AUTOMATIC CLUSTERING OF CELTIC COINS BASED ON 3D POINT CLOUD PATTERN ANALYSIS

Sofiane Horache[1], François Goulette[1], Jean-Emmanuel Deschaud [1], Thierry Lejars[2], Katherine Gruel[2]

[1] Centre de Robotique, CAOR, Mines Paristech, PSL University (name.surname)@mines-paristech.fr
[2] - AOROC, ENS Ulm, PSL University((name.surname)@ens.fr)

Commission IIWG II/8

KEY WORDS: 3D Point Cloud, Registration, Celtic coins, Pattern Recognition, Clustering, Archaeology

ABSTRACT:

The recognition and clustering of coins which have been struck by the same die is of interest for archeological studies. Nowadays, this work can only be performed by experts and is very tedious. In this paper, we propose a method to automatically cluster dies, based on 3D scans of coins. It is based on three steps: registration, comparison and graph-based clustering. Experimental results on 90 coins coming from a Celtic treasury from the II-Ith century BC show a clustering quality equivalent to expert's work.

1. INTRODUCTION

The Celts are well-known for the high quality of their metallurgical productions (shields, weapons, jewels, helmets, coins, and so on). In order to better understand their manufacturing methods, archaeologists rely on detailed observations of objects, in which 3D scans are becoming increasingly important. Celtic coins can give very useful information to better understand the socio-economic context. Coins are among the first mass-produced objects with repeatability constraints imposed on composition, shape, size, weight and image. The image is the mark of the issuing power. For archaeologists, detailed analysis of coins are therefore crucial (Richard, Lopez, 2014, Gruel, 1981, Brousseau et al., 2009). One aim is to estimate the number of dies[1] used in the process of fabrication to estimate the volume of money supply. To do so, archaeologists compare the pattern of coins and try to see if they are struck by the same die. As we can see in Figure 1, the problem is difficult because the differences between patterns are hardly perceptible for an untrained eye (the coins are about 2 cm wide). Moreover, the coins may be damaged, worn or even fractured. Archaeologists can find thousands of coins and it would take too much time to compare every pair of coins. That's why, they need automatic tools to help them to know whether the coins come from the same die or not. To solve this challenging task, we propose a rather simple but effective method to know automatically whether two coins come from the same die and to cluster them. To recognize dies, we first align the patterns using a registration algorithm. Then, by analyzing the histograms of point to point distance maps, we can tell whether two coins are struck by the same dies or not. With all these comparisons, it is possible then to cluster coins that have the same die. Registration algorithms are algorithms which align 3D point cloud and are usually used for reconstruction, SLAM (Simultaneous Odometry And Mapping) or even 3D object tracking (Vongkulbhisal et al., 2017). But aligning partially scanned scenes or 3D models and aligning patterns are different. Indeed, registration algorithms usually align shapes, whereas in our problem, we want to align patterns in arbitrary shapes. That's why we need to adapt registration algorithms to perform this task.

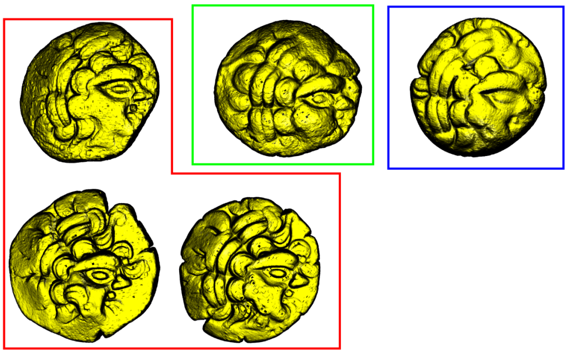


Figure 1. Example of 3D point cloud model of coins. The 3 coins framed in red have the same die. The two other coins have been struck each by a different die. You can see the similarity of the pattern for all coins.

Our contribution is the following:

- We propose a method that performs accurate coin die clustering with small annotated dataset

In order to realize this objective, we adapted the registration algorithm for the pattern alignment, and we also define a metric to know whether two patterns are similar. Extensive experiments have been performed to show the interest of our method.

2. RELATED WORK

2.1 Pattern recognition applied on point cloud

Researchers like (Zambanini et al., 2014, Zambanini, Kampel, 2013, Schlag, Arandjelovic, 2017, Salgado, 2016) usually work with images because it is easier to acquire images and pattern recognition techniques work very well on images. According

[1] A die is a stamping tool containing the image of the pattern. With a high pressure, pattern can be stamped in the coin

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2020-973-2020 | © Authors 2020. CC BY 4.0 License.

to (Salgado, 2016) and to our knowledge, no work has been published on 3D scans of coins for classification or clustering. Moreover, researchers usually try to get a rough classification of the very different patterns. But in our dataset, coins have the same class of patterns and we want to know if they come from the same die or not (a sample of our dataset can be found in Figure 1). In our problem, it is much harder to use RGB images. In Figure 2, we can see rust in the coins and other color artifacts. With 3D scanner, we have access to the geometry of the pattern. Therefore, it seems to us that it is easier to solve the problem with very accurate 3D data instead of images. (Salgado, 2016) performs classification of coins with descriptors, using a popular method like SIFT (Lowe, 2004). With descriptors, they can use Bag of Visual Word for each image and use SVM for the classification. (Schlag, Arandjelovic, 2017) use Deep Learning to perform classification on image. But direct classification is ineffective in our case because we don't know in advance the number of classes and, we don't have so many labeled data. To our knowledge, no work has been done on coin die clustering.

