2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Region Based Fusion of 3D and 2D Visual Data for Cultural Heritage Objects

Robert Frohlich*, Zoltan Kato*‖, Alain Tremeau†, Levente Tamas‡, Shadi Shabo§, and Yona Waksman¶
*Institute of Informatics, University of Szeged P.O. Box 652, H-6701, Szeged, Hungary. Email: frohlich,kato@inf.u-szeged.hu
‖ Department of Mathematics and Informatics, J. Selye University, Komarno, Slovakia.
†Laboratoire Hubert Curien UMR CNRS 5516, Université Jean Monnet, Saint Etienne, France.
‡Robotics and Nonlinear Control Research Group, Technical University of Cluj-Napoca, Cluj-Napoca 400019, Romania.
§Laboratoire Archéorient, UMR CNRS 5133-Maison de l'Orient et de la Méditerranée, Université Lumière Lyon2, France.
¶Laboratoire Archéométrie et Archéologie, UMR CNRS 5138-Maison de l'Orient et de la Méditerranée, Lyon, France.

***Abstract*—A workflow is proposed for Cultural Heritage applications in which the fusion of 3D and 2D visual data is required. Using data acquired by cheap, standard devices, like a 3D scanner having a low quality 2D camera in it, and a high resolution DSLR camera, one can produce high quality color calibrated 3D model for documenting purpose. The proposed processing workflow combines a novel region based calibration method with an ICP alignment used for refining the results. It works on 3D data, that do not necessarily contain intensity information in them, and 2D images of a calibrated camera. These can be acquired with commercial 3D scanners and color cameras without any special constraint. In contrast with the typical solutions, the proposed method is not using any calibration patterns or markers. The efficiency and robustness of the proposed calibration method has been confirmed on both synthetic and real data.**

I. Introduction

Recently, as more and more devices are available for archeologists to document important cultural heritage (CH) objects, the need for effective software solutions is also increasing. Capturing an object with different modalities giving different levels of detail, the fusion of these data is inevitable at a given point. Reviewing recent CH publications we can observe, that a large variety of devices are used, starting from low cost, entry level devices, till expensive, high end solutions that require laboratory conditions for the best possible results. As these conditions can rarely be ensured on the field, some compromises have to be made. Most of the works rely on either laser or structured light based 3D scanners, or photogrammetry to obtain the 3D model of an object. Though the latter is widely used, it has some clear disadvantages: a large number of images has to be captured and the processing part is time consuming. Thus the results cannot be seen on the field, it cannot be verified if the necessary level of details has been captured. In order to overcome this issue, the authors of [1] have experimented with a mathematical positioning procedure to reduce the required number of images captured. Others usually use various software solutions to do the 3D reconstruction using more images taken from arbitrary positions [2]. Since most commercial softwares rely on keypoints, problems can occur with objects having little or no texture at all. In these cases, the best practice is placing external markers near or on the object, visible on the captured images. A good example is presented in [3], where geotagged marker points were used for both photogrammetric and laser scanning techniques.

Most of the 3D scanners include a built-in RGB camera capable of recording color information together with the coordinates for every point, but in most cases, these are poor quality, low resolution sensors intended primarily to facilitate registering the scans into a complete 3D color model. As a result, textured models based on such RGB data are not satisfactory for most CH documentation applications.

In archaeological CH study 3D modelling has become a very useful process to obtain indispensable data for 3D documentation and visualization. The purpose of this work was to provide archeologists belonging to laboratories Archéométrie et Archéeologie (www.arar.mom.fr) and Archéorient (www.archeorient.mom.fr) with a solution for 3D reconstruction of ceramics from the eastern Mediterranean from 3D textured model. For archeologists 3D models of ceramics or fragments of potteries, including textural details, represent another way of documenting ceramics next to the traditional 2-D representations through technical drawings. Beyond the accuracy of the 3D features such as structural surfaces and shapes, archeologists are also concerned by the accuracy of color features, especially color patterns and color

