POLITECNICO DI MILANO

A robot trajectory programming method using multi-camera systems

SILVIO GIANCOLA
DAVIDE CHIARION
REMO SALA

Introduction

Study of a need : Tiles decoration

	Hand made	Serigraphy
Advantages	Authenticity and unicity High quality product	Fast method High quantity
Inconvenient	Poor quantity Necessity of a qualified person	Poor value Printed

Innovative decoration process?

➢ Use of anthropomorphic robot for the tile painting

Robot programming method?

➢ Paintbrush trajectory registration

Tool Identification

PROPOSED SOLUTION

STEREOSCOPIC SYSTEM

PROS

NON INTRUSIVE

PRECISION

CONS

LIGHT CONDITION

AMBIGUITY

OCCLUSION

Tool Identification

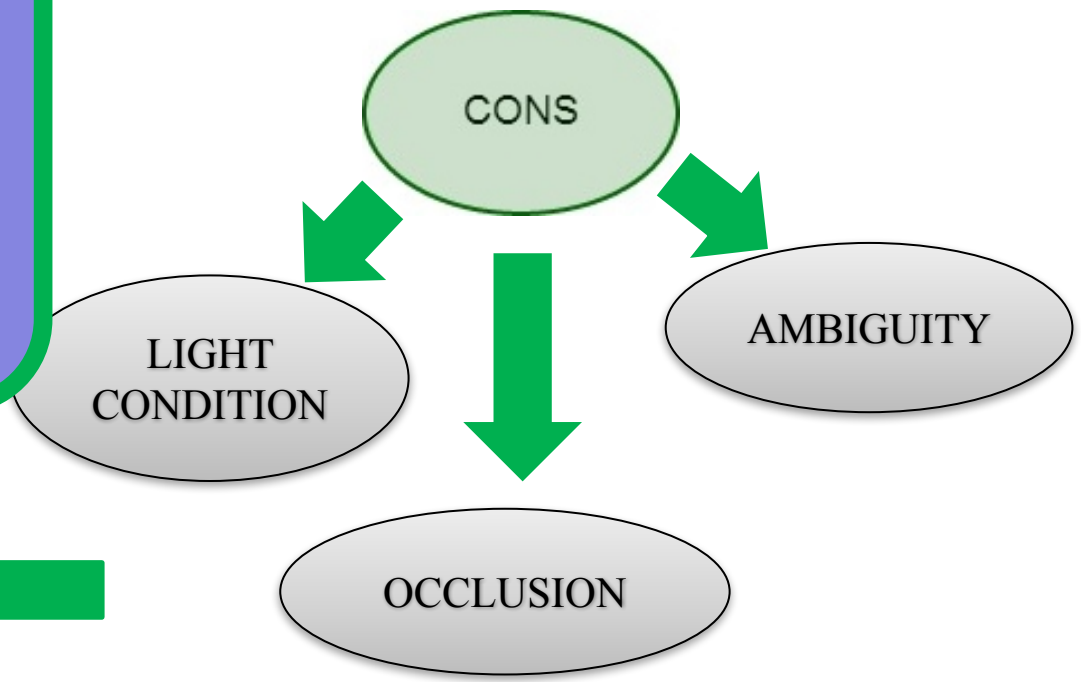
PROPOSED SOLUTION

STEREOSCOPIC SYSTEM

ENLIGHTENED TRINOCULAR SYSTEM

+

REFLECTIVE MARKER SENSORIZED TOOL



Tool Identification

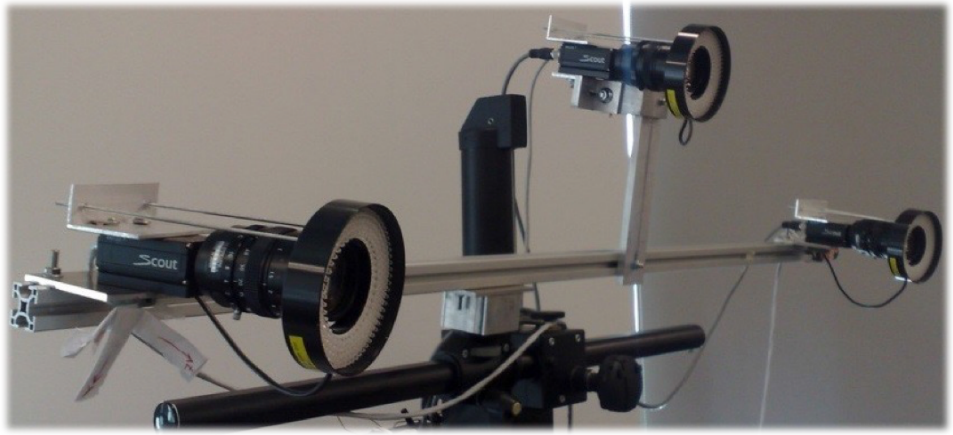
PROPOSED SOLUTION

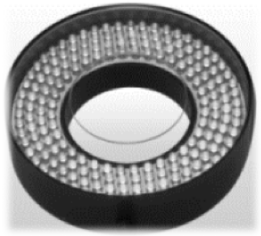
STEREOSCOPIC SYSTEM

ENLIGHTENED TRINOCULAR SYSTEM

+

REFLECTIVE MARKER SENSORIZED TOOL

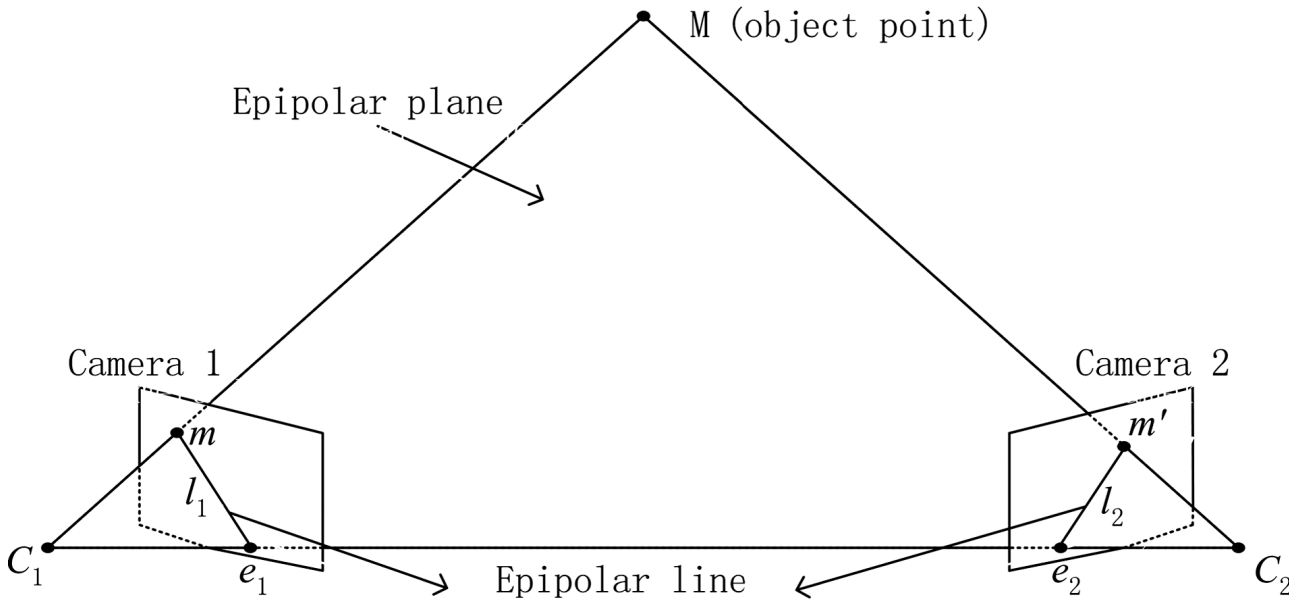


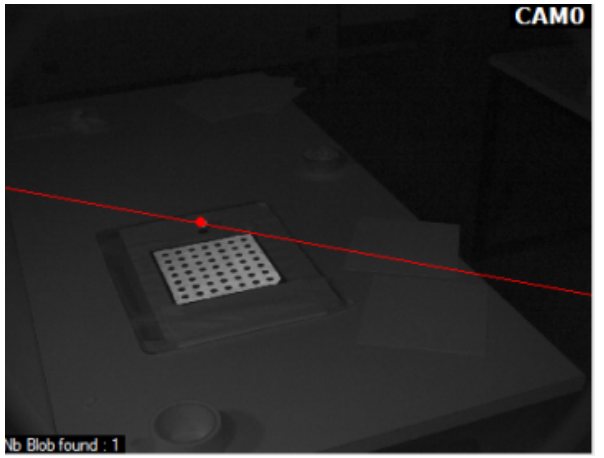


