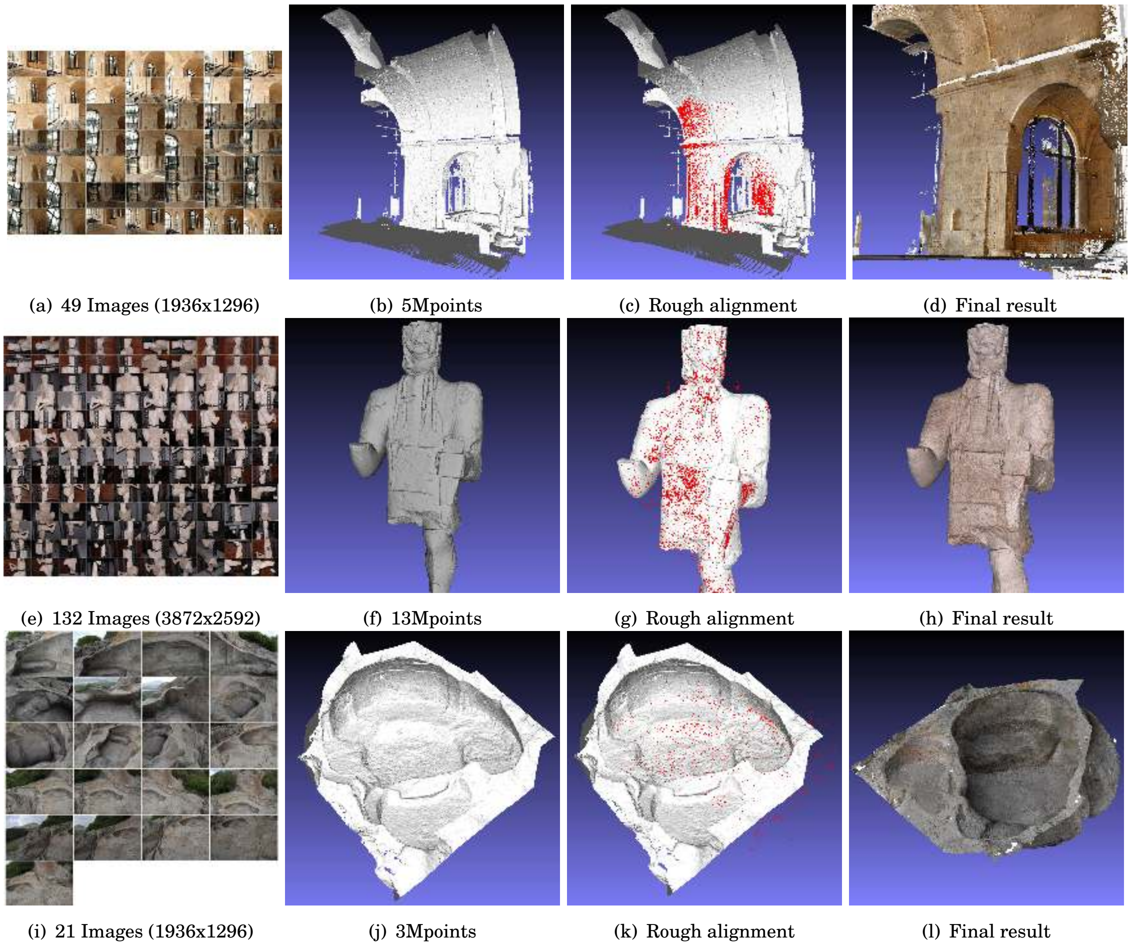

(a) 48 Images (1920x1080)
(b) 170Kpoints
(c) Rough alignment
(d) Final result
(e) 11 Images (3072x2048)
(f) 13Mpoints
(g) Rough alignment
(h) Final result
(i) 8 Images (3072x2048)
(j) 18Mpoints
(k) Rough alignment
(l) Final result

Fig. 4. **Experimental ground truth datasets:** In each row we present the details of a ground truth 2D/3D registration dataset used for testing, respectively (top to bottom): *Armadillo* (courtesy of the Stanford 3D Scanning Repository [Levoy et al. 2005]), *Fountain-P11* (courtesy of Strecha et al. [2008; 2011]), and *Herz-Jesu-P8* (courtesy of Strecha et al. [2008; 2011]). For each model we show (left to right) the input image set, the original model, the rough alignment between the original model and the sparse point set from SfM, and the final colored result.

## 8. RESULTS

Our technique was implemented on Linux using C++. The SfM software used for our tests is *Bundler* [Snavely et al. 2006; Snavely 2013]. For the minimization problem in the refinement step, we employ a C/C++ package for generic SBA based on the Levenberg-Marquardt algorithm, developed by Lourakis and Argyros [2009]. Approximate nearest neighbors are computed with the GPU-accelerated RBC library [Cayton 2010]. The user interface to manually calibrate input photos is built using OpenGL and Qt tools. Our benchmarks were executed on a PC with 12 Intel Core i7-3930K 3.2 CPU Processor, 32GB RAM,and a NVidia GeForce GTX 560. We use this method in our production pipe-line, that has been applied to many acquisition campaigns. All the results presented in the following sub-sections are obtained using the new proposed automatic stochastic approach for coarse alignment.



