Computer vision-based approach for rite decryption in old societies

Jilliam María Díaz Barros
Le2i Laboratory/CNRS
Université de Bourgogne, FR
jilliam.diaz@iee.lu

Cédric Demonceaux
Le2i Laboratory/CNRS
Université de Bourgogne, FR
cedric.demonceaux@u-bourgogne.fr

Adlane Habed
ICube Laboratory/CNRS
Université de Strasbourg, FR
habed@unistra.fr

Alamin Mansouri
Le2i Laboratory/CNRS
Université de Bourgogne, FR
alamin.mansouri@u-bourgogne.fr

Abstract

This paper presents an approach to determine the spatial arrangement of bones of horses in an excavation site and perform the 3D reconstruction of the scene. The relative 3D positioning of the bones was computed exploiting the information in images acquired at different levels, and used to relocate provided 3D models of the bones. A novel semi-supervised approach was proposed to generate dense point clouds of the bones from sparse features. The point clouds were later matched with the given models using Iterative Closest Point (ICP).

1 Introduction

In the current project, the interest of the archaeologists lay in studying rites performed in Iron Age societies through the analysis of spatial arrangements of excavated relics and, more importantly, animal skeletons. The spatial arrangement of animal bones might indeed shed some light on certain religious practices [1], [2], [3]. This generally required working on replicas of the excavated bones, or better, digitized instances of these. To this end, a comprehensive library of 3D models representing the entire skeleton of a horse obtained using a 3D laser scanner was built. However, in order to recover the entire 3D model of the animal's skeleton, the digitized bones needed to be positioned in space by relying on photographs captured in situ while documenting the excavation process.

Thus, the primary objective was to determine the spatial arrangement of bones of horses found in an archaeological site, to obtain the 3D reconstruction of the scene. The reconstruction was limited by the low number of images, acquired at different levels during the excavation process, some of them with changes in the scene lighting and with incomplete or partially occluded bones. Furthermore, the calibration of the camera was not available and according to the EXIF tags, the focal length was changed when acquiring different views of the scenes.

2 Proposed approach

Considering the aforementioned limitations, a semi-supervised method for the 3D reconstruction of the scene was proposed. The approach can be summarized in five steps, depicted in Figure 1:

(1) Object extraction.

(2) Feature detection and matching.

(3) Generation of additional points.

(4) 3D reconstruction.

(5) Point cloud matching.

In order to perform the 3D reconstruction, at least two views of the scene had to be available.

2.1 Object extraction

The first step was to specify the location of each bone within each image and subtract the background. This process was realized by interactively extracting a binary mask for each bone in each pair of images. One example of this procedure can be observed in Figure 2.

Figure 2. Original image (left). Binary mask denoting the location of the skull (center). Skull extracted using the binary mask (rigth).

2.2 Feature detection and matching

As a second step, the feature points of the resulting images were extracted by using the following detectors:

- Features from Accelerated Segment Test (FAST) by Rosten and Drummond[4].
- Speeded-Up Robust Features (SURF) by Bay et al[5].
- Maximally Stable Extremal Regions (MSER) by Matas et al[6].
- Harris corners by Harris and Stephens[7].
- Minimum eigenvalue by Shi and Tomasi[8].

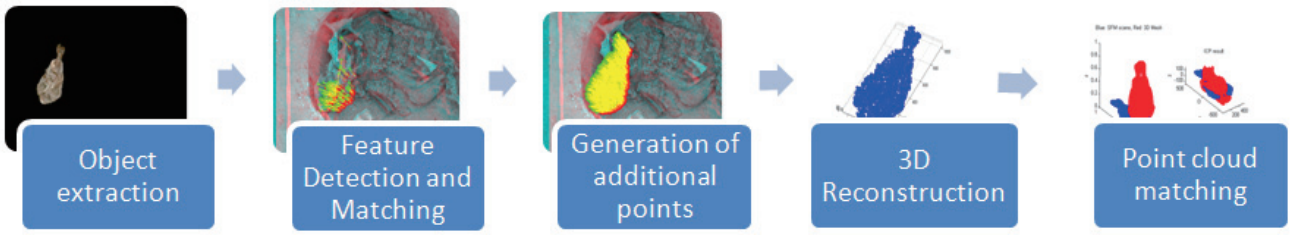

Figure 1. Proposed approach.

It was necessary to implement all these methods since the texture in some regions was homogeneous, and consequently, the number of features points was sparse.

