

# Reconnaissance automatique de motifs sur des nuages de points 3D et application dans l'art celtique

Sofiane Horache<sup>1</sup>, François Goulette<sup>1</sup>, Jean-Emmanuel Deschaud<sup>1</sup>, Thierry Lejars<sup>2</sup>,  
Katherine Gruel<sup>2</sup>

<sup>1</sup> Centre de Robotique, CAOR, Mines Paristech <sup>2</sup> - AOROC, ENS Ulm

06 février 2020



# Contexte

# Introduction



(a)



(b)



(c)

FIGURE

# Pièces de Monnaie



FIGURE – Pièces de monnaie

# Objectifs

But : mieux comprendre l'artisanat celte.

# Objectifs

But : mieux comprendre l'artisanat celte.

- ▶ Combien de motifs ont été frappés par le même poinçon (ou coin) ?
- ▶ Combien de poinçons (coin) nécessaires dans la fabrication d'un objet ou d'une série ?
- ▶ Dans le cas des pièces, reconstitution du coin possible ?

# Objectifs

But : mieux comprendre l'artisanat celte.

- ▶ Combien de motifs ont été frappés par le même poinçon (ou coin)?
- ▶ Combien de poinçons (coin) nécessaires dans la fabrication d'un objet ou d'une série?
- ▶ Dans le cas des pièces, reconstitution du coin possible?

Nous allons nous concentrer sur les pièces de monnaie.

## Objectifs techniques

Nous disposons d'un trésor d'environ 1000 pièces. Nous cherchons à **automatiser** la reconnaissance de coin.

## Objectifs techniques

Nous disposons d'un trésor d'environ 1000 pièces. Nous cherchons à **automatiser** la reconnaissance de coin.

- ▶ Quelles pièces ont été fabriquées par le même coin ?
- ▶ Reconstitution du motif en entier.

## Objectifs techniques

Nous disposons d'un trésor d'environ 1000 pièces. Nous cherchons à **automatiser** la reconnaissance de coin.

- ▶ Quelles pièces ont été fabriquées par le même coin ?
- ▶ Reconstitution du motif en entier.

Données : scan 3D des pièces (nuage de point, écart-type sur mesure distance environ  $50 \mu\text{m}$ )

