

Université Toulouse III - Paul Sabatier

M2 IM - Master Informatique Image et Multimédia

*Registration of Cardiac US and CT Data Using
Fourier Descriptors and Dynamic Time Warping*

Rapport de Recette

Equipe: ALBERTUS Kévin BOUKAIS Dyhia DRZAZGA Pascal NGUYEN Thi thanh thuy	Superviseur : KOUAME Denis	Client : ALACOQUE Xavier CUTTONE Fabio
--	--------------------------------------	---



Préambule	2
Introduction	2
Génération des données de test	3
Génération du phantom dans Houdini	3
Génération des fichiers DICOM	4
Implémentation dans Mevislab	4
Génération des contours US	5
Binarisation	5
Contours	5
Export Matlab	5
Génération des contours CT	6
Segmentation par “Region Growing”	6
Maillage du volume	7
Itération par exécution d’un script en python	7
Positionnement initial du plan de coupe	7
Génération des coupes de translation	7
Génération des coupes de rotation	8
Exécution des scripts Matlab	8
Tests et Descriptif du code Matlab	9
Structuration des données	9
Calcul des descripteurs de Fourier	9
Calcul de la distance	11
Calcul du poids	11
Calcul d’une métrique par “Dynamic Time Warping”	11
Calcul de la minimisation	12
Conclusion	12

1. Préambule

Au moment de l'écriture de ce rapport, et bien que la totalité de l'implémentation de la publication "*Dynamic Registration of Cardiac US and CT Data Using Fourier Descriptors and Dynamic Time Warping*" soit réalisée ; nous ne disposons pas encore de l'ensemble des résultats de test. Ceux-ci seront adjoints ultérieurement dans une annexe.

2. Introduction

Dans le cadre des cardiopathies congénitales, et en particulier dans le cadre des pathologies cardiaques rares (quelques dizaines de cas par an en France) il est crucial pour les équipes médicales d'obtenir des images aptes à faire un bilan fiable pour prendre des décisions offrant les meilleurs compromis. Cet enjeu est d'autant plus crucial, car les cas rares sont des cas pour lesquels les références anatomiques sont par définition biaisées. Pour augmenter sa capacité de diagnostic et de prise en charge de ses patients, le service de chirurgie pédiatrique du CHU Toulouse Purpan mise depuis de nombreuses années sur le développement de l'échographie 3D pour les cardiopathies congénitales.

L'examen tomographique (CT) est d'un usage généralisé dans le cadre des cardiopathies congénitales, car cet examen permet d'obtenir un volume 3D de la totalité du volume cardiaque et des gros vaisseaux. Il facilite une appréhension globale de l'anatomie du patient et permet de faciliter la détection d'un grand nombre des pathologies cardiaques.

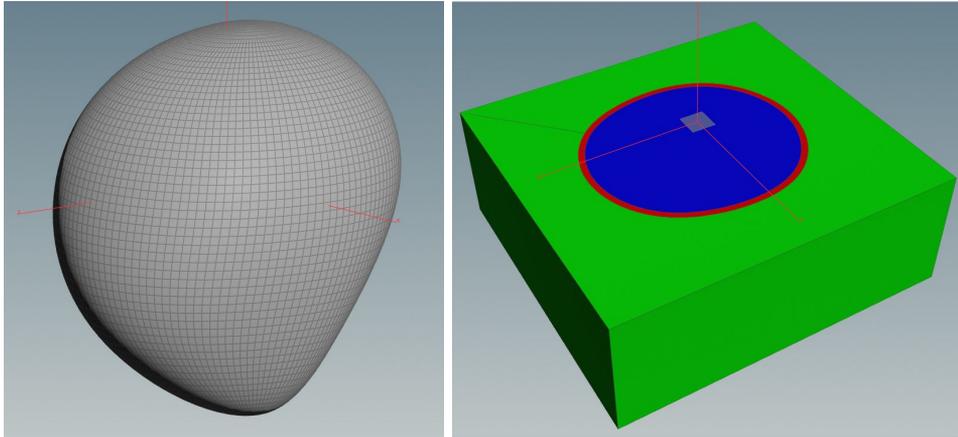
En pratique, les équipes médicales sont obligées d'utiliser les images échographiques et tomographiques conjointement pour couvrir l'ensemble de leurs besoins, les avantages de l'un étant les inconvénients de l'autre. Il devient donc de plus en plus nécessaire de croiser les informations entre les deux procédés. C'est dans ce cadre que se définit le projet.