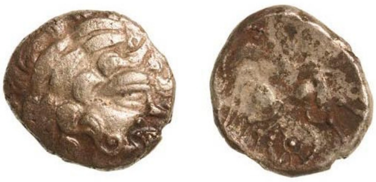


Figure 2. Example of a RGB image of both side of a coin.

2.2 Point Cloud Registration

Point Cloud registration[2] is still a hot topic in the literature and there are still a lot of challenges unsolved. Usually, registration methods are classified in two families: local registration and global registration. Local registration is an easier problem because we suppose that the 3D point cloud are almost aligned.

Local registration The most popular method in local registration is Iterative Closest Point (Besl, McKay, 1992, Chen, Medioni, 1992) and a lot of variants exist (Chen, Medioni, 1992, Rusinkiewicz, Levoy, 2001, Pavlov et al., 2017, Chetverikov et al., 2002). For example, (Babin et al., 2019) compares different cost functions, in order to make ICP robust to outliers. But in our case, we want to align patterns. Outliers are the border of the coins here so, the outliers are not random points in our case. An other family of method uses probability density distribution (especially Gaussian mixture) to model the point cloud (Tsin, Kanade, 2004, Jian, Vemuri, 2011, Myronenko, Song, 2010, Horaud et al., 2011, Evangelidis et al., 2014, Gao, Tedrake, 2019, Eckart et al., 2018). If the point cloud is modeled as a Gaussian mixture, the transformation can be found using Expectation Maximization algorithm. The advantage of Gaussian mixture model methods is that it's more robust to noise and outliers. It is also possible to adapt the method to irregular point cloud, see (Järemo Lawin et al., 2018) for more details. But the main drawback is the slowness of the method so it only works for small point cloud, even if recently some works (Gao, Tedrake, 2019, Eckart et al., 2018) have addressed this problem. Recently, some Deep Learning approaches like (Aoki et al., 2019) or (Lu et al., 2019) have been developed for point cloud registration (mainly registration of scene and 3D models). But the main drawback is that we need a lot of annotated data which is not our case.

Global registration The previous methods only perform local registration ie the initialization needs to be closed to the right solution. Descriptors like FPFH (Hutchison et al., 2010), SHOT (Hutchison et al., 2010), Spin Image (Johnson, Hebert, 1999) can be used to compute matches between the two points cloud or more and then robust estimation (RANSAC or M-Estimator (Zhou et al., 2016)) can be performed to keep inliers and exclude outliers. GoICP (Yang et al., 2016) performs branch and bound to compute the right transformation. It divides the space of solution into hypercube and then it computes a bound. Then with respect to the bound, it excludes part of the search space or it subdivides again the space. GOGMA (Campbell, Petersson, 2016) performs branch and bound methods but models the point clouds using Gaussian Mixture models. (Liu et al., 2018, Yang, Carlone, 2019) improves this method by searching only translations. (Yang, Carlone, 2019) also use a truncated least square to be robust to outliers. It reduces the dimension of the solution search, so it is faster. Then, it is easier to find the right rotation. The main drawback with branch and bound methods is that a good bound is needed to exclude candidates. In the case of flat objects like coin, we can reduce the dimension of search space but it is more difficult to find the appropriate bound. Discriminative optimization (Vongkulbhisal et al., 2017) and inverse discriminative optimization (Vongkulbhisal et al., 2018) learn a sequence of update that lead to a stationary transformation instead of optimizing a cost function.

3. METHOD

To perform coin die recognition on a coin set, our method consists of 3 main steps:

1. Pattern matching using registration
2. Automatic computing of probability that two coins come from the same die
3. Coin clustering

The whole method can be summed up in Figure 3.

3.1 Registration algorithm

Local Registration Let X and Y, be point cloud of size N and M, we want to align. The goal is to find the translation and the rotation such that:

$$R^*, t^* = \underset{R \in SO(3), t \in \mathbb{R}^3}{\arg\min} \sum_{i=1}^{N} \sum_{i=1}^{M} \delta(x_i, y_j) ||Rx_i + t - y_j||^2 \quad (1)$$

where δ is a function which is equal to 1 if the points are matched and 0 if the points are not matched. In the case of Gaussian Mixture model, δ is a soft assignment function (Myronenko, Song, 2010). For the assignment between X and Y, we usually use the nearest neighbor. Equation 1 can be rewritten as:

$$(R^*, t^*) = \underset{R \in SO(3), t \in \mathbb{R}^3}{\arg\min} \sum_{i=1}^{N} ||Rx_i + t - NN_Y(Rx_i + t)||^2 \quad (2)$$

and with:

$$NN_Y(x_i) = \underset{y \in Y}{\arg\min} ||x_i - y||^2 \quad (3)$$

Where NN_Y is a function that compute the nearest neighbor on the point cloud Y. The problem with the Equation 2 is that we

[2] In this case, we mean rigid registration.

