Noname manuscript No.
(will be inserted by the editor)

Fully Automatic Registration of Image Sets on Approximate Geometry

M. Corsini · M. Dellepiane · F. Ganovelli · R. Gherardi · A. Fusiello · R. Scopigno

Received: date / Accepted: date

Abstract The photorealistic acquisition of 3D objects often requires color information from digital photography to be mapped on the acquired geometry, in order to obtain a textured 3D model. This paper presents a novel fully automatic 2D/3D global registration pipeline consisting of several stages that simultaneously register the input image set on the corresponding 3D object. The first stage exploits Structure From Motion (SFM) on the image set in order to generate a sparse point cloud. During the second stage, this point cloud is aligned to the 3D object using an extension of the 4 Point Congruent Set (4PCS) algorithm for the alignment of range maps. The extension accounts for models with different scales and unknown regions of overlap. In the last processing stage a global refinement algorithm based on mutual information optimizes the color projection of the aligned photos on the 3D object, in order to obtain high quality textures. The proposed registration pipeline is general, capable of dealing with small and big objects of any shape, and robust. We present results from six real cases, evaluating the quality of the final colors mapped onto the 3D object. A comparison with a ground truth dataset is also presented.

Keywords 2D-3D registration, image-geometry alignment, texture registration, 3D scanning

M. Corsini , M. Dellepiane , F. Ganovelli, R. Scopigno
Visual Computing Lab (ISTI-CNR), Pisa, Italy
E-mail: massimiliano.corsini@isti.cnr.it

R. Gherardi
Toshiba Cambridge Research Laboratory, UK
E-mail: riccardo.gherardi@crl.toshiba.co.uk

A. Fusiello
University of Verona, Italy
E-mail: andrea.fusiello@univr.it

1 Introduction

The digitalization of real objects into virtual models often entails acquiring not only the shape of the object but also its color. This is mostly done by taking pictures with a digital camera and solving the 2D/3D registration problem between the images and the 3D model obtained with, for example, a 3D scanner.

Some methods to solve this problem require user intervention, or they are based on strong assumptions on the type of object or the properties of the images to be mapped. A simple solution, adopted in [54, 56, 20], consists of taking calibrated pictures during the scanning phase from a known (relative) position, for example by mounting a digital camera on the laser scanner. In principle, this strategy is limited by the fact that the working conditions may not be optimal for taking photographs during the 3D scanning. However, Yang et al. [22] in particular have demonstrated that the color information acquired during the acquisition phase can also be used later to map digital photos with a very different visual appearance. Hence, the direct acquisition of colored point clouds/range maps can be very useful for solving the 2D/3D registration problem. In any case, for certain applications, color information is not available. For example, in some sites of interests, scanning can be performed only at night (e.g. Piazza della Signoria in Florence). In addition, the images and the geometry may have to be collected separately, for example by a professional photographer and a team that scans for other purposes, such as in a project aimed at visualizing David's restoration [14], which was achieved by mapping two different photographic datasets onto a previously acquired geometry without color information. Another issue is that some 3D scanners provide very poor color information (such as some triangulation scanners) or have problems setting up a reliable co-located calibrated camera (e.g. an airborne Lidar). In this paper we propose a novel global 2D/3D registration pipeline, which is so general that it can handle any type of application. The proposed pipeline simultaneously aligns a set of images on the 3D model of an object without any user intervention, with no prior knowledge and with no requirements regarding the geometry and the visual features involved.

The underlying idea behind our proposed registration algorithm is to exploit the improvements in image-based 3D reconstruction, where many robust Structure From Motion (SFM) algorithms are now available. The input photos are processed by an SFM algorithm, and the output is the position and orientation of the cameras at the time of shooting, along with a set of sparsely reconstructed 3D points. The idea is to use this data to compute the 2D/3D registration, by recasting it as a problem of aligning the 3D point cloud produced by the SFM to the geometry of the object. The transformation that aligns the point cloud to the object is applied to the extrinsic parameters of the cameras. Due to the sparseness and noise of these point clouds, the resulting alignment may be rather approximate. We therefore use a global refinement method based on mutual information to improve the accuracy of the final 2D/3D alignment. We will show that this registration pipeline provides high quality results and consistent color mapping all over the surface of the model.

1.1 Contribution

In this work we propose a fully automatic 2D/3D registration pipeline consisting of three processing stages. Two of the three processing stages can themselves be regarded as a contribution. Thus, the main contributions of this work are:

- An algorithm for the alignment of partially overlapping point clouds. The input point clouds can have different non-homogeneous sampling densities and different scales.
- A novel global registration refinement algorithm based on a statistical analysis capable of re-positioning a set of cameras in order to obtain very accurate and globally coherent color mapping on the 3D model.
- These two algorithms, have been integrated in a general, fully automatic, 2D/3D registration pipeline in order to robustly align a set of photos on a 3D model. No prior assumptions or additional information regarding the input geometry and the set of digital photos is required. Another important feature of our approach is that it is global, whereas many other approaches align one image at a time.

While the idea of exploiting multi-view geometry to improve 2D/3D registration is not new (Stamos et al. [61,38] and Wu et al. [66]), our approach is not bound by any strong assumptions regarding the shape of the object to be acquired or the information available, such as colored point clouds, thus making it more general than other state-of-the-art algorithms.

2 Related Work

Our pipeline exploits results from Structure from Motion (SFM) and point clouds alignment, thus important studies in these fields are also outlined below, together with a complete overview of 2D/3D registration methods.

2.1 Structure from Motion

Images are becoming the preferred way for the ubiquitous, low cost acquisition of quality three dimensional data. Several Structure from Motion (SFM) pipelines have been proposed [7,31,59,64,28]. They usually process images in batches and handle the reconstruction process without making assumptions about the image in the scene or the acquisition rig.

A key issue is the scalability of the SFM pipeline. One strategy is to use *partitioning methods* [18], which reduce the reconstruction problem to smaller and better conditioned subproblems which can be then optimized [62,45].

Another strategy is to select a subset of input images and feature points which represent the entire solution. Hierarchical sub-sampling was pioneered by Fitzgibbon [18], using a balanced tree of trifocal tensors over a video sequence; this approach was subsequently refined by Nister [46]. In Shum et al. [57] the sequence is divided into segments, which are resolved locally. They are then merged hierarchically, if necessary using a representative subset of the segment frames. A similar approach is followed by Gibson [24]. A recent paper [59] that works with sparse datasets describes a method of selecting a subset of images whose reconstruction approximates the result obtained using the entire set.

Gherardi et al. [23] proposed a hierarchical and parallelizable scheme for SFM. The images are organized into a hierarchical cluster tree, and the reconstruction then proceeds from the leaves to the root. Partial reconstructions correspond to internal nodes, whereas images are stored in the leaves. The SFM stage we use is based on this approach. We chose this approach since, thanks to its hierarchical nature, the subsets of images that provide a good reconstruction can be aligned independently, if the overall reconstruction fails.

2.2 Point Clouds Registration

The problem of aligning point clouds has long been studied, particularly in the case of range maps obtained by digital 3D scanning.

One of the most well known is the Iterative Closest Point (ICP) algorithm [3,55], which, given a set of *roughly* aligned range maps, refines the alignment by minimizing the Hausdorff distance between the overlapping range maps following an iterative approach. The global registration of point clouds can be found by extracting local shape descriptors, matching points with similar signatures, and using these matches to choose the best alignment transformation. Some examples of methods that use this approach include Spin Images (Johnson et al. [29]), which uses a cylindrical projection of local sets of surface points represented as an image, the methods based on SIFT and RIFT descriptors [36,58], and Kalogerakis et al's method [30], which extracts of line features directly on the point cloud data. Makaida et al. [41] developed a method for the fully automatic alignment of point clouds by correlating Extended Gaussian Images (EGI) in the Fourier domain. Pottman et al. [51] performed the optimization directly on the affine space, by applying the rigidity constraint only toward the end of the optimization. Krishnan et al. [34,33] proposed a framework to perform the optimization explicitly on the manifold of rotations through an iterative scheme based on the Gauss-Newton optimization method, thus obtaining a quadratic convergence rate. Bonarrigo et al. [4] improved the optimization-on-a-manifold approach by boosting its performance. None of the above methods is very robust against noise or the presence of outliers. However, one of the most robust algorithms is the 4-Point Congruent Set (4PCS) which combines a non-local descriptor that is simple and fast to compute (four coplanar points in the point cloud) and uses a RANSAC scheme to choose couples of descriptors. We propose here an extension of the 4PCS algorithm just mentioned, which accounts also for the estimation of different scale between the point clouds to align.

2.3 2D/3D Registration

Fixed-relative methods. One of the simplest approach for the registration of images on range maps (or other geometric data) is the *fixed-relative position* or *co-located camera* approach, where it is assumed that the pose of each camera is relative to a known position [54,56,20]. In principle, this approach cannot always be applied although studies such as Yang et al. [22] demonstrate that large number of applications can by exploiting color information, as stated in the Introduction.

Semi-automatic methods. Semi-automatic approaches, where the user supports the registration process, are generally robust. They are usually based on the setting of several 2D-3D correspondences: one of the most recent method of this type is the one of Franken et al. [19]. However, the procedure can be very time-consuming, especially when tens or hundreds of images need to be aligned. Automatic planning of the images required could minimize image acquisition and remove the need for registration, as in Matsushita et al. [43] where the camera is positioned manually and the pose is optimized in advance. Obviously, this approach can only be used in controlled environments.

Features-based methods. Automatic image-geometry registration can be achieved by analyzing the image and geometric features in order to estimate the 2D-3D transformation. Features can be points, lines, rectangles, circles, etc. Neugebauer et al. [44] employed edge intensity in their registration method. Liu et al. [37] assume that the 3D scene contains clusters of vertical and horizontal lines, and thus they used orthogonality constraints for the registration. Parallelepipeds are extracted from the range maps, and subsequently matched to rectangles extracted from the input images. This method is particularly suitable for urban scenes. Unfortunately, the assumptions of the above methods only hold in certain types of scenes.

Color-based methods. One way to make these methods more robust and reliable is to exploit the reflectance values (laser intensity) or color information that some 3D scanners acquire. This helps the feature extraction and the establishment of correspondences [27]. In fact, Yang et al. [22] made their co-located camera approach robust using this method. Wu et al. [66] exploited color information to align two 3D scenes even from significant viewpoint changes. Wu proposed a new local feature called VIP (Viewpoint Invariant Patch) computed by normalizing the local viewpoint and orientation using local texture rectification and a dominant image gradient. A single VIP is sufficient to estimate the similarity transformation to align two 3D models. This method is particularly suitable for large-scale image-based reconstruction where several reconstructed parts of the whole scene need to be aligned.

Silhouette-based methods. Several other algorithms use a *silhouette based* approach to find the camera transformation by minimizing the error between the contour of the object in the image and the contour of the projected 3D model [39,8,35]. Lensch [35] proposed a robust implementation of these techniques by introducing a similarity measure to compare them. Silhouette-based methods require the entire object to be present in each image. However, this may be a severe limitation for large scale objects. Moreover, there normally has to be a manual pre-processing step on the images to remove the background.

Statistical methods. One of the mathematical tools typically used for registration is Mutual Information (MI), which catches the non-linear correlations between the image and the geometric properties of the target surface. This approach, which is extensively used in medical imaging (see [48] for a survey), was pioneered by Viola and Wells [65] and by Maes et al. [40]. Viola and Wells [65] suggested registering the image by maximizing the mutual information of the surface normals with the corresponding gradient variations of the image. Corsini et al. [13] extended this algorithm by including other geometric properties in the alignment algorithm, such as ambient occlusion and reflection directions. Cleju et al. [11] also extended Viola and Wells's work to align more than one image simultaneously. We propose a similar approach to refine the global registration based on a completely different optimization framework.

Multi-view methods. Our proposal is to exploit multiview geometry to simultaneously align the entire set of images of interest. This is in contrast with almost all of the methods presented so far, which register one image at a time. Moreover, in many methods the convergence depends on the initial 3D model position. Our idea is not entirely new. In fact, the main works which exploit Structure From Motion (SFM) during 2D/3D registration process are those of Zhao et al. [67], Stamos et al. [61] (which is an extension of the work of Liu et al. [38]), Zheng et al. [68] and Pintus et al. [47].

The aim of Zhao et al.'s study [67] is to register a video onto a point cloud. To achieve this goal, a point cloud is computed from the video sequence using motion stereo and camera pose estimation techniques. The point cloud obtained is then registered with the target 3D model using the ICP algorithm. This is similar to our idea, i.e. to recast the 2D/3D registration problem so that the aim becomes to align two point clouds. The main difference lies in the alignment algorithm, which is a standard ICP. This simple approach suffices thanks to the fact that the coherency of the video sequence exploited to achieve the sparse 3D reconstruction enables good initial registration to be directly obtained. Intrinsic camera parameters must be known beforehand.