3. Génération des données de test

Afin de tester et de mesurer les performances de notre solution, nous avons l'obligation de générer nos propres données de simulation. Le choix a été fait de partir d'un modèle 3D simplifié d'un volume cardiaque nous permettant de générer des fichiers DICOM de scanner tomographique et d'échographie qui soient corrélés.

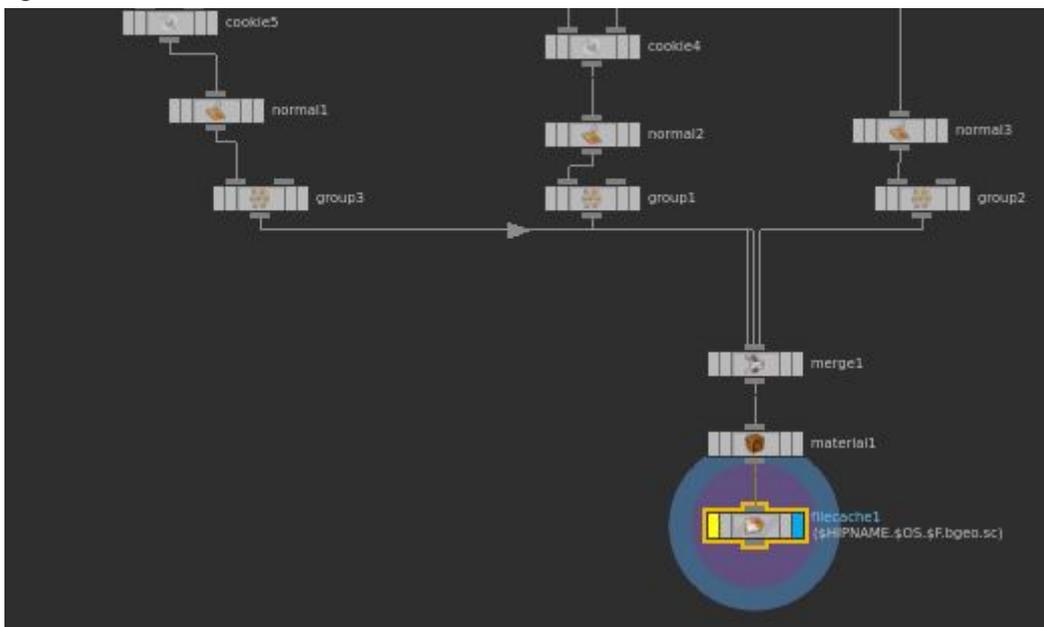
3.1. Génération du phantom dans Houdini

Le logiciel d'animation **Houdini** de chez **SideFx** est utilisé pour la création, l'animation et le rendu 4D servant à la simulation CT et US.



Le principal défi de cette étape vient du besoin de créer un volume "plein" et animé. En effet, les techniques classique de définition des surfaces par maillage ne permettent pas par définition de "remplir" le volume qu'elles représentent. A contrario, les techniques volumique (par voxélisation par exemple) peinent à animer les surfaces solides.

Notre solution consiste dans un réseau d'opérations booléennes animées puis rendues sous forme de séquence d'images. Ce principe nous permet de simuler les variations de densité des organes.



3.2. Génération des fichiers DICOM

Le deuxième défi consiste à générer des fichiers DICOM comparables par leur structure à ceux provenant d'appareils d'imagerie médicale. Cette opération est réalisée dans **Mevislab** par le biais d'un code python dont la fonction est d'envoyer vers un écouteur les images calculées dans Houdini dans un ordre instant, tranche. Cet ordonnancement est réalisé grâce à un formatage du nom des images lors de leur calcul sous la forme *timeId_sliceId_filename.tif*

Ensuite, le volume CT de 512x512x512x voxels est étalonné en intensité avec des valeurs entre 0 et 4095 en unsigned int16 (soit 65535 nuances), puis un LUT 1D est appliqué pour correspondre aux données réelles. Un code temporel est également ajouté à chaque instant image.

Finalement, en utilisant le module "**itkImageFilterWriter**" et le module "**DicomTagModify**" le fichier est enregistré au format DICOM en respectant les normes.