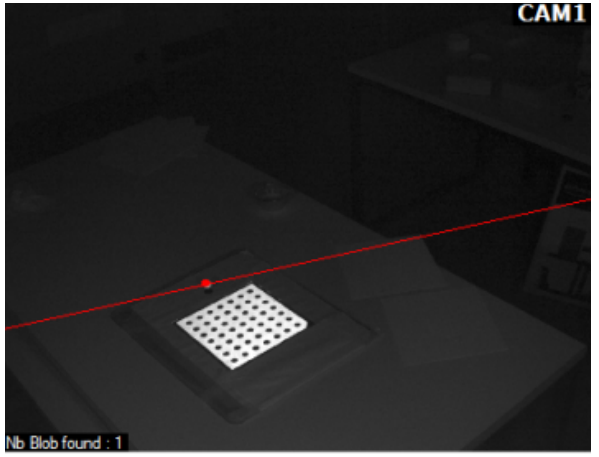
Tool Identification

Stereoscopic system

- *Based on 2 cameras*

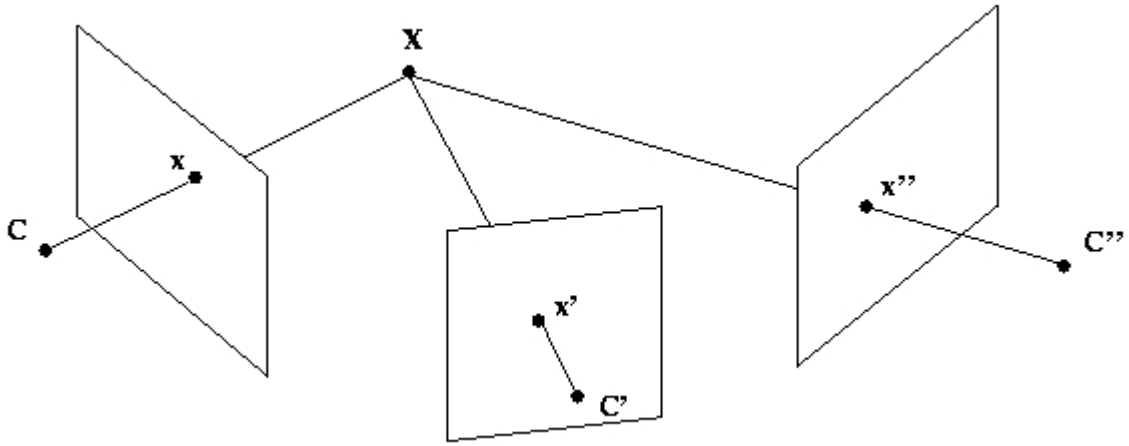


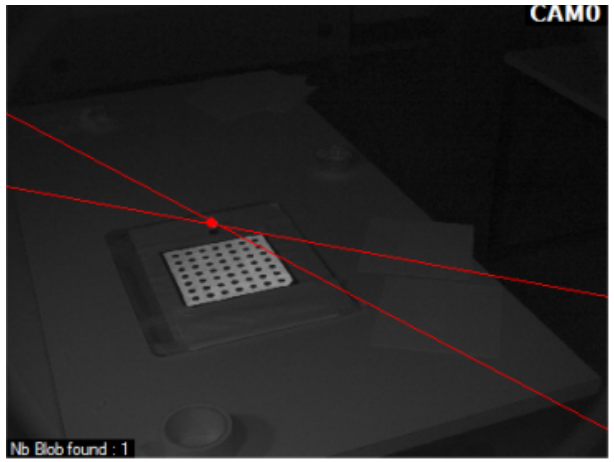


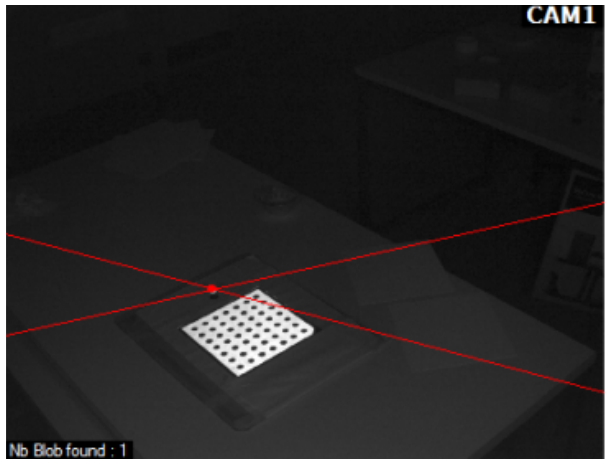


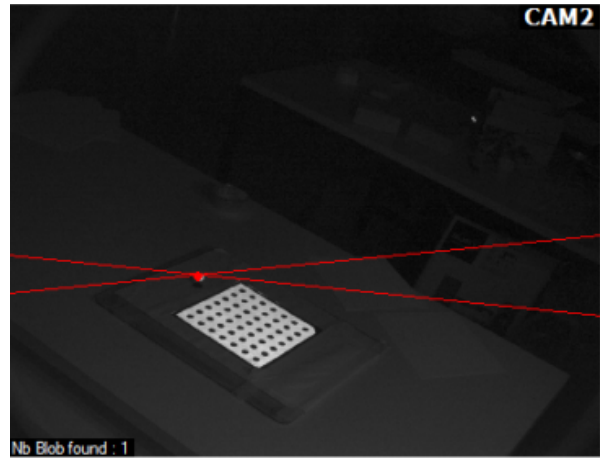
Tool Identification

Trinocular stereoscopy







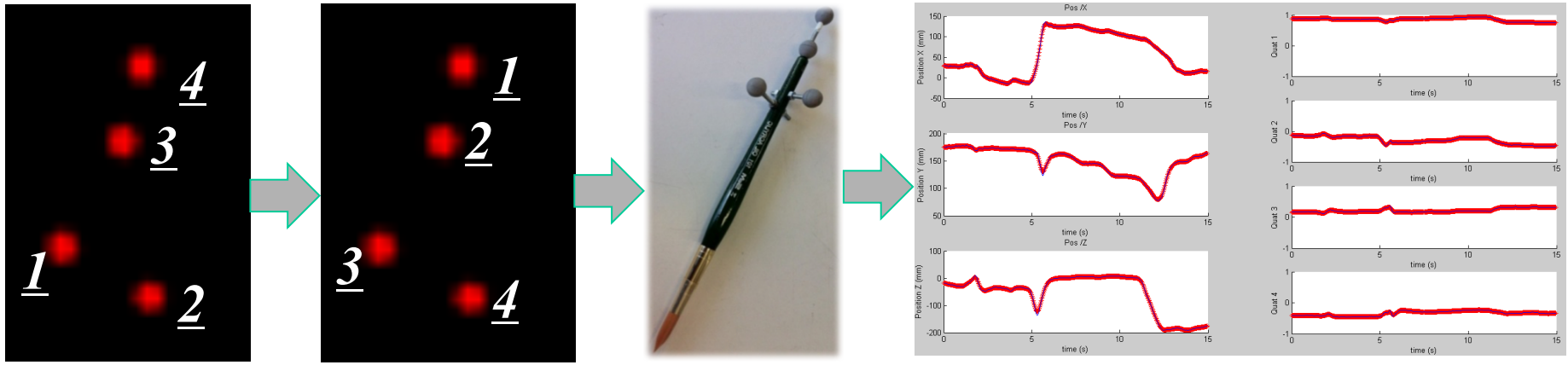


SUMMARY

Tool Identification

Trajectory Acquisition from Trinocular Vision System

- *Point Cloud Indexing*
  - *Rigid Body Registration*
    - *Trajectory Smoothing*