(a) 49 Images (1936x1296) (b) 5Mpoints (c) Rough alignment (d) Final result

(e) 132 Images (3872x2592) (f) 13Mpoints (g) Rough alignment (h) Final result

(i) 21 Images (1936x1296) (j) 3Mpoints (k) Rough alignment (l) Final result

Fig. 5. **Experimental CH datasets:** In each row we present the details of a CH 2D/3D registration dataset used for testing, respectively (top to bottom): *Church*, *Archer* and *Grave*. For each model we show (left to right) the input image set, the original model, the rough alignment between the original model and the sparse point set from SfM, and the final colored result.

## 8.1 Evaluation Datasets

Here we present some experimental results on selected datasets (Fig. 4 and 5).

**Armadillo.** This synthetic dataset contains a 170K point geometry from Stanford 3D Scanning Repository [Levoy et al. 2005] and 48 perfectly aligned Full HD images computed using a rendering pipeline with known camera parameters, ambient occlusion lighting and dark background.

**Fountain-P11.** This model is a ground truth test from the datasets in the multi-view evaluation repository by Strecha et al. [2008; 2011]. Fountain-P11 data is provided with all the detailed information about extrinsic and intrinsic ground truth camera parameters for each photo. The image dataset contains eleven 6Mpixel images acquired with a Canon D60 digital camera. The geometry size is 13Mpoints with a resolution of 3mm, and it was captured with a Zoller-Frölich LIDAR laser scanner.

**Herz-Jesu-P8.** This is another ground truth model from the multi-view evaluation repository by Strecha et al. [2008; 2011]. Its 18M point geometry has a resolution of 4mm. The eight input photos, with a size of 3072x2048, and the dense geometry are acquired with the same devices as *Fountain-P11*. For each photo extrinsic and intrinsic ground truth camera parameters are included either.

**Church.** This CH dataset is a good test for our algorithm in terms of noise sensitivity and robustness, since it is a non-uniformly sampled model with a high percentage of outliers (see also Fig. 6). These points arise from the complexity of the input image set (e.g., reflections on mirrors and specular highlights), or belong to object parts not acquired with the time-of-flight scanner. It represents a part of the left nave in the Romanesque San Saturnino Basilica in Cagliari (Italy). It was acquired using a time-of-flight laser scanner Leica ScanStation2, and the 5Mpoints geometry, resulting from geometric reconstruction [Cuccuru et al. 2009], has a resolution of 2mm. The photographic dataset was acquired with a Nikon D200 camera and is composed by 49 2.5Mpixel images.

**Archer.** This is an example of sculpture acquisition. It is an item from the Mont'e Prama collection of 37 statues[Bettio et al. 2013; Marton et al. 2014]. The geometry was acquired with a Minolta Vivid9i Laser scanner at a quarter-millimeter resolution. The 132 images composing the dataset are captured using a Nikon D200 camera, and have a size of 3872x2592.

**Grave.** This CH dataset has been chosen because its geometry has a lot of smooth regions that don't contain well defined planes or straight lines. It is the digital model of an archaeological site (grave from a prehistoric necropolis) [Pintus et al. 2012]. The 2D/3D dataset is made up of 21 images (1936x1296), and a geometry of 3Mpoints with a 2mm resolution. It was acquired using a time-of-flight laser scanner Leica ScanStation2, and a Nikon D200 camera.

In each column of Fig. 4 and 5 we present respectively the input photo dataset, the rendering of the dense geometry, the alignment between the dense geometry and the pruned sparse point cloud (red points), and the registered images finally blended over the dense model by applying the photo blending framework of Pintus et al. [2011b; 2011a].