4. Implémentation dans Mevislab

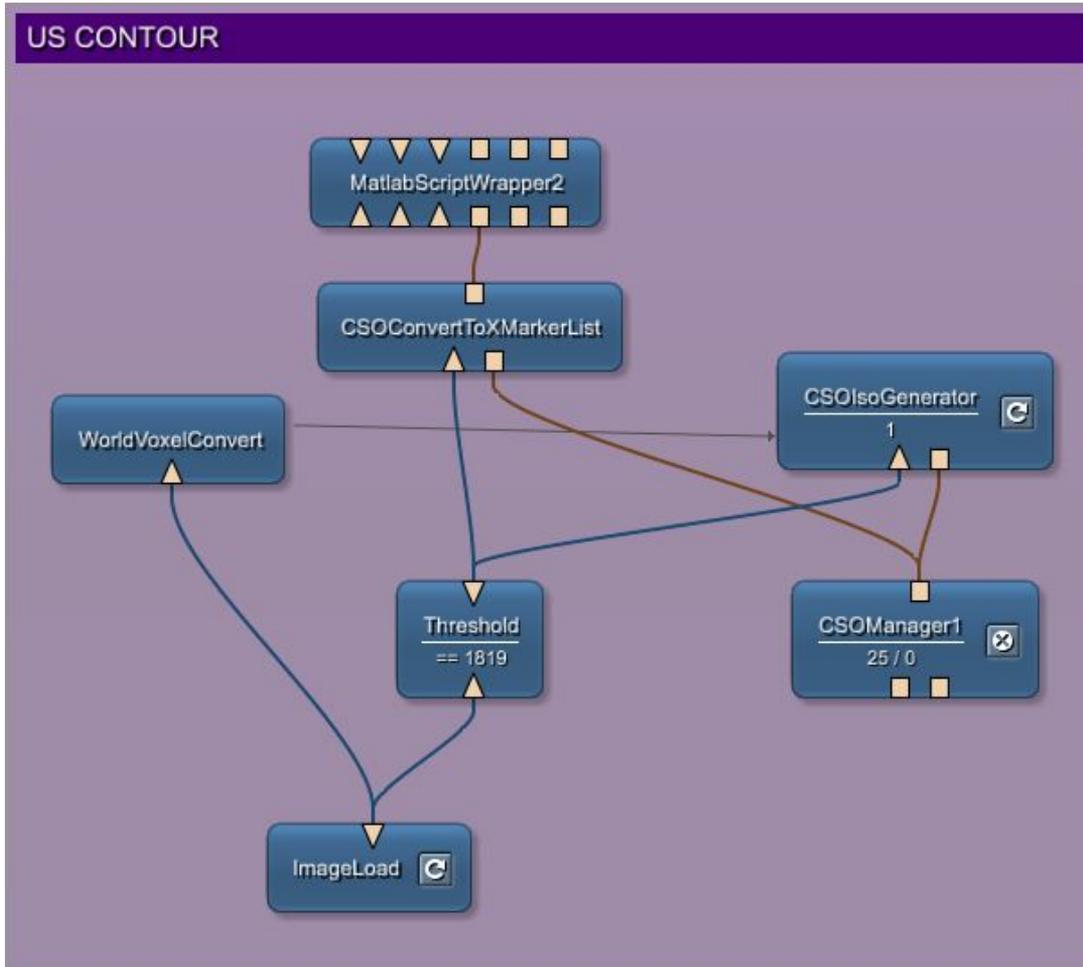
Dans cette section, nous nous attacherons à définir les principales étapes de l'algorithme de la publication scientifique réalisées directement dans **Mevislab**.

Cette implémentation fait intervenir le système de connection de modules, une automatisation des itération par scripts python et un appel **Matlab** au travers du module "**MLMatlabScriptWrapper**". Les codes appelés par ce module sont détaillés dans la section "Tests et Descriptif du code **Matlab**".

Ce module optionnel est disponible à l'adresse :

<https://github.com/MeVisLab/communitymodules.git>

4.1. Génération des contours US



4.1.1. Binarisation

L'endocarde du cycle cardiaque est binarisée par l'utilisateur qui définit la position d'une graine de segmentation et d'une plage seuil.