The authors would like to thank i3mainz - Institute for Spatial Information and Surveying Technology for the Chinese warrior public data set, the Maison de l'Orient et de la Méditerranée Jean Pouilloux for the ceramic fragments, and Rada Deeb for her help. The 3D acquisition was done thanks to the funding of the project PALSE IPEM (ANR-11-IDEX-0007). Robert Frohlich was partially supported by the COST Action TD1201 - COSCH (Colour and Space in Cultural Heritage) through an STSM grant. Levente Tamas was partially supported by the grant nr. C.I.2/1.2./2015 of the Technical University of Cluj-Napoca. This work was partially supported by the NKFI-6 fund through project K-120366; the Research & Development Operational Programme for the project "Modernization and Improvement of Technical Infrastructure for Research and Development of J. Selye University in the Fields of Nanotechnology and Intelligent Space", ITMS 26210120042, co-funded by the European Regional Development Fund; and by the Agence Universitaire de la Francophonie (AUF) and the Romanian Institute for Atomic Physics (IFA), under the AUF-RO project NETASSIST.

inclusions. Indeed spatial and color features are important factors for specialists in ceramics to analyze fragments, make hypothesis about 3D objects/shapes from sets of fragments, and in general as educational and research tools.

A possible solution is to fuse high resolution color calibrated RGB images with the 3D data. The main challenge of such a fusion is the estimation of the camera's relative pose to the reference 3D coordinate system. In the computer vision community many solutions are available solving this problem based on: finding point or line correspondences between the two domains [4], using mutual information [5], and large number of solutions relying on specific artificial landmarks or markers [6]. There are also expensive software solutions (*e.g.* [7] used Innov Metric Polyworks, [8] used Photomodeler) that solve this problem. However, these also require good quality RGB information in the 3D data, hence a pure geometric data with no RGB information is not enough to solve the fusion.

In contrast, our method works without color information in the 3D data and uses regions instead of matching key-points, which can be easier to detect in case of CH objects with homogeneous surface paintings. One region visible on both the 2D images and the 3D point cloud is already enough to solve the pose estimation, but with more regions the method becomes more robust [9]. In 2D, these regions can be easily segmented using standard segmentation methods, while in 3D, they can usually be segmented based on the 3D model's surface parameters or based on color information, if it is available. We show on synthetic benchmarks the performance of our method, including the robustness against segmentation errors that can occur in real world situations. We also validate the method on real data test cases which confirms that with good quality input data we can achieve high quality results, as well as moderate errors in the 3D model are well tolerated.

II. Fusion workflow

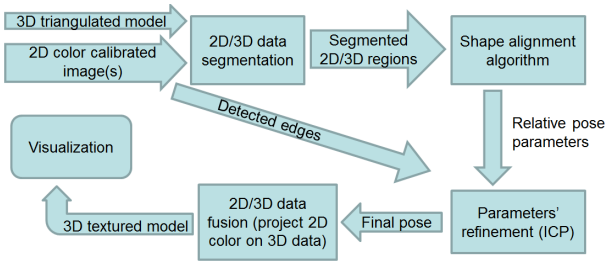


Fig. 1. Workflow diagram of the proposed method.

The diagram of the proposed workflow is shown in Fig. 1. The input consists of a 3D triangulated mesh and a set of 2D spectral images of the CH object. These images go through a 4-step processing pipeline in order to obtain a precise textured 3D model. We assume that the 2D cameras are color calibrated and their internal projection parameters are known, furthermore the acquired 3D point cloud has been preprocessed into a triangulated mesh (this is typically done by the 3D device's own software). In the following, we will present each processing step.