Liu et al. [38] presented a feature-based method that extends their previous work [37]. Due to the rigid geometry constraints of this method (orthogonality constraints between line features, three vanishing points needed in the images) only a subset of the input images can be aligned. SFM is used to register the remaining uncalibrated images. The subset of aligned images is used to estimate the similarity transformation (scale/rotation/translation) which maps the coordinate frame of the cameras in the reference frame of the 3D model. The geometric constraints limit this system to aligning images of 3D buildings on the corresponding 3D models.

Stamos et al. [61] extended this system in order to relax the orthogonality constraint so that the algorithm can be used not only in strictly urban scenes, but for example in indoor architectures. In any case, the architectures remain the main target of this pipeline. Zheng et al.'s registration algorithm [68] is a features-based method, which requires the parameterization of the input model in order to extract features based on surface normals. Corresponding features are extracted in the images that are also calibrated using an SFM algorithm. How the corresponding 2D features for the 2D/3D registration are extracted is not clearly explained in the paper, thus making it difficult to carry out a real comparison with the proposed approach. Moreover, the problem of different scale factors between the image-based reconstructed points and the 3D model is neglected.

Pintus et al. [47] proposed a method for registering images on point clouds, which is based on the use of SFM. However, the rough alignment step entails the user to manually align one or more images onto the point cloud, and the global refinement step relies only on the point cloud generated by the SFM algorithm, which is not always reliable. We believe that the global refinement should use the images and the 3D model rather than the reconstructed points, as in Cleju et al. [11] and in our method.

3 Overview

The alignment of 2D images on 3D models shares some problems with range maps alignment. The procedure can be divided into *rough aligment* and *fine alignment*; the algorithmic solutions for these two phases are usually different. In the first phase, an initial alignment position is found, given that no previous information regarding the object and the photographic set is known. The alignment is then refined to an optimal position by the second phase.

Our global registration pipeline (Figure 1) takes as input a 3D model of an object of interest, **P**, repre-

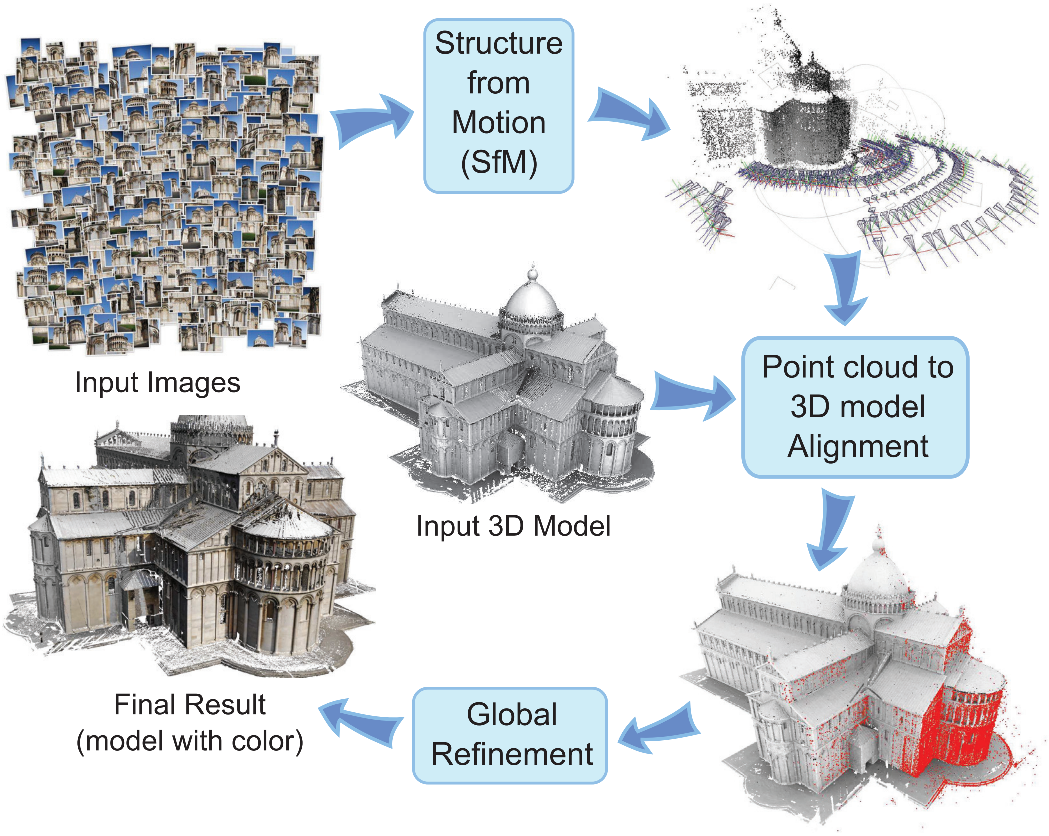

Structure from Motion (SfM)
Input Images
Input 3D Model
Point cloud to 3D model Alignment
Final Result (model with color)
Global Refinement

Fig. 1 Overview of the proposed global 2D/3D registration pipeline.

sented as a point cloud or a triangular mesh, and a set of images to be registered on it, $\mathcal{S}$. The pipeline has three processing stages. In the first stage an SFM algorithm is used to calibrate the images of $\mathcal{S}$ and to obtain a sparse 3D reconstruction denoted by $\mathbf{Q}$. Note that $\mathbf{Q}$ can only cover a part of $\mathbf{P}$ (partial photo coverage) but it might also contain points that do not belong to $\mathbf{P}$ (everything in the pictures is reconstructed, including objects in the background). In the second stage, $\mathbf{Q}$ is aligned onto $\mathbf{P}$, taking into account the aforementioned problems and the fact that the scale factor of $\mathbf{Q}$ is unknown. This is achieved by extending the 4 Point Congruent Set (4PCS) algorithms by Aiger et al. [1]. The output of this stage provides a rough alignment, and is very important since a good initial position of the image set is fundamental to obtain an accurate final registration. However, small errors are allowed. During the third stage this approximate registration of the images on the geometry of the 3D model is then globally refined. Each camera parameter (position, orientation and focal length) is adjusted to obtain a globally coherent color projection on the model. This is done by combining a previous algorithm for the fine alignment of 2D/3D registration based on mutual information [13] with a graph-based optimization framework typically employed in the global refinement of range maps [53].

In the rest of the paper we provide further details of these three stages.

4 Structure From Motion: obtaining a sparse 3D Reconstruction

The Structure From Motion algorithm we use in the first stage is called SAMANTHA [17,23], and it is devoted to orient the cameras and recover the sparse structure of the scene, up to a similarity. It consists of three sub-stages: keypoint matching, clustering, and geometric processing.

The keypoint matching sub-stage is fairly standard, and mainly follows [6] and [59]; the details are reported in [17]. The output of this sub-stage is a set of *tracks*, i.e., keypoints that match in more than three images, and a set of fundamental matrices and homographies linking pairs of views.

In the second sub-stage the views are organized into a hierarchical cluster structure that guides the reconstruction. The method starts from an affinity matrix among views, computed using the following measure,

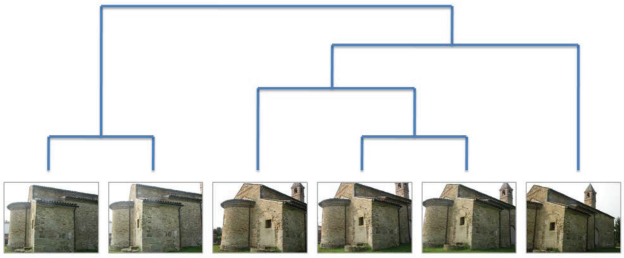


Fig. 2 An example of dendrogram for a six-views set.

which takes into account the number of common keypoints and how well they are spread over the images:

$$a_{i,j} = \frac{1}{2}\frac{|S_i \cap S_j|}{|S_i \cup S_j|} + \frac{1}{2}\frac{CH(S_i) + CH(S_j)}{A_i + A_j} \tag{1}$$

where S_i and S_j are the set of matching keypoints in image I_i and I_j respectively, $CH(\cdot)$ is the area of the convex hull of a set of points and A_i (A_j) is the total area of the image.

Views are grouped together by agglomerative clustering, which produces a binary cluster tree, called a *dendrogram* (see Figure 2). The clustering algorithm proceeds in a bottom-up manner: starting from all singletons, each sweep of the algorithm merges the pair with the smallest cardinality of the ℓ closest pair of clusters. The distance between two clusters is determined by the distance of the two closest objects, as in the simple linkage rule.

This means that we soften the "closest first" agglomerative criterion by introducing a competing "smallest first" principle, which tends to produce better balanced dendrograms. The amount of balancing is regulated by the parameter ℓ: when $\ell = 1$ this is the standard agglomerative clustering with no balancing; when $\ell \geq n/2$ (n is the number of views) a perfectly balanced tree is obtained, but the clustering is poor, since the distance is largely disregarded. We found that balanced results are obtained when $\ell = 5$.

This procedure enables us to decrease the computational complexity compared to a sequential SFM pipeline, from $O(n^5)$ to $O(n^4)$ in the best case, i.e. when the tree is well balanced (see [23] for a complete proof).

The dendrogram produced by the clustering substage constraints the order in which SAMANTHA processes the views. Each cluster is initialized with a two-view reconstruction, after which the reconstruction of a cluster is enhanced by adding new view from the same cluster, as in the sequential pipelines. Alternatively, two clusters are merged.

Each node is upgraded, as soon as possible, to a Euclidean frame. If cameras are calibrated (the intrinsic parameters are known) then the Euclidean frame is available from the beginning. If not, autocalibration is run on nodes with a minimum of m views, where m depends on the conditions (for example, autocalibration with known skew and aspect ratio require a minimum of four views to obtain a unambiguous solution).

This stage can be replaced by other SFM algorithms. However, we decide to adopt this solution due to its hierarchical nature, which, in principle, enables to provide a good reconstruction even when the reconstruction of the whole object fails, as highlighted by the results presented in Section 8.

5 Point Cloud to 3D Model Alignment

The aim of this stage is to align the sparse point cloud obtained by SAMANTHA, hereafter **Q**, to a more accurate 3D model **P**. Compared to the well known problem of registering couples of range maps, this case has serious complications.

First, the point clouds produced by the SFM methods are a sampling of the real object up to an arbitrarily, and unknown, *scale factor*. If **Q** and **P** were complete samplings of the same surface, we could (at least approximately) recover the scale factor as the ratio between the size of the oriented bounding boxes of the two models. Unfortunately, and this is the second difficulty, *the two models can share any fraction of the surface*. Figure 3 shows an example where neither of the two models includes the other.

The third complication is that the density of the point cloud **Q** varies unevenly over all the model, thus we cannot use it with any certainty to find the scale factor. Finally, the point cloud produced by the SFM is generally noisy and the approaches based on local descriptors would be difficult to exploit in this case.

5.1 Alignment with 4 Points Congruent Sets (4PCS)

Our proposal is based on a recent work by Aiger et al. [1]. They implemented a RANSAC approach to align pairs of surfaces in arbitrary initial poses. Their idea is to pick a fixed number of quadruples of coplanar points on **P** and then, for each of these quadruples, to look for all the quadruples in **Q** that are approximately congruent, i.e. that can be transformed into the quadruple in **P** with a rigid transformation. For all the candidate quadruples in **Q** the respective transformation is applied to the whole point cloud. Then the number transformed points of **Q** which is within a predefined threshold from their closest point in **P** is calculated. The candidate quadruples with the highest number of such points define the chosen transformation.