## Pourquoi c'est difficile?

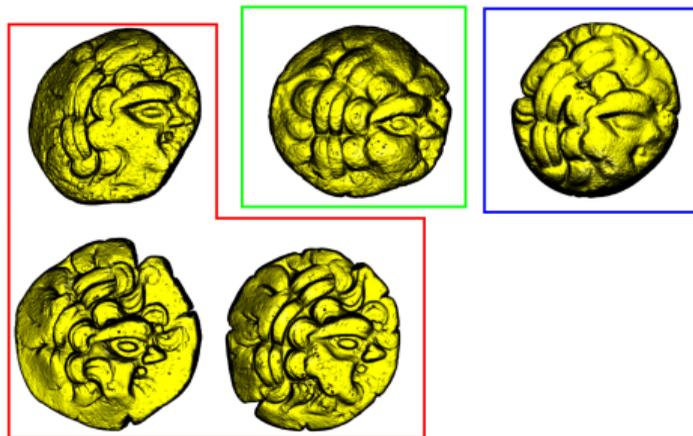


FIGURE – Exemple de scans de pièces

## Pourquoi c'est difficile?

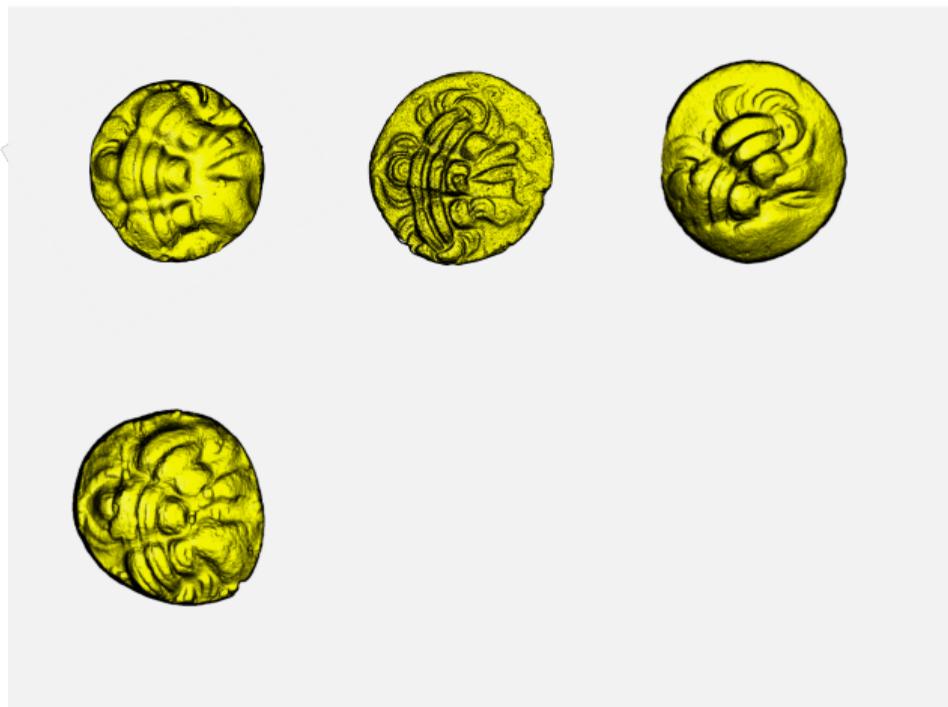


FIGURE – Pièces abîmées, fracturées

## Pourquoi c'est difficile?



0504\_07-03813.jpg

FIGURE – Pièces rouillées

## Pourquoi c'est difficile?

- ▶ Motifs très proche (indistinguable pour un œil non entraîné)
- ▶ Pièces fracturées
- ▶ Comparaison deux à deux (trésor de 1000 pièces = 500 000 comparaisons)
- ▶ On ne dispose que de 92 pièces labellisées

# Chaîne de traitement

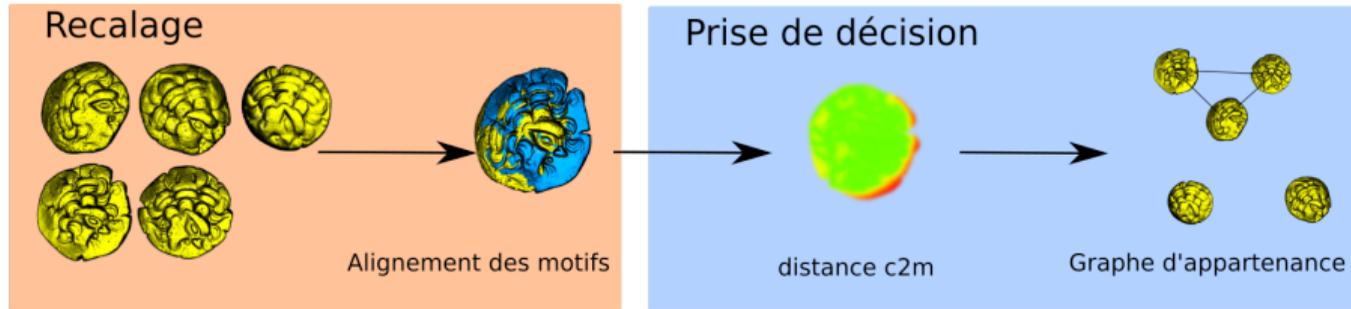


FIGURE – Chaîne de traitement pour la reconnaissance de coin

# Recalage



# C'est quoi le recalage?

## But

- ▶ Soit  $X$  et  $Y$  deux nuages de points 3D.
- ▶ Le but est de trouver une transformation  $T$  qui permet d'aligner  $X$  et  $Y$ .