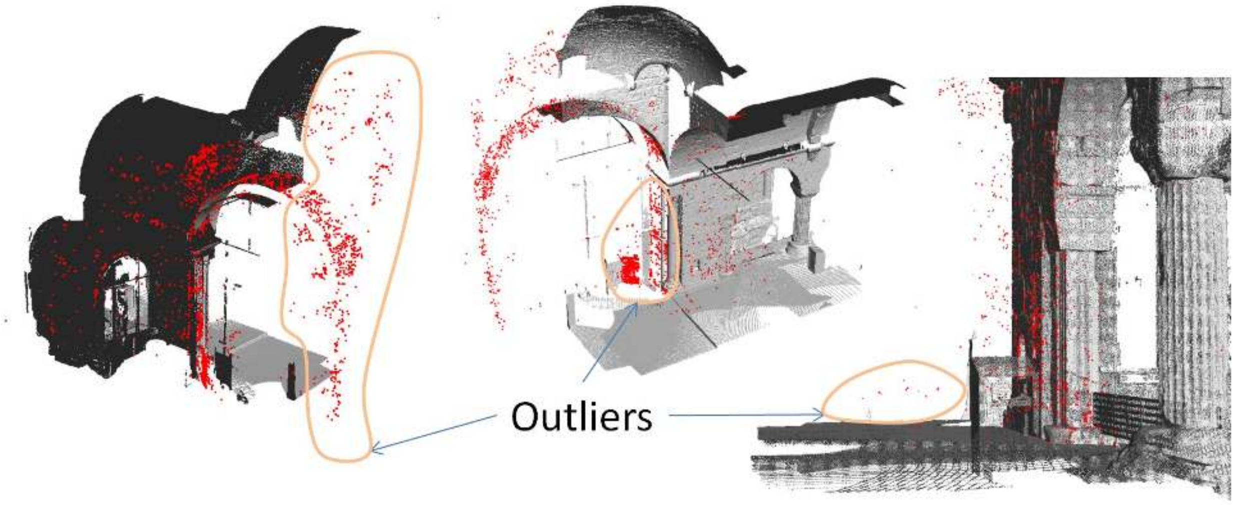


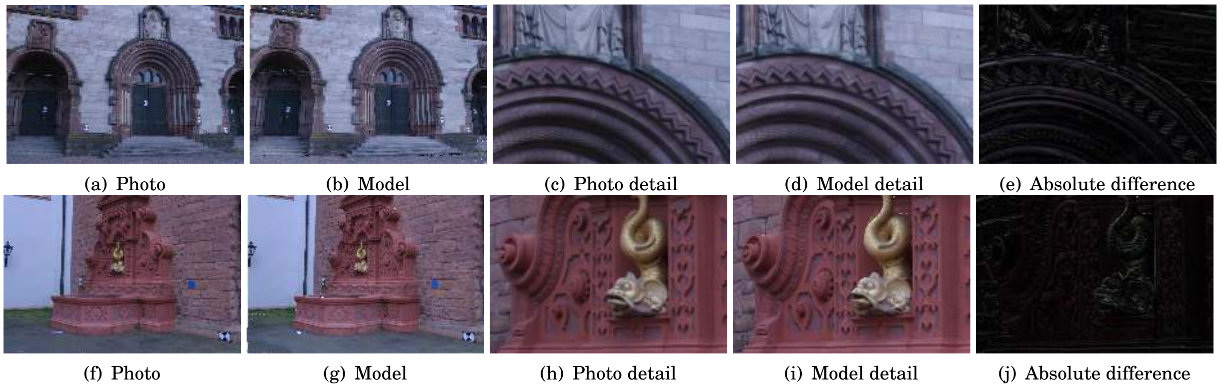
Fig. 6. **Outliers.** Registration of sparse (red) and dense (white) geometries of a *Church's Detail*. SfM geometry contains a lot of outliers. Our method proves to be robust in this non-ideal case, without requiring more user intervention. Here the sparse point cloud is shown not pruned from outliers.

Table I. Time statistics (all times in minutes).

| Dataset | #Images | #Points | SfM | Manual Coarse Reg. | Auto Coarse Reg. | Fine Reg. | Total Semi-Auto | Total Auto |
|---|---|---|---|---|---|---|---|---|
| Armadillo | 48 | 170K | 6 | 5 | 12 | 1 | 12 | 19 |
| Fountain-P11 | 11 | 13M | 11 | 4 | 121 | 8 | 24 | 140 |
| Herz-Jesu-P8 | 8 | 18M | 7 | 4 | 33 | 5 | 16 | 45 |
| Church | 49 | 5M | 59 | 8 | 80 | 17 | 84 | 156 |
| Archer | 13 | 13M | 360 | 6 | 29 | 27 | 393 | 416 |
| Grave | 21 | 3M | 9 | 5 | 119 | 1 | 15 | 129 |

## 8.2 Time

Table I shows the time necessary for semi-automatic or fully automatic alignment. The Structure-from-motion task takes from few minutes to hours, depending on the number and the size of input photos, and on matching complexity. In the semi-automatic approach, for all the presented datasets, the user manually aligned only one photo to the geometry. This operation took on average about five minutes, a time comparable to per-image manual alignment times presented in Franken et al. [2005]. The completely automatic method replaces manual alignment with stochastic global optimization, which has always succeeded for all the test dataset. Computational times are higher than semi-automatic alignment, but they are competitive with previous fully automatic alignment pipe-lines [Corsini et al. 2013]. For instance, the processing time to align the reconstructed point cloud to the input model for the Herz-Jesu dataset is 60 minutes for Corsini et al. and 33 minutes for our method. It should be noted that it is in general not possible to infer performance characteristics from timings obtained on different 3D models, because the computational effort varies not only with the number of vertices or images, but mostly on the complexity of surface shape. For more details about geometric and topological dependency, see [Corsini et al. 2013]. Our final refinement step typically converges in few minutes. The main advantage of the fully automatic pipeline is the possibility of using it in a completely unsupervised manner, e.g., as a web service for color mapping, similarly to what done for automated 3D reconstruction services [Vergauwen and Van Gool 2006].


(a) Photo (b) Model (c) Photo detail (d) Model detail (e) Absolute difference

(f) Photo (g) Model (h) Photo detail (i) Model detail (j) Absolute difference

Fig. 7. **Visual Quality Evaluation:** Comparison between original photos (a)(f) and the rendering of the colored model from the same view point (b)(g). The last columns show these results for a smaller region of the datasets, and their relative absolute differences. Original datasets courtesy of Strecha et al. [2008; 2011].

Table II. Position, orientation, and re-projection error.

| Dataset<br>Name | Average camera<br>distance (cm) | Position<br>error (cm)<br>[Corsini et al. 2013]/Our | Orientation<br>error (degree)<br>[Corsini et al. 2013]/Our | Re-projection<br>error (pixels)<br>[Corsini et al. 2013]/Our |
|---|---|---|---|---|
| Armadillo | 82 | N.A./1.35 | N.A./0.40 | N.A./3.77 |
| Fountain-P11 | 658 | 10.92/3.19 | 0.27/0.26 | 21.28/5.19 |
| Herz-Jesu-P8 | 765 | 15.40/11.33 | 0.56/0.09 | 4.91/5.57 |

### 8.3 Visual Quality Evaluation

The estimation of 2D/3D registration quality is not straightforward at all. To our knowledge there is no standard way to measure the alignment accuracy, so, to make a comparison between our method and the state-of-the-art works we adopt the two kinds of evaluation proposed in Corsini et al. [2013]: visual quality and quantitative evaluation.