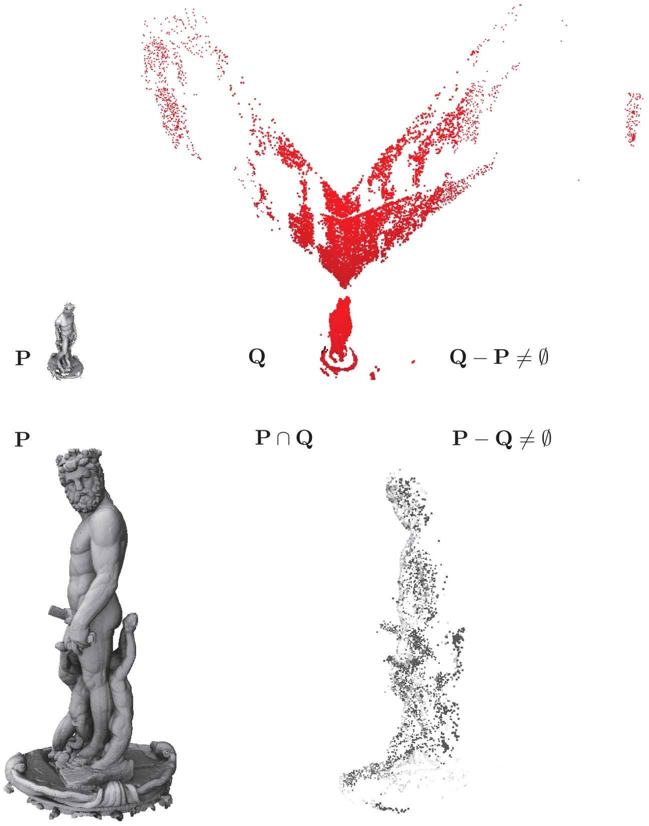

$\mathbf{P}$
$\mathbf{Q}$
$\mathbf{Q} - \mathbf{P} \neq \emptyset$
$\mathbf{P}$
$\mathbf{P} \cap \mathbf{Q}$
$\mathbf{P} - \mathbf{Q} \neq \emptyset$

Fig. 3 An example of a partial overlap of the 3D model (**P**) and the point cloud obtained by SFM (**Q**). Note that **Q** also covers part of the building behind the statue which is not represented by **Q**. On the other hand, the back of the statue is covered by **P** but not by **Q** .

The 4-points congruent sets method has proven to be fairly robust on noisy data, mostly because it does not need to compute fragile local descriptors based, for example, on normals, curvature and so on. However, the efficiency of the 4PCS algorithm relies on *a priori* knowledge of the size of the overlap of the two objects, and on the assumption that there is no scale factor between the objects. Since we cannot make these assumptions, the complexity of the most time consuming task of the algorithm, i.e. testing a candidate quadruple, increases by a factor $O(n)$ where n is the number of vertices in **Q** (a more detailed explanation of the original algorithm and the reasons for these assumptions can be found in the original paper [1] and are concisely summarized in Appendix A).

5.2 Our Approach

We propose a scale independent version of the 4PCS algorithm by introducing two modifications. The first is a preprocessing step to overcome the difference in sampling between the scanned model and the point cloud obtained by SFM. This is because the sampling of the latter does not depend on object geometry alone, as it does with the sampling with laser scanners, but on the reconstruction stage as well. By expressing the point cloud as a set of planar regions and resampling them uniformly, we try to obtain a representation that is as dependent as possible on the actual shapes and not on the sampling provided by the SFM algorithm. This consists in partitioning the two point clouds into a set of quasi planar regions using the Variational Shape Approximation algorithm (VSA) [12] and then resampling the clouds uniformly with respect to their area (details in Section 5.3).

The second modification consists of introducing a rasterization based algorithm that considerably reduces the time needed to test candidate transformations. The algorithm estimates if the integral of the distance between the two point clouds is below a certain threshold using a *hardware occlusion query* and exploiting the phenomenon known as *z-fighting*. Details of this technique can be found in Section 5.4. As in the original version, we refine the result by applying the Iterative Closest Point algorithm [55].

Listing 1 shows the steps of our algorithm:

Listing 1 Scheme of the proposed algorithm

```
ScaleIndependent4PCS(P,Q)
{
  Pr = VSA_Resample(P);
  Qr = VSA_Resample(Q);

  nBest = 0;
  for i = 0 to L
    B = CoplanarBase(Pr);
    Ui = FindCongruents(Qr);
    forall Ui in U
      T = FindTransformation(Ui,B);
      if (ZFightingRejectionTest(T))
        nClosest = CountClosest(T,$\delta$);
          if(nClosest > nBest)
            nBest = nClosest;
            Tbest = T;

  RefineWithICP(T);
}
```

5.3 Resampling by the approximation-driven Variational Shape Approximation (VSA)

The original version of the VSA algorithm takes as input the number of regions into which the surface must be partitioned, and returns a partition into planar regions and an approximation error. We rewrote the VSA algorithm so that it takes an approximation error ϵ as input and provides a partition into planar regions so that the approximation error is below ϵ. The approximation error we use is critical. We cannot use an absolute value, because the unity of measure of $\mathbf{Q}$ is unknown, and neither can we choose a value that is dependent on the size of a bounding volume (for example a percentage of the bounding box), because we do not know the overlap between the two point clouds *a priori.*

Instead, we find the value for ϵ by analyzing the histogram of the approximation error of fitting a plane to a point and its neighbors. For each point q let $Pl_h(q)$ be the plane best fitting the h closest points to q and q itself, and let $E(Pl_h(q))$ be the approximation error of that plane. In an ideal situation of an infinitely dense sampling, the error E would be zero except when the surface is not G^1 (e.g. on the ridges and apexes). In a non ideal but optimal situation, there is a very small approximation error on all G^1 points and higher value on the non-G^1 points, and the number of non-G^1 points should be less than the G^1 points by a quadratic factor. In other words, in an optimal sampling we assume that a point and its neighbors describe a portion of plane except for those points that are close to a feature point. We apply these considerations to $\mathbf{Q}$, which is not an optimal sampling, by computing the error E for all its points and taking the average value of the 80 percentile as a value for ϵ. Thus, we try to filter out those sampling points that are not likely to be on a planar region, and we take the approximation error of the remaining points as a value that accounts for sampling precision. Figure 4 shows the result of the VSA with our computation of ϵ for the scanned model and the point cloud of two 3D models. Although the partitions are different the size of the regions is similar in the two datasets.

The partition of the point clouds into the coplanar regions is used to prune the number of quadruples generated in $\mathbf{P}$ and $\mathbf{Q}$, and, consequently, the number of quadruples tested for the alignment. More specifically, we restrict the choice of the points of a quadruple to those that do not belong to the same planar region. This assumes that the overlap region between $\mathbf{P}$ and $\mathbf{Q}$ does not consist of a single planar region and it means that we do not have to generate n^2/r couples, where r is the number of planar regions. Note that if the only overlapping between the two point clouds were a planar region, our algorithm, like any other based on a distance between surfaces, would fail.

Testing for a congruent basis on $\mathbf{Q}$

Testing if a quadruple U_i in $\mathbf{Q}$ is congruent with a quadruple B in $\mathbf{P}$ means to find if the transformation that brings the two quadruples to coincide can be expressed as a composition of rotation, translation and uniform scaling.

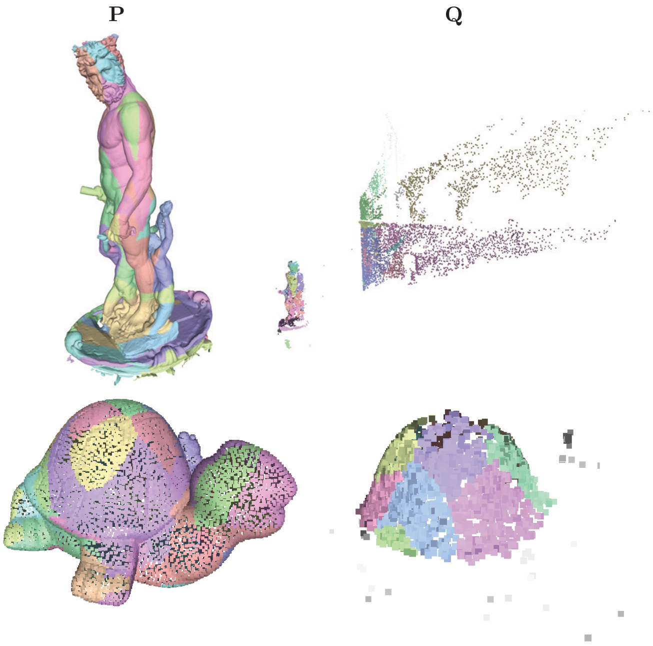


Fig. 4 Partition of the point clouds with the VSA algorithm. (Left) The reference model. (Right) The point cloud obtained by SFM.

Before trying to compute the transformation we use two simple tests to discard non-congruent quadruples, by computing and comparing two quantities that should be the same for congruent couples. The first is the angle α_p between two segments (a, b) and (b, c) and the second is the ratio between their lengths $R_p = d_{1p}/d_{2p}$ (see Figure 5). If the quadruple is not discarded, we compute the 4×4 matrix that brings U_i on B as:

$$M = RTS \tag{2}$$

where S is a uniform scaling calculated as $(d_{1p}/d1_q + d_{2p}/d_{2q})/2$, i.e. the average ratio between the segment lengths of the two quadruples, T is the translation that makes the two intermediate point to coincide, and R is the rotation around the (now common) intermediate point that makes s_{1q} to coincide with s_{1p} and s_{2q} to coincide with s_{2p}. Note that this is an heuristic solution and the more precise Horn method [26] can be used instead. However, under the hypothesis that the quadruple has not been discarded by the tests on angle or the ratio of the lengths, we estimated experimentally that the error produced is negligible. So, we decided to adopt the heuristic since it is computationally cheaper than the Horn method.

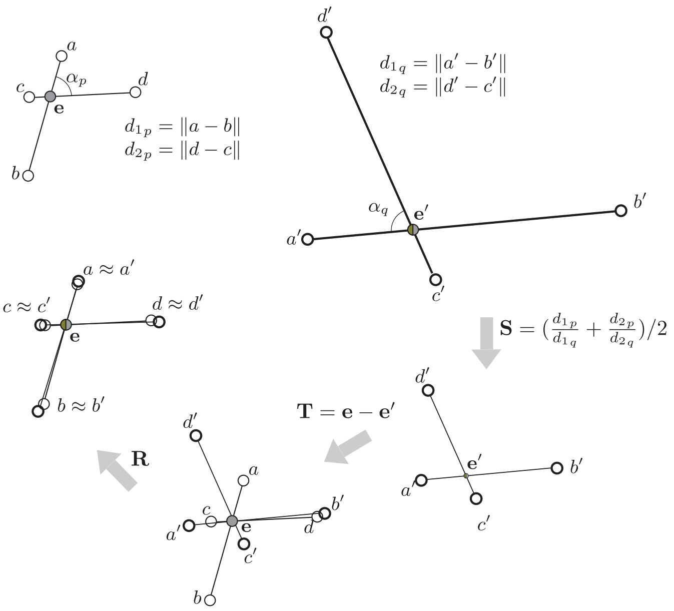


Fig. 5 Test for early rejection of non congruent basis and transformation.

5.4 Rasterization-based fast rejection test

Testing wether a transformation that coincides two quadruples also brings the two surfaces close to each other would be the most costly operation of the process, since we need to apply the transformation matrix to every single vertex of $\mathbf{Q}$ and then find its distance by $\mathbf{P}$. Using a search data structure to keep the vertices of $\mathbf{P}$, there is a cost of $O(n \log n)$ for each closest-point test.

Instead, we harness the larger FLOPS and the parallelism of the GPU to make a very fast rejection test by adapting the method presented in [50] (in the context of fitting geometric primitives) to our case.

Let $\mathbf{Q}' = M\mathbf{Q}$ be the point cloud $\mathbf{Q}$ transformed by the candidate transformation M and suppose the two point clouds are visualized as in Figure 6(a) twice: the first by translating $\mathbf{Q}'$ along the viewing direction v by a small amount $+\delta$ (Figure 6(b)) and the second by translating it by $-\delta$ (Figure 6(c)). The first time the points of $\mathbf{Q}'$ have a greater chance to occlude the points of $\mathbf{P}$, and the second time the opposite is true. If $\mathbf{P}$ and $\mathbf{Q}'$ are far from each other, i.e. if the transformation is not a good one, the two renderings will produce a very similar, if not equal, result. This is because if they are far, the translation of $\mathbf{Q}'$ by a small amount will make no difference. The nearer $\mathbf{P}$ and $\mathbf{Q}'$, the more the translation of $\mathbf{Q}'$ will affect the result of the rendering.