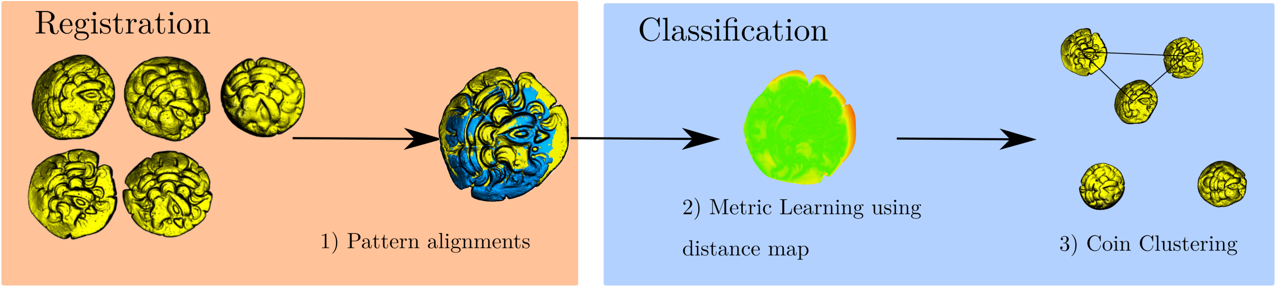


Figure 3. Pipeline of the method. First we perform registration to align patterns, then we compute a point to point distance map and we use machine learning to discriminate similar patterns from different patterns. Then we can group coins with similar patterns and coins with different patterns.

need the right matches to compute the correct transformation. The principle of ICP is the following:

- We compute matches using the nearest neighbors.
- With the matches, we can compute the right transformation using the Kabsch algorithm.

Therefore, in ICP, at each iteration, we want to solve:

$$y_i^{(n)} = NN_Y(R^{(n)}x_i + t^{(n)}) \tag{4}$$

$$(R^{(n+1)}, t^{(n+1)}) = \underset{R \in SO(3), t \in \mathbb{R}^3}{\arg\min} \sum_{i=1}^{N} ||Rx_i + t - y_i^{(n)}||^2 \tag{5}$$

An interesting variant of ICP is point to plane ICP (Chen, Medioni, 1992, Rusinkiewicz, Levoy, 2001). Instead of solving the Equation 5, we solve the following equation:

$$y_i^{(n)} = NN_Y(R^{(n)}x_i + t^{(n)}) \tag{6}$$

$$n_{y_i}^{(n)} \text{ is the normal of point } y_i^{(n)} \tag{7}$$

$$(R^{(n+1)}, t^{(n+1)}) = \underset{R \in SO(3), t \in \mathbb{R}^3}{\arg\min} \sum_{i=1}^{N} ((Rx_i + t - y_i^{(n)}).n_{y_i}^{(n)})^2 \tag{8}$$

Equation 8 has no analytic solution, but we can linearize the equation and use the Least Square Estimation to estimate the rotation and the translation. In that case, we need to suppose that the angle of rotation is small.

Due to the irregularity of the shape of each coin, we cannot apply ICP directly. Indeed, the border of coins will influence a lot the registration, whereas it doesn't belong to the patterns. But we cannot use robust methods, because, here outliers are well organized shapes and not random points. Registration algorithms are generally sensible to the shape of the 3D Point Cloud. One way to deal with this problem is to simply remove the borders because in a coin, the pattern is usually at the center of the coin. To remove the border, we can compute the center of the coin and only keep the points that are within a ball of radius r. This simple trick allow ICP to be robust to borders.

Global registration As we've said in the previous sections, local registration works if the point cloud are partially aligned. It can be useful if we want to refine manual registration but if we want a fully automatic method, a global registration method is necessary. The problem of ICP is that it finds a minimum of the equation 3 but the minimum is local. If we want to find a global solution of the equation 3, we have several possibilities:

- We can try random initializations and then for each initialization, perform an ICP. We'll call this method Random search ICP.
- We can try every initialization on a grid: we'll call it Grid Search ICP
- We can perform branch and bound optimization: it is GoICP (Yang et al., 2016).

We have tested the three methods. We will discuss the results in the experiments section, and we will see that Random Search ICP performs the best. We have tested another algorithm of global registration based on descriptors. For the 3D descriptors, FPFH descriptors can be used for example. Then, we can compute the closest feature to obtain matches, and then filter the matches using the reciprocity test. With the matches, a robust method such as RANSAC can be used to estimate the transformation and filter the correct matches. For 2D descriptors, we can project the point cloud into plane, then SIFT descriptors (Lowe, 2004) can be used. As we will see in the experiment section, descriptors did not work on our dataset.

3.2 Metric learning on patterns

Now, we suppose that patterns of two coins are well aligned. First, we compute the point to point distance map D between the two aligned point cloud X and Y, using this formula:

$$D_i = ||x_i - NN_Y(x_i)||^2 \tag{9}$$

Then, we compute the histogram of distance h from D. h is a distribution of the point to point distance between X and Y. If the patterns are well aligned we should see a lot of points with small distances (see Figure 4 for examples of point to point distance maps and their respective histograms). In order to decide whether the aligned patterns are the same or not, we propose to use machine learning procedure and just with the histogram of distance we can know if the pattern is the same or not. The machine learning algorithm will measure the probability that two patterns are similar. It can be seen as a metric learning procedure.