# C'est quoi le recalage?

## But

- ▶ Soit  $X$  et  $Y$  deux nuages de points 3D.
- ▶ Le but est de trouver une transformation  $T$  qui permet d'aligner  $X$  et  $Y$ .

## Transformations rigide

C'est une rotation  $R \in SO(3)$ , une translation  $t \in \mathbb{R}^3$ .

# Principe

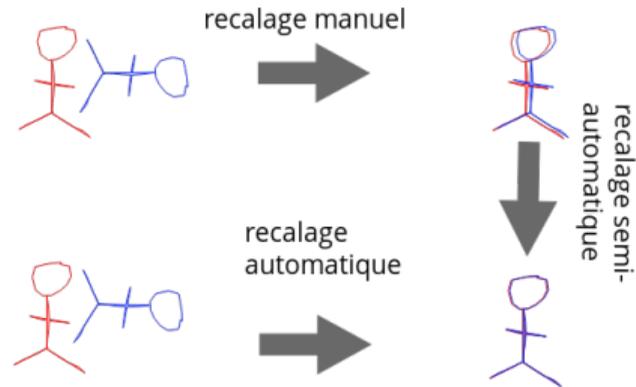


FIGURE – Différents types de recalages

# Iterative Closest Point

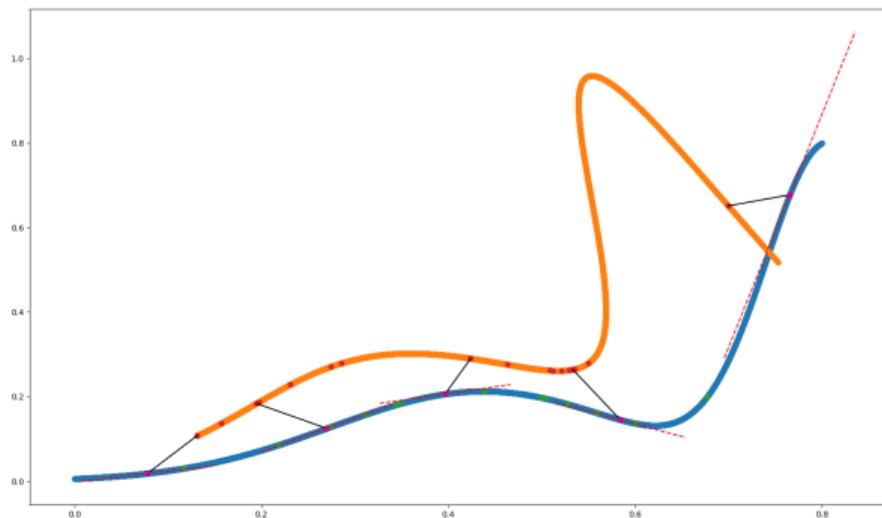


FIGURE – ICP en 2D

# ICP

- ▶ Rapide
- ▶ Facile à comprendre et à coder
- ▶ Plein de variantes
- ▶ Donne de bons résultats localement
- ▶ Mais la convergence est locale...

# Problèmes

- ▶ Habituellement, on recale des formes et pas des motifs...
- ▶ ICP ne marche que localement

# Problèmes

- ▶ Habituellement, on recale des formes et pas des motifs...
- ▶ ICP ne marche que localement

## Idée

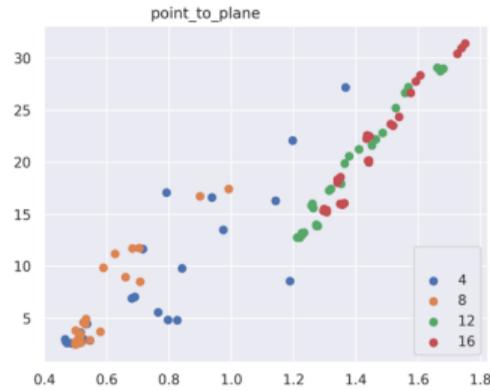
- ▶ Supprimer les bords
- ▶ Tester plusieurs initialisations