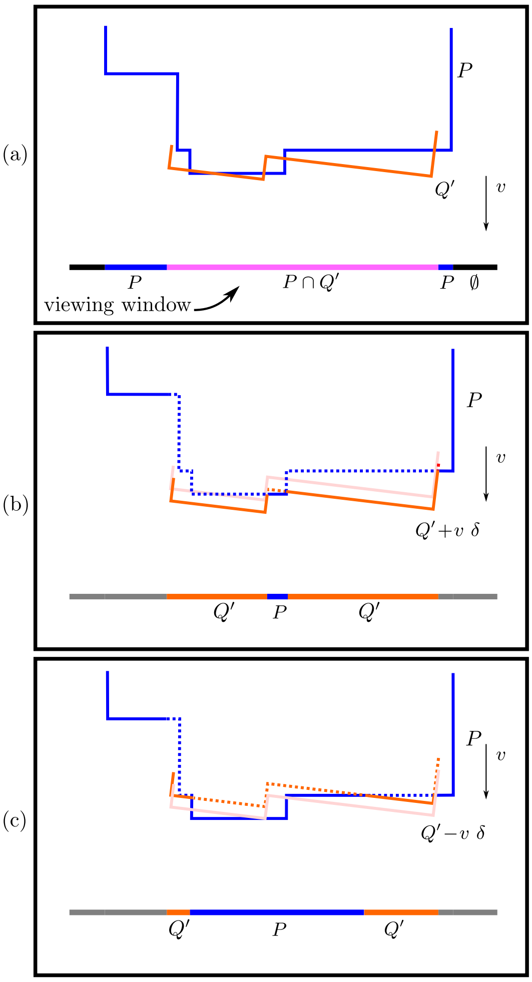


Fig. 6 Estimating the overlapping between $\mathbf{P}$ and $\mathbf{Q}'$. (a) Definition of the area of the screen where both projects (b) Fragments distribution when $\mathbf{Q}'$ is moved towards the viewing plane (c) Fragments where $\mathbf{Q}'$ is moved away from the viewing plane.

Following these considerations we define the overlap between $\mathbf{P}$ and $\mathbf{Q}'$ as:

$$Ov(v, \mathbf{P}, \mathbf{Q}') = \frac{\|F_{\mathbf{Q}}(v, +\delta) - F_{\mathbf{Q}}(v, -\delta)\|}{F_{\mathbf{P}\cap\mathbf{Q}}(v, 0)} \tag{3}$$

$$Ov(\mathbf{P}, \mathbf{Q}') = \max_{v \in V} Ov(v, \mathbf{P}, \mathbf{Q}') \tag{4}$$

where $F_s(v, x)$ is the number of pixels of the rendering belonging to the surface $s \in (\mathbf{P}, \mathbf{Q}')$ when $\mathbf{Q}'$ is translated along v by x and V is a set of directions. The implementation to compute $Ov(v, P, Q')$ is straightforward. We first set the viewing transformation, then render $\mathbf{P}$, apply the translation $+\delta/-\delta$ and set the starting point of the occlusion query. The *occlusion query* is a GPU function which counts the number of fragments that have passed the depth test since the query was started. In this way we know how many fragments were produced by rendering $\mathbf{Q}'$, i.e. the value of $F_{\mathbf{Q}}(v, +\delta)$.

Note that this algorithm finds a lower estimate of how much the portion of surfaces of $\mathbf{Q}$ and $\mathbf{P}'$ are closer than δ to each other. In the example in Figure 6, if the view direction was horizontal towards the right, the estimated overlap would have been 0. Therefore the higher the number of directions tested, the more precise the result and the higher the cost, because testing for each direction costs two rendering passes. Since the test is based on occlusion we do not directly render $\mathbf{P}$ and $\mathbf{Q}$ as point clouds but we generate two low resolution meshes using a Poisson reconstruction [32]. This is not mandatory, we can also employ a simple splatting rendering technique as long as the rendering of the point cloud produces a reasonable occlusion.

6 Global Refinement

Once the transformation between the point cloud and the 3D model has been found, by applying the same transformation to the set of cameras estimated by SFM, an initial alignment will be reached. As previously stated, the point cloud resulting from SFM is sparse and less accurate than the scanned model, thus we can only obtain a rough alignment of the two.

The next step in the pipeline focuses on optimizing the parameters (both extrinsic and intrinsic) of each camera to obtain a coherent color projection of all the images on the 3D model. This goal would be the same as estimating the actual camera parameters at shooting time only if no approximation errors were committed either in the 3D scanning pipeline or in the SFM reconstruction, which is never the case.

Our strategy is based on Mutual Information (MI) among all the images in the set. Section 2.3 already detailed some of the approaches using the maximization of Mutual Information to align a 2D image on a 3D model. Although these methods are reliable and robust, they only enable the alignment of a single image at a time, and only exploit the geometric properties of the object (see [65, 11, 13]). The use of these methods on a group of images will likely lead to a lower quality color projection, for example when a few images are not perfectly aligned.

In a global registration framework, the goal is to "distribute" the alignment error among all the images, in order to minimize the inaccuracies and improve the quality of the final color of the model. This is also the aim of the work by Cleju et al. [11] although the optimization proposed, the stochastic gradient descent method, is different from our solution. The core of our global refinement lies in the maximizing the MI for each image, calculated between the image itself and a rendering of the projection of the other images of the dataset on the 3D model. We apply a graph-based global registration which is similar to an approach originally proposed by Kari Pulli [53], in the context of the global alignment of range maps. In our graph, the nodes are the images and the links connect the images which projection on the model overlap. Our method is presented in the next section, after a brief description of the single-image method of Corsini et al. [13] which is useful for the clarification of aspects of the proposed algorithm.

6.1 Single-image alignment using Mutual Information

Mutual Information (MI) measures the information shared by two random variables A and B. Mathematically, this can be expressed using entropy or joint probability. Following this interpretation, the mutual information $\mathcal{MI}$ between two images I_A and I_B can be defined as:

$$\mathcal{MI}(I_A, I_B) = \sum_{(a,b)} p(a,b) \log \left(\frac{p(a,b)}{p(a)p(b)} \right) \tag{5}$$

where $p(a)$ ($p(b)$) is the probability that the value of the pixel I_A (I_B) is a (b) and $p(a,b)$ is the joint probability of the event (a,b). The joint probability distribution can be easily estimated by evaluating the joint histogram of the two images and then dividing the number of occurrences of each entry by the total number of pixels. A joint histogram is a bi-dimensional histogram made up of $N \times N$ bins; the occurrence (a,b) is associated with the bin (i,j) where $i = \lfloor a/m \rfloor$ and $j = \lfloor b/m \rfloor$ and m is the width of the bin. This measure can be seen as a measure of *nonlinear correlation* between the variables A and B.

The image-to-geometry registration problem is cast in this framework by determining the parameters of the camera model that produce, the image I_B that maximizes MI with respect to the image to align (I_A). The main problem is the generation of the image I_B; the lack of a-priori knowledge regarding lighting, color and material properties of the model prevents realistic renderings from being generated. However, the goal of the rendering cycle is not to generate a photorealistic rendering but to synthesize an image which has a high correlation (even nonlinear) with the input picture under a wide range of lighting conditions and material appearances. Corsini et al. [13] proposed several rendering types that make the geometry of the model correlate well with the images. For example, ambient occlusion correlates well since the occluded parts of the geometry often correspond with the dark parts in the real image due to the poor illumination arriving at these points.

In this context the registration can be formalized as an optimization problem in a 7D space:

$$\mathcal{C}^* = \arg\max_{\mathcal{C}\in\mathbb{R}^7} \mathcal{MI}(I_A, I_B(\mathcal{C})) \tag{6}$$
$$\mathcal{C} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, f)$$

where f is the focal length, (t_x, t_y, t_z) and $(\theta_x, \theta_y, \theta_z)$ define the position and orientation of the camera, I_A is the image to align and I_B is the rendering (e.g. the ambient occlusion map) of the 3D model. Obviously, I_B depends on the camera parameters $(\mathcal{C})$. The Equation (6) can be solved by a non-linear optimization algorithm such as NEWUOA [52].

6.2 Extension to multiple images

Having more than one image turns out to be an advantage for the aforementioned approach. With only one image to align and the geometry of the 3D model we had to analyze the geometric features that could correlate with real photographs even in absence of color. However, since we have a set of images, we can project them on the surface and obtain the color information. Obviously, at the beginning the projection will be inaccurate, which is where the graph-based global optimization comes into play to distribute and progressively minimize the alignment error. In other words, the graph is a structure where each node corresponds to an image. The nodes are connected if the images overlap, and a weight is associated with each arc. The value of the arc between an image I_1 and another image I_2, indicated with $w(I_1, I_2)$, corresponds to

$$w(I_1, I_2) = \mathcal{MI}\left(I_1, \mathrm{proj}(I_2, I_1)\right)) \, O(I_1, I_2)) \tag{7}$$

where the first term is the MI calculated between the image I_1 and the projection of the image I_2 on the image plane of I_1. The projection is achieved as follows: if the 3D model is “covered” by image I_2, the corresponding pixel value is used, otherwise the *combined rendering* (ambient occlusion + normals map) proposed by Corsini et al. [13] is used. The value of the arc is also weighted by the term O, which represents the amount of overlap between the images, and is the ratio between the pixel on I_1 image plane which is covered by I_2, and the total number of pixels covered by the 3D model. According to this definition, the graph related to each dataset considers each image, and creates an arc for each couple of images where there is enough overlap (the threshold value for function $O(.)$ is set to 0.2). The result of the building phase is a weighted directed graph. The 3D model is not represented in the graph, but it plays the role of a “medium” due to the projections involved.

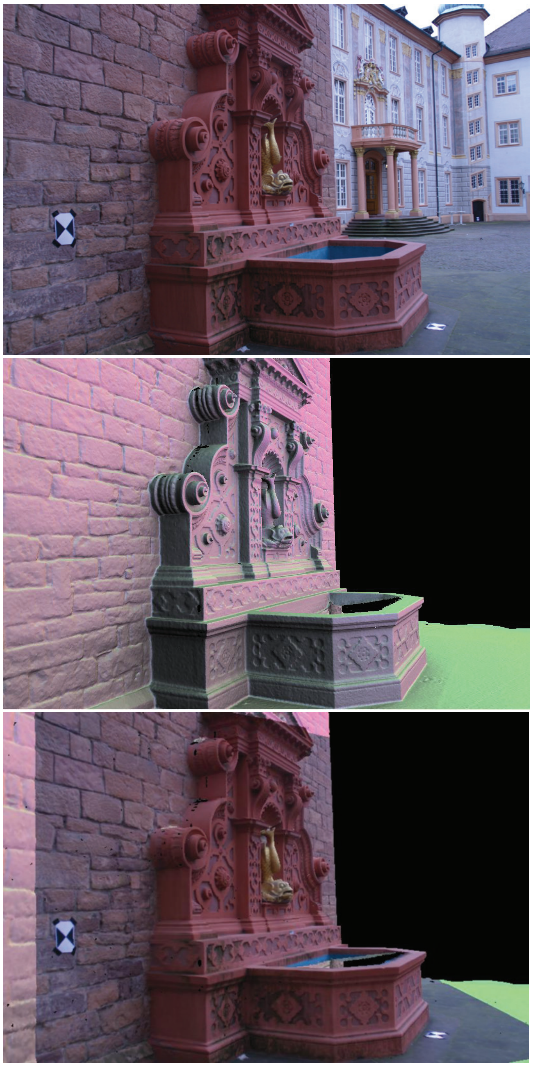


Fig. 7 (Top) One of the images of the dataset. (Middle) The rendering proposed by Corsini et al [13]. (Bottom) The rendering used to guide the global refinement.

6.3 Global graph-based refinement

The refinement approach is similar to the approach usually applied for the global adjustment of range map registration, when the registration error between pairs of range maps calculated as the Haussdorf distance is minimized.

The refinement is obtained by considering one node at a time, and refining all the nodes in the graph. The procedure is repeated until convergence, which is when the difference between the camera parameters before and after the refinement operation is below a defined threshold. The difference between two sets of camera parameters can be evaluated in many ways. In our case, this value is calculated by projecting a set of samples from the target 3D model onto the image plane, before and after refinement. The average difference in pixels

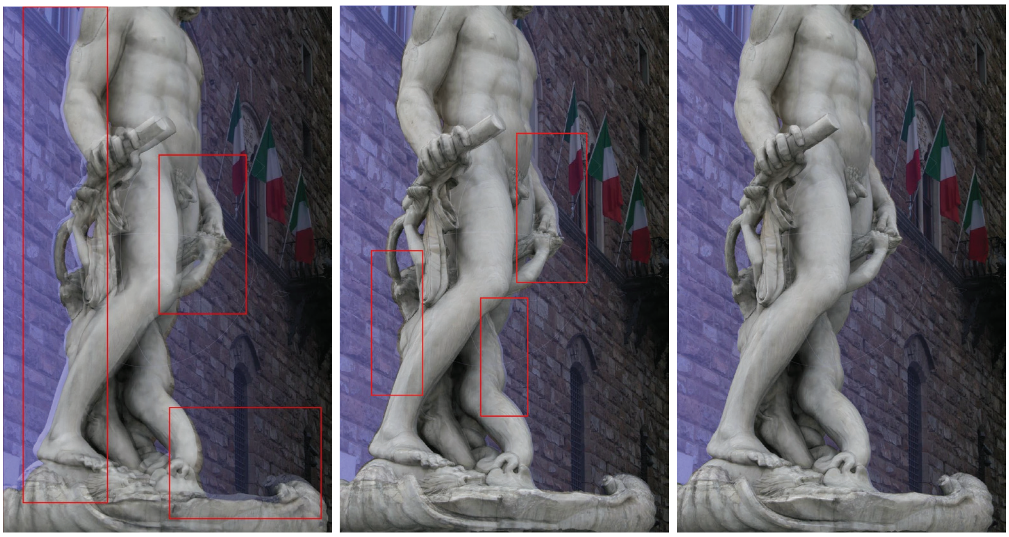


Fig. 8 (Left) Starting alignment. (Center) Result using Corsini et al. [13]. (Right) Result using global refinement.

gives a reasonable measurement of the difference between the camera parameters.