Tool Identification

Point Cloud Ordering

- *Experimental method using distances matrix*

$$Distances = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix}$$

- *Ex:*

$$\begin{bmatrix} 0 & d_{13} & d_{12} & d_{14} \\ d_{31} & 0 & d_{32} & d_{34} \\ d_{21} & d_{23} & 0 & d_{24} \\ d_{41} & d_{43} & d_{42} & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & d_{12} & X & d_{14} \\ d_{21} & 0 & X & d_{24} \\ X & X & 0 & X \\ d_{41} & d_{42} & X & 0 \end{bmatrix}$$

(a) Inversion of 2^{nd} and 3^{rd} marker

(b) Occlusion of 3^{rd} marker

Tool Identification

Point Cloud Ordering – Algorithm

- *If n is the number of marker, we split the n x n matrix in n rows. Be row_i the i^{th} row.*
- *For each row_i, we define an empty 2 x n array named ind_i, linked to the i^{th} row.*
- *For each element d_{ij} of row_i, its value is matched to a correspondent value d_{kl} of the model. The rows and columns indexes (k and l) are then saved in the columns of ind_i.*
- *For each ind_i, we define i_{sol} the largest occurrence of an index in this matrix.*
  - *If all the element of ind_i is 0, then a marker is missing.*
  - *Else, the largest occurrence of an index different to 0 returns the correct index of i.*

Tool Identification

Point Cloud Ordering – Example

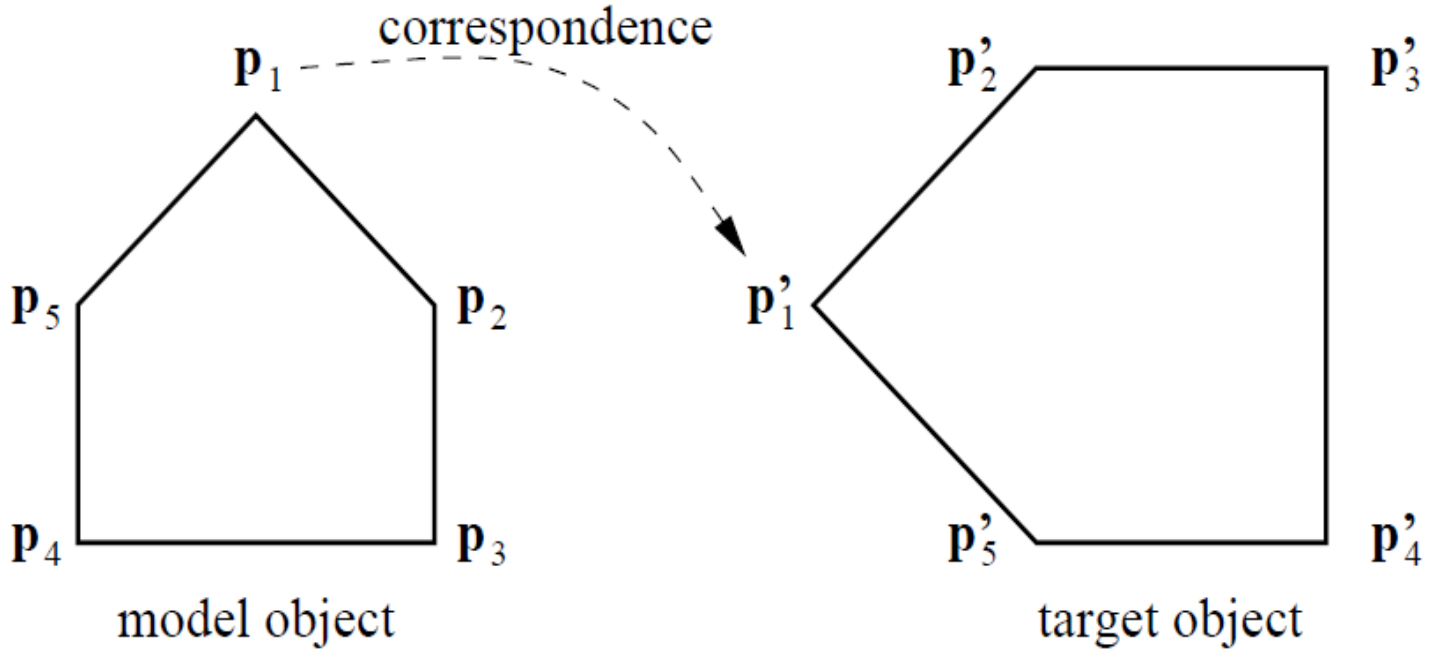
$$\begin{bmatrix} 0 & d13 & d15 & d12 & X \\ d31 & 0 & d35 & d32 & X \\ d51 & d53 & 0 & d52 & X \\ d21 & d23 & d25 & 0 & X \\ X & X & X & X & 0 \end{bmatrix}$$

- $ind_1 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 3 & 5 & 2 & 0 \end{bmatrix}$, the 1st marker corresponds to the 1st one
- $ind_2 = \begin{bmatrix} 3 & 0 & 3 & 3 & 0 \\ 1 & 0 & 5 & 2 & 0 \end{bmatrix}$, the 2nd marker corresponds to the 3rd one
- $ind_3 = \begin{bmatrix} 5 & 5 & 0 & 5 & 0 \\ 1 & 3 & 0 & 2 & 0 \end{bmatrix}$, the 3rd marker corresponds to the 5th one
- $ind_4 = \begin{bmatrix} 2 & 2 & 2 & 0 & 0 \\ 1 & 3 & 5 & 0 & 0 \end{bmatrix}$, the 4th marker corresponds to the 2nd one
- $ind_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$, a marker is missing (the 4th).