From these points, features vectors were extracted using SURF [2]. During the feature matching, the outliers were eliminated using RANSAC.

2.3 Generation of additional points

In most cases, regardless of using all the algorithms of point detection, the number of feature points was sparse and not uniformly distributed in the whole region. This led to some mismatching during the comparison of the clouds of points.

In order to automatically avoid the previous problem, additional points were added uniformly to the area of interest in one image and projected in the second one assuming an affine transformation F between the two images, i.e., $\begin{pmatrix} u \\ v \end{pmatrix} = F \begin{pmatrix} x \\ y \end{pmatrix}$ [9], with (u, v) and (x, y) the coordinates of the same point in two images. This process was done after verifying that the number of matched points was at least 5.

The transformation can also be expressed as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = s \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

If $a = \cos(\theta)$ and $b = \sin(\theta)$, the previous equation becomes the linear system presented in (1), or $Ax = b$, and can be solved using linear least squares (2).

$$\begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \\ & \cdots & & \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} u \\ v \\ \vdots \end{pmatrix} \quad (1)$$

$$\hat{x} = \left(A^T A\right)^{-1} A^T b \quad (2)$$

2.4 3D reconstruction

The extrinsic parameters were not available in the real scenes, and despite neither the intrinsic parameters, a rough calibration matrix could be computed using the EXIF tags under certain assumptions. However, in most cases the focal length was modified from one image to another.

Considering these limitations, the scenes were reconstructed from the points using Structure from Motion (SfM), assuming that the camera was not projective. The toolbox of Vincent Rabaud[10] was employed in this section.

Within this toolbox, if the number of frames was equal to 2, the Gold Standard for Affine camera matrix method was implemented to find the projection matrices. If the number of frames was greater than 2, Tomasi-Kanade and autocalibration methods were used.

The projection matrices, P and P', were computed using the toolbox and the original set of points were extracted from each pair of images. Afterwards, the 3D scene was reconstructed with the full set of points using Direct Linear Transformation (DLT), equation (3), from the homogeneous coordinates of the points in both images $(x, y, 1)$ and $(x', y', 1)$. The reconstructed scene X_n was computed by using SVD of the 4x4 matrix and selecting the singular vector of the smallest singular value[9]. It corresponded to the cloud of points used in the next step.

$$\begin{bmatrix} xp^{3T} - p^{1T} \\ yp^{3T} - p^{2T} \\ x'p^{3T} - p'^{1T} \\ y'p^{3T} - p'^{2T} \end{bmatrix}_{4\times4} X_{n_{4\times1}} = 0_{4\times1} \quad (3)$$

2.5 Matching of point clouds

The previously reconstructed bones were normalized and matched with their correspondences in the set of 3D models, using Iterative Closest Point (ICP). The task was performed with the ICP toolbox of Wilm and Kjer [11], which included three different matching methods: Brute force, Delaunays and K-D tree.

Finally, the 3D models were repositioned according to the transformation provided by the toolbox. The sizes of the given models were resized to fit the extracted cloud of points.

3 Set of images

The set of images can be divided in two main scenes, as shown in Figure 3:

(a) Arrangements of bones of the leg (Scene 1 and 2).

(b) Arrangements of bones at different excavation levels (Scenes 3, 4 and 5).

A pair of images was provided for scenes 1, and 3 - 5, and three images for scene 2. One image per scene is displayed from Figure 6 to 10.

4 Experiments and Results

Below are presented the results obtained at different stages.

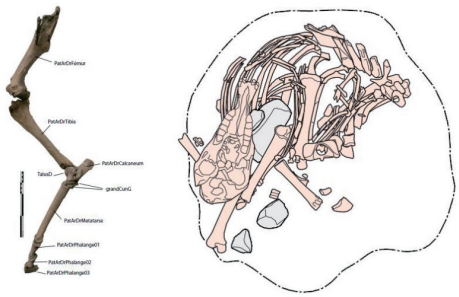

Figure 3. Main scenes: bones of the leg (left) and diverse bones (right).

4.1 Feature detection and matching

As mentioned before, the feature points were extracted using FAST, SURF, MSER, Harris corners and Minimum eigenvalues, and the matching was performed from feature vectors computed with SURF. The outliers were eliminated with RANSAC, using the epipolar constraint. The inliers of 'Scene 4' can be observed on the top-left image of Figure 4. The matching of the additional points generated assuming an affine transformation is shown at the top-right image of the same Figure, and a close-up of them are presented at the bottom images.