The graph refinement follows this loop:

1. **Selection of the node**: of the nodes with the biggest number of already refined neighbors, the node to be refined is the one that, has the biggest number of entering arcs.
2. **Node refinement**: the refinement is obtained by maximizing the MI (solving (6)) between the image requiring alignment and a rendering of the 3D model where all the images associated with connected nodes are projected on the geometry. Since several images can be projected onto the same portion of geometry, the color assigned is a weighted contribution of all the images, based on the value of the arc connecting the two nodes. Using this approach, the other images should "guide" each node to find a common alignment, thus reducing or distributing the alignment error throughout all the nodes. Figure 7 shows an image (top), the *combined rendering* of the model proposed by Corsini et al [13], and the corresponding rendering used to guide the alignment. If portions of the geometry are not covered by any other image in the set, *combined rendering* is used.
3. **Node labeling**: when the maximization procedure ends, the node is labeled as *refined*, and the graph is updated (all the weights of the arcs involving the node are re-calculated). The procedure goes back to step 1, until all the nodes are refined.

An example of the results on a single image, using the proposed approach, is shown in Figure 8. The top image shows the initial alignment of the model (in transparency) with respect to an image. Severe misalignments are visible, indicated by the red squares. The middle image is the result of the alignment algorithm proposed by Corsini et al. [13]. The alignment is improved, but there are still inaccuracies near the left-hand side and in the rear area of the statue: this happens because the information provided by the geometry in this case is not sufficient. Using our approach (see Figure 8 on the right) the image set helps to ensure a much more accurate alignment. However, before running the graph-based optimization framework, there is a *pre-alignment step* where Corsini et al.'s algorithm is applied to each image separately. This improves the initial estimation of the camera parameters estimation. Our experiments show that the final quality of the results is slightly improved by this pre-alignment step.

6.4 MI vs NCC

It is important to underline here, that the novelty of the approach lies in how we distribute the alignment error among all the adjacent photos irrespectively of the distance metrics used between them . Hence, in principle, other similarity metrics between images, such as the NCC, can be used. NCC is usually employed in different ways in many Multi-View Stereo (MVS) matching algorithms (e.g. [21, 5, 25, 49]) to optimize the color projection. In any case, for various reasons the quality of our results is different, and in some cases worse than the results obtained with the MI.

The first problem is that the pre-aligned phase can be only done using the MI, since the combined looks very different from the original image. Despite this, the rendering correlates non-linearly with the input images [13]; the dark parts are dark due to the ambient oc-

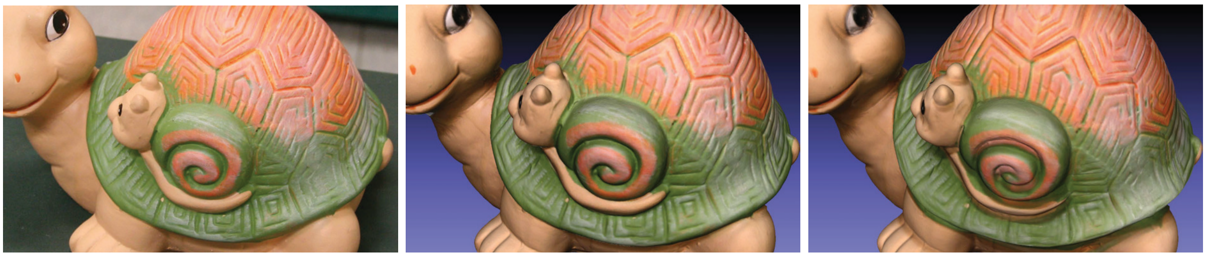


Fig. 9 MI vs NCC evaluation. (Left) Input image (Center) Alignment with the MI. (Right) Alignment with the NCC. Note that the color projection created while using the NCC does not respect the geometric features (note the red pattern near the small slug and the rear leg of the tortoise) due to a slight misalignment of the silhouette of the object.

clusion, normals are correlated with the shading. Hence, we would expect a lower performance using NCC, and preliminary tests confirm this.

The second problem in using the NCC in our optimization framework is that the MI is evaluated between the rendering of the 3D model obtained by projecting the color of several overlapping images and the input image, and not locally as in many MVS approaches. Simply replacing the MI with the NCC RGB (using the formulation proposed in Goesele et al. [25]) produced problems for some datasets. For example, for the Tortoise dataset we obtain a good result in terms of color projections, however the silhouette of the object is not respected (as shown in Figure 9).

In conclusion, although the proposed framework is specifically designed to work with the MI, a detailed comparison of some NCC-based MVS approaches (properly adapted to the framework) would be interesting for future research.

7 Experimental Results

In this section we present several experiments performed on real data to assess the performance of the proposed registration pipeline. The datasets used are listed below, with a brief explanation for each one. The first two are taken from the benchmark datasets proposed by Stretcha et al. [63] to evaluate the performance of multi view reconstruction algorithms. We use the registered cameras provided by these datasets to compare their color projection with our approach.

Fountain-dense This dataset represents a fountain and is made up by eleven 12 MPixel images acquired with a Canon D60 digital camera. The geometry of the model was measured with a Zoller-Frölich LIDAR laser scanner. The software provided by the manufacturer was used to generate the final 3D model. More information regarding the camera calibration (both extrinsic and intrinsic parameters are provided) can be found in the original paper.

Herzjesu-P8 This dataset (Figure 10) represents the facãde of a church and it is composed by eight 6 MPixel images. The model was acquired with the same procedure as the Fountain model.

Shell This dataset regards a small object with highly reflective material. It is composed by 35 (1728 × 1152) images taken with a Canon EOS 350D. The object's geometry has been acquired with a Konica Minolta Vivid 910 laser scanner and the final 3D model assembled using Meshlab-open source software for geometry processing [10].

Tortoise-top This dataset (Figure 10) regards another small object, a painted clay statue of a tortoise. It is made up of 19 (3456 × 2304) images taken with a Canon EOS 350D. These images regard only the top part of the object. The object's geometry was generated in the same way as the Shell dataset.

Tortoise-bottom This dataset is made up of 13 images of the bottom part of the object described above. We separated the photographic campaign of the tortoise into two different datasets to show how to join different image sets to form the final color of the input model. The geometry used for the alignment is the same as the Tortoise-top set.

Neptune Neptune (Figure 10) is a famous statue placed on a large fountain in Piazza della Signoria, in Florence. This object was selected due to its size (more than 5 meters tall) and its shape, which is very complex. This model was generated starting with eight scans acquired with a RIEGL LMSZ390i time-of-flight scanner processed with Riegl software. A Canon EOS 350D was used for the photographic campaign (44 photos at 12 Mpixels).

Duomo This dataset is one of the most ambitious in terms of resources. It regards a huge church located in Piazza dei Miracoli, in the center of Pisa. The full model of the Duomo, made up of about 350 millions of triangles with an accuracy of about 2 cm, was acquired using a Leica time-of-flight scanner. The photographic dataset is made up of 309 (1936 × 1296) images, acquired on a sunny day and depicting only the rear of the Duomo. The results were used to draw the scheme in Figure 1.

As is clear from the description of the datasets used in our experiments, the geometry data came from different devices. The images were also generated by different digital machines with different resolutions. This makes our tests particularly suitable in understanding the applicability of the proposed global alignment pipeline in a real production scenario. In addition, we selected a wide range, from small to big models, from simpler to more complex shapes, thus highlighting that our approach does not rely on assumptions regarding the size and shape of the 3D objects to be aligned. The only requirement is that the object's surface can be sparsely reconstructed by the SFM stage. In the next section, we present the results obtained against the ground truth data mentioned, we evaluate the quality of the colored model produced before and after the global fine alignment and we discuss the processing time required by the different stages.

7.1 Visual Quality Evaluation

In the case of the image-to-geometry alignment, it is difficult to measure the performances of a system, due to the lack of a corresponding measure of the alignment error of the range scans. Moreover, it is also difficult to evaluate against a ground truth: even the data proposed by Strecha did not prove completely reliable in terms of the color projection, especially for the Fountain-dense dataset (see Figure 14). Hence, most of the evaluation lies in the visual quality, although we will present some numerical analysis in the next section. The third column in Figure 10 shows that the point cloud to 3D geometry registration was able to estimate an accurate similarity transformation, thus finding an optimal starting point for the global fine alignment.

In order to test the quality of registration, we projected the set of images on the 3D model using the approach proposed by Callieri et al. [9], which provides a robust framework for projecting an arbitrary number of images onto detailed 3D models. A color is assigned for each vertex as a weighted sum of the contributions of all the images. Please refer to the original paper for further details about this procedure.

The first evaluation was related to the capacity of the global fine alignment to converge to a very good result regardless of the accuracy of the initial alignment. Figure 11 shows the result of the projection of the set of images on the models before and after the global refinement stage in two datasets: Duomo and Fountain. It is evident how severe initial misalignments were recovered during the last stage of our system, which proves robust even when the initial alignment is not accurate.

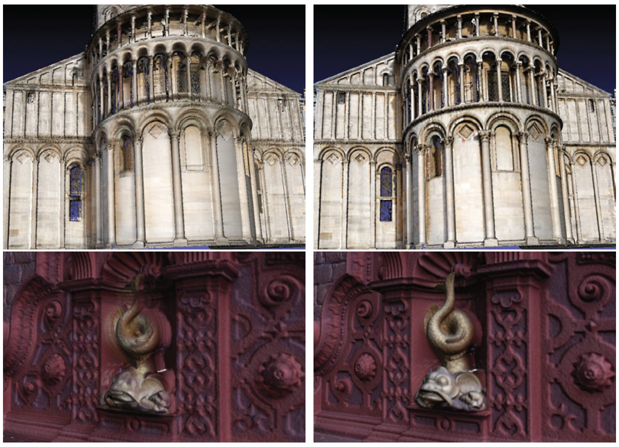


Fig. 11 Some examples of colored models before and after the global fine alignment. (Top Row) Duomo. (Bottom Row) A particular of the Fountain.

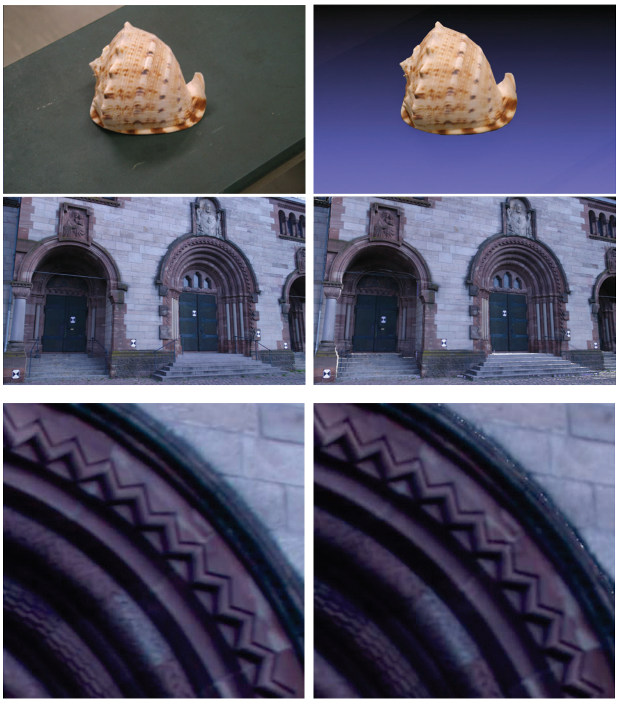


Fig. 12 (First Column) One of the image of the dataset. (Second Column) A rendering from the corresponding point of view of the 3D model.

