Automatic mapping of uncalibrated pictures on dense 3D point clouds

Davide Davelli, Alberto Signoroni
Università degli Studi di Brescia
Dipartimento di Ingegneria dell’Informazione
via Branze 38, I25123, Brescia, Italy
davellidavide@gmail.com, alberto.signoroni@ing.unibs.it

Abstract—This paper presents a solution for the automated texturing of models obtained from 3D scanners by using uncalibrated pictures taken with a digital camera. The proposed method exploits calibrated light intensity images given by the scanning device to automatically extract a set of correspondences with the uncalibrated pictures, such correspondences are used to give an accurate estimate for the camera pose. In a second phase the set of calibrated pictures is projected onto the model to build the texture, with a proper merging of overlapping images. The image blending is carried out by weighting the pixels of each picture depending on their depth and normal orientation. Results are proposed for sculpture datasets where a method to assess the accuracy of the mapping is applied.

Index Terms—cultural heritage, 2D-3D registration, camera calibration, point cloud coloring, photo blending

I. INTRODUCTION

In the last few years, 3D scanning technology has set a new standard in the Cultural Heritage field, allowing the acquisition and digitization of artifacts with reducing costs and efforts [1]. Usually, professional 3D scanning devices are designed to only acquire the geometry of an object. In the cultural heritage field, however, it is often important to give a photorealistic representation of the scanned object. It is then necessary to merge high resolution geometrical data with high resolution and high colorimetric quality textures [2]. A way to acquire color information could be the use of the optical sensors mounted on 3D scanners, taking advantage of an intrinsic calibration with the acquired depth maps. However, this method has several drawbacks. Firstly the cameras mounted on structured light and laser scanners have not the same quality when compared to professional digital cameras. Secondly, photographic campaigns usually take place in environments with controlled lighting, which usually is diffuse and intense; such light can interfere with the acquisitions of the scanner. Another reason is the fact that the users of scanning devices are not necessarily experts in digital photography and viceversa, so that the presence of different expertises are likely to be required. Lastly, a digital camera offers greater flexibility: it allows the easy acquisition of many views, whereas, 3D scans generally require more handwork to be carried out. The high degree of automation we obtain and the ability to work with a set of pictures taken from photo services that fully cover the artworks from freely chosen viewpoints, allow us to propose a solution which is a step further with respect to techniques that approached the same kind of problem [3], [4], [5]. For all these reasons, it seems to be advisable and more appropriate to carry out two separate acquisition campaigns using high-end 3D scanners and professional cameras for acquiring the geometry and the color texture of the artworks respectively. This assumption also allows us to offer the same solution in most general cases, where the two campaigns are conceived and done independently (i.e. one pre-exists at the time when the other is planned and carried out). Thus, the problem we face is the mapping of a set of uncalibrated pictures on a 3D model and we want to do this in an automatic way (see the graphical abstract of Fig.1). To solve this problem we propose to exploit the light intensity images obtained from a structured light scanner and epipolar geometry constraints to perform feature-based automatic image alignments. This allows to project color pictures onto a point cloud model where a texture is then built merging the contributions of the photographic images. The exploitation of additional information obtained from the scanning device allows to follow an alternate approach with respect to the one which tries to build a sparse model using a Structure From Motion (SfM) method [6] and subsequently to align it to the dense model produced by the scanner. With respect to this approach, our method uses an higher amount of information, also serviceable to decrease the overall computational complexity.
II. RELATED WORK

The problem of exploiting information from pictures to build or improve 3D models of the represented scene has always been a topic of great interest for the Computer Vision community. As mentioned, one of the most studied subjects is the one that tries to extract 3D modeling information from a group of pictures, where the Structure From Motion method is the most popular approach. SfM and like-minded approaches [6] do not require a-priori 3D information but, as a drawback, they can be computationally demanding and generally produces not completely reliable and/or sparse 3D models. A general pipeline which allows to combine SfM inferred 3D models and color information with more detailed geometry information (e.g. coming from aligned point clouds or preexisting 3D models) has been recently proposed [2].