Tool Identification

Rigid Body Registration

- *Find Rotation and Translation of a point cloud respect to a model*



Tool Identification

Rigid Body Registration

- *Find Rotation Translation and Scaling of a point cloud respect to a model*
- *Singular Value Decomposition (SVD)*
  - $M = \sum_{i=1}^{n} r_i' * r_i{}^T = U * S * V,$
    with r_i' *and* r_i *the deviation of points of a cloud respect to its barycenter*
  - $Rotation = U * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & sign(det(U * V^T)) \end{bmatrix} * V^T$
  - $Translation = \hat{p}' - Rotation * \hat{p}$
    with $\hat{p}'$ *et* $\hat{p}$ *the barycenter of the point clouds*

Trajectory Smoothing

Filtering and Interpolations :

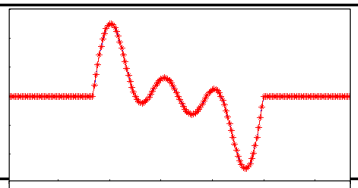
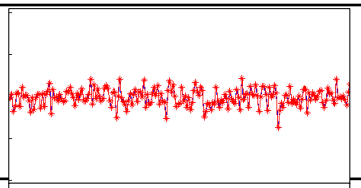
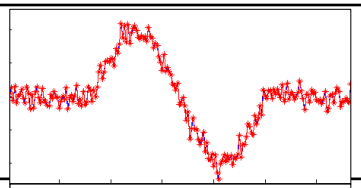
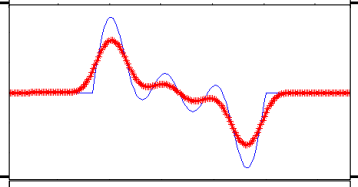
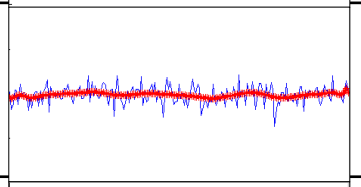
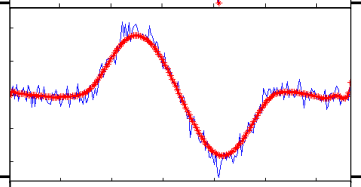
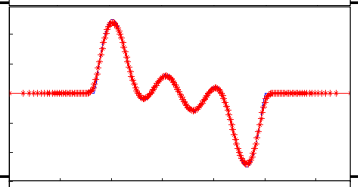
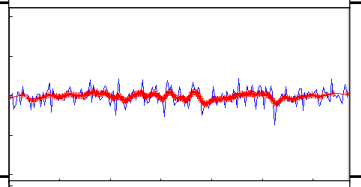
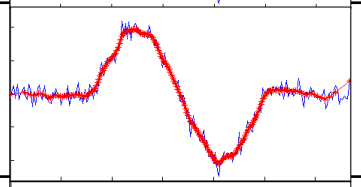
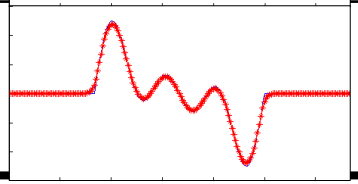
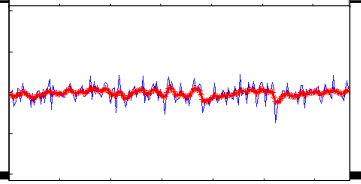
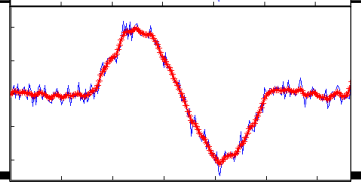
- *Cubic / Spline interpolation*
- *Bézier Curves interpolations*
- *B-Spline interpolations (NURBS)*
- *Low Pass filter*

Tests :

- *Sinus Following*
- *White Noise reduction*
- *Sinus and White Noise Combination*
- *Acquired Data*