Machine Learning algorithm We decide to use Logistic regression to solve our problem. Let $h_1 \dots h_K$ be the histograms of a pair of coins (feature of dimension d) and let $l_1 \dots l_K$ be their respective labels (1 if it is the same pattern, 0 if it is not). The objective is to find the parameters $\theta \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such

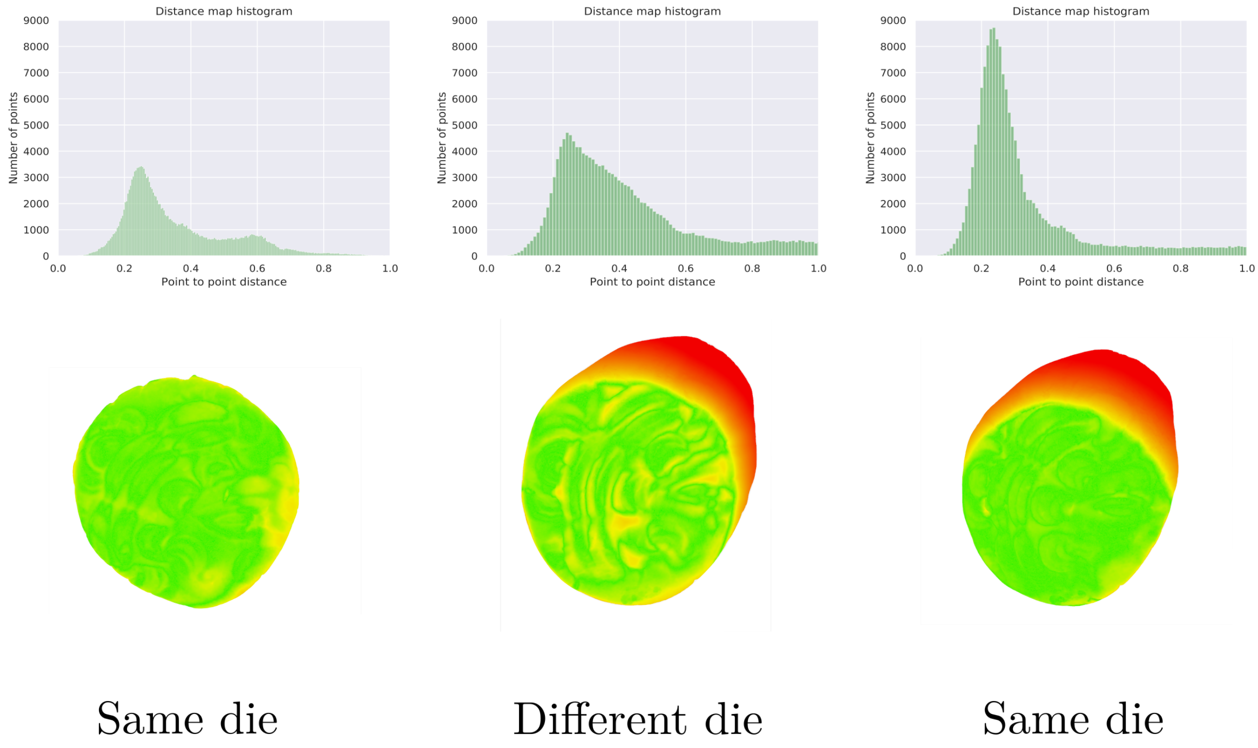


Figure 4. 3 examples of distance map between two pair of coins on the bottom and their respective histogram on the top.

that:

$$\theta^*, b^* = \underset{\theta \in \mathbb{R}^d, b \in \mathbb{R}}{\arg\min} \sum_{i=1}^{K} \log(1+\exp(-(\theta^T h_i+b))-l_i(\theta^T h_i+b)+||\theta||^2 \tag{10}$$

It can be minimized using gradient descent or Newton methods. After finding the right parameters with the training set, we can evaluate quickly and automatically the probability that the patterns are similar.

3.3 Clustering of coins

We have described a way to get the probability that two coins have the same die. With this data, we can cluster coins in different groups corresponding to the die they are coming from. Let $X_1, \dots X_n$ be the coins. We can now compute p_{ij} the probability that X_i and X_j have the same pattern. Let $P = (p_{ij})_{i=1,\dots,N, j=1\dots,N}$ the matrix of pair comparisons. P can also be seen as a weighted graph where each node is a coin. To perform graph clustering, we can threshold graph weights to get a non weighted graph. In other words, we compute :

$$A_{ij} = \mathbb{1}_{p_{ij} \geq \alpha} = \begin{cases} 1 & p_{ij} \geq \alpha \\ 0 & p_{ij} < \alpha \end{cases} \tag{11}$$

then we can compute the connected components of the graph A to group coins with the same die pattern.

4. EXPERIMENTS

In this section, we will describe the experiments we've done to evaluate our method. The main questions is: How can we evaluate the whole pipeline ? To do so, we need to evaluate the components of the pipeline separately. We will first describe how the data have been acquired, then, we will explain how to evaluate each step of our method.

4.1 Dataset

The coins have been scanned with the AICON Smartscan (with surface measurement distance error around 50 μm). 3D Point clouds have around 300k points for one side of the coin. 881 coins have been scanned but only a small part of that dataset is annotated. For 66 coins, we have all the pair transformations. We have therefore $\frac{N(N-1)}{2}$ comparisons ie 2145 comparisons for the metric learning. To evaluate the whole pipeline, we have 90 coins (we just need the labels to evaluate the coin die clustering). Because annotating this treasury manually is complicated and tedious (we remind the readers that only an expert can annotate the data), we could not evaluate our method on the whole dataset.

4.2 Implementation details

We use Python and C++ to implement our method. We use Open 3D, a modern library in C++ to implement our own version of ICP. To perform logistic regression, we used scikit-learn with the default parameters. To plot the final graph, we use the Plotly library and Networkx library.