A. Segmentation(2D-3D)

As our method works with regions, we have to segment a corresponding set of regions both in the 2D images and 3D data. Since every test case is unique, we have to choose the segmentation method according to the surface properties. On the 2D images, any standard segmentation method (*e.g.* [10]) could be used (in our experiments, we used the *Fuzzy selection tool* of the free *Gimp* software). For the 3D data as well, we can choose based on the type of data that we have. When RGB information is available, we can simply use color based segmentation methods as in 2D. If it is not available, we can use 3D region growing, like the Minimum Covariance Determinant based algorithm [11] or interactive graph cut like [12]. Manual selection can also be used, for example in our experiments the *Z-painting* tool of *Meshlab* has been used for interactive selection of regions, which works well regardless of the availability of RGB data. Let us emphasize, that no matter how many regions we extract from the data, they only have to correspond as a *whole*, a pairwise correspondence is not needed (even the number of regions can be different)! Hence, given a corresponding set of 2D regions $\{\mathcal{D}_i\}_{i=i}^{N}$ and 3D regions $\{\mathcal{F}_j\}_{j=i}^{M}$, they only have to satisfy the following constraint:

$$\mathcal{D} = \mathbf{P}\mathcal{F}, \text{ with } \mathcal{D} = \cup_{i=1}^{N}\mathcal{D}_i \text{ and } \mathcal{F} = \cup_{j=1}^{M}\mathcal{F}_i \qquad (1)$$

where $\mathbf{P}$ is the camera projection matrix (see next Section).

B. Pose estimation

Given a corresponding set of segmented 2D-3D regions $\mathcal{D}$ and $\mathcal{F}$, we propose an extension of our plane-based Lidar-perspective camera pose estimation algorithm [9] to the data fusion problem. While the method in [9] was used strictly on planar regions, we show that the method can be extended to curved (but smooth) surfaces. This way it can be used in Cultural Heritage applications, since most of the objects, ceramics have smooth regions.

Assuming that each of the segmented 3D regions $\{\mathcal{F}_j\}_{j=i}^{M}$ are smooth enough (*i.e.* they satisfy (1)), let us express a 3D point $\mathbf{X}$ with its homogeneous world coordinates $\mathbf{X} = (X_1, X_2, X_3, 1)^T$. The perspective camera sees the same world point $\mathbf{X}$ as a homogeneous point $\mathbf{x} = (x_1, x_2, 1)^T$ in the image plain obtained by the perspective projection

$$\mathbf{x} = \mathbf{K}\mathbf{R}[\mathbf{I}|\mathbf{t}]\mathbf{X} \qquad (2)$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{K}$ is the 3×3 upper triangular *calibration* matrix containing the camera's *intrinsic* parameters, while $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation, respectively, aligning the *camera coordinate system* with the *world coordinate frame* (*i.e.* the coordinate system of the 3D points). Since we are using calibrated cameras, $\mathbf{K}$ is known, thus the only unknown parameters are the 6 pose parameters (3 angles of rotation in $\mathbf{R}$, 3 components of the translation in $\mathbf{t}$). Therefor the effect of $\mathbf{K}$ can be inverted in order to obtain a camera-independent normalized image

$$\mathbf{K}^{-1}\mathbf{x} = \mathbf{R}[\mathbf{I}|\mathbf{t}]\mathbf{X} = \mathbf{P}\mathbf{X} \qquad (3)$$

Classical solutions would now establish a set of 2D-3D point matches (*e.g.* using special calibration targets or markers), and then solve for $\mathbf{R}$ and $\mathbf{t}$ via a system of equation based on (3).