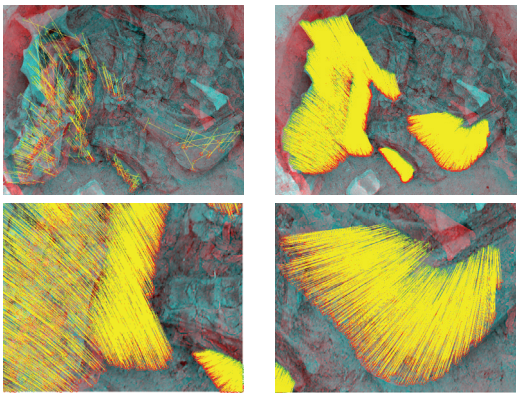

Figure 4. Matching of feature points and additional generated points for the scene 4: features points (top left), generated points (top right), close-up of generated points (bottom).

4.2 3D reconstruction

Initially, a synthetic scene created from one of the 3D models was used to test different reconstruction algorithms. The intrinsic and extrinsic parameters were assumed to be known. Two resulting images were computed using equation (4), where X represents the coordinates of the points in the 3D world and x the 2D points in the pixel coordinates[9].

$$\boldsymbol{x} = K\,[R|T]\,X \tag{4}$$

As the calibration was available, the essential matrix was computed from equation (5) and the fundamental matrix from equation (6), using the essential matrix.

$$E = [\boldsymbol{t}]_x\, R \tag{5}$$

$$F = K^{T^{-1}} \cdot E \cdot K^{-1} \tag{6}$$

The 3D scene X_n was estimated with the Direct Linear Transformation method as from equation (3).

Afterwards, the back projection was computed from $\boldsymbol{x} = PX_n$, to calculate the residual error using the projection matrices.

The process was repeated one more time, assuming that the essential matrix was not available, and therefore, using the feature points. For the selected model, the error using the fundamental matrix obtained from the essential matrix was of 0.00554, and using the feature points, of 0.0082. These results showed the validity of implementing Direct Linear Transformation method for the reconstruction of synthetic scenes, using the essential matrix or the feature points.

For the real scenes, as explained in subsection 2.4, the projection matrices were computed with the SfM toolbox, using the methods for uncalibrated cameras. These matrices were later implemented to perform the 3D reconstruction of the scene by using DLT.

4.3 Matching of point clouds

The cloud of points generated for each bone was matched with the corresponding 3D model using three algorithms of ICP: Brute force, Delaunays and K-D tree. One result after using 'K-D tree' is shown in Figure 5.

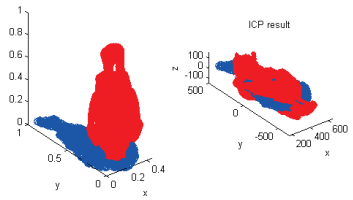

Figure 5. Point clouds matching. Blue: Reconstructed bone. Red: Provided 3D model.

Afterwards, the set of 3D models were arranged to reconstruct the scenes. The resulting reconstructions using the three ICP algorithms were nearly the same, with differences in the execution time. Figures 6 to 10 show the reconstructed scenes using the 'K-D tree' algorithm.

The execution time in seconds of each ICP algorithm is presented in Table 1.

Table 1. Execution time (s) of each ICP algorithm.

	Scene				
	1	2	3	4	5
Brute Force	24.97	16.79	23.99	28.13	23.03
Delaunay's	99.49	56.31	79.98	78.65	78.48
K-D tree	8.22	2.73	4.05	5.25	4.52

Since the acquired images did not cover many different views from the bones, the real depths of the bones

were not fully reconstructed. This led to the mismatching in the orientation of some bones, such as the bone in Figure 7 which was upside down in the reconstructed scene, but in general, the results were satisfactory.

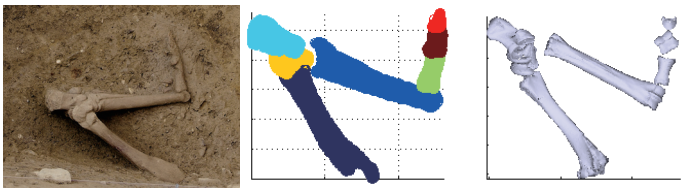

Figure 6. Scene 1: Original image (left), reconstructed scene (center) and point clouds matching results using K-D tree.