When 3D geometry information is available, many approaches and techniques for the registration of 2D images to 3D models have been developed over the years. A family of works such as [7], [8], [9], [10] and [11] propose methods for camera calibration and pose estimation from a set of correspondences between geometric visual cues such as lines, vanishing lines, circles, t-sections and rectangular shapes. These methods can be very effective in registering pictures of urban and architectural environments where such shapes can be easily found, but fail in registering images on objects, such as sculptures, that do not have regular geometric features. In the latter case such methods may require the placements of landmarks on the object or near it in order to be able to extract enough correspondences. Other possible approaches are the ones in which a set of punctual correspondences is set between the images and the 3D model and subsequently camera calibration algorithms, such as DLT [6] or Tsai’s [12], are applied. In the case studied here, where images uncalibrated to the model are registered by exploiting the presence of some calibrated pictures, some solution have already been proposed. In [3] and [4], a set of correspondences between the two sets of images is manually given in order to calibrate the pictures to be mapped, and the calibrated images are blended to form a texture. Also in [13] color accumulation through multi-band blending assumes a series of already calibrated pictures. In this work, we propose a 2D-3D registration method between high quality photo and 3D point cloud which is capable of automatically extract the correspondences by bridging on the color information coming from the 3D scanner. This differs from what proposed in [5], where an automatic approach for correspondences extraction is applied to single range scans and single pictures. The proposed method allows the registration of multiple pictures which are used to attribute a color to the point of the model by weighted color blending and point visibility understanding. Automated geometry aware camera calibration is also meaningful for image localization [14].

III. AUTOMATIC 2D-3D REGISTRATION

In order to be as flexible as possible, the algorithm has been subdivided in three well separated parts. Figure 2 show an high level block diagram of the algorithm, highlighting its three main phases. The algorithm uses the following data:

- 3D model of the object (sculpture) in a point cloud format;
- Collection of 3D scans of the model as range images;
- Photographic pictures of the object taken with a Nikon D70 camera (from now on camera pictures/images);
- Pictures of the sculpture taken with the imaging apparatus of the 3D scanner (from now on scanner pictures/images).

![Fig. 2. Block scheme of the algorithm](image1)

A. Correspondences setting

The presented method for automatic correspondences setting exploits the fact that scanner pictures and depth images are aligned by construction, since they are taken by the same sensor. Therefore, for each pixel we get six values: the $x$, $y$ and $z$ coordinates of the point and its color values. Note that, when the scanner failed to retrieve a measure for the point, some pixels may contain only color values. We set correspondences between camera pictures and scanner pictures, and the latter will give 3D coordinates of the position of the pixel on the model. In order for the processing chain to be consistent, all the depth images are be aligned among themselves. We assume that a point cloud 3D model has already been built starting from the single depth images. The automatic correspondence phase is based on the automatic features extraction and is composed by the following steps: 1) preprocessing, 2) features extraction and description from camera picture and from scanner picture and 3) features matching and matches validation. Figure 3 shows a block scheme of the automatic correspondence setting algorithm.

![Fig. 3. Block scheme of the automatic correspondences setting algorithm](image2)
1) Pre-processing: The pre-processing consists in the conversion to grayscale of the two images, followed by a low-pass filtering (smoothing) to mitigate Gaussian noise in the images, which may lead to bad matches in later steps. The camera picture is also subsampled, in order to have resolution comparable to the one of the scanner image. The subsampling also greatly reduces the computation time of the entire elaboration chain. In order to enhance the features extraction step, the pixels on the scanner images that have an invalid 3D coordinate are masked with 0, since it would be useless to match these points since they do not have an associated depth image 3D point.

2) Feature detection and description: For the feature detection and description, Lowe’s SIFT keypoints [15] has been used. This choice was based on two principles: firstly SIFT has the property to extract point-type features (also referred ad keypoints), which in our case can be directly used as calibration points for the camera calibration phase; secondly, as the acronym suggests, they are invariant (up to a certain degree) to some transformation such as scaling, rotations and illumination changes. Note that SIFT (as well as other similar keypoint features), are robust to rotations around the z axis of the camera reference system, but they are much more sensible to rotations of the image plane around the other two axes (changes of perspective). Such sensitivity is emphasized when the image subject has a highly variable geometry, i.e. a complex three-dimensional structure, so that even a small perspective change can lead to local great changes in the imaged shape of the object (and possibly large variations of its projected shadows).

3) Features matching: Given a set of descriptors for a couple of images (camera image and scanner image), they can be mutually matched by finding, for each point of interest, the keypoints in the other image type that minimizes the Euclidean distance between the respective descriptors (represented as 128-dimensional vectors). For every keypoint on the first image, a k-nearest neighbors algorithm is preformed to find the 2 nearest keypoint in the second image, and viceversa. In this way we can suppress matches that could be regarded as possibly ambiguous, by applying the principle suggested by Lowe in [16] to reject matches which ratio between the distances of the nearest and the next nearest match is more than 0.8. A second step is the matches validation. Since we are comparing different views of the same subject, the matching points, in order to be correct, must satisfy an underlying geometric structure.