However, in many CH applications, it is not always possible to attach markers to the object's delicate surface. Furthermore, the 3D scans and camera images might be acquired at different times, using different lighting conditions for optimal results. Our pose estimation method, based on the 2D shape registration approach presented in [9], proposes a solution in these challenging situations. Since point correspondences are not available, we cannot use (3) directly. However, individual point matches can be integrated out [9] yielding the following integral equation:

$$\int_{\mathcal{D}} \mathbf{x} d\mathbf{x} = \int_{\mathbf{P}\mathcal{F}} \mathbf{z} d\mathbf{z}, \tag{4}$$

where $\mathcal{D}$ corresponds to the regions visible in the *camera* image and $\mathbf{P}\mathcal{F}$ is the virtual image of the *3D region*s projected by $\mathbf{P}$. We can clearly see that the above integral equation stays valid for curved, smooth surfaces as well, as long as $\mathcal{D}$ and $\mathcal{F}$ are satisfying (1) (*i.e.* no self-occlusion of points takes place). There are 2 issues with the above equation:

1) it corresponds to a system of 2 equations only, which is clearly not sufficient to solve for all 6 parameters of the camera pose;
2) the evaluation of the right hand side requires the explicit projection of the 3D regions $\mathcal{F}$, which might be computationally expensive.

To resolve 1), observe, that (3) remains valid when a function $\omega : \mathbb{R}^2 \to \mathbb{R}$ is acting on both sides of the equation [9]

$$\omega(\mathbf{x}) = \omega(\mathbf{PX}), \tag{5}$$

and the integral equation of (4) becomes

$$\int_{\mathcal{D}} \omega(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{P}\mathcal{F}} \omega(\mathbf{z}) d\mathbf{z}. \tag{6}$$

Thus adopting a set of nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, each ω_i generates a new equation yielding a system of ℓ independent equations. Hence we are able to generate sufficiently many equations. The $\mathbf{R}$ and $\mathbf{t}$ parameters of the camera pose are then simply obtained as the solution of the nonlinear system of equations (6). In practice, an overdetermined system is constructed, which is then solved by minimizing the algebraic error in the *least squares sense* via a standard *Levenberg-Marquardt* algorithm.

To resolve 2), let us choose power functions for ω_i

$$\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i}, \quad n_i \leq 3 \text{ and } m_i \leq 3, \tag{7}$$

which yields the 2D geometric moments of the projected 3D region $\mathbf{P}\mathcal{F}$, that can be computed efficiently. Since $\mathcal{F}$ consists of triangulated surface patches, their projections is a set $F^{\triangle}$ of triangulated planar patches, thus the final form of the equations becomes

$$\int_{\mathcal{D}} x_1^{n_i} x_2^{m_i} d\mathbf{x} = \int_{\mathbf{P}\mathcal{F}} z_1^{n_i} z_2^{m_i} d\mathbf{z} = \sum_{\forall \triangle \in F^{\triangle}} \int_{\triangle} z_1^{n_i} z_2^{m_i} d\mathbf{z}. \tag{8}$$

The integrals over the triangles are various geometric moments, which can be computed for order $(p+q)$ using the closed form formula [9], [13]

$$2 \sum_{k=0}^{p} \sum_{l=0}^{q} \frac{(-1)^{k+l} \binom{p}{k} \binom{q}{l} \nu_{kl}}{k+l+2} {z_1}_0^{p-k} {z_2}_0^{q-l} \tag{9}$$

where

$$\nu_{kl} = \sum_{i=0}^{k} \sum_{j=0}^{l} \frac{\binom{k}{i} \binom{l}{j}}{k-i+l-j+1} (z_{10} - z_{11})^i (z_{11} - z_{12})^{k-i} (z_{20} - z_{21})^j (z_{21} - z_{22})^{l-j} \tag{10}$$

with the notation z_{1_i} and z_{2_i}, $i = 0 \ldots 2$ being the vertices of the triangle.