4.3 How to evaluate registration?

In order to justify the validity of our method, we manually registered point clouds and measured the angle error (in degree) and the translation error (in mm). To measure errors, we use the following metrics:

$$\delta t_{12} = ||t_2 - t_1||_2 \tag{12}$$

$$\delta R_{12} = ||\log_{SO3}(R_1 R_2^T)||_2 \tag{13}$$

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2020-973-2020 | © Authors 2020. CC BY 4.0 License.

By knowing the typical error of manual registration, we can correctly evaluate our method. If $\delta t_{12} < \delta t_{manual}$ and if $\delta R_{12} < \delta R_{manual}$, we can consider that the automatic registration succeed. For the rest of the experiments, we'll use $\delta R_{manual} = 5$ degrees and $\delta t_{manual} = 0.5mm$ which is a good order of magnitude.

Evaluation of local registration To evaluate local registration, we've used 100 pairs of coins with their transformations computed manually. For the local registration problem, we've evaluated different variants of the ICP to see which parameters are important:

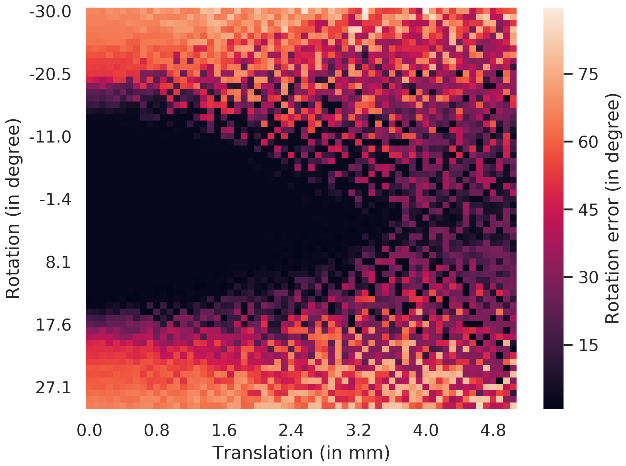


Figure 5. Convergence basin of ICP with respect to the initialization for pair of coin coming from the same die. In absciss axis, the norm of the translation varies from 0 to 5 mm. In ordinate axis, the rotation varies from -30 to 30 degrees. The dark part in this map represents the convergence basin of ICP.

1. The influence of border
2. The influence of the cost function (Point to Point or Point to Plane)

We can see in Figure 6 the influence of border exclusion and that point to plane ICP is better than point to point ICP. Indeed, according to (Rusinkiewicz, Levoy, 2001), point to plane ICP work very well on flat structures. By removing the borders of the coins, we have lower errors. But for some coins, we can see that error is still big even with point to plane and by removing borders. Indeed, if the patterns are not similar (coming from different dies), the registration will fail because we're trying to registrate different patterns. But if the patterns are coming from the same die, by using the criterion defined above, **98%** of coins are correctly registered. An ICP can be done in approximately 300 ms. For all other experiments, we keep the best radius for border exclusion as 8 mm.

Influence of the initialization To evaluate the influence of the initialization, we tried to compute the convergence basin of ICP. To what extent does ICP converge to the right solution? We can see in Figure 5, a visualization of the convergence basin by varying two parameters: for the translation, we've taken a random direction and the norm of this translation has been varied. Also, the rotation around the z axis has been varied. We can conclude that if the distance between the initialization and the global minimum is less that 10 degrees in rotation and 3 mm in translation, we are nearly sure that we will converge with ICP. This information is very useful for the global registration.

Comparison between different global methods We've experimented different algorithms for the global registration.

Because, we work on flat point clouds, we can reduce the dimension of space search (instead of 6 degrees of freedom, we have 3 degrees). In Table 1, we can see which global algorithms converge to the right solution. The descriptors approach doesn't work here. One possible reason is that the coin contains repetitive patterns. The descriptors are too local therefore, it is hard to take into account the global structure of the pattern. Even when there are some missing parts or when the coin is partially damaged, the registration still succeed in aligning patterns. GoICP performs also well but it is too slow, we've realized that GoICP was equivalent to Grid Search because, the bound was not accurate enough to exclude some candidates. Random search here has the best results. In Figure 7, we can see the registration of pair of coins using Random Search ICP.