Two different views of the same object can be seen as having two different cameras (i.e. image planes) looking at some of the same points in the scene. Exploiting this consideration, we can use the epipolar geometry model, that is more robust than the homographic one when validating our kind of matches. In epipolar geometry, the fundamental matrix $F$ describes the relation between the same point seen by two cameras. If a point $X$ in 3D space is projected as $x$ in the first view, and as $x'$ in the second, then there exists a matrix $F$ so that $x, x'$ and $F$ satisfy the relation $x'^T F x = 0$. The matches are validated by executing the RANSAC algorithm [17], fitting the points to a epipolar geometry model (i.e. finding the fundamental matrix $F$), and discarding the outliers. In order to get the maximum amount of correspondences, each camera picture is matched with all the scanner pictures. Such procedure is quite demanding in terms of computational time: to speed up the process the extracted keypoints and descriptors of the scanner images have been saved as files. In this way they must be computed only at the first iteration, doubling up the speed of the feature extraction and matching steps in the next iterations.

B. Camera calibration

From the previous step, we obtain a set of 2D-2D correspondences between camera image and scanner image. The points on the scanner image locate points on the domain (grid) of the depth image, directly related to the model points. With this process we can retrieve a set of 2D-3D correspondences between camera pictures and 3D model points, which can be fed to a camera calibration algorithm, which allows the estimate of the $3 \times 4$ camera matrix $P$ describing the projection of 3D points on the image plane. The algorithm adopted for calibration is the normalized DLT with geometric error minimization [6]. With this, the following steps must be accomplished:

1. Translate and scale image and object points so that their centroid is at the origin and their RMS from the origin is $\sqrt{2}$ and $\sqrt{3}$ respectively. Let the normalized image points be $\tilde{x}_i = T x_i$, and the normalized space points be $\tilde{X}_i = U X_i$, the matrices $T$ and $U$ being defined as follows:

$$T = \begin{bmatrix} s & 0 & -s \cdot c_x \\ 0 & s & -s \cdot c_y \\ 0 & 0 & 1 \end{bmatrix}, U = \begin{bmatrix} S & 0 & -S \cdot C_z \\ 0 & S & -S \cdot C_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $s, S, c$ and $C$ are the scaling factor and the centroid for the image points and object points respectively.

2. Each correspondence generates a constraint for the system:

$$\begin{bmatrix} M_i^T & 0 & -u_i M_i^T \\ 0 & -M_i^T & v_i M_i^T \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = 0$$

where $M_i$ is the vector of homogeneous coordinates of a point in the 3D space, $u_i$ and $v_i$ are the coordinates of the projection of $M_i$ on the image plane and $p_1$, $p_2$, $p_3$ are the rows of the camera matrix $P$ that must be estimated. Given a set of $n$ correspondences, a $2n \times 12$ matrix $A$ can be built. The least-square solution of $A p = 0$, subject to $\|p\| = 1$, is obtained from the unit singular vector of $A$ corresponding to the smallest singular value. The vector $p$ contains the 12 elements of $P$ listed by rows.

3. Using the linear estimate as a starting point, minimize the geometric error

$$\sum_{i=0}^{n} d(m_i, PM_i)^2$$

where $m_i$ is the homogeneous 3 elements vector of coordinates representing the projection of $M_i$ on the image plane.
4. The camera matrix $P$ for the unnormalized coordinates is obtained from the camera matrix $\tilde{P}$ of the normalized coordinates, as $P = T^{-1}\tilde{P}U$.

C. Model texturing

Using the matrix $P$ obtained from the camera calibration phase, the 3D object points can be projected on the image plane by multiplying their homogeneous coordinates vector on the left by $P: x = PX$.

The operation of projection from a three dimensional space to a 2D space (the image plane), comes with the loss of the depth due to the reduction in the space dimensions: points that should have been occluded in a 3D space are actually projected on the image plane. In our case we are working on a point cloud, which lacks of the topology of mesh surfaces, so that when projecting points on the image plane we must have a procedure that allow us to detect occluded points. A solution method we envisage is articulated as follows:

1. Candidates to be projected are only those points which normal has negative inner product with respect to the image plane normal (green points in Figure 4).
2. Considering all candidates: build a depth map by projecting each candidate on the image plane. If more than one point are projected on the same pixel, keep only the point nearest to the camera centre (blue lines in Figure 4).
3. Perform a neighbourhood search of $9 \times 9$ pixels in the depth map and, for every neighbour, if the depth difference between the pixel and its neighbour is higher than a given threshold, discard the neighbour.