C. ICP refinement

In the previous step, we have obtained a camera pose by minimizing the *algebraic error* of the system in (8). Although this is already a good quality estimate, we can further refine it by minimizing a relevant *geometric error*. In the following, we will show how a standard Iterative Closest Point (ICP) [14] algorithm can be used, if color information, even if it is of poor quality, is also available at each 3D point. In our workflow, ICP is used to align the 3D edge lines' projection with the 2D edge map (denoted by $\mathbf{x_e}$) of the camera image. There are different approaches to detect edges in a 3D pointcloud based on geometric properties, but for our purpose, we have to rely solely on the color information to be able to detect the same edges as in the 2D image. We have tackled this by simply projecting the 3D data onto an image with the initial camera pose using (2), then running Matlab's edge detection function on that image, resulting the edge points. The corresponding 3D points $\mathbf{X_e}$ will be the detected 3D edge points. The algorithm then iteratively projects the 3D $\mathbf{X_e}$ edge points using the current $\mathbf{K}\mathbf{R}^n[\mathbf{I}|\mathbf{t}^n]$ camera matrix, that has only the camera pose parameters $(\mathbf{R}^n, \mathbf{t}^n)$ changing between iterations, giving the reprojected edge points ${\mathbf{z_e}}^n$ at iteration n:

$${\mathbf{z_e}}^n = \mathbf{K}\mathbf{R}^n[\mathbf{I}|\mathbf{t}^n]\mathbf{X_e} \tag{11}$$

The ICP algorithm will align this ${\mathbf{z_e}}^n$ projection to $\mathbf{x_e}$, the edge map of the 2D image. We can clearly see that ICP will actually minimize the *backprojection error* this way.

D. Data fusion

The final step of the workflow is the data fusion itself. Using the estimated relative pose and the calibration matrix of the camera, we can project (with (2)) the 3D points onto the 2D

image. Since these do not necessarily project to exact pixel coordinates, we can interpolate the neighbouring pixels' color to find the best RGB value for every projected point. If we had multiple 2D input images, then we can fuse all images with the 3D data. For those 3D points, that are visible in more camera images, we have to decide which camera has the best view of it. For this purpose, let us calculate the normal vector $\mathbf{n}_i$ for each 3D point $\mathbf{X}_i$. In our experiments, we have used Meshlab's *Compute normals for point sets* function, which fits a local plane to every point's small neighborhood (10 neighbours). Then for every point $\mathbf{X}_i$ we compute the angle of its normal $\mathbf{n}_i$ with the orientation vector $\mathbf{c}_j$ of each camera's optical axis as

$$\cos\theta = \frac{\mathbf{c_j} \cdot \mathbf{n_i}}{\|\mathbf{c_j}\|\|\mathbf{n_i}\|}, \tag{12}$$

and the camera image j with maximal $\cos\theta$ value is used to colorize the 3D point $\mathbf{X}_i$. As a result, we get a good quality textured 3D model of the object. Since the 2D images are color calibrated, no color shift will appear, no transitions will be visible between regions that get RGB information from different images, if we assume a good uniform lighting was used when capturing the images. For easier examination of the results, we only used a single camera image for fusion in the test cases shown in Fig. 6.

III. Evaluation on synthetic data

For a quantitative and qualitative evaluation of the proposed pose estimation algorithm in Section II-B, we have generated a benchmark set using 16 different shapes (such as in Fig. 2a). The 3D data was generated by projecting a 2D shape on a virtual spherical surface (having a Gaussian curvature of $K = 1/r^2 = 1/10000$). The 2D image of such an object was captured with a virtual camera having the intrinsic parameters of a standard 1Mpx camera and a random pose by rotating it with $(-25^\circ \cdots 25^\circ)$ along all three axis and translating it randomly along all three axis with the maximum possible translation being equal to the size of the object. A data set consists of 100 such images.

The results are presented in Fig. 2b - Fig. 2f. To evaluate the precision of the pose parameters, we backprojected the 3D points on the image plane, and calculated the percentage of the non-overlapping area (δ error) between the projection and the original observation. Ideally these should overlap perfectly, but experimentally we have found that 5% δ error or lower can be considered a correct result. We have also calculated the errors of the 3 rotation angles, and the translation error (see Fig. 2c) as the distance between reference and estimated position.