In this section we are going to show some results that help us analyzing our algorithm in terms of visual similarity between original photos and the rendering of the colored 3D digital model. For this purpose, we projected the aligned images onto the geometry by using a robust photo blending framework by Pintus et al. [2011b; 2011a]. The color is assigned to the geometry with a per-vertex strategy; redundant information is weighted based on a per-pixel image quality estimation. Please see the original paper for a detailed description of the seamlessly blending algorithm.

The quality of our registration is evaluated in a visual manner by rendering the colored model from the same viewpoint of the original photo. In fig. 7 we compare the original photo (first column) of *Herz-Jesu-P8* and *Fountain-P11* with their rendered 3D colored models. The last columns show the high quality result of the proposed method in a restricted part of our datasets, and their relative absolute differences.

### 8.4 Quantitative Evaluation

As stated above, in order to quantitatively validate our contribution, we have tested the algorithm with ground truth datasets: *Armadillo*, *Fountain-P11* and *Herz-Jesu-P8*. We used the last two models to compare the results of our techniques with those in Corsini et al. [2013]. In order to have a strong consistency in the numerical comparison between the registration quality of our approach and that presented in Corsini et al. [2013], we used fine aligned datasets provided by Corsini, and all the results reported in Tables II and III are computed using the same algorithm.

In Table II we evaluate the accuracy of the camera parameters estimation. We compute the following errors: the median position error, i.e., Euclidean distance between the camera position and the ground truth position; the median orientation error, i.e., the angle between the optical axis of the ground truth cameras and the estimated one; the median re-projection error, i.e., the distance in pixel between the re-projected vertex computed using ground truth and estimated extrinsic and intrinsic camera parameters. While Corsini et al. [2013] use the average error, we evaluate our and their registrations by computing median values, which are more robust to outliers. In the *Armadillo* model, the position error is less than $2\%$ of the average distance between camera positions, and the re-projection error of about 4 pixels is less than the $0.4\%$ of the image height. Compared to Corsini et al. [2013], in the *Fountain-P11* dataset we decreased the re-projection error by more than 15 pixels, while the accuracies in the registration of *Herz-Jesu-P8* are comparable.

Another way to measure the quality of a 2D/3D registration is to compute the squared standard deviation of the re-projected color, assuming that for a diffuse environment under constant illumination the same point looks similar from all point of views. This kind of evaluation, also applied by Corsini et al. [2013] does not require a ground truth data. For each vertex we compute the variance of the pixel re-

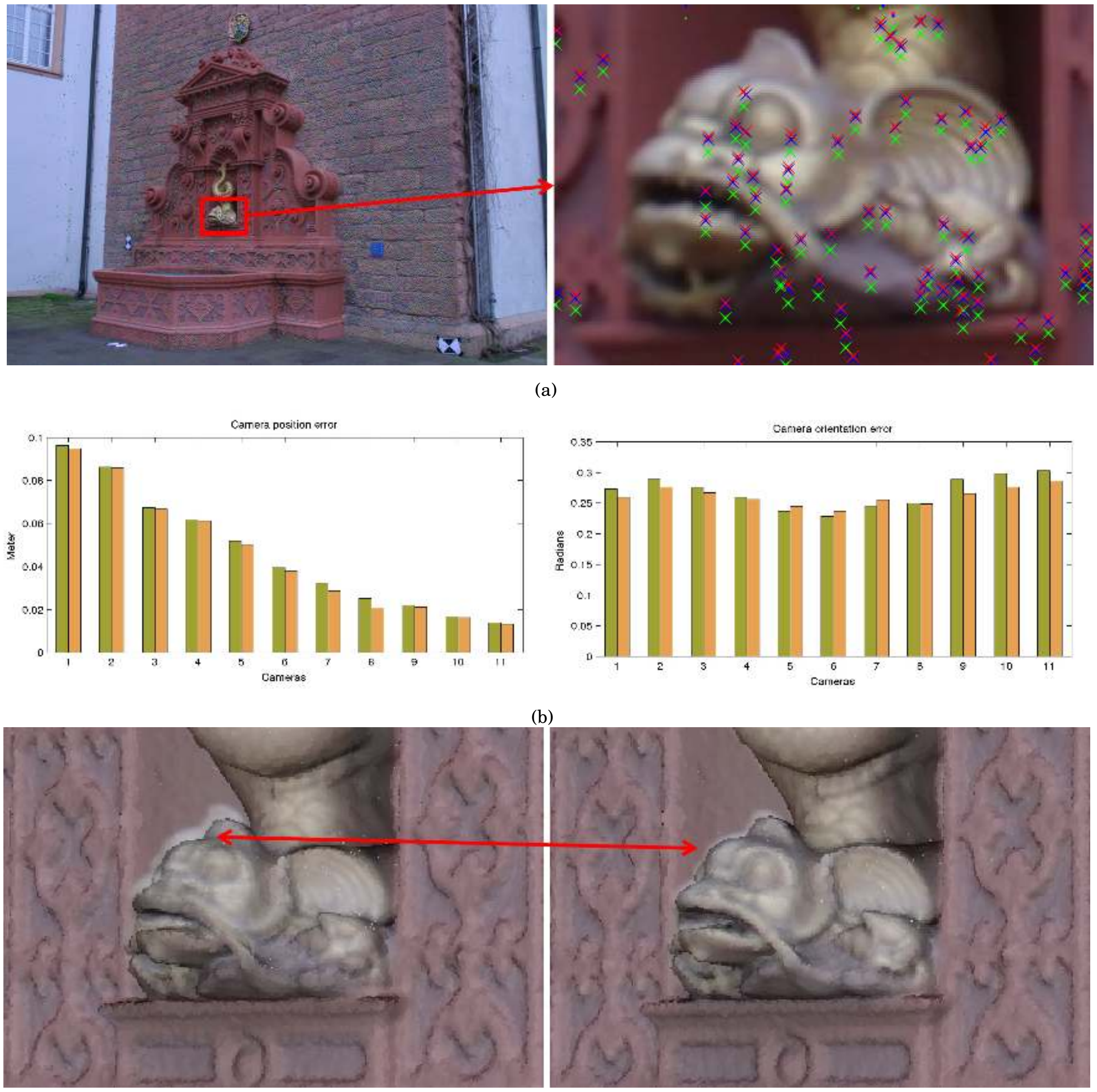
Table III. Quality of the 2D/3D registration.