Figure 7. Scene 2: Original image (left), reconstructed scene (center) and point clouds matching results using K-D tree.

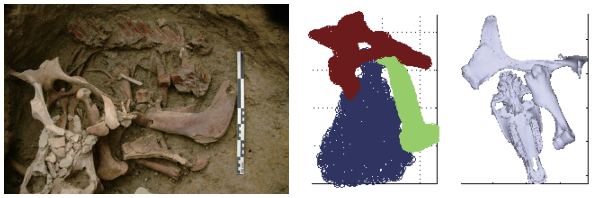

Figure 8. Scene 3: Original image (left), reconstructed scene (center) and point clouds matching results using K-D tree.

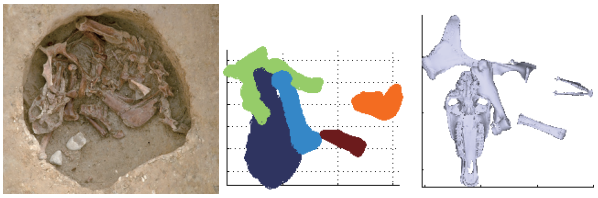

Figure 9. Scene 4: Original image (left), reconstructed scene (center) and point clouds matching results using K-D tree.

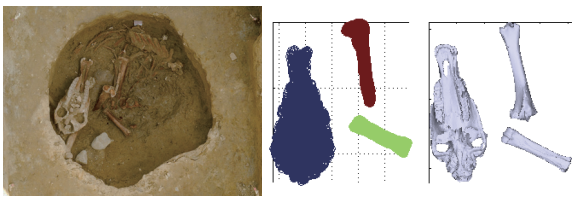

Figure 10. Scene 5: Original image (left), reconstructed scene (center) and point clouds matching results using K-D tree.

5 Conclusions

In this paper, a set of computer vision tools was implemented to determine the spatial arrangement of bones of horses during the excavations. One of the goals was to create cloud of points of the bones, by using the information of the images, to relocate the 3D models. In order to do so, features points were extracted for each bone using different detectors. Due to the low number of points to construct a point cloud, additional points were generated assuming an affine transformation. Despite having some outliers in the feature points, the resulting generated points were accurate.

SfM for uncalibrated cameras was implemented to find the projection matrices and the Direct Linear Transformation method was used for the reconstruction. The results obtained after applying three algorithms of ICP, Brute force, Delaunays and K-D tree were the same, despite the execution time varied from one to other, being K-D tree the fastest and Delaunays the slowest.

Future research will aim to automate the segmentation and object extraction step. Additional features detectors will be tested in order to find the most suitable for this type of application.

References

[1] A. Pluskowski: "The Ritual Killing and Burial of Animals: European Perspectives," *Oxbow Books*, 2012.

[2] G. Auxiette and P. Meniel: "Les dpts danimaux en France : de la fouille linterprtation," *Actes de la table ronde de Bibracte, 15-17 octobre 2012. Mergoil, Archologie des plantes et des animaux 4*, Montagnac, 286 p., 2013.

[3] E. Dietrich, G. Kaenel, D. Weidmann, P. Jud, P. Mniel and P. Moinat: "Le sanctuaire helvte du Mormont," *Archologie Suisse*, vol. 30, no. 1, pp. 2-14, 2007. `http://www.archeodunum.ch/FILES/mc9/56_tmp_430.pdf`

[4] E. Rosten and T. Drummond: "Fusing points and lines for high performance tracking," *IEEE International Conference on Computer Vision*, pp. 15081511, 2005.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool: "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, pp. 346-359, 2008.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla: "Robust wide baseline stereo from maximally stable extremal regions," *Proceedings of British Machine Vision Conference*, pp. 384-396, 2002.

[7] C. Harris and M. Stephens: "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, p. 147151, 1988.

[8] J. Shi and C. Tomasi: "Good Features to Track," *9th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd Ed. Cambridge University Press, 2003.

[10] V. Rabaud: "Vincent's Structure from Motion Toolbox," `http://vision.ucsd.edu/~vrabaud/toolbox/`, 2009.

[11] J. Wilm and H. M. Kjer: "Iterative Closest Point," *MATLAB Central,* `http://www.mathworks.com/matlabcentral/fileexchange/27804-iterative-closest-point`, 2010.