Since in real cases both the 2D and 3D regions are affected by segmentation errors, we have also evaluated the robustness of our pose estimation method against such errors. For this purpose we have generated two different data sets: one with synthetically generated segmentation errors on 3D regions and another one with the 2D images being corrupted by it. An example synthetic data pair is shown in Fig. 2a, on top the 3D curved surface is shown, while below two images of the region, one with 10% simulated segmentation error. As we can see from the error plots in Fig. 2b - Fig. 2f, the method is more robust for 2D segmentation errors. The same median δ error is achieved with 10% 2D segmentation error as with 5% error in the 3D segmentation, but with the 3D case we see more bad results. This also reflects on the rotation and translation error plots, while the median values are similar for the two cases, the number of incorrect results is higher in the 3D case. Nevertheless a median delta error of below 2% in using only a single curved region, is considered satisfactory.

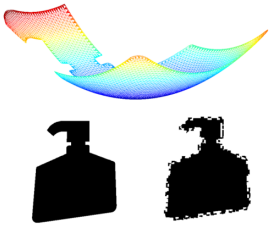
IV. Real data test cases

We have verified our workflow on different real data test cases, of which 2 are presented in detail here: one using high precision data inputs, while the other using more affordable acquisition solutions.

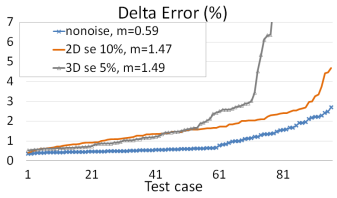
A. The Chineese warrior test case

The object used for this test case is a small (18 cm tall) figurine. The 2D images were taken with a calibrated Nikon D800 DSLR camera having a full frame 36Mpx sensor, while the 3D data has been produced with a high precision marker based Structure-from-Motion software solution in strict laboratory conditions, giving us a perfect reference data in this case. While usual software solutions use markers or key-points to produce such fused data, our method uses only the color images and a raw pointcloud (it doesn't even has to include RGB information!). In the first step of our workflow, we have to segment a few regions in 2D and 3D. Since the test object has a more complex, rugged surface, we have to concentrate on the smooth, well defined regions, where self-occlusion doesn't occur. Best choice in this case is segmenting the straps and bands on the clothes, since these are smooth regions, raised from their neighbors, with clearly visible ends. In 2D, a region growing segmentation tool was used, while in 3D an interactive selection method in Meshlab was adopted. Using the segmented data pairs, the second step estimates the pose of the camera relative to the 3D pointcloud with good precision. This is illustrated in Fig. 3 by backprojecting a few hand-picked 3D key-points with the estimated camera pose to the 2D image (red dots) - which are close to their reference location (green dots). The measured average error was around $20 - 30$px, which translates to approx. 1mm real world error. In case we don't have access to intensity information in the 3D data, or if the object itself does not have a rich texture on its surface, then this is the final result. Note that if there is no intensity information, using a commercial software solution to align such data would also be challenging.

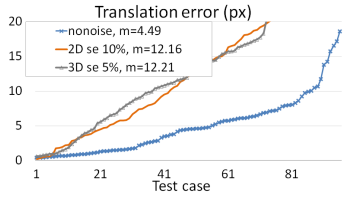
As in this test case we have color information too in the 3D input data, we can apply the ICP refinement step proposed in our workflow. The algorithm refines the relative pose based on the edge-map of the 2D image, and the projection of the 3D edge points. At this step, edges detected on smooth, mostly planar surfaces are desirable. To measure the benefits of using the ICP refinement step, we backprojected the same 3D key-points and calculated the backprojection error. In Fig. 3, we can see that landmark points are projected closer