| Dataset | Ground Truth | Registration | Registration |
|---|---|---|---|
| Name | Quality (QC) | Quality (QC) | Quality (QC) |
| | | [Corsini et al. 2013] | Our |
| Armadillo | 49.86 | N.A. | 629.36 |
| Fountain-P11 | (43.80, 40.25, 52.81) | (114.68, 104.02, 136.91) | (46.9927, 42.148, 55.2615) |
| Herz-Jesu-P8 | (101.25, 99.93, 128.87) | (102.40, 100.85, 129.59) | (107.895, 110.074, 140.9) |
| Church | N.A. | N.A. | (1277.29, 1153.07, 1000.57) |
| Archer | N.A. | N.A. | (468.31, 426.64, 373.13) |
| Grave | N.A. | N.A. | (213.25, 206.89, 203.78) |

flectance values that contribute to its final color across the image dataset, and we report the resulting median variance from all vertices. We perform this computation for each channel, resulting in a final 3-component vector of median squared standard deviations as a quality measure. As before, rather than computing average values as in Corsini et al. [2013], we evaluate the median values. Table III illustrates the quality measure for all datasets, and confirms the results in Table II. The re-projection quality of our algorithm is comparable to the one achieved by Corsini et al. [2013] in the *Herz-Jesu-P8* dataset, while is much better in the *Fountain-P11* colored model. The performances in the case of the *Armadillo* dataset and the other CH datasets in Table III are similar. In general, when a vertex belongs to a silhouette in a particular image, and the color gradient is high across the silhouette (e.g., when foreground and background are very different), even a single pixel mis-alignment will produce a high error. This is the case of the Armadillo, a bright object on a dark background (see Fig. 4a), which, despite an extremely high registration quality, has non negligible errors in color re-projection (see Table II). On the other hand, the registration quality of the Grave is higher than the others, because the color gradient in the image dataset is low, so small errors in the alignment do not produce high errors in color re-projection. Another important element that affect the re-projection error is the image sampling. The quality of the Church dataset is worse than the Archer because 49 low resolution images (2.5Mpixels) were used for a bigger object, while for the Archer (a 2 meter statue) 132 10Mpixel images are employed.

Finally, we measure how the proposed algorithm is able to converge to a good alignment while starting from an initial, coarse registration. In the first row of Fig. 8 we project onto one image the sparse points before (green crosses) and after (blue crosses) the refinement, and we mark as red crosses the image features corresponding to the sparse points. As it is easy to see in the zoomed image, the refinement algorithm reduces the total re-projection error, recovering the initial severe misalignment. Further comparisons are presented in the last two rows, where we plot the camera position and orientation errors before (green bins) and after (orange bins) fine alignment, and we show the contribution of fine alignment in the case of Camera #2. In this case, the camera position error is similar, but we decrease the orientation error by about one degree; the resulting visual improvement in terms of color re-projection is clearly visible by superimposing the original photo and the dense geometry (third row), both with coarse (left) and refined (right) registration.

Moreover we use ground truth information from the *Fountain-P11* dataset to compare the performances of the proposed algorithm and our previous approach [Pintus et al. 2011c]. In Fig. 9 we show for each camera its position error obtained by applying the method in Pintus et al. [Pintus et al. 2011c] (blue bins) and our technique (red bins). The figure shows how the more robust approach presented here reduces maximum errors, while making the average camera position error decrease by about 1.5cm with respect to our previous approach [Pintus et al. 2011c], and by over 7cm with respect to the method of Corsini et al. [2013] (see Table II).



(a)

(b)

(c)

Fig. 8. **Coarse vs Fine alignment:** (First row) Re-Projection of the sparse points before (green crosses) and after (blue crosses) refinement. Red crosses are image features corresponding to the sparse points. (Second row) Camera position and orientation errors before (green bins) and after (orange bins) fine alignment. (Third row) Comparison between coarse (left) and fine (right) registration of camera #2, obtained by superimposing the rendered geometry and the original photo. Dataset courtesy of Strecha et al. [2008; 2011].

## 9. CONCLUSIONS AND FUTURE WORK

We have presented an efficient, fast and robust technique for registering a set of images with a 3D geometry. Our approach minimizes or eliminates the user intervention and is generally applicable to different kinds of 3D models.