The last step is necessary due to the discrete nature of the point cloud model; points which normal has negative inner product with respect to the image plane normal, but are lying behind other points are also projected since the ray connecting them to the camera centre is actually passing through a "hole" between the foreground points (yellow points in Figure 4). When projecting a group of pictures on the same 3D model, parts of the point cloud are likely viewed by more than one image, so that one point is projected on pixels belonging to more than one picture. Several methods have been studied to solve this ambiguity; effective ones associate to the point a weighted average of the colors of the pixels in which it has been projected. Our approach is similar to the one proposed in [18]. It consist in building weight masks for each pixel of the image based on the distance between the point (which was projected on that pixel) and the camera centre, and the angle between the point normal and the image plane normal.

IV. REGISTRATION ERROR EVALUATION

In order to test the performances of the calibration algorithm, it has been decided to compute the projection error of some points projected from the point cloud to the image plane. Reference points have been established on the sculpture using adhesive circular markers. In this way the 3D scanner was able to make measurements in the region inside the marker. This would not have been possible with, as instance, cross-type markers, because the scanner treats dark points as invalid; on the other hand using light color markers would have made difficult their unambiguous detection on the images. For each marker, the 3D coordinates of its center are extracted and projected on the calibrated picture. The distance between the projected center and the real center coordinates on the picture is computed. Figure 5 shows the marker placement on the sculpture.

![Fig. 4. Graphical representation of the occluded points removal problem.](image)

![Fig. 5. Markers displacement on the object](image)

In order to find the markers’ centers in the images, an algorithm for automatic centers extraction has been developed. The algorithm is structured as follows:

1. Select a region of interest (ROI) around the circle.
2. Perform a thresholding segmentation of the ROI, obtaining a binary map separating the circle from the background.
3. Compute the cumulative sums of the columns and the rows of the binary map. The coordinates of the center are calculated as the centroid of the region where the cumulative sum of the rows (and columns) is $> 0$.

Once the center of the markers are correctly extracted from each image, the average error in pixel along $x$ and $y$ coordinates can be computed as the absolute difference between the coordinates:

$$D_x = |x - \hat{x}|, \quad D_y = |y - \hat{y}|,$$

the algebraic mean is computed to retrieve the average error along both coordinates:

$$e_x = \frac{1}{N} \sum_{i=1}^{N} D_x(i), \quad e_y = \frac{1}{N} \sum_{i=1}^{N} D_y(i),$$
where \( N \) is the number of reference points (markers) which are visible in the picture.

V. RESULTS

The evaluation of the proposed technique has been carried out using visual comparisons and quantitative assessment. The tests were made on two datasets generated from 3D scans and photographic campaigns of the sculptures of a Putto and a Lion’s head. The Putto dataset comprises a set of 29 depth images with as much scanner pictures, a set 20 of uncalibrated photographic images and a point cloud 3D model. The Lion’s head dataset includes 25 depth images and associated images, 16 photographic images and a point cloud 3D model. The camera pictures have a resolution of 3000 \( \times \) 2008 pixels, whereas the scanner pictures have a resolution of 1280 \( \times \) 1024 pixels. Figure 6 shows a comparison between the same view of the sculpture of the Lion’s head. The first is a photographic image, while the other two are the renderings of the point cloud textured with scanner images and with the photographic pictures according to the proposed method. Since we are working on point clouds, some black regions appear on the rendered models, due to the background being visible through the model itself. Figure 7 shows two views of the point cloud from the Putto dataset textured using camera pictures from the dataset. It can be seen that the model is uniformly colored over the entire surface. Therefore, the models textured with the photographic images are visually accurate (no visual artifacts are visible, just a little blurring in some areas) and they are much more similar to the original appearance of the sculptures than the ones textured with the scanner pictures. In fact, the imaging sensor of the scanner has a high color dominant in the red channel, causing the reddish color of the model. On the other hand, the model textured with the scanner pictures might result more sharp and detailed. This is only partially due to the fact that there are no misalignments in texturing with scanner images. Mainly, the feeling of sharpness is due to the presence of shading effects caused by the (unnatural) direct bright illumination produced by the scanner, which is optimized for pattern based geometry acquisition but, as already stated, not ideal for photographic image acquisition, where diffuse lighting is advisable. Table V presents the projection errors in pixels, obtained from the registration error evaluation algorithm presented in the previous section, for the 16 camera picture used in the texturing of the Lion’s head sculpture. The second and third column account for the average registration errors computed for the \( x \) and \( y \) coordinates. The last column shows the number of correspondences used to calibrate the camera image. The last row contains the average error for the \( x \) and \( y \) coordinates. The measurement results show, in almost all cases, a quite low error when compared to the actual images resolution, with only few exceptions in which the average error is higher than 20 pixels. These anomalies are mainly due to the fact that the calibration points obtained from the automatic process were less uniformly distributed with respect to the other cases. In fact, it is important that such points are distributed as uniformly as possible over the camera image, otherwise the estimate of the camera parameters might be polarized to well fit points around the region where the calibration points are