# Résultats

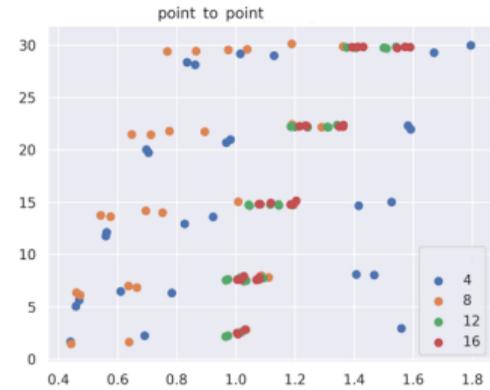


FIGURE – Résultats qualitatifs

# Résultats



(a) Point to Plane ICP



(b) Point to Point ICP

FIGURE – Moyenne d'erreur de rotation (axe des ordonnées) et translation (axes des Abscisses) pour 100 paires de pièces

## Apprentissage d'une probabilité

# Comparaison

Comment savoir si deux pièces recalées ont le même coin ?

# Carte de distance

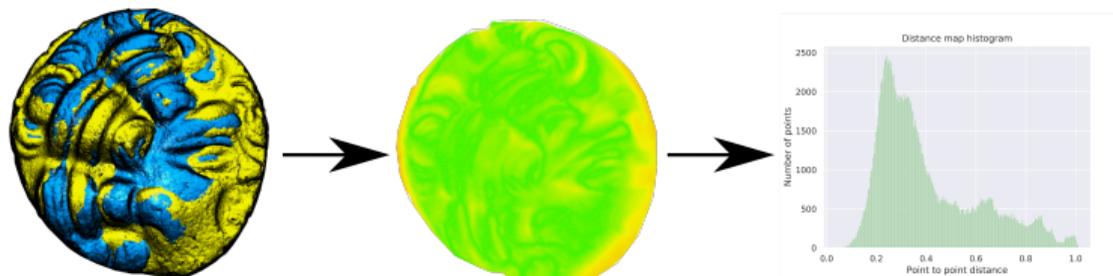


FIGURE – 3 exemples de carte de distances avec les histogrammes

# Histogramme

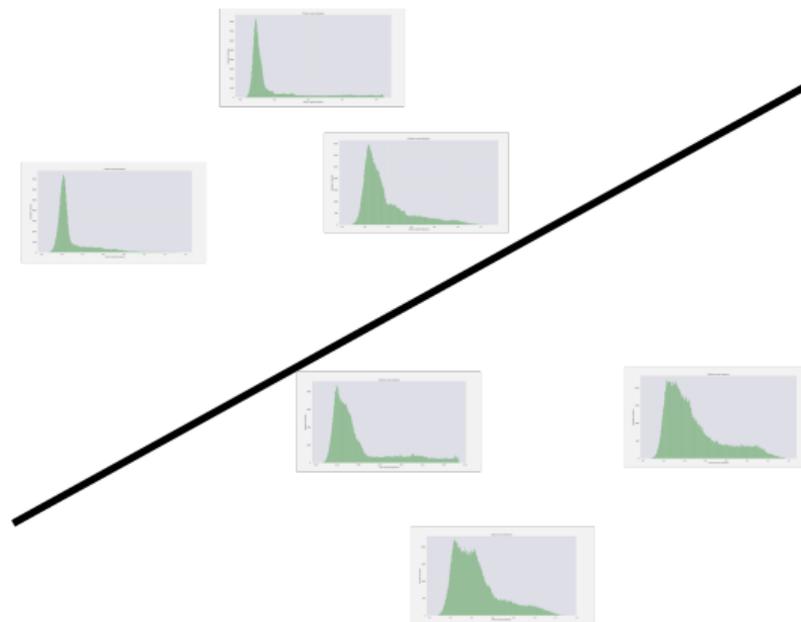


FIGURE – Classification binaire des histogrammes

## Résultats

- ▶ On dispose de 2145 paires de pièces (66 pièces) dont on connait la transformation pour les recalcr.