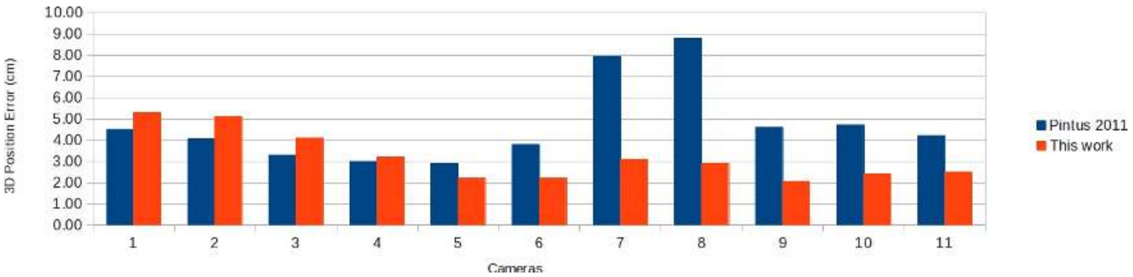


Fig. 9. **Camera position error - *Fountain-P11*:** For each of the eleven cameras we show the position error computed with the algorithm of Pintus et al. [2011c] (blue bins) and the proposed one (red bins). The average camera position error is decreased by over 1.5cm.

Compared to state-of-the-art stochastic optimization approaches [Papazov and Burschka 2011], and image-to-geometry alignment methods [Pintus et al. 2011c], our improved automatic coarse registration proved to be more robust and reliable in terms of noise sensitivity and presence of outliers. Further, compared to the state-of-the-art, good alignment results are presented in a variety of cases, including a series of real-world Cultural Heritage 2D/3D datasets. These input datasets range from synthetic data to real 3D geometries from different types of scanning technology, different level of noise and resolution, different size of 3D and 2D datasets, and different amount of outliers both in the images and in the 3D models (see Fig. 2 and Fig. 6). However, a quantitative evaluation of the algorithm robustness, as a study of the dependency function between the quality of coarse registration and the final alignment, is out of the scope of this paper, and it would be a very interesting topic and a possible direction for future developments and investigations. Finally it proved to be suitable to produce a good input data for photo blending framework.

Of course, the image resolution and the accuracy of the SfM also play an important role in this automatic process, by strongly affecting the reconstruction of the 3D sparse point cloud. With a higher 2D sampling of the scene and a more precise multi-view stereo approach, the better quality of the sparse geometry will result in a more likely speed-up of the stochastic optimization process, and an increased chance of its success. In the future, we are planning to investigate different SfM solutions to improve the robustness and quality in reconstructing the point cloud from images, e.g., by employing hierarchical SfM algorithms [Farenzena et al. 2009]. Further, a speed up in the multi-view stereo will allow us to use higher resolution images. These elements will produce a more accurate and detailed 3D reconstruction with less outlier points, and result in an improvement of the coarse and fine alignment. We will also study a way to relax the constraints on the estimation of the scaling range in the stochastic search space, e.g., by employing an error metric that will properly take into account and avoid extreme cases, as the trivial solution of a null scaling.

**Acknowledgments.** This work is partially supported by the EU FP7 Program under the DIVA project (290277) and by Soprintendenza per i Beni Archeologici per le Province di Cagliari ed Oristano. We also acknowledge the contribution of Sardinian Regional Authorities. We thank Massimiliano Corsini for providing us with their image-to-geometry registration datasets of *Fountain-P11* and *Herz-Jesu-P8*.

## REFERENCES

D. Aiger, N. J. Mitra, and D. Cohen-Or. 2008. 4-points Congruent Sets for Robust Surface Registration. *ACM Trans. Graph.* 27, 3 (2008), #85, 1–10.

Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. 1998. An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions. *J. ACM* 45, 6 (1998), 891–923.

Atsuhiko Banno and Katsushi Ikeuchi. 2010. Omnidirectional texturing based on robust 3D registration through Euclidean reconstruction from two spherical images. *Computer Vision and Image Understanding* 114, 4 (2010), 491–499.

Paul J Besl and Neil D McKay. 1992. Method for registration of 3-D shapes. In *Robotics-DL tentative*. International Society for Optics and Photonics, 586–606.

Fabio Bettio, Enrico Gobbetti, Emilio Merella, and Ruggero Pintus. 2013. Improving the digitization of shape and color of 3D artworks in a cluttered environment. In *Proc. Digital Heritage*. 23–30.

G.L. Bilbro and W.E. Snyder. 1991. Optimization of functions with many minima. *IEEE Trans. Sys., Man and Cybernetics* 21, 4 (1991), 840–849.

Louis Borgeat, Guillaume Poirier, J.-Angelo Beraldin, Guy Godin, Philippe Massicotte, and Michel Picard. 2009. A framework for the registration of color images with 3D models. In *ICIP*. 69–72.

Thomas Breuel. 2003. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding* 90, 3 (2003), 258 – 294.

L. Brunie, S. Lavallée, and R. Szeliski. 1992. Using force fields derived from 3D distance maps for inferring the attitude of a 3D rigid object. In *Proc. ECCV*. 670–675.

L. Cayton. 2010. A nearest neighbor data structure for graphics hardware. In *Proc. Accelerating Data Management Systems Using Modern Processor and Storage Architectures*. 1–10.