4.1.2. Contours

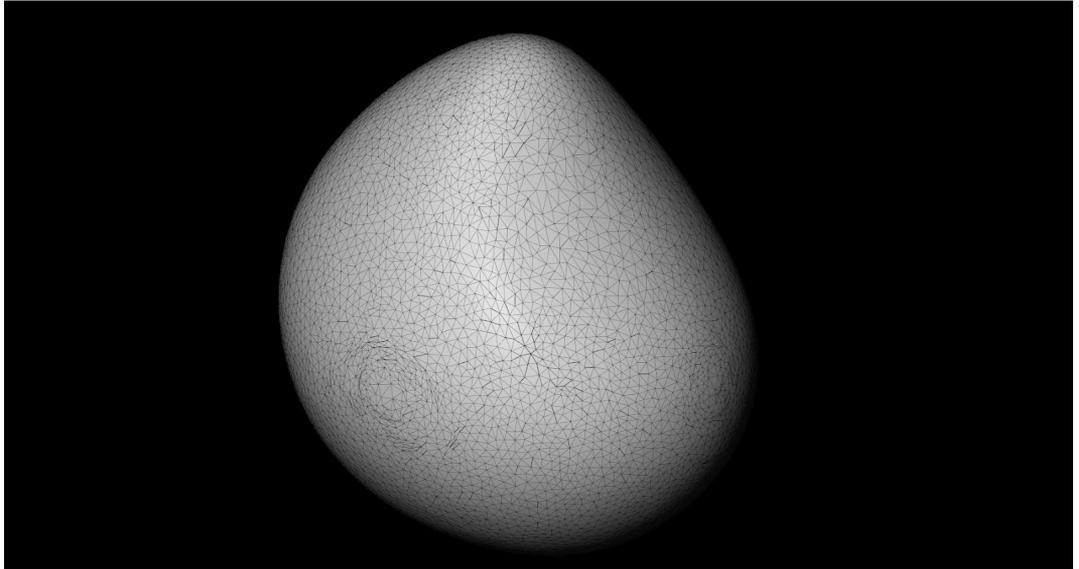
Le module "**CSOIsoGenerator**" génère une courbe de contours à partir de la binarisation pour tous les instants du cycle cardiaque.

4.1.3. Export Matlab

Le module "**MLMatlabScriptWrapper**" est utilisé pour exporter les coordonnées des points de tous les contours US.

4.2.2. Maillage du volume

Le volume segmenté est converti en maillage polygonal par le module “WEMIsoSurface”.



4.2.3. Itération par exécution d'un script en python

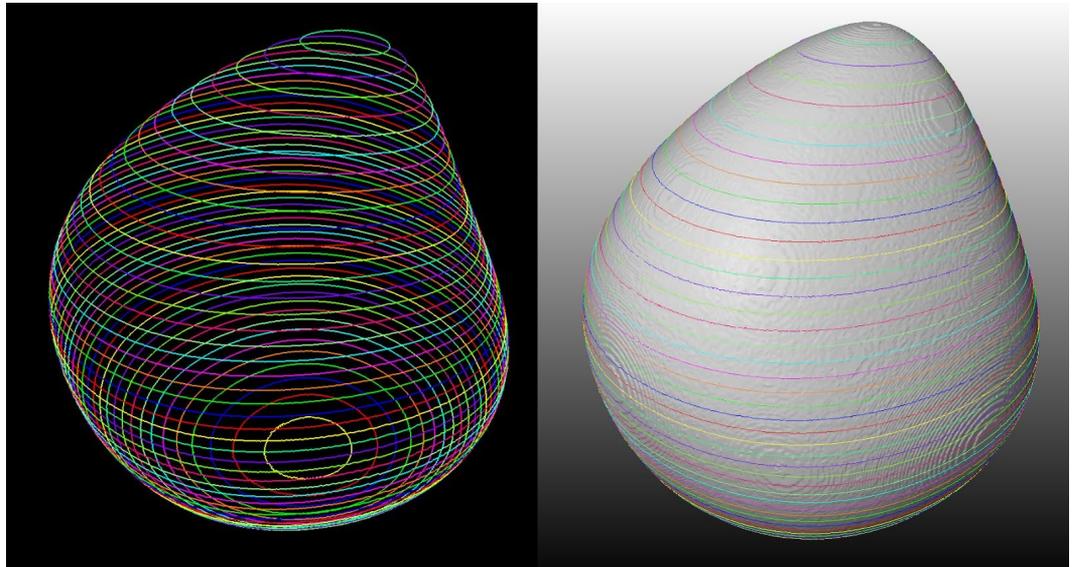
```
1 nw = ctx.parent()
2
3 nw.field("CSOManager2.removeAllCSOsAndGroups").touch()
4
5 nw.field('ComposePlane.normal').value = [0, 0, 1]
6 nw.field('ComposePlane.point').value = nw.field('VolumeCenter.volumeCenterWorldCoordinates').value
7 sliceCenter = int(nw.field('VolumeCenter.volumeCenterVoxelCoordinates').value[1])
8 sliceMax = nw.field("VolumeCenter.Info.sizeX").value
9 sliceDelta = 30
10 sliceStep = int(sliceMax)/int(sliceDelta)
11
12
13 mode = 0
14 id = 0
15
16 if mode == 0 :
17     for i in range(1, int(sliceMax), sliceStep):
18         print i, id
19         nw.field('ComposePlane.point').value = [256, i, 256]
20         nw.field('WEMClipPlaneToCSO.plane').value = [0, 0, 1, i]
21         nw.field('WEMClipPlaneToCSO.apply').touch()
22         nw.field('CSOMerge.apply').touch()
23         id += 1
24 elif mode == 1 :
25     for i, j in zip(range(0,10), range(10,0,-1)):
26         k=float(i)/10.0
27         l=float(j)/10.0
28         print i, j
```