<table>
<thead>
<tr>
<th>Picture</th>
<th>( \text{avg } x ) error (pixels)</th>
<th>( \text{avg } y ) error (pixels)</th>
<th>Calibration points</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>1.84</td>
<td>1.47</td>
<td>319</td>
</tr>
<tr>
<td>02</td>
<td>0.99</td>
<td>1.69</td>
<td>161</td>
</tr>
<tr>
<td>03</td>
<td>0.81</td>
<td>1.22</td>
<td>539</td>
</tr>
<tr>
<td>04</td>
<td>0.89</td>
<td>1.08</td>
<td>281</td>
</tr>
<tr>
<td>05</td>
<td>1.67</td>
<td>0.72</td>
<td>106</td>
</tr>
<tr>
<td>06</td>
<td>24.41</td>
<td>12.56</td>
<td>604</td>
</tr>
<tr>
<td>07</td>
<td>8.79</td>
<td>6.28</td>
<td>255</td>
</tr>
<tr>
<td>08</td>
<td>1.81</td>
<td>0.8</td>
<td>253</td>
</tr>
<tr>
<td>09</td>
<td>0.89</td>
<td>1.26</td>
<td>700</td>
</tr>
<tr>
<td>10</td>
<td>7.58</td>
<td>5.23</td>
<td>207</td>
</tr>
<tr>
<td>11</td>
<td>1.17</td>
<td>1.07</td>
<td>180</td>
</tr>
<tr>
<td>12</td>
<td>1.4</td>
<td>0.73</td>
<td>126</td>
</tr>
<tr>
<td>13</td>
<td>13.42</td>
<td>8.99</td>
<td>573</td>
</tr>
<tr>
<td>14</td>
<td>9.39</td>
<td>8.9</td>
<td>334</td>
</tr>
<tr>
<td>15</td>
<td>11.11</td>
<td>7.93</td>
<td>284</td>
</tr>
<tr>
<td>16</td>
<td>10.45</td>
<td>3.82</td>
<td>236</td>
</tr>
<tr>
<td>Global average</td>
<td>6.04</td>
<td>3.98</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7. Two views of the point cloud obtained textured using 15 camera pictures from the Putto dataset.
more concentrated, thus giving bad estimate for the other points. This might be a concern when using the automatic correspondences extraction, since there is not a control over how such points must be spatially distributed. In such cases, users might want to manually set some additional points to improve the alignment precision. Overall, the proposed method reaches the objective sought, that is a photorealistic texturing of the 3D models, capable of satisfying the most common visual rendering applications. For some specific applications, such as the ones aimed to make accurate measurements on textured 3D models, the registration error shown in Table V might still be too high. In these cases improvements could be made by increasing the image calibration accuracy, by either using more advanced algorithms such as the one proposed by Tsai [12], or exploiting approaches, such as the one recently presented in [19], directed to refine the image registration through locally optimized mappings.

VI. CONCLUSIONS

The method presented in this paper offers a solution for the registration of uncalibrated pictures to a 3D model, using images from an optical 3D scanner as support data. Starting from uncalibrated pictures, the implemented algorithm automatically carries out all the steps necessary to build a texture for a three dimensional model. From the evaluation of the results, we can state that the proposed algorithm offers a good solution for texturing 3D models for visualization and rendering purposes. The whole process, as opposed to other solutions, is fully automated in all its steps. Moreover, the algorithm has been designed in a modular way, so that improvements in all its stages can easily be made, leading to an increase of its speed and precision in registering images to the model.

REFERENCES