Yang Chen and Gérard Medioni. 1992. Object modelling by registration of multiple range images. *Image and vision computing* 10, 3 (1992), 145–155.

Ondrej Chum, Jiri Matas, and Josef Kittler. 2003. Locally Optimized RANSAC. In *DAGM-Symposium*. 236–243.

I. Cleju and D. Saupe. 2007. Stochastic optimization of multiple texture registration using mutual information. *Pattern Recognition* (2007), 517–526.

David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. 2004. Variational shape approximation. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 905–914.

Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, Riccardo Gherardi, Andrea Fusiello, and Roberto Scopigno. 2013. Fully automatic registration of image sets on approximate geometry. *International journal of computer vision* 102, 1-3 (2013), 91–111.

Gianmauro Cuccuru, Enrico Gobbetti, Fabio Marton, Renato Pajarola, and Ruggero Pintus. 2009. Fast low-memory streaming MLS reconstruction of point-sampled surfaces. In *Graphics Interface*. 15–22.

Matteo Dellepiane, Marco Callieri, Federico Ponchio, and Roberto Scopigno. 2008. Mapping highly detailed color information on extremely dense 3D models: the case of Davids restoration. *Computer Graphics Forum* 27, 8 (2008), 2178–2187.

Michela Farenzena, Andrea Fusiello, and Riccardo Gherardi. 2009. Structure-and-motion pipeline on a hierarchical cluster tree. In *Proc. ICCV Workshops*. IEEE, 1489–1496.

Martin A. Fischler and Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (1981), 381–395.

Thomas Franken, Matteo Dellepiane, Fabio Ganovelli, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. 2005. Minimizing user intervention in registering 2D images to 3D models. *The Visual Computer* 21, 8-10 (2005), 619–628.

Natasha Gelfand, Niloy J. Mitra, Leonidas J. Guibas, and Helmut Pottmann. 2005. Robust Global Registration. In *Proc. SGP*. 197–206.

George and Stockman. 1987. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing* 40, 3 (1987), 361 – 387.

Riccardo Gherardi, Michela Farenzena, and Andrea Fusiello. 2010. Improving the efficiency of hierarchical structure-and-motion. In *Proc. CVPR*. IEEE, 1594–1600.

Sébastien Granger and Xavier Pennec. 2002. Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. In *Proc. ECCV*. 418–432.

Chad Hantak and Anselmo Lastra. 2006. Metrics and Optimization Techniques for Registration of Color to Laser Range Scans. In *3DPVT*. 551–558.

Yaron C Hecker and Ruud M Bolle. 1994. On geometric hashing and the generalized Hough transform. *IEEE Trans. Sys., Man and Cybernetics* 24, 9 (1994), 1328–1338.

Paul W Holland and Roy E Welsch. 1977. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory and Methods* 6, 9 (1977), 813–827.

Berthold KP Horn. 1987. Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4, 4 (1987), 629–642.

PJ Huber and EM Ronchetti. 2009. *Robust Statistics*. Wiley.

Katsushi Ikeuchi, Takeshi Oishi, Jun Takamatsu, Ryusuke Sagawa, Atsushi Nakazawa, Ryo Kurazume, Ko Nishino, Mawo Kamakura, and Yasuhide Okamoto. 2007. The Great Buddha Project: Digitally Archiving, Restoring, and Analyzing Cultural Heritage Objects. *IJCV* 75, 1 (2007), 189–208.

Bing Jian and Baba C. Vemuri. 2005. A Robust Algorithm for Point Set Registration Using Mixture of Gaussians. In *Proc. ICCV*. 1246–1251.

Andrew Edie Johnson and Martial Hebert. 1999. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 5 (1999), 433–449.

Ryan S Kaminsky, Noah Snavely, Steven M Seitz, and Richard Szeliski. 2009. Alignment of 3D point clouds to overhead images. In *Proc. CVPR Workshops*. IEEE, 63–70.

Kenichi Kanatani. 1990. *Group-theoretical methods in image understanding*. Vol. 2. Springer-Verlag Berlin.

H.P.A. Lensch, W. Heidrich, and H.P. Seidel. 2000. Automated texture registration and stitching for real world models. In *Proc. Pacific Graphics*. IEEE, 317–452.

M. Levoy, J. Gerth, B. Curless, and K. Pull. 2005. The Stanford 3D scanning repository. http://www-graphics.stanford.edu/data/3dscanrep. (2005).

Hongdong Li and R. Hartley. 2007. The 3D-3D Registration Problem Revisited. In *Proc. ICCV*. 1 –8.

Yunzhen Li and Kok-Lim Low. 2009. Automatic registration of color images to 3D geometry. In *CGI*. 21–28.

Lingyun Liu and Ioannis Stamos. 2005. Automatic 3D to 2D registration for the photorealistic rendering of urban scenes. In *Proc. CVPR*, Vol. 2. IEEE, 137–143.

Lingyun Liu, Ioannis Stamos, Gene Yu, George Wolberg, and Siavash Zokai. 2006. Multiview geometry for texture mapping 2D images onto 3D range data. In *Proc. CVPR*, Vol. 2. IEEE, 2293–2300.

M.I. A. Lourakis and A.A. Argyros. 2009. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software* 36, 1 (2009), 1–30.

David G. Lowe. 1991. Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 5 (1991), 441–450.

David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV* 60, 2 (2004), 91–110.