4.2.3.1. Positionnement initial du plan de coupe

Le centroïde du volume sanguin est utilisé comme centre d'initialisation du plan de coupe. Comme proposé dans la publication, deux options d'axes de coupe sont proposés selon que l'échographie à recaler soit une vue 2 ou 4 chambres.

4.2.3.2. Génération des coupes de translation

En fonction d'un pas de translation arbitraire, le module “WEMClipPlanToCSO” génère de manière itérative les courbes de contour résultantes de l'intersection entre le plan virtuel et le maillage du volume sanguin.



4.2.3.3. Génération des coupes de rotation

En fonction d'un pas arbitraire du vecteur normal du plan de coupe, le module "**WEMClipPlanToCSO**" génère de manière itérative les courbes de contour résultantes de l'intersection entre le plan virtuel et le maillage du volume sanguin.

4.2.3.4. Exécution des scripts **Matlab**

Le module "**MLMatlabScriptWrapper**" est utilisé pour exporter les coordonnées des points de tous les contours CT.

Le même module sert ensuite de "**main**" et appelle les fonctions **Matlab** détaillés dans la section suivante.

En sortie, les meilleures minimisations en translation et rotation sont retournées.

5. Tests et Descriptif du code Matlab

Fonctions :

Function [matriceContours_CT, matriceContours_US] = listePointsToStructure(listePoints_CT, listePoints_US)

Function [vecteurDescripteur_n] = calculDescripteursDeFourier(matriceContours_n, nbDescripteurs)

Function [vecteurDistance] = calculDistance(vecteurDescripteur_CT, matriceDescripteur_US)

Function [vecteurWeight] = calculWeight(matrice_CT, matrice_US)

Function [scalaireMetric] = DTW(vecteurDistance, vecteurWeight)

Function [vecteurGradient] = calculDescenteGradient(vecteurMetricAxe)

Main.m :

Contour US -> function calculDescripteursDeFourier() : { matriceDescripteur_US}

Contour CT -> function calculDescripteursDeFourier() : vecteurDescripteur_CT

For chaque Axe

For i = 1 : nb_Slice

 vecteurDescripteur_CT, { vecteurDescripteur_US_n} -> function calculDistance() : vecteurDistance

 matrice_CT, matrice_US_n -> function calculWeight() : vecteurWeight

 vecteurDistance, vecteurWeight -> function DTW() : scalaireMetric

 vecteurMetricAxe(i) = scalaireMetric

end

 vecteurMetricAxe -> function calculDescenteGradient() : vecteurGradient

end

5.1. Structuration des données

En entrée, une liste de points des contours US ainsi qu'une liste de points des contours CT sont fournies. On va couper les données entrée de US pour recevoir une matrice du contour US ou chaque élément est liste des points pour un contours US. On continue pareil avec les données entrée de CT.

5.2. Calcul des descripteurs de Fourier

La résolution spatiale des données CT ainsi que des données US étant différentes, un moyen de décrire les contours de chacun est d'utiliser les descripteurs de Fourier. Cette technique permet aussi d'égaliser les différents contours au même niveau de détail.