4.4 Metric learning evaluation

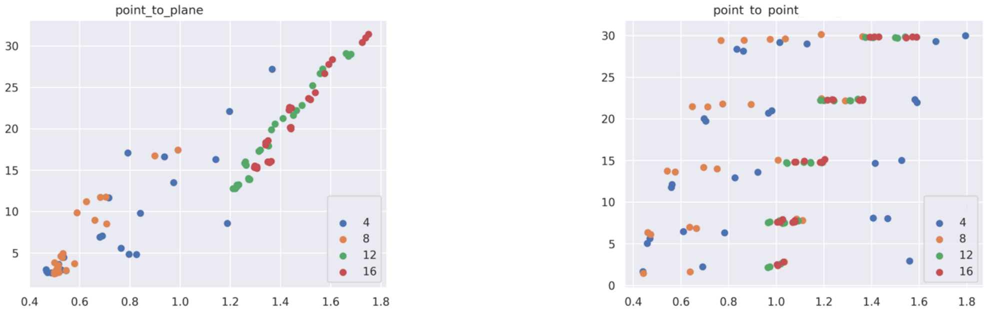
We have seen in the previous section that the decision whether two coins have the same die is a problem of binary classification. To correctly evaluate this part, we have aligned by hand the patterns, then we perform an ICP to refine and then, we can test if two patterns are similar or not. As it said in previous section, the dataset is composed of 2145 comparisons so 2145 histograms. In our dataset, we have 1436 negative labels and 709 positive labels. We divide our dataset randomly in a training set and a test set: half of our dataset is used for training, the other half is used for testing. We can simply compute the accuracy on the test set to see if our method is correct or not. When the patterns are correctly registered using ICP, the accuracy is **98%** on the test set. But if we registrate the patterns by hand the accuracy drop to **72%** on the test set. It shows that ICP allows a better repeatability in decision. We can see that the method is very effective when the patterns are well aligned. When patterns are the same, the histograms of distances are good features to see if the patterns are similar or not. But it will fail when the coins are not well aligned. We have also shown that the metric learning part doesn't need a huge dataset to learn an appropriate metric.

4.5 Clustering evaluation

We did the evaluation on 90 labeled coins. This part is the most difficult part to evaluate because, the treasury is not fully labeled yet and also, the archaeologists could not have done all the comparisons between coins. We can see in Figure 8 a representation of the graph. Each vertex is a coin and the connected components are a die. Normally, every cluster should be a complete graph (ie all coins in a cluster connected two by two), but the registration can sometimes fail. Also, if the coin is damaged, we can see that the connected component graph will not be complete. Indeed, if a coin is damaged or fractured, the algorithm can fail recognizing the die. Quantitatively, there are several methods to evaluate our clustering method,we decide to use the adjusted rand index (ARI) (Hubert, Arabie, 1985). It is defined as :

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{14}$$

RI is called the rand index. ARI is more relevant because, for random labels, the value is closed to 0. The best value we can



(a) Point to Plane ICP

(b) Point to Point ICP

Figure 6. Translation error (absciss axis) and rotation error (ordinate axis) with different sizes of border exclusion (the radius is in cm). Each point represents the mean of error on pairs of coin with 100 random initializations. We can remark that Point to Point ICP doesn't converge. Point to Plane ICP converges only if we exclude the border but if we exclude too much, it doesn't work. The best radius in our experiments is 8 mm.

	Error < 1 deg	Error < 5 deg	Error < 0.5 mm
2D descriptors	<1%	<1%	<1%
3D descriptors	<1%	<1%	<1%
grid search	18%	84%	75%
goICP	18%	84%	75%
random search	**22%**	**94%**	**86%**

Table 1. Success rate in % of global registration. We measure the error using Equation 12 and 13. For grid search and random search, we perform the same number of trial (hundred of trials). GoICP has the same results as grid search if we limit the number of iterations for GoICP.

have is 1. In our case, the ARI has a value of **0.86**. Most of the time, the method is correct but there are still some mistakes.

Where do the mistakes come from ? As we have said earlier, some coins are damaged or even fractured, it is therefore more difficult to registrate the patterns. Also, the metric learning method can indicate that two coins with the same die are different, because one coin is too damaged or has a fracture. These particular cases explain why there are still some mistakes and each sub-graph is not always complete in Figure 8. Even if there are still some errors, the clusters are still meaningful and can be useful for experts.

5. CONCLUSION AND DISCUSSIONS

This work introduces the challenging problem of coin die clustering and a method to solve it on real data. To our knowledge, no automatic method exists in the literature to cluster patterns on 3D point cloud that only experts can identify. To do so, we can perform coin registration no matter the shape of coins. Then, with small annotated dataset, we can learn an accurate metric which gives a probability that two coins have the same die. Finally, we can obtain clusters of coins and, they are relevant for archaeologists. This work will allow them to analyze automatically treasuries and to know how many dies have been used at that time. In future work, we are interested in generalizing pattern recognition on every kind of shapes and not just coins.

REFERENCES

Aoki, Y., Goforth, H., Arun Srivatsan, R., Lucey, S., 2019. Pointnetlk: Robust and efficient point cloud registration using pointnet. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Babin, P., Giguère, P., Pomerleau, F., 2019. Analysis of robust functions for registration algorithms. *2019 International Conference on Robotics and Automation (ICRA)*, 1451–1457.

Besl, P. J., McKay, N. D., 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2), 239–256. https://doi.org/10.1109/34.121791.

Brousseau, L., Téreygeol, F., Faucher, T., Arles, A., 2009. À la recherche des ateliers monétaires grecs : l'apport de l'expérimentation.

Campbell, D., Petersson, L., 2016. GOGMA: Globally-Optimal Gaussian Mixture Alignment. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, 5685–5694.

Chen, Y., Medioni, G., 1992. Object Modelling by Registration of Multiple Range Images. *Image Vision Comput.*, 10(3), 145–155. https://doi.org/10.1016/0262-8856(92)90066-C.

Chetverikov, D., Svirko, D., Stepanov, D., Krsek, P., 2002. The Trimmed Iterative Closest Point algorithm. *Object recognition supported by user interaction for service robots*, 3, IEEE Comput. Soc, Quebec City, Que., Canada, 545–548.

Eckart, B., Kim, K., Kautz, J., 2018. Fast and Accurate Point Cloud Registration using Trees of Gaussian Mixtures. *arXiv:1807.02587 [cs]*. http://arxiv.org/abs/1807.02587. arXiv: 1807.02587.