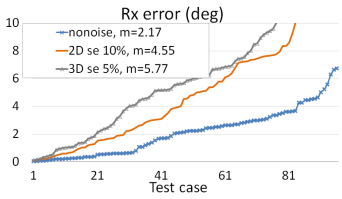
(a) Sample synthetic data



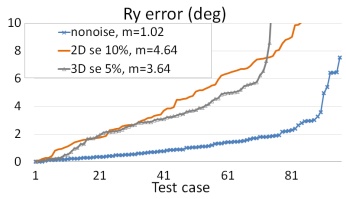
(b) Percentage of non-overlapping area



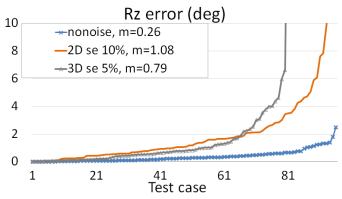
(c) Distance between reference and estimated camera position



(d) Rotation error along x axis



(e) Rotation error along y axis



(f) Rotation error along z axis

Fig. 2. (a) Sample synthetic data, (b-f) error plots of the results on synthetic data (*2D se* stands for 2D segmentation error, *3D se* for segmentation error on the 3D data and *m* stands for median value).

to their correct location, reducing the average distance to 8px, equivalent to 0.2mm projection error. This can be considered good precision for most CH applications. Final fused result from only a single camera image can be seen in Fig. 6.

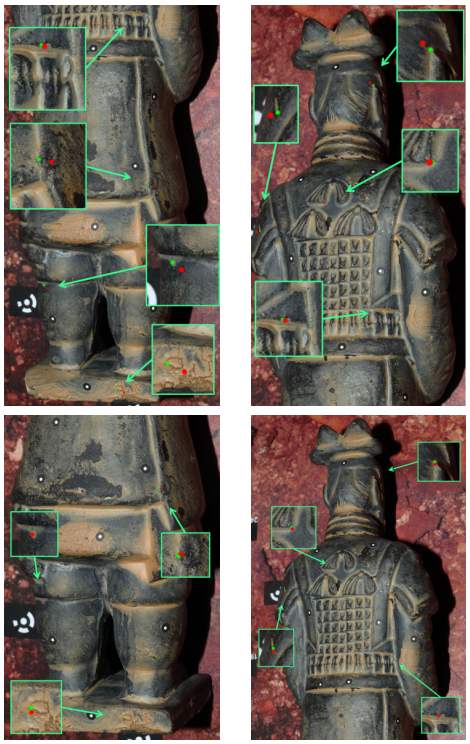


Fig. 3. Precision of the region based pose estimation's results in first row, and results of ICP refinement in second row. Green dots are the reference locations while red dots are the back-projections of the 3D landmarks.

B. Ceramic fragments test cases

The objects used in this test case are small fragments of ceramic bowls and vases. The 2D images were captured with a standard Canon 1000D DSLR camera, having 2.5Mpx resolution. The 3D data was produced by a handheld Artec Spider scanner and its bundled software. In this case, using a relatively cheap and easy to use scanner solution, we cannot expect perfect 3D data. The software uses a key-point based algorithm to align partial scans and build the complete 3D model. Since the scanner only has a low resolution RGB camera built in, this process can get cumbersome in some situations. As we have found, even if the software produces a visually pleasing, watertight 3D model, it may lack precision. Of course a perfect alignment was not possible with these incorrect 3D data, but we have shown, that in spite of the imperfect 3D model, our algorithm is robust enough to produce a good fused result. The segmented regions used for the pose estimation are shown in Fig. 4. The backprojection error of the two test cases can be seen in Fig. 5. The average error was 33px and 28px respectively.

Fig. 4. 2D segmentation example.