The second evaluation was intended to analyze the accuracy of the image alignment. The first column of Figure 12 shows an image taken from the Shell and Herzjesu dataset, and there is a rendering of the 3D model with color from the same viewpoint in the second column. The quality of the color is perfectly comparable to the initial images (which were of poor quality, in

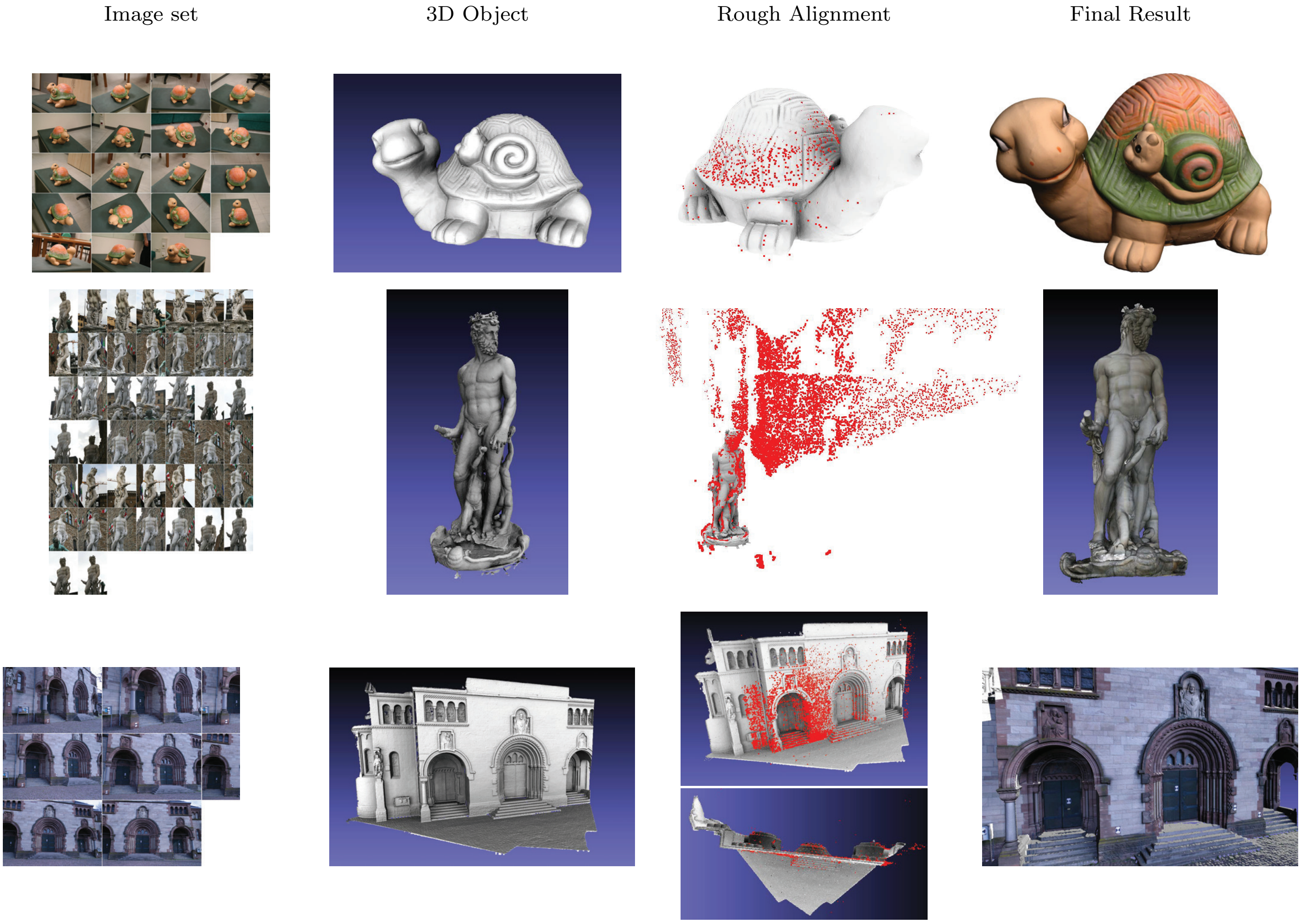


Fig. 10 Some of the dataset used in the experiments and visual results. The output of the second stage is shown in the third column; the cloud point coming from the SFM stage is depicted in red color. The colored model obtained using the calibrated images after the fine alignment is shown in the fourth column.

the case of the Shell). In the detail from Herzjesu, the fine details are also perfectly preserved. This denotes an extremely accurate alignment of the images, due to the fact that one of the limitations of Callieri's approach was that fine details were blurred in the presence of small image misalignments.

Finally, the third evaluation regarded the improvements in global refinement w.r.t. state-of-the-art methods. Figure 13 shows the results of the color projection using the approach by Corsini et al. [13], when each image was aligned on the geometry (first column), and using our global refinement approach (second column). The quality of the projected color shows that the fine details were preserved with greater accuracy (Tortoise dataset, first row). In addition, given the not very accurate geometry and a severely misaligned starting point (Neptune dataset), the single-image geometry-related approach was not able to converge, while the global refinement generated a much better result, and all the images were projected correctly.

7.2 Quantitative Performance Evaluation

In order to quantitatively evaluate the performance of the final alignment, we present two sets of numerical evaluations. First, we provide an evaluation of the accuracy of the camera parameters estimation, then we evaluate the precision of alignment by measuring the color coherency of the projected colors among the images.

This first evaluation can be performed only if ground truth data are available. Hence, as previously mentioned, we used the camera parameters of the Fountain-dense

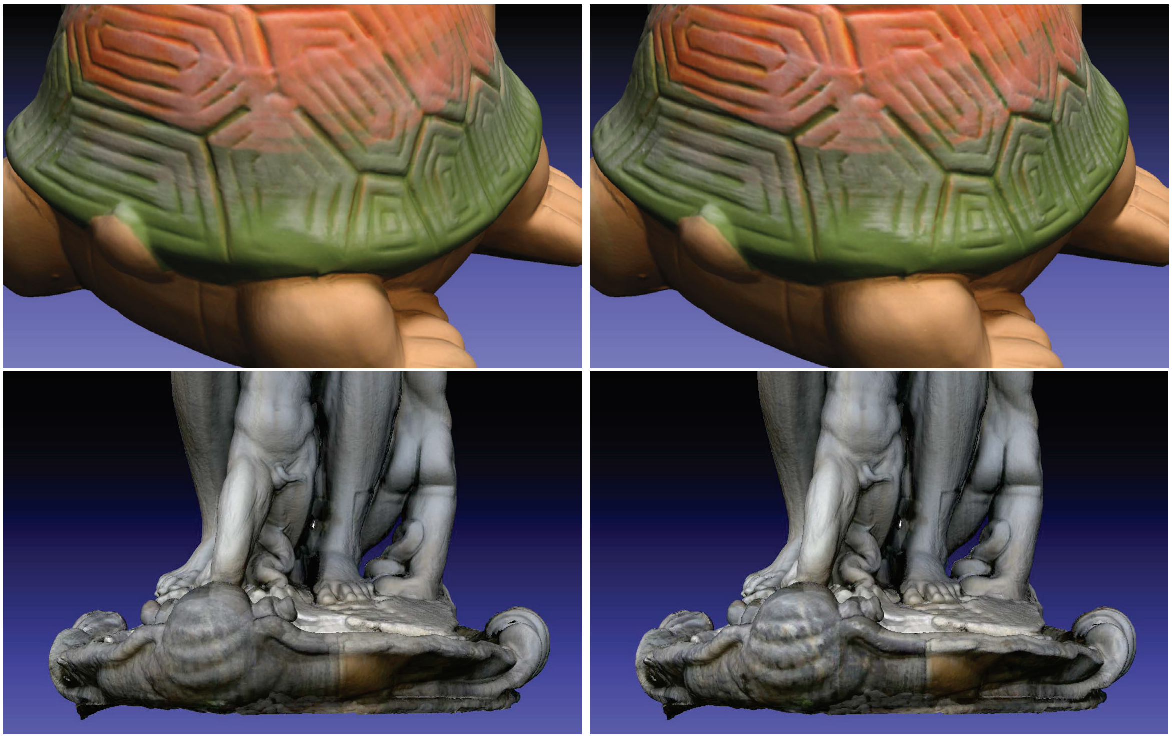


Fig. 13 Color projection after single image-to-3D fine alignment [13] (first column) and using the proposed global refinement algorithm (second column). Note that the color projected using the proposed global refinement algorithm has an increased overall sharpness making the texture details more readable.

and Herzjesu-P8 datasets provided by Stretcha et al. [63]. Unfortunately, when we used the reference camera parameters to project the images on the 3D model, we discovered that the projection was not perfectly accurate (see Figure 14). This is probably due to the fact that also small errors in camera position and orientation can lead to a severe projection misalignment due to the distance of the cameras from the models. We recall that this benchmark is related to multi-view 3D reconstruction and not to texture registration, but this demonstrates the high quality of results that the proposed pipeline is able to achieve.

Table 1 shows the errors with respect to the ground truth data for the results of both the first and the second stages of our method. Three types of errors are reported: the mean of the position error, in terms of the Euclidean distance between the camera position from the ground truth position, the mean orientation error, in terms of the angle between the Z-axis of the ground-truth orientation and the estimated orientation, and the re-projection errors, calculated by re-projecting each vertex of the model in the corresponding image plane using both the ground truth and the estimated camera parameters and calculating the distance in pixels. Analysis of the table shows that the global refinement step reduces the average error by about 50% in terms of the position. Regarding the re-projection error, which seems to result in quite high values, it is necessary to take into account that the final color projection gives a better visual quality compared to the ground truth data (see Figure 14).

The second type of results regards the quality of color projection obtained from the registered image set. Since, to the best of our knowledge, a universally accepted method to numerically evaluate this quality does not exist, for each vertex we calculated the variance of colors projected. We used the following formula:

$$\bar{C} = \sum_{i=1}^{N} I_i(x_p(v), y_p(v)) \tag{8}$$

$$Q_C = \sum_{i=1}^{N} \left(\bar{C} - I_i(x_p(v), y_p(v))\right)^2 \tag{9}$$

where N is the number of images projected onto the vertex v (occlusions are taken into account), and $x_p(v)$, $y_p(v)$ are the pixel coordinates of the image plane obtained by projecting the vertex v onto the image I_i using the i-th camera. This quantity is evaluated for each image channel, hence we have a different quality for the red, green and blue channel. Table 2 gives results of this metric calculated on the datasets used in our experiments. The table shows that the fine alignment step always improves the quality. In addition, when the ground truth was available, the final result was even

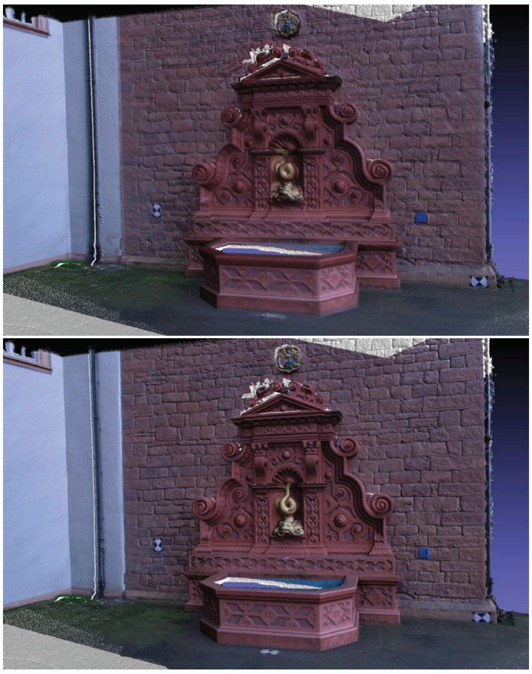


Fig. 14 (Top) Color projection for Fountain dataset, using ground truth camera data. (Bottom) Color projection using the camera data estimated with the proposed method.

better than the reference. This further shows that the ground truth data are probably not accurate enough to be used for color projection.

The quality values are extremely good for the final results in all the cases, except for the Neptune and Duomo datasets, where they are larger than average. For the Neptune dataset, the values of Q_C are quite high compared to the others although the visual results are pleasing. This is because the input images have a very different luminance due to the different position where the pictures were taken. In fact, if we evaluate the variance of mean luminance of each image (in the HLS space), we obtain a value of $L = 671.808$, while the value of L calculated on the projected color in the same way as the Q_C metric is $L = 765.261$, that is close to the luminance variance of the images.

The higher value of the Duomo is essentially due to the low resolution of the photographic dataset (1936×1296), which led to the low quality of fine detail. Moreover, further errors were caused by some misalignments that produced serious changes in the color projections, i.e. the blue background which is sometimes projected in some small parts of the model, and the black-white differences around the columns of the Abside.