Considérons N points ordonnées de coordonnées (x, y) d'un contour quelconque. La première étape consiste à passer ces points dans l'espace complexe tel que :

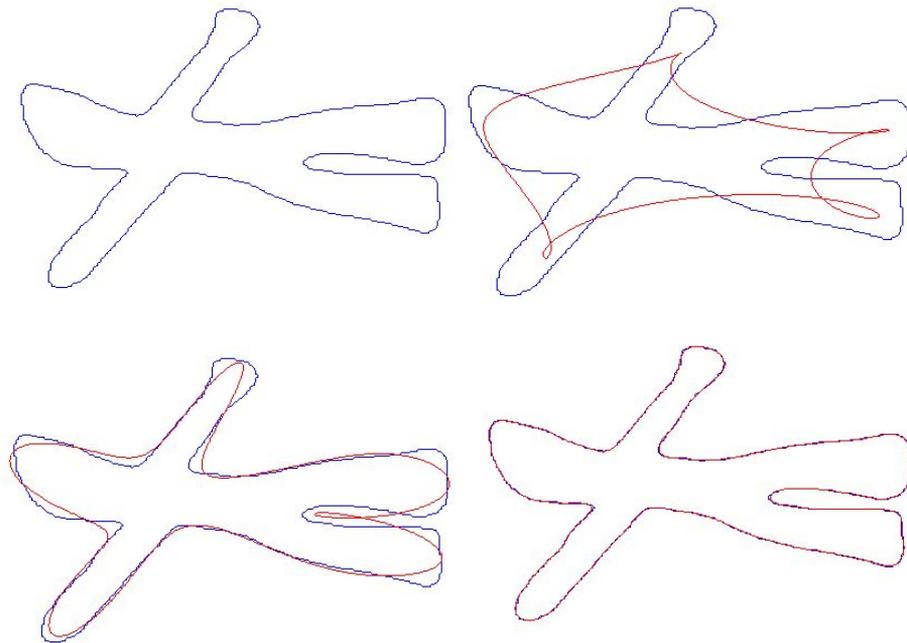
$$z_n = x_n + i * y_n \text{ avec } n \in \{ 0, \dots, N-1 \} .$$

Le passage dans l'espace fréquentiel se fait à l'aide de la transformée de Fourier discrète habituelle :

$$Z_k = \frac{1}{N} \sum_{m=0}^{N-1} z_m e^{(-\frac{2i\pi}{N} km)} , k = (0, \dots, N-1) .$$

La première composante (DC) représente la position du centre de gravité. Cette valeur n'est donc pas forcément utile pour décrire la forme du contour. La seconde composante permet de décrire la taille de la forme. C'est pour cela que l'on obtient un cercle si l'on désire ne récupérer qu'un seul descripteur de Fourier pour représenter un contour.

Les fréquences suivantes permettent de représenter plus ou moins les niveaux de détails. Les basses fréquences vont permettre d'apporter la majeure partie de l'information, tandis que les hautes fréquences vont s'attarder sur les détails. Ces hautes fréquences vont aussi être plus sensibles au bruit. C'est donc sur ces fréquences là que nous allons travailler pour les descripteurs de Fourier. Au fur et à mesure que le nombre de descripteurs de Fourier augmente, le contour de la courbe se rapproche de plus en plus de l'originale. Avec 50 descripteurs, la courbe apporte assez de précision.



Si l'on prend $M < N$ composantes basses fréquences, il est donc possible de reconstituer une approximation du contour plus ou moins précise à l'aide de la Transformée de Fourier Inverse discrète :

$$z_m = \sum_{k=-M/2}^{M/2} Z_k e^{j2\pi mk/N}, \quad m = (0, \dots, N-1).$$

Enfin, afin d'ajouter de l'importance sur les premières composantes, on ajoute un poids proportionnel à l'ordre tel que $Z'_k = \frac{1}{k+1} \|Z_k\|$.