V. Conclusion

A workflow has been proposed for the 2D-3D visual data fusion. While most of the current solutions struggle in situations when RGB information is missing, or the surface color of the object is too homogeneous, or the surface is too reflective, our method has a clear advantage by not relying on 3D color information to solve the problem. It is only necessary to detect

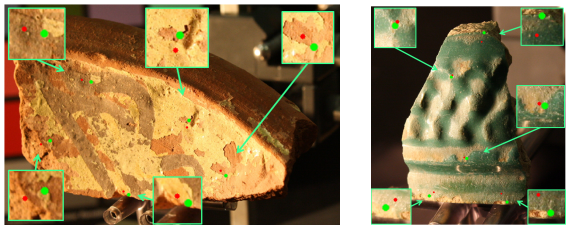


Fig. 5. Final precision achieved using the ICP refinement step. Green dots are the selected specific landmarks, red dots are the back-projection of the same landmarks in 3D.

at least a single smooth region visible in both 3D and 2D domains, and a good pose estimation is obtained. When RGB information is also available with the 3D data, it can be used at the segmentation step and also for the refinement of the final parameters. Of course, the precision depends on the quality of the input data, but we have shown experimentally that using cheap devices we can produce relatively good quality outputs, in terms of texture resolution and color precision. Furthermore, since we are not directly using the RGB color values, the method potentially works with infrared images, or even with hyperspectral images.

REFERENCES

[1] L. Grosman, O. Smikt, and U. Smilansky, "On the application of 3-d scanning technology for the documentation and typology of lithic artifacts," *Journal of Archaeological Science*, vol. 35, no. 12, pp. 3101 – 3110, 2008.

[2] A. Koutsoudis, B. Vidmar, and F. Arnaoutoglou, "Performance evaluation of a multi-image 3d reconstruction software on a low-feature artefact," *Journal of Archaeological Science*, vol. 40, no. 12, pp. 4450 – 4456, 2013.

[3] P. Grussenmeyer, T. Landes, T. Voegtle, and K. Ringle, "Comparison methods of terrestrial laser scanning, photogrammetry and tacheometry data for recording of cultural heritage buildings," *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37(B5), pp. 213–218, 2008.

[4] A. Mastin, J. Kepner, and J. W. F. III, "Automatic registration of lidar and optical images of urban scenes," in *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*. Miami, Florida, USA: IEEE, June 2009, pp. 2639–2646.

[5] P. Viola and W. M. Wells, III, "Alignment by maximization of mutual information," *Int. Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, Sep. 1997.

[6] H. S. Alismail, L. D. Baker, and B. Browning, "Automatic calibration of a range sensor and camera system," in *Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission*. Zurich, Switzerland: IEEE, October 2012, pp. 286–292.

[7] R. Kadobayashi, N. Kochi, H. Otani, and R. Furukawa, "Comparison and evaluation of laser scanning and photogrammetry and their combined use for digital recording of cultural heritage," *Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. 5, pp. 401–406, 2004.

[8] S. Al-kheder, Y. Al-shawabkeh, and N. Haala, "Developing a documentation system for desert palaces in jordan using 3d laser scanning and digital photogrammetry," *Journal of Archaeological Science*, vol. 36, no. 2, pp. 537 – 546, 2009.

[9] L. Tamas and Z. Kato, "Targetless Calibration of a Lidar - Perspective Camera Pair," in *Int. Conf. on Computer Vision, Bigdata3dcv Workshops*, Sydney, Australia, December 2013, pp. 668–675.

[10] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1454–1466, Oct 2001.

Fig. 6. Final fusion results from single viewpoint, using the ICP refinement step. The ceramics 3D data are available from the authors affiliated to the Maison de l'Orient et de la Méditerranée.

[11] A. Nurunnabi, D. Belton, and G. West, "Robust segmentation in laser scanning 3d point cloud data," in *Proc. of Int. Conf. on Digital Image Computing: Techniques and Applications*, Dec 2012, pp. 1–8.

[12] D. Sedlacek and J. Zara, "Graph cut based point-cloud segmentation for polygonal reconstruction," in *Proc. of the 5th Int. Symposium on Advances in Visual Computing: Part II*, ser. ISVC '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 218–227.

[13] G. C. Best, "Helpful formulas for integrating polynomials in three dimensions (in Technical Notes and Short Papers)," *Int. Journal of Mathematics and Computer Science*, vol. 18, no. 86, pp. 310–312, April 1964.

[14] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.