7.3 Processing Time

We now present the processing times of the three stages of the proposed pipeline. The tests were performed on an Intel Dual Core 2.33GHz machine, with 6GB of RAM and a NVidia GeForce GTX 260.

For details on the processing time and complexity issues for the SFM stage we refer to the original paper [23]. In any case, the processing time depends mainly on the number of images and to what extent the dendrogram is balanced. In the implementation used, mainly on CPU, the reconstruction got around one minute for the Tortoise-top dataset (19 images only) to the two hours and a half for the Duomo dataset (309 images).

The time required to align the reconstructed point cloud to the input model can vary a lot due to the RANSAC approach. It also depends on the number of vertices of the two point clouds, on the size of the overlapping region between them and on the shape of the object. For example, although the Shell dataset is made up of only a few vertices and with an almost complete overlap between the point clouds (i.e. few missing regions, few spurious data) it required almost 300 minutes because the symmetrical shape of the object made the GPU based rejection test ineffective in distinguishing bad quadruples from good ones. On the other hand much bigger datasets, such as Herzjesu, were completed in less than 60 minutes. The most representative performance index is the time required to estimating the overlap with our GPU based test against the corresponding CPU based Hausdorff distance. The GPU test has a constant of approximately 4 milliseconds, while computing the CPU Hausdorff distance for an average size model such as the Neptune, requires from 3 to 4 seconds. This 10^3 factor is the main ingredient to fight the curse of dimensionality caused by introducing the scale on the 4PCS algorithm.

The time required for the global refinement is partly dependent on the number of input images, but also the overlap between them influences the complexity of the graph. Moreover, the initial misalignment determines the convergence time for each MI maximization due to the steps required by the NEWUOA to converge. For the biggest dataset, the Duomo, this stage completed the final alignment of 309 images in 130 minutes. The Tortoise-top dataset was completed in nearly 12 minutes.

In conclusion, the proposed pipeline automatically manages very complex tasks within a few hours of processing.

Dataset Name	Position error (cm)	Orientation error (degree)	Re-projection error (pixels)
Fountain-dense (before)	11.671	0.248	48.33
Fountain-dense (after)	6.941	0.275	31.3
Herzjesu-P8 (before)	31.126	0.739	40.22
Herzjesu-P8 (after)	15.403	0.558	32.49

Table 1 Position, orientation, and re-projection error before and after the global fine alignment.

Dataset Name	Ground Truth Quality (Q_C)	Rough Alignment Quality (Q_C)	Fine Alignment Quality (Q_C)
Fountain-dense	(151.791, 128.073, 171.849)	(166.925, 151.025, 199.352)	(107.614, 96.8494, 125.24)
Herzjesu-P8	(137.172, 133.097, 181.063)	(194.082, 201.918, 249.1)	(128.719, 128.96, 163.468)
Shell	n.a.	(327.643, 262.55, 286.673)	(298.344, 256.597, 290.528)
Tortoise-top	n.a.	(531.674, 355.197, 364.923)	(343.254, 256.33, 262.746)
Tortoise-bottom	n.a.	(1310.46, 637.017, 572.498)	(362.296, 240.542, 224.799)
Neptune	n.a.	(993.332, 1028.991, 1167.722)	(706.911, 735.617, 846.415)
Duomo	n.a.	(1799.06, 1763.01, 1734.5)	(1033.62, 998.716, 956.778)

Table 2 Quality of the 2D/3D registration before and after the fine alignment stage. Color variance of the incident pixels coming from different photos is used as quality factor. The mean of $\sigma_r, \sigma_g, \sigma_b$ computed on the model surface is reported.

8 Discussion

The results on the above datasets highlight the strenghts of the proposed method. In particular, it is:

- **General**: there is no strong assumption regarding the object, since no particular feature, size or shape is taken into account. This enables very different cases to be handled, from small statues to entire buildings.
- **Automatic**: the procedure requires no user intervention. The total processing time is in the order of a few hours even for very complex cases.
- **Robust and flexible**: the method can be adapted for non-ideal cases, where the images and/or the 3D model are not fully reliable. The point cloud generated by SFM is used only for the initial registration, and the 3D model is used only as a ”medium” during the global refinement. The alignment thus tries to find the solution that fits the input data, even such data are not of a high quality.

A further benefit is that the last part of the system can be applied to photo sets from different SFM reconstructions. Figure 15 shows the results of merging of the Tortoise-top and Tortoise-bottom sets, which could not be processed together because the object had to be turned on the table to acquire different parts of the surface. Starting from the two initial alignments provided by the second step in the system, and applying the global refinement to all the images at the same time, complete coverage of the object is achieved. Overlapping parts of images from different datasets contributed in obtaining an almost perfect alignment.

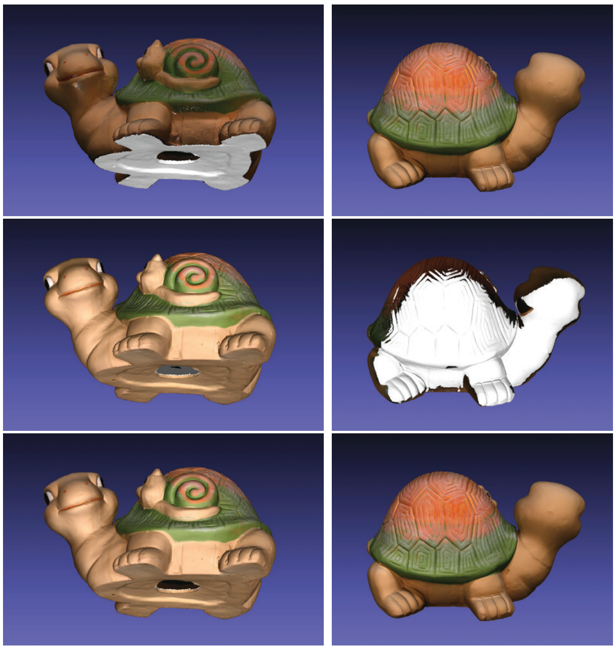


Fig. 15 (First Row) Tortoise-top colored model. (Second Row) Tortoise-bottom colored model. (Third Row) The result of the global alignment applied on the two merged photo sets.

Since MI is very robust for correlating images with very different visual appearances, it is reasonable to argue that the proposed pipeline can be used to map sets of images with very different lighting conditions. This can be used for relighting large structures, such as buildings or plazas, as a matter of interesting future applications (for example taking inspiration from approaches such as the Polynomial Texture Maps [42]).

Despite all these advantages, some limitations still remain.

One limitation is the generation of the point cloud through the SFM stage. In some cases the photos may not be suitable for an image-based reconstruction, in which case nothing can be done. This depends both on how the photos were taken and on the reflectance properties of the material of the object. For example, an object made with very reflective materials (i.e. a skyscraper full of glass) or an object made by transparent materials cannot be reconstructed effectively. Another potential problem is related to the cloud point alignment stage; since it is based on a RANSAC scheme, the processing time required for the alignment is not easily predictable and can vary a lot even in the case of similar datasets. Moreover, in some cases, such as the Tortoise dataset where the shape is close to a sphere, the transformation could be wrongly calculated. This can be solved by simply re-launching the pipeline, skipping the SFM stage.

The fine alignment step is very robust, and does not suffer from particular limitations. Even if it is true that the geometry can not correlate well with the images, so that hybrid approaches could be necessary [60], the use of the information coming from many incident images on the same parts of the surface prevents alignment problems. In other words it is very unlikely that the final alignment will be worse than the initial alignment, provided that the initial alignment is near to the best solution. In the case of severe misalignments of a group of images, or in the presence of strong repeating patterns, a group of wrongly aligned images may cause the other images to give an inaccurate result. However this should only happen in very particular cases, which would be extremely challenging regardless of what approach was chosen.

9 Conclusions

We have presented a global 2D/3D registration pipeline for the simultaneous alignment of an image set on a 3D object acquired through laser scanning. The main application of this pipeline is in the context of photorealistic 3D object acquisition, for color mapping or for the image-based estimation of surface properties.

The main advantage of the proposed approach is its generality since no particular assumption regarding the size and the shape of the object is necessary. The only requirement is that it must be possible to even sparsely reconstruct the object starting from a set of images with standard SFM algorithms. The proposed pipeline is also very flexible thanks to the combination of the hierarchial SFM chosen and the cloud point alignment stage, which also handles cases where the reconstruction is a sub-set or a super-set of the 3D object to be aligned. In addition, the processing time is relatively low, taking into account the complexity of the problem and the amount of data to be processed. To conclude, the proposed pipeline is able to obtain high quality color mapping results, as demonstrated by the experiments presented on real datasets.

An interesting direction for future research would be to add an analysis step, in the pre-alignment phase, to remove the images with high misalignments, since such images can compromise the quality of the final alignment. Approaches involving small warping of the images [16,15] could further improve the quality of the projected color, and thus more accurately preserve the fine details. In addition, the SFM stage could be replaced with a dense multi-view reconstruction algorithm. In some cases, this should improve the performance of the subsequent stages. This replacement could also be used in a more interesting way, i.e. to enrich the geometric details of the 3D model by integrating a dense reconstruction and scanned geometry, in those areas where the geometry acquired is missing or approximate, for example due to the difficulties in positioning the scanner with respect to the object.

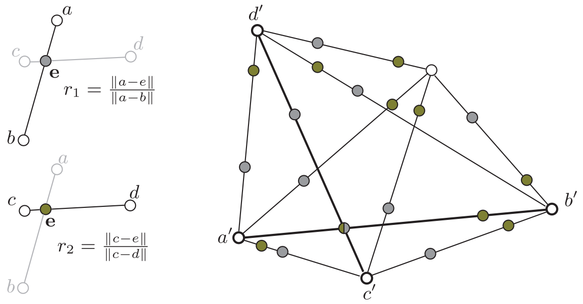
Acknowledgements This research work was partly funded by the EU Community FP7 ICT under the V-MUST.net project (Grant Agreement 270404). We would also like to thank the anonymous reviewers for their feedback which enabled us to improve the final version of the paper.

References

1. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust pairwise surface registration. ACM Trans. Graph. **27**, 85:1–85:10 (2008)
2. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM **45**, 891–923 (1998)
3. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
4. Bonarrigo, F., Signoroni, A.: An enhanced 'optimization-on-a-manifold' framework for global registration of 3D range data. In: Proc. of 3DIMPVT '11, pp. 350–357. IEEE Computer Society, Washington, DC, USA (2011)
5. Bradley, D., Boubekeur, T., Heidrich, W.: Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008), pp. 1–8 (2008)
6. Brown, M., Lowe, D.: Recognising panoramas. In: Proc. Int. Conf. Computer Vision, vol. 2, pp. 1218–1225 (2003)
7. Brown, M., Lowe, D.G.: Unsupervised 3D object recognition and reconstruction in unordered datasets. In: Proc. Int. Conf. on 3D Digital Imaging and Modeling (2005)