Les descripteurs de Fourier possèdent plusieurs propriétés intéressantes notamment en matière de translation et de rotation. Par exemple, lors d'une translation du contour 2D avec $z_0 = x_0 + j * y_0$, on obtient :

$$Z'_k = \frac{1}{N} \sum_{m=0}^{N-1} [z[m] + z_0] e^{-j2\pi mk/N} = \frac{1}{N} \sum_{m=0}^{N-1} z[m] e^{-j2\pi mk/N} + \frac{1}{N} \sum_{m=0}^{N-1} z_0 e^{-j2\pi mk/N} = Z[k] + z_0 \delta[k]$$

On remarque donc que la translation affecte la composante DC de la transformée de Fourier et qu'elle n'affecte pas l'information contenant le contour.

5.3. Calcul de la distance

On a en entrée un vecteur de 50 descripteurs de Fourier pour un contour CT et 24 vecteurs de 50 descripteurs pour les 24 contours US.

On calcule la distance euclidienne entre le contour CT et chaque contour US, ce qui nous donne en sortie un vecteur de distance à 24 éléments.

Pour trouver la meilleure position du plan dans le volume CT, nous avons comparé à chaque itération un contour CT issu d'une coupe du volume 3D à une séquence de contours US.

Comme vu précédemment, nous avons calculé les descripteurs de Fourier de chaque contour pour pouvoir les comparer entre eux. Puisque la séquence de contours US est la même à chaque itération, nous les avons calculés à l'extérieur de la boucle.

Chaque contour est transformé en une courbe de descripteurs. Le calcul de la distance entre 2 contours consiste donc à calculer la distance euclidienne entre deux vecteurs de descripteurs.

$$d(i_n, j_n) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$$

5.4. Calcul du poids

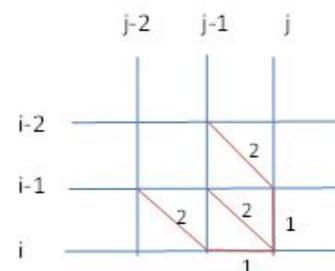
Cette fonction a pour but de compenser le décalage temporel entre la séquence US et les instants images CT en attribuant un "poids d'influence". Cette fonction fait sens pour des données réelles, dans la mesure où même recalées par rapport à leur pic QRS, l'échantillonnage temporel CT et US diffèrent.

$$w(i_n, j_n) = (i_n - i_{n-1} + (j_n - j_{n-1})) \text{ où } w \text{ est le poids, } i \text{ et } j \text{ les données CT et US.}$$

5.5. Calcul d'une métrique par "Dynamic Time Warping"

La fonction DTW donne la distance minimale entre la séquence US et la séquence CT qui change avec le temps en utilisant l'équation de la programmation dynamique.

$$g(i, j) = \min \begin{bmatrix} g(i-1, j-2) + 2d(i, j-1) + d(i, j), \\ g(i-1, j-1) + 2d(i, j), \\ g(i-2, j-1) + 2d(i-1, j) + d(i, j) \end{bmatrix}$$



Au lieu de prendre la distance entre chaque point de contour CT et US dans le temps (comme la distance Euclidienne), on va comparer la distance entre un contour CT à l'instant i avec les

contours US voisins de cet instant i et prendre la distance minimale. C'est-à-dire, elle donne la distance par le somme des distances entre des contours CT et US, qui se rapprochent le plus.

5.6. Calcul de la minimisation

Afin de s'assurer d'avoir la valeur minimale optimale, on applique une descente de gradient sur la courbe générée par les métriques des différentes coupes. On va appliquer cette minimisation sur les 3 courbes différentes obtenues après translation d'un plan selon l'axe des abscisses, rotation autour de l'axe des ordonnées et rotation autour du grand axe.

Le minimum de chacune de ces 3 courbes correspond à la nouvelle position de l'échographie dans le repère du coroscanner qui sera renvoyée à **Mevislab**.

6. Conclusion

Comme nous l'avions anticipé en conclusion de notre rapport de méthodes et algorithmes, nous nous sommes confronté à une implémentation étapes par étapes liée à l'interdépendance des concepts mathématiques de l'étude. Dans le sens où chaque étape dépend grandement des résultats qui la précède, nous avons dû valider chaque étape avec certitude, ce qui nous a limité dans le nombre de tests de l'ensemble du système.

A ce jour, nous n'avons pu réaliser des tests que pour des translations. Nous planifions bien évidemment de poursuivre par des tests de rotation, puis par des tests combinés. Ensuite, nous nous attacherons à altérer les données par l'adjonction des bruits caractéristiques de chaque modalité, puis par l'augmentation des divergences temporelles et enfin, par des tests sur des données réelles.