## Résultats

- ▶ On dispose de 2145 paires de pièces (66 pièces) dont on connait la transformation pour les recalcr.
- ▶ On coupe le jeu de données en un jeu de données l'entraînement et de test.

## Résultats

- ▶ On dispose de 2145 paires de pièces (66 pièces) dont on connait la transformation pour les recaler.
- ▶ On coupe le jeu de données en un jeu de données l'entraînement et de test.
- ▶ Le score est de **98%** pour la comparaisons deux à deux.
- ▶ Avec le recalage manuel uniquement, le score tombe à **72 %**.

## Résultats

- ▶ On dispose de 2145 paires de pièces (66 pièces) dont on connaît la transformation pour les recalés.
- ▶ On coupe le jeu de données en un jeu de données d'entraînement et de test.
- ▶ Le score est de **98%** pour la comparaison deux à deux.
- ▶ Avec le recalage manuel uniquement, le score tombe à **72 %**.

Pour regrouper les pièces du même coin, on peut mettre un seuil sur la probabilité et obtenir un graphe.

# Clustering

# Résultats

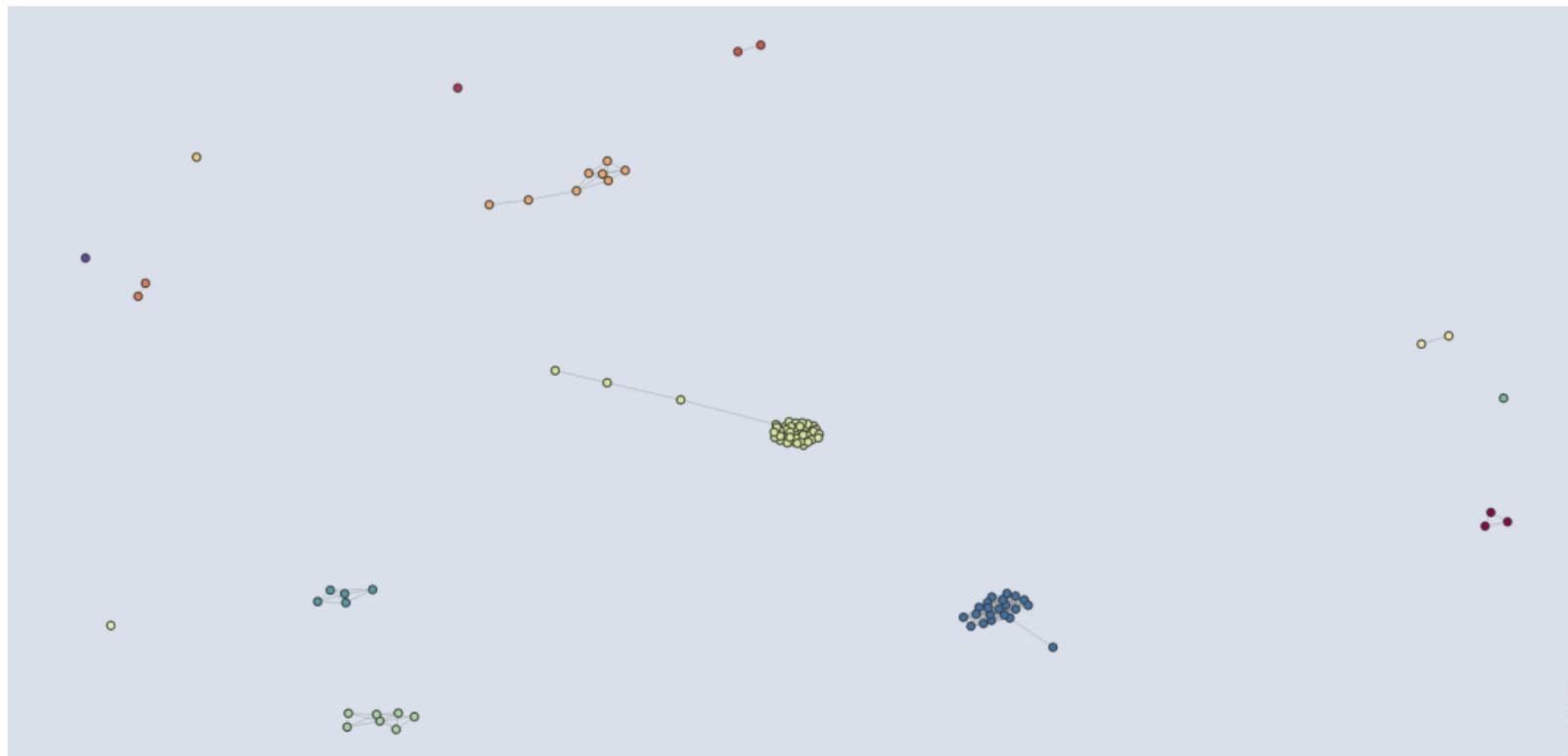


FIGURE – Clustering des 90 pièces.



## Analyse du graphe

- ▶ Le clustering est quasiment similaire à celui de l'expert (Adjusted Rand Index = **0.89**)
- ▶ On voit des points isolés (pièce à coin unique)
- ▶ On trouve des graphes presque complets (pièces appartenant à un même coin)
- ▶ On trouve des graphes non complets (?)

## Analyse du graphe

- ▶ Le clustering est quasiment similaire à celui de l'expert (Adjusted Rand Index = **0.89**)
- ▶ On voit des points isolés (pièce à coin unique)
- ▶ On trouve des graphes presque complets (pièces appartenant à un même coin)
- ▶ On trouve des graphes non complets (?)

Pourquoi des sous-graphes non complets ?