8. Brunie, L., Lavallée, S., Szeliski, R.: Using force fields derived from 3D distance maps for inferring the attitude of a 3D rigid object. In: Proc. of the Second European Conference on Computer Vision (ECCV'92), pp. 670–675. Springer-Verlag (1992)
9. Callieri, M., Cignoni, P., Corsini, M., Scopigno, R.: Masked photo blending: mapping dense photographic dataset on high-resolution 3D models. Computer & Graphics **32**(4), 464–473 (2008)
10. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: Meshlab: an open-source mesh processing tool. In: Sixth Eurographics Italian Chapter Conference, pp. 129–136. Eurographics (2008)
11. Cleju, I., Saupe, D.: Stochastic optimization of multiple texture registration using mutual information. In: Proceedings of the 29th DAGM conference on Pattern recognition, pp. 517–526. Springer-Verlag (2007)
12. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. ACM Trans. Graph. **23**, 905–914 (2004)
13. Corsini, M., Dellepiane, M., Ponchio, F., Scopigno, R.: Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. Computer Graphics Forum **28**(7), 1755–1764 (2009)
14. Dellepiane, M., Callieri, M., Ponchio, F., Scopigno, R.: Mapping highly detailed colour information on extremely dense 3d models: The case of david's restoration. Computer Graphics Forum **27**(8), 2178–2187 (2008)
15. Dellepiane, M., Marroquim, R., Callieri, M., Cignoni, P., Scopigno, R.: Flow-based local optimization for image-to-geometry projection. Visualization and Computer Graphics, IEEE Transactions on **18**(3), 463 –474 (2012)
16. Eisemann, M., De Decker, B., Magnor, M., Bekaert, P., de Aguiar, E., Ahmed, N., Theobalt, C., Sellent, A.: Floating textures. Computer Graphics Forum (Proc. of Eurographics) **27**(2), 409–418 (2008)
17. Farenzena, M., Fusiello, A., Gherardi, R.: Structure-and-motion pipeline on a hierarchical cluster tree. In: IEEE Int. Workshop on 3-D Digital Imaging and Modeling. Kyoto, Japan (2009)
18. Fitzgibbon, A.W., Zisserman, A.: Automatic camera recovery for closed and open image sequencese. In: Proc. Europ. Conf. Computer Vision (ECCV1998), pp. 311–326 (1998)
19. Franken, T., Dellepiane, M., Ganovelli, F., Cignoni, P., Montani, C., Scopigno, R.: Minimizing user intervention in registering 2D images to 3D models. The Visual Computer **21**(8-10), 619–628 (2005)
20. Früh, C., Zakhor, A.: Constructing 3D city models by merging aerial and ground views. IEEE Computer Graphics and Applications **23**, 52–61 (2003)
21. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)
22. Gehua Yang, G.Y., Becker, J., Stewart, C.V.: Estimating the location of a camera with respect to a 3d model. In: Proc. of the Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM2007), pp. 159–166. IEEE Computer Society (2007)
23. Gherardi, R., Farenzena, M., Fusiello, A.: Improving the efficiency of hierarchical structure-and-motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), pp. 1594–1600 (2010)
24. Gibson, S., Cook, J., Howard, T., Hubbold, R., Oram, D.: Accurate camera calibration for off-line, video-based augmented reality. Mixed and Augmented Reality, IEEE / ACM Int. Symp. on (2002)
25. Goesele, M., Curless, B., Seitz, S.M.: Multi-view stereo revisited. In: Proc. of CVPR '06, vol. 2, pp. 2402–2409. IEEE Computer Society (2006)
26. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Am. A **4**(4), 629–642 (1987)
27. Ikeuchi, K., Nakazawa, A., Hasegawa, K., Ohishi, T.: The great buddha project: Modeling cultural heritage for vr systems through observation. In: Proc. of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'03), pp. 7–. IEEE Computer Society (2003)
28. Irschara, A., Zach, C., Bischof, H.: Towards wiki-based dense city modeling. In: Proc. Int. Conf. Computer Vision (ICCV2007), pp. 1–8 (2007)
29. Johnson, A.: Spin-images: A representation for 3-d surface matching. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997)
30. Kalogerakis, E., Nowrouzezahrai, D., Simari, P., Singh, K.: Extracting lines of curvature from noisy point clouds. Comput. Aided Des. **41**(4), 282–292 (2009)
31. Kamberov, G., Kamberova, G., Chum, O., Obdrzalek, S., Martinec, D., Kostkova, J., Pajdla, T., Matas, J., Sara, R.: 3D geometry from uncalibrated images. In: Proc. 2nd Int. Symp. on Visual Computing (2006)
32. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: A. Sheffer, K. Polthier (eds.) Eurographics Symposium on Geometry Processing (SGP2006), pp. 61–70. Eurographics Association, Cagliari, Sardinia, Italy (2006)
33. Krishnan, S., Lee, P.Y., Moore, J.B., Venkatasubramanian, S.: Global registration of multiple 3D point sets via optimization-on-a-manifold. In: Proc. of the 3rd Eurographics symposium on Geometry processing (SGP2005). Eurographics Association (2005)
34. Krishnan, S., Lee, P.Y., Moore, J.B., Venkatasubramanian, S.: Optimisation-on-a-manifold for global registration of multiple 3D point sets. Int. J. Intell. Syst. Technol. Appl. **3**(3/4), 319–340 (2007)
35. Lensch, H.P.A., Heidrich, W., Seidel, H.P.: Automated texture registration and stitching for real world models. In: PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications, p. 317. IEEE Computer Society (2000)
36. Li, X., Guskov, I.: Multi-scale features for approximate alignment of point-based surfaces. In: Proc. of the 3rd Eurographics symposium on Geometry processing (SGP2005). Eurographics Association (2005)
37. Liu, L., Stamos, I.: Automatic 3D to 2D registration for the photorealistic rendering of urban scenes. In: CVPR, vol. 2, pp. 137–143. IEEE Computer Society (2005)
38. Liu, L., Stamos, I., Yu, G., Wolberg, G., Zokai, S.: Multiview geometry for texture mapping 2D images onto 3D range data. In: Proc. of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp. 2293–2300. IEEE Computer Society (2006)
39. Lowe, D.G.: Fitting parameterized three-dimensional models to images. IEEE Trans. Pattern Anal. Mach. Intell. **13**, 441–450 (1991)
40. Maes, F., Collignon, A., Vandeermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. IEEE Transactions in Medical Imaging **16**, 187–198 (1997)

41. Makadia, A., Patterson, A., Daniilidis, K.: Fully automatic registration of 3d point clouds. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1297–1304 (2006)
42. Malzbender, T., Gelb, D., Wolters, H.: Polynomial texture maps. In: SIGGRAPH '01, pp. 519–528. ACM (2001)
43. Matsushita, K., Kaneko, T.: Efficient and handy texture mapping on 3D surfaces. Computer Graphics Forum **18**(3), 349–358 (1999)
44. Neugebauer, P.J., Klein, K.: Texturing 3D models of real world objects from multiple unregistered photographic views. Computer Graphics Forum **18**(3), 245–256 (1999)
45. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3D reconstruction. In: Proc. Int. Conf. Computer Vision, pp. 1–8 (2007)
46. Nistér, D.: Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In: Proc. Europ. Conf. Computer Vision (ECCV2000), pp. 649–663 (2000)
47. Pintus, R., Gobbetti, E., Combet, R.: Fast and robust semi-automatic registration of photographs to 3D geometry. In: The 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (2011). To appear
48. Pluim, J., Maintz, J., Viergever, M.: Mutual-information-based registration of medical images: a survey. IEEE Transactions on Medical Imaging **22**(8), 986–1004 (2003)
49. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Int. J. Comput. Vision **72**(2), 179–193 (2007)
50. Portelli, D., Ganovelli, F., Tarini, M., Cignoni, P., Dellepiane, M., Scopigno, R.: A framework for user-assisted sketch-based fitting of geometric primitives. In: Proceedings of WSCG, the 18th Int. Conference on Computer Graphics, Visualization and Computer Vision (2010)
51. Pottmann, H., Huang, Q.X., Yang, Y.L., Hu, S.M.: Geometry and convergence analysis of algorithms for registration of 3d shapes. Int. J. Comput. Vision **67**(3), 277–296 (2006)
52. Powell, M.J.D.: Developments of NEWUOA for minimization without derivatives. IMA Journal of Numerical Analysis **28**(4), 649–664 (2008)
53. Pulli, K.: Multiview registration for large data sets. In: Proc. of the 2nd international Conference on 3-D digital imaging and modeling (3DIM'99), pp. 160–168. IEEE Computer Society, Washington, DC, USA (1999)
54. Pulli, K., Abi-Rached, H., Duchamp, T., Shapiro, L.G., Stuetzle, W.: Acquisition and visualization of colored 3D objects. In: Proc. of the 14th International Conference on Pattern Recognition (ICPR'98), vol. 1, pp. 11–. IEEE Computer Society (1998)
55. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Proc. of the Third International Conference on 3-D Digital Imaging and Modeling, pp. 145–152 (2001)
56. Sequeira, V., Goncalves, J.G.: 3D reality modelling: Photo-realistic 3d models of real world scenes. 3D Data Processing Visualization and Transmission, International Symposium on **0**, 776 (2002)
57. Shum, H.Y., Ke, Q., Zhang, Z.: Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In: Proc. Int. Conf. Computer Vision and Pattern Rec. (1999)
58. Skelly, L., Sclaroff, S.: Improved feature descriptors for 3-D surface matching. In: Proc. SPIE Conf. on Two- and Three-Dimensional Methods for Inspection and Metrology V, vol. 6762 (2007)
59. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: SIGGRAPH'06, pp. 835–846 (2006)
60. Sottile, M., Dellepiane, M., Cignoni, P., Scopigno, R.: Mutual correspondences: an hybrid method for image-to-geometry registration. In: Eurographics Italian Chapter Conference 2010, pp. 81–88. EG (2010)
61. Stamos, I., Liu, L., Chen, C., Wolberg, G., Yu, G., Zokai, S.: Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes. Int. J. Comput. Vision **78**, 237–260 (2008)
62. Steedly, D., Essa, I., Dellaert, F.: Spectral partitioning for structure from motion. In: Proc. Int. Conf. Computer Vision (ICCV2003), pp. 649–663 (2003)
63. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR'08, pp. 1 –8 (2008)
64. Vergauwen, M., Gool, L.V.: Web-based 3D reconstruction service. Machine Vision and Applications **17**(6), 411–426 (2006)
65. Viola, P., William M. Wells, I.: Alignment by maximization of mutual information. Int. J. Computer Vision **24**(2), 137–154 (1997)
66. Wu, C., Clipp, B., Li, X., Frahm, J.M., Pollefeys, M.: 3d model matching with viewpoint-invariant patches (vip). In: CVPR'08. IEEE Computer Society (2008)
67. Zhao, W., Nister, D., Hsu, S.: Alignment of continuous video onto 3d point clouds. Pattern Analysis and Machine Intelligence, IEEE Transactions on **27**(8), 1305–1318 (2005)
68. Zheng, H., Cleju, I., Saupe, D.: Highly-automatic mi based multiple 2D/3D image registration using self-initialized geodesic feature correspondences. In: H. Zha, R. ichiro Taniguchi, S.J. Maybank (eds.) ACCV (3), *Lecture Notes in Computer Science*, vol. 5996, pp. 426–435. Springer (2009)


a
c
d
e
$r_1 = \frac{\|a-e\|}{\|a-b\|}$
b
a
c
d
e
$r_2 = \frac{\|c-e\|}{\|c-d\|}$
b
d'
b'
a'
c'

$B_p = (a, b, c, d)$
selected quadruple in
P

$B_q = (a', b', c', d')$
candidates congruent
quadruples in
Q

Fig. 16 Quadruple characterization in 4PCS algorithm.

Appendix A

A coplanar quadruple (a, b, c, d) is expressed as the combination of the two segments $s_1 = (a, b)$ and $s_2 = (c, d)$ and characterized by their length $d_1 = \|a - b\|$ and

$d_2 = \|c - d\|$. Since the segments are coplanar, they will meet at an *intermediate point* e, so the quadruple is further characterized by the points along the two segments where they meet, i.e. ratios:

$$r_1 = \|a - e\| / \|a - b\| \quad (10)$$

$$r_2 = \|c - e\| / \|c - d\| \quad (11)$$

Aiger et al. [1] use this simple characterization of a quadruple to prune the number of possibly congruent quadruples on $\mathbf{Q}$ and hence to speed up the process. Consider a couple in $\mathbf{Q}$ as the segment (q_1, q_2) of a candidate congruent quadruple (q_1, q_2, q_3, q_4). If this quadruple is congruent to (a, b, c, d) it means that if we generate the intermediate points on (q_1, q_2) and (q_3, q_4) with the ratios r_1 and r_2 they will coincide, because affine transformations preserve ratio of the distances. Their approach consists in generating the intermediate points for all the segments in $\mathbf{Q}$ and inserting them into a range-query data structure [2]. This can be built in $O(k \log k)$ time and accessed in $O(\log k)$ (where k is the number of the intermediate points) Then, they only test the quadruples made of two segments which intermediate points coincide. If we consider all the couples in $\mathbf{Q}$ as potential segments of a congruent quadruple k is $O(n^2)$, but if we restrict the choice to a couple of points at a distance d_1 or d_2 from each other and assume a uniform distribution of points k will be $O(n)$. Therefore they have $O(n^2)$ for finding $O(n)$ segments/intermediate points plus $O(n \ \log n)$ for building and accessing the search data structure, and hence the global complexity is $O(n^2)$.

In principle, we could almost apply it to our case without any change, simply by including scaling in computing the transformation between candidate congruent quadruples. The problem with this is that we cannot limit the list of couples in $\mathbf{Q}$ to those within a distance d_1 or d_2 because there is a unknown scale factor between $\mathbf{Q}$ and $\mathbf{P}$. Therefore the order of magnitude of k is $O(n^2)$ and the global complexity becomes $O(n^2 \log n)$. Although the total asymptotic complexity raised "only" by a factor $\log n$ the actual time for computing the result becomes very large. This is because the number of quadruples to be tested for affinity with the base in $\mathbf{P}$ raised by $O(n)$ to $O(n^2)$ and the cost of evaluating the transformation between two quadruples is high.