Evangelidis, G. D., Kounades-Bastian, D., Horaud, R., Psarakis, E. Z., 2014. A Generative Model for the Joint Registration of Multiple Point Sets. D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (eds), *Computer Vision – ECCV 2014*, 8695, Springer International Publishing, Cham, 109–122.

Gao, W., Tedrake, R., 2019. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. *Conference On Computer Vision and Pattern Recognition (CVPR)*.

Gruel, K., 1981. Le trésor de Trébry (Côtes-du-Nord) - Ier siècle avant notre ère (Études de Numismatique celtique 1). *Collection de l'Institut des Sciences et Techniques de l'Antiquité*.

Horaud, R., Forbes, F., Yguel, M., Dewaele, G., Jian Zhang, 2011. Rigid and Articulated Point Registration with Expectation Conditional Maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3), 587–602. http://ieeexplore.ieee.org/document/5444893/.

Hubert, L., Arabie, P., 1985. Comparing partitions. *Journal of Classification*, 2(1), 193–218. https://doi.org/10.1007/BF01908075.

Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Tombari, F., Salti, S., Di Stefano, L., 2010. Unique Signatures of Histograms for Local Surface Description. K. Daniilidis, P. Maragos, N. Paragios (eds), *Computer Vision – ECCV 2010*, 6313, Springer Berlin Heidelberg, Berlin, Heidelberg, 356–369.

Jian, B., Vemuri, B. C., 2011. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 1633-1645.

Johnson, A. E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 433–449.

Järemo Lawin, F., Danelljan, M., Shahbaz Khan, F., Forssén, P.-E., Felsberg, M., 2018. Density adaptive point set registration. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y., Wang, C., Song, Z., Wang, M., 2018. Efficient Global Point Cloud Registration by Matching Rotation Invariant Features Through Translation Search. V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (eds), *Computer Vision – ECCV 2018*, 11216, Springer International Publishing, Cham, 460–474.

Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91–110. http://link.springer.com/10.1023/B:VISI.0000029664.99615.94.

Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., Song, S., 2019. DeepICP: An End-to-End Deep Neural Network for 3D Point Cloud Registration. *CoRR*, abs/1905.04153. http://arxiv.org/abs/1905.04153.

Myronenko, A., Song, X., 2010. Point-Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), 2262–2275. http://arxiv.org/abs/0905.2635. arXiv: 0905.2635.

Pavlov, A. L., Ovchinnikov, G. V., Derbyshev, D. Y., Tsetserukou, D., Oseledets, I. V., 2017. AA-ICP: Iterative Closest Point with Anderson Acceleration. *arXiv:1709.05479 [cs]*. http://arxiv.org/abs/1709.05479. arXiv: 1709.05479.

Richard, J.-C., Lopez, C., 2014. Technique moderne de reconstitution d'empreintes monétaires. Application à un type monétaire préaugustéen des Rutènes.

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 145-152.

Salgado, L., 2016. Medieval coin automatic recognition by computer vision. *computer vision*, 98.

Schlag, I., Arandjelovic, O., 2017. Ancient Roman Coin Recognition in the Wild Using Deep Learning Based Recognition of Artistically Depicted Face Profiles. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, IEEE, Venice, 2898–2906.

Tsin, Y., Kanade, T., 2004. A Correlation-Based Approach to Robust Point Set Registration. T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, T. Pajdla, J. Matas (eds), *Computer Vision - ECCV 2004*, 3023, Springer Berlin Heidelberg, Berlin, Heidelberg, 558–569.

Vongkulbhisal, J., Torre, F. D. l., Costeira, J. P., 2017. Discriminative Optimization: Theory and Applications to Point Cloud Registration. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, 3975–3983.

Vongkulbhisal, J., Ugalde, B. I., De la Torre, F., Costeira, J. P., 2018. Inverse composition discriminative optimization for point cloud registration. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2993–3001.

Yang, H., Carlone, L., 2019. A polynomial-time solution for robust registration with extreme outlier rates. Technical report. arxiv preprint: 1903.08588.

Yang, J., Li, H., Campbell, D., Jia, Y., 2016. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11), 2241–2254. http://arxiv.org/abs/1605.03344. arXiv: 1605.03344.

Zambanini, S., Kampel, M., 2013. Coarse-to-Fine Correspondence Search for Classifying Ancient Coins. J.-I. Park, J. Kim (eds), *Computer Vision - ACCV 2012 Workshops*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 25–36.

Zambanini, S., Kavelar, A., Kampel, M., 2014. Classifying Ancient Coins by Local Feature Matching and Pairwise Geometric Consistency Evaluation. *2014 22nd International Conference on Pattern Recognition*, IEEE, Stockholm, Sweden, 3032–3037.

Zhou, Q.-Y., Park, J., Koltun, V., 2016. Fast Global Registration. B. Leibe, J. Matas, N. Sebe, M. Welling (eds), *Computer Vision – ECCV 2016*, 9906, Springer International Publishing, Cham, 766–782.