# Conclusion

## Conclusion

Nous avons une simple méthode qui permet de répondre à une problématique difficile données par les archéologues :

- ▶ Fonctionne en 3 étapes.
- ▶ Donne un résultat cohérent par rapport à un expert.
- ▶ Automatique et nécessite peu de données labellisées.

## Conclusion

Nous avons une simple méthode qui permet de répondre à une problématique difficile données par les archéologues :

- ▶ Fonctionne en 3 étapes.
- ▶ Donne un résultat cohérent par rapport à un expert.
- ▶ Automatique et nécessite peu de données labellisées.
- ▶ Mais ne se généralise pas à d'autres objets de forme plus arbitraire

# Des Questions ?

# Évaluation

Un algorithme quelconque nous donne  $R$  et  $t$ . Alors pour les évaluer, on calcule ces erreurs :

$$\delta t = ||t - t_{man}|| \quad (1)$$

$$\delta R = ||\log_{SO3}(RR_{man}^T)|| \quad (2)$$

# Évaluation

Un algorithme quelconque nous donne  $R$  et  $t$ . Alors pour les évaluer, on calcule ces erreurs :

$$\delta t = ||t - t_{man}|| \quad (1)$$

$$\delta R = ||\log_{SO3}(RR_{man}^T)|| \quad (2)$$

1. L'erreur de translation se mesure en millimètre.
2. L'erreur de rotation se mesure en degré

# Évaluation

Un algorithme quelconque nous donne  $R$  et  $t$ . Alors pour les évaluer, on calcule ces erreurs :

$$\delta t = ||t - t_{man}|| \quad (1)$$

$$\delta R = ||\log_{SO3}(RR_{man}^T)|| \quad (2)$$

1. L'erreur de translation se mesure en millimètre.
2. L'erreur de rotation se mesure en degré

On peut comparer le recalage manuel et automatique.

## Recalage local

On ne sais pas comment les points se correspondent.

## Recalage local

On ne sais pas comment les points se correspondent.