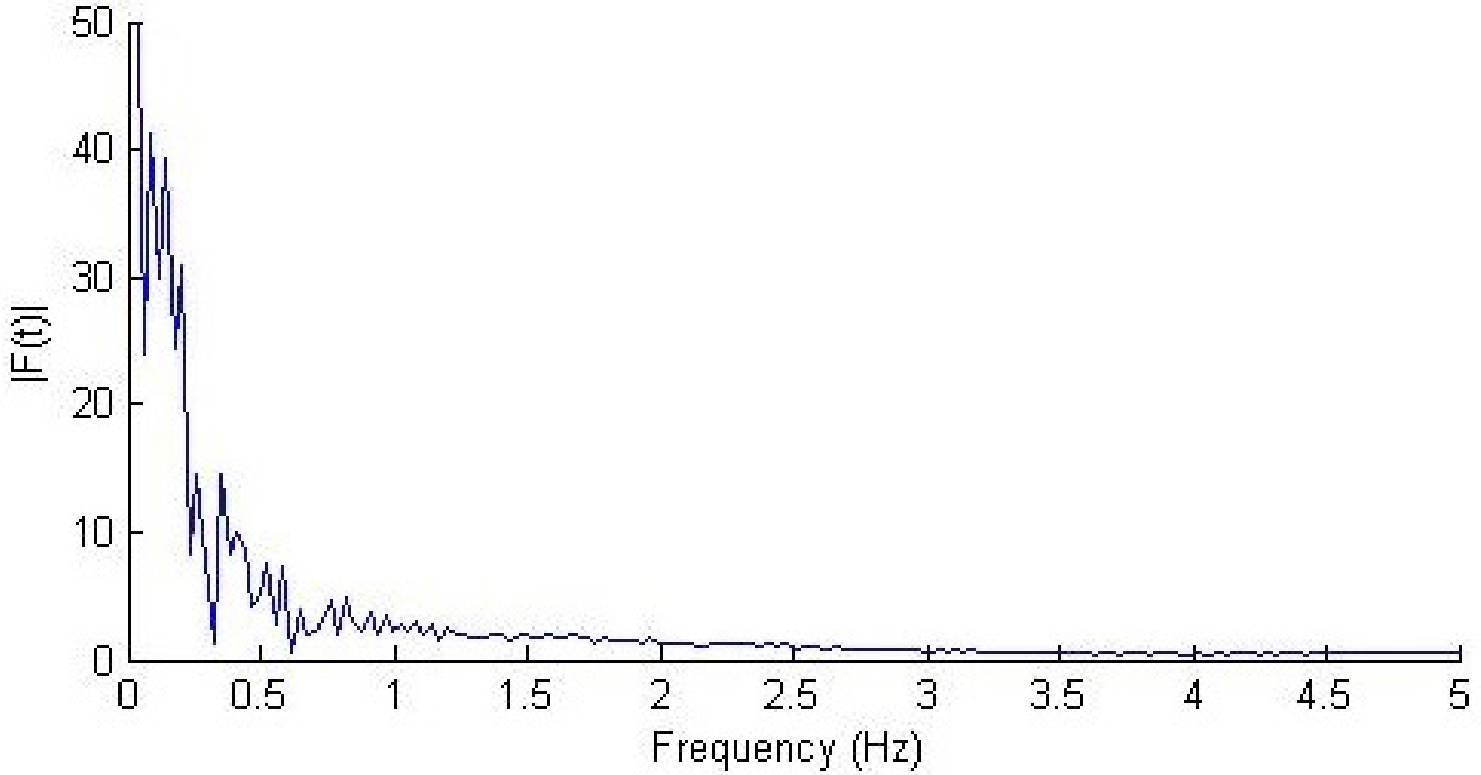
Trajectory Smoothing

Results :

	Sinusoids	*White noise*	*Combination*	*Acquired data*
Cubic / Spline				
Bézier curves				
B-Spline (NURBS)				
Low Pass				

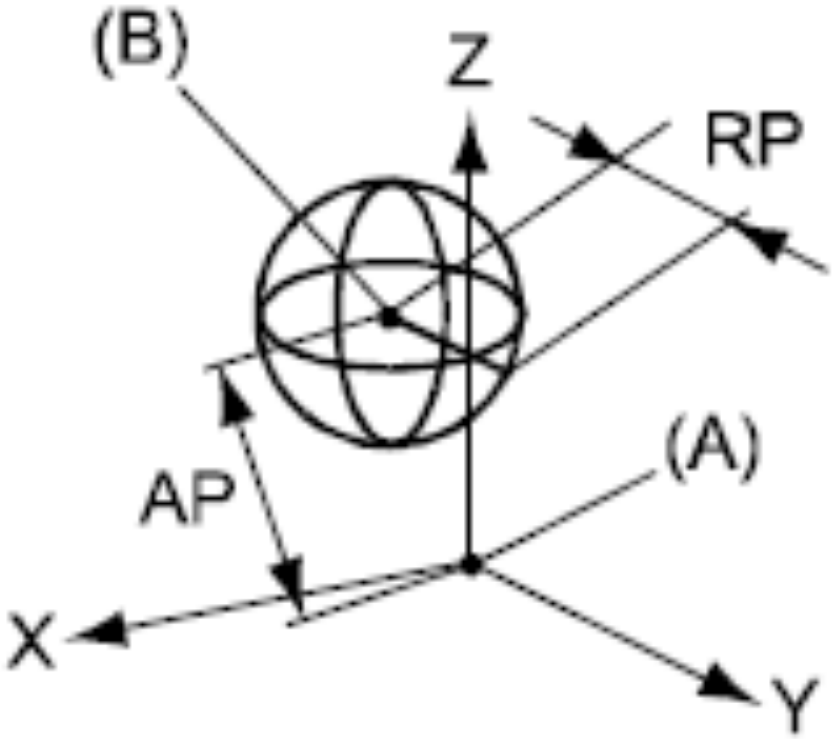
Trajectory Smoothing

Amplitude Spectrum of an acquisition:



Metrologic Analysis

Metrologic Analysis

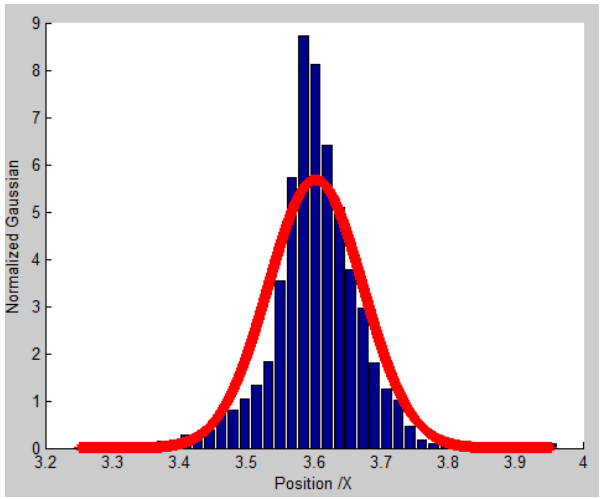


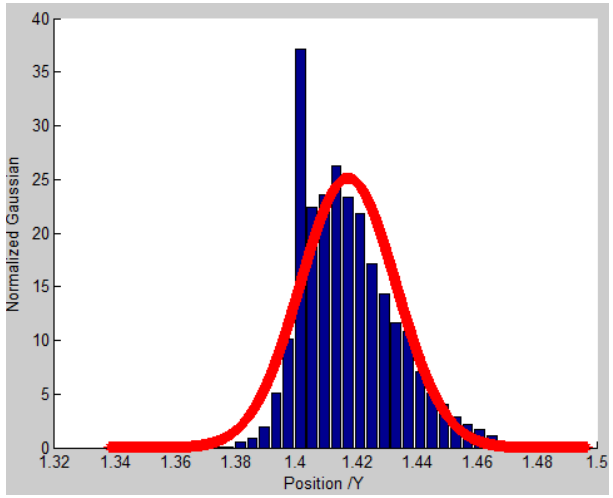
- *Accuracy (AP) : proximity of measurement results to the true value*
- *Repeatibility (RP) : precision of the measurement*

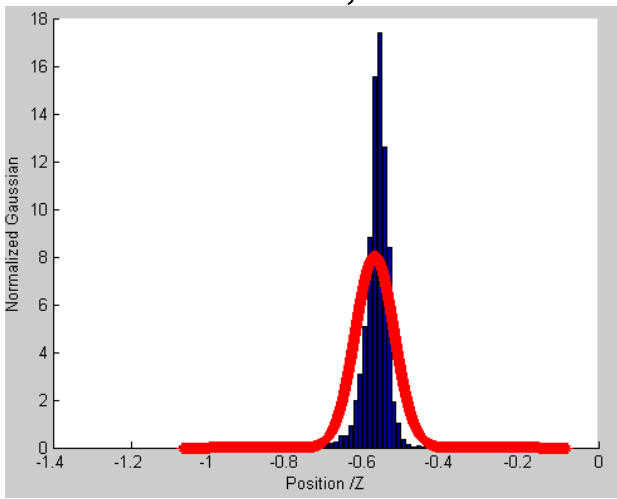
Metrologic Analysis

Metrologic Analysis

- *Repeatability*
  - *Analysis of the reconstruction of a marker at the origin position*
  - *Statistical study (gaussian)*
- *Results*
  - *Uncertainty respect to the axis :*
  - *X axis : 0,21 mm*
  - *Y axis : 0,05 mm*
  - *Z axis : 0,15 mm*