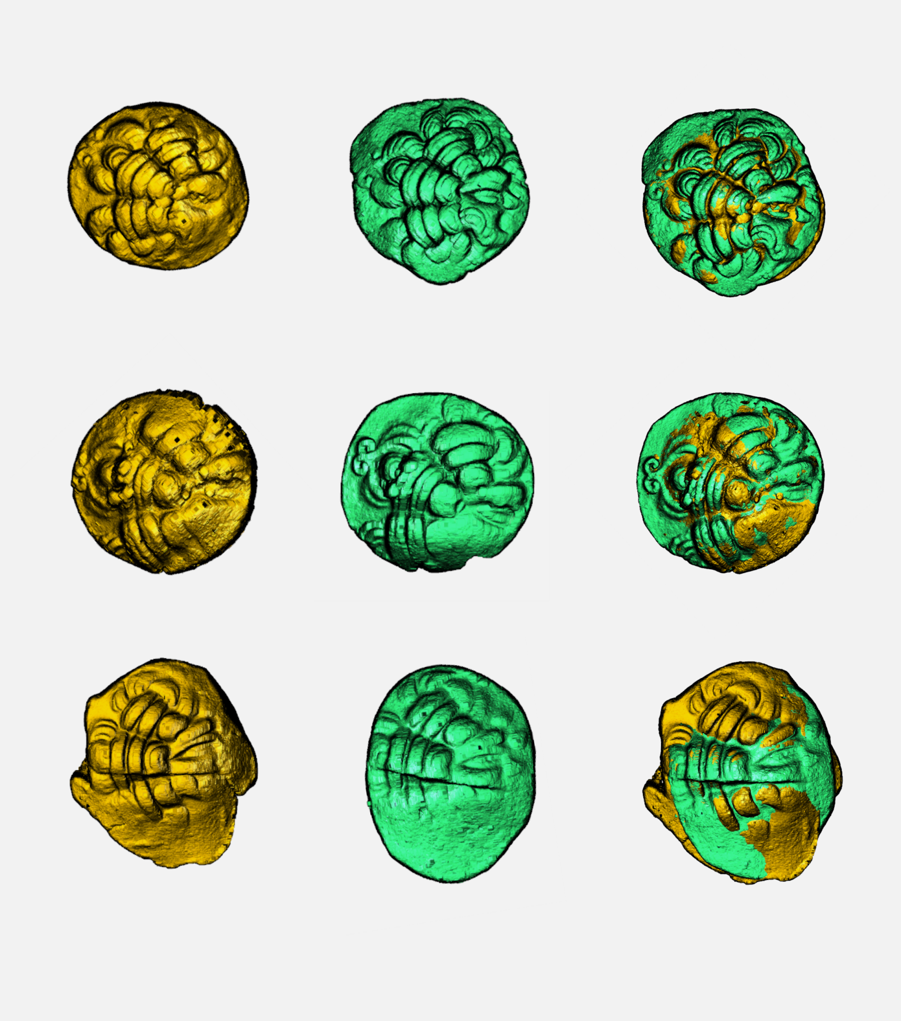


Figure 7. Qualitative results of global registration of pair of coins using Random Search ICP.

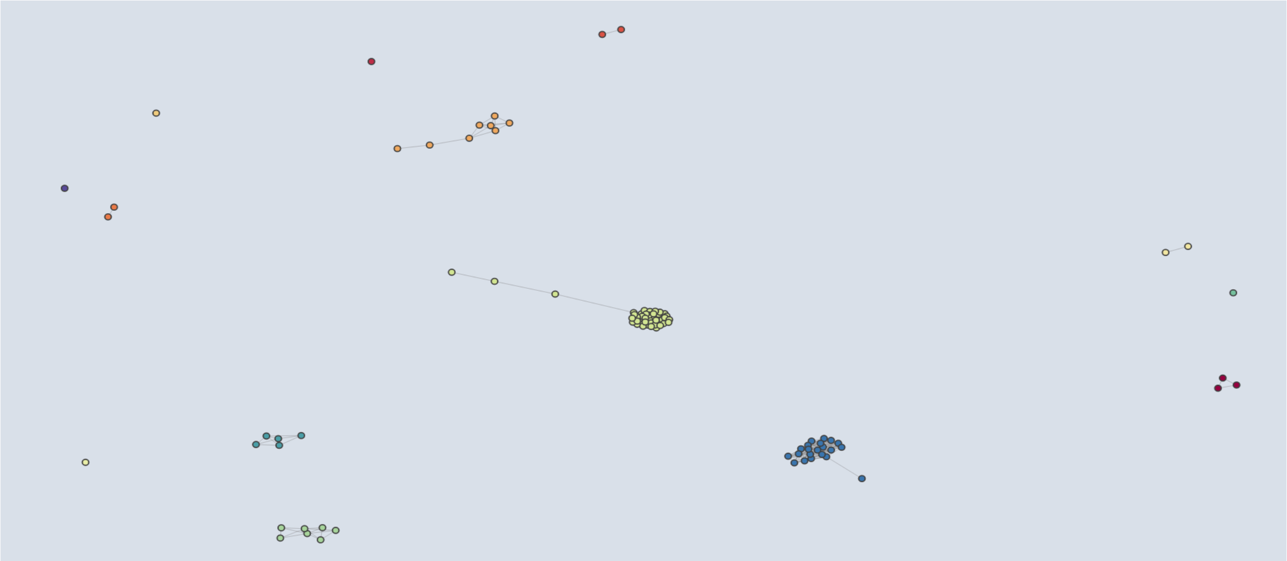


Figure 8. Clustering of 90 coins. Each node represents a coin and coins of the same cluster are linked together and have the same color.