- Pour avoir de bonne correspondances, il faut une bonne transformation!

## Recalage local

On ne sais pas comment les points se correspondent.

- ▶ Pour avoir de bonne correspondances, il faut une bonne transformation!
- ▶ Pour avoir une bonne transformation..., il faut de bonne correspondances...

## Recalage local

On ne sais pas comment les points se correspondent.

- ▶ Pour avoir de bonne correspondances, il faut une bonne transformation!
- ▶ Pour avoir une bonne transformation..., il faut de bonne correspondances...

Idée : les plus proches voisins comme correspondances (Iterative Closest Point) :

## Recalage local

On ne sais pas comment les points se correspondent.

- ▶ Pour avoir de bonne correspondances, il faut une bonne transformation !
- ▶ Pour avoir une bonne transformation..., il faut de bonne correspondances...

Idée : les plus proches voisins comme correspondances (Iterative Closest Point) :

$$y_i^{(n)} = NN_Y(R^{(n)}x_i + t^{(n)}) \quad (3)$$

$$(R^{(n+1)}, t^{(n+1)}) = \arg \min_{R \in SO(3), t \in \mathbb{R}^3} \sum_{i=1}^N \|Rx_i + t - y_i^{(n)}\|^2 \quad (4)$$

## Point à Plan

On minimise la distance "point à plan"

$$y_i^{(n)} = NN_Y(R^{(n)}x_i + t^{(n)}) \quad (5)$$

$$n_{y_i}^{(n)} \text{ is the normal of point } y_i^{(n)} \quad (6)$$

$$(R^{(n+1)}, t^{(n+1)}) = \arg \min_{R \in SO(3), t \in \mathbb{R}^3} \sum_{i=1}^N ((Rx_i + t - y_i^{(n)}) \cdot n_{y_i}^{(n)})^2 \quad (7)$$

Marche beaucoup mieux sur des structures plates!!

# Convergence d'ICP local

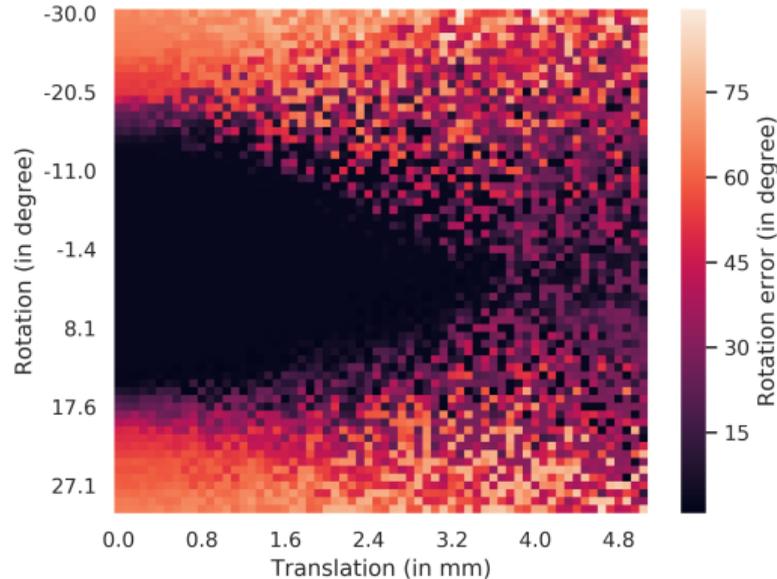


FIGURE – Bassin de convergence d'ICP selon l'initialisation pour une paire de pièce ayant le même contour

## Histogramme des distance

On dispose de 2145 paires de pièces (66 pièces) dont on connait la transformation pour les recaler.

- ▶ on calcule les histogrammes des cartes de distances.
- ▶ On peut séparer les paires en jeux de données entraînement et jeux de données test
- ▶ On entraîne un algorithme de classification binaire (Régression Logistique)

On peut ainsi obtenir la probabilité que deux pièces sont du même coin.