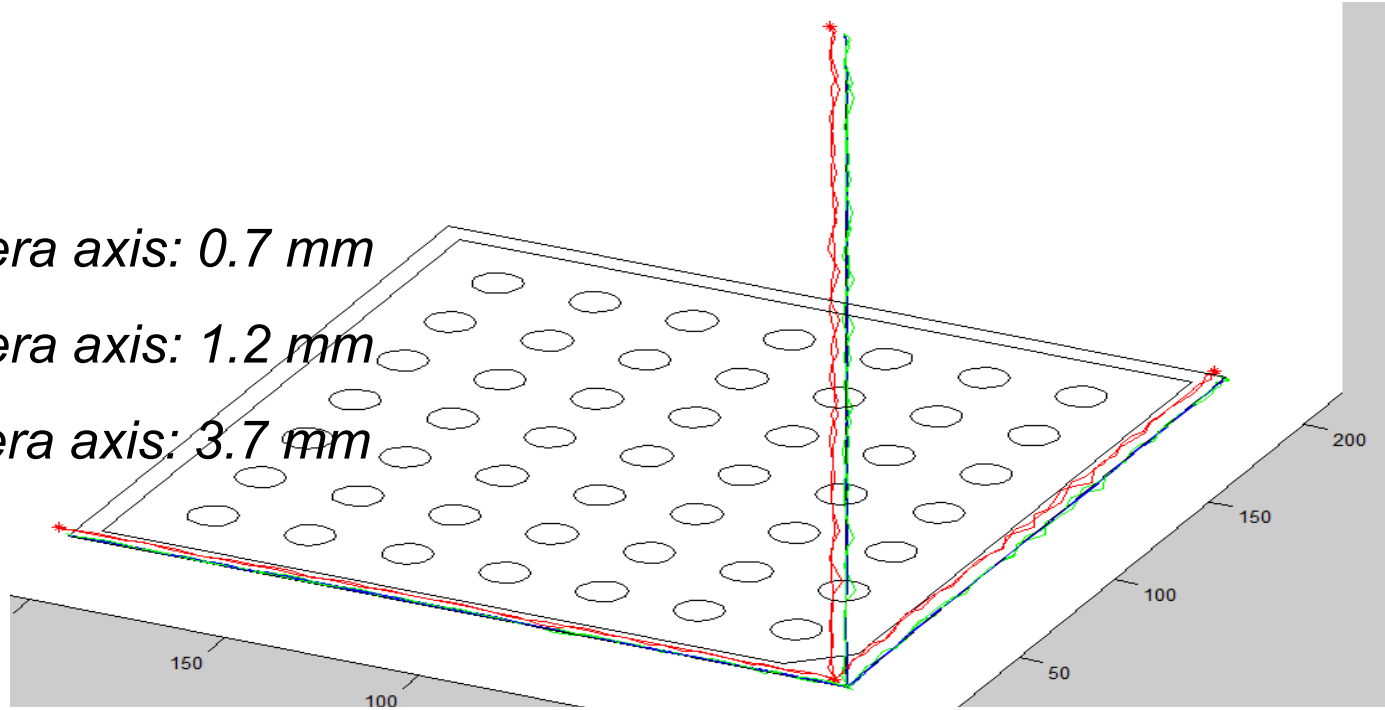


Metrologic Analysis

Metrologic Analysis

- *Accuracy*
  - *Trajectory analysis among the coordinate system axis*
  - *Use of the Rigid Registration algorithm on the point cloud defining the trajectory*

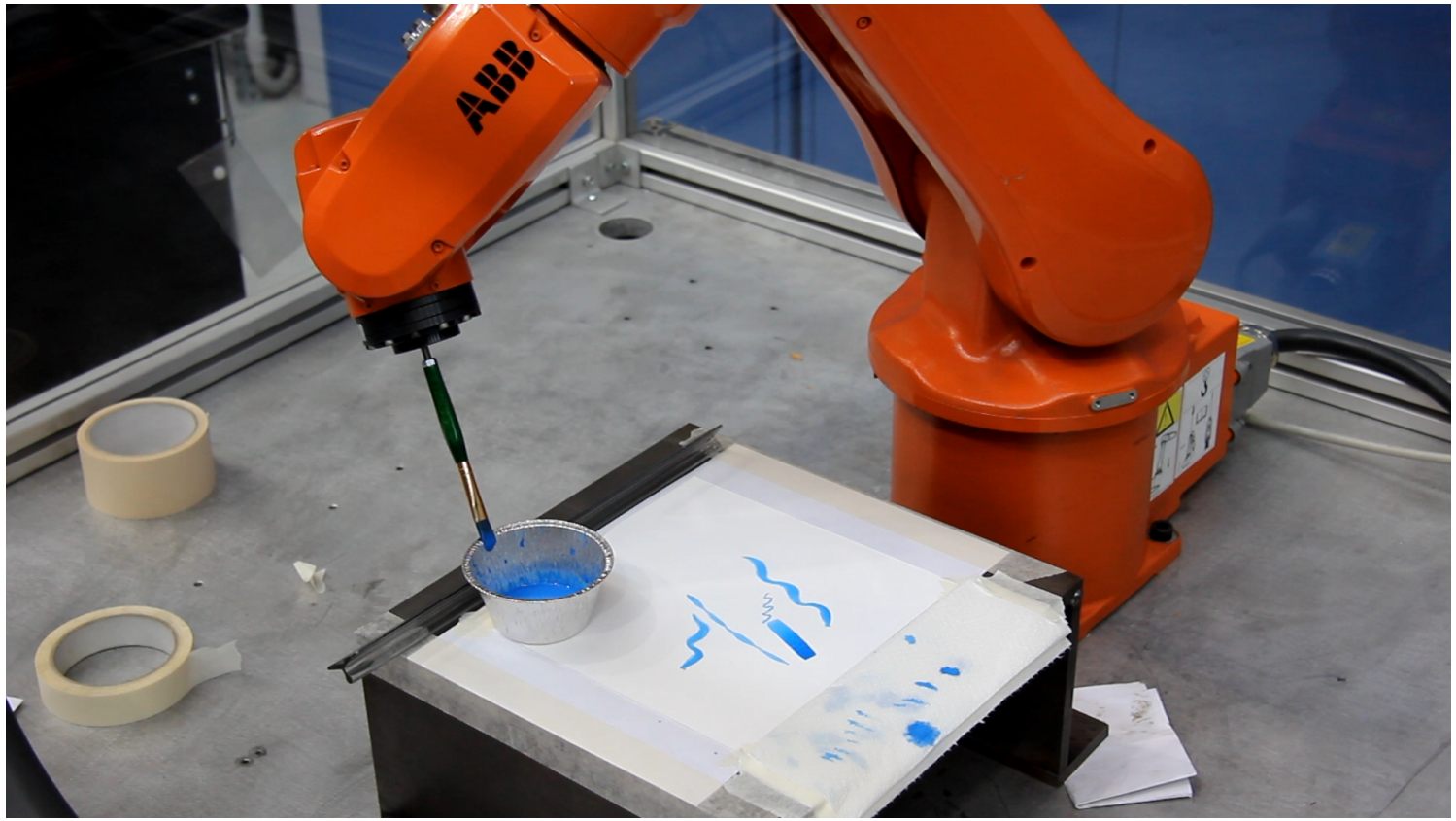
- *Results*
  - *Error among X camera axis: 0.7 mm*
  - *Error among Y camera axis: 1.2 mm*
  - *Error among Z camera axis: 3.7 mm*



Application

Application

- *Model of the robot : ABB IRB 120*



Application

Application

Conclusion

Conclusion

- *Feasibility **OK***
- *Precision **< 0.3 mm***
- *Accuracy to improve with correct registration between the 2 setups*