Fabio Marton, Marcos Balsa Rodriguez, Fabio Bettio, Marco Agus, Alberto Jaspe Villanueva, and Enrico Gobbetti. 2014. IsoCam: Interactive Visual Exploration of Massive Cultural Heritage Models on Large Projection Setups. *ACM Journal on Computing and Cultural Heritage* (2014). To appear.

Andrew Mastin, Jeremy Kepner, and J Fisher. 2009. Automatic registration of LIDAR and optical images of urban scenes. In *Proc. CVPR*. IEEE, 2639–2646.

Minolta/Range7. 2013. http://www.konicaminolta.eu/en/measuring-instruments/products/3d-measurement/range-7/introduction.html. (2013).

P.J. Neugebauer and K. Klein. 1999. Texturing 3d models of real world objects from multiple unregistered photographic views. In *Computer Graphics Forum*, Vol. 18. 245–256.

Jorge Nocedal and Stephen Wright. 2006. *Numerical Optimization* (2nd ed.). Springer.

Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. 2009. Branch-and-Bound Methods for Euclidean Registration Problems. *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (May 2009), 783–794. Issue 5.

Chavdar Papazov and Darius Burschka. 2011. Stochastic global optimization for robust point set registration. *Comput. Vis. Image Underst.* 115 (2011), 1598–1609.

Ruggero Pintus, Enrico Gobbetti, and Marco Callieri. 2011a. Fast Low-Memory Seamless Photo Blending on Massive Point Clouds using a Streaming Framework. *ACM Journal on Computing and Cultural Heritage* 4, 2 (2011), Article 6.

Ruggero Pintus, Enrico Gobbetti, and Marco Callieri. 2011b. A Streaming Framework for Seamless Detailed Photo Blending on Massive Point Clouds. In *Eurographics Areas Papers*. 25–32.

Ruggero Pintus, Enrico Gobbetti, and Roberto Combet. 2011c. Fast and Robust Semi-Automatic Registration of Photographs to 3D Geometry. In *Proc. VAST*. 9–16.

Ruggero Pintus, Enrico Gobbetti, Giuseppa Tanda, and Massimo Vanzi. 2012. Acquisizione digitale multi-sensore di siti archeologici: il caso di Montessu, in "La Preistoria e la protostoria della Sardegna. In *Atti della XLIV Riunione Scientifica dell'Istituto Italiano di Preistoria e Protostoria, Cagliari-Barumini-Sassari 23-28 novembre 2009*. Vol. 3. I.I.P.P., 963–968.

Sudipta N Sinha, Drew Steedly, and Richard Szeliski. 2012. A multi-stage linear approach to structure from motion. In *Trends and Topics in Computer Vision*. Springer, 267–281.

Noah Snavely. 2013. http://phototour.cs.washington.edu/bundler/. (2013).

Noah Snavely, Steven M. Seitz, and Richard Szeliski. 2006. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph.* 25, 3 (2006), 835–846.

Noah Snavely, Steven M Seitz, and Richard Szeliski. 2008. Modeling the world from internet photo collections. *International Journal of Computer Vision* 80, 2 (2008), 189–210.

Ioannis Stamos and PK Alien. 2001. Automatic registration of 2-D with 3-D imagery in urban environments. In *Proc. ICCV*, Vol. 2. IEEE, 731–736.

Ioannis Stamos, Lingyun Liu, Chao Chen, George Wolberg, Gene Yu, and Siavash Zokai. 2008. Integrating Automated Range Registration with Multiview Geometry for the Photorealistic Modeling of Large-Scale Scenes. *International Journal of Computer Vision* 78, 2-3 (2008).

Christoph Strecha, Wolfgang von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proc. CVPR*. IEEE, 1–8.

C. Strecha, W. von Hansen, L.J. Van Gool, P. Fua, and U. Thoennessen. 2011. Dense multi-view stereo evaluation. http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html. (2011).

Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. 2013. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.* 19, 7 (2013), 1199–1217.

Toru Tamaki, Miho Abe, Bisser Raytchev, and Kazufumi Kaneda. 2010. Softassign and EM-ICP on GPU. In *Proc. ICNC*. 179–183.

Yanghai Tsin and Takeo Kanade. 2004. A Correlation-Based Approach to Robust Point Set Registration. In *Proc. ECCV*. 558–569.

Maarten Vergauwen and Luc Van Gool. 2006. Web-based 3D reconstruction service. *Machine vision and applications* 17, 6 (2006), 411–426.

Paul Viola and William M III Wells. 1997. Alignment by maximization of mutual information. *International journal of computer vision* 24, 2 (1997), 137–154.

Nathaniel Williams, Kok-Lim Low, Chad Hantak, Marc Pollefeys, and Anselmo Lastra. 2004. Automatic Image Alignment for 3D Environment Modeling. In *SIBGRAPI*. 388–395.

Haim J. Wolfson and Isidore Rigoutsos. 1997. Geometric Hashing: An Overview. *Computing in Science and Engineering* 4 (1997), 10–21.

Changchang Wu, Brian Clipp, Xiaowei Li, J-M Frahm, and Marc Pollefeys. 2008. 3D model matching with viewpoint-invariant patches (VIP). In *Proc. CVPR*. IEEE, 1–8.

Wenyi Zhao, David Nister, and Steve Hsu. 2005. Alignment of continuous video onto 3D point clouds. *IEEE Trans. Pattern Analysis and Machine Intelligence* 27, 8 (2005), 1305–1318.