3D Geometric Analysis of Tubular Objects Based on Surface Normal Accumulation

Bertrand Kerautret, Adrien Krähenbühl, Isabelle Debled-Rennesson et Jacques-Olivier Lachaud

LORIA, Université de Lorraine
Equipe ADAGIO

Colloque de clôture du projet DigitalSnow
Autrans, 8-9 Juillet 2015

Loria
Laboratoire lorrain de recherche en informatique et ses applications

LAMA
Laboratoire de Mathématiques
Université de Savoie

Overview of the presentation

- 1. Introduction
  - 1.1 Tubular Object.
  - 1.2 Objective of geometric measure.
  - 1.3 Existing works.
- 2. Extraction of Centerline from Tubular Objects.
  - 2.1 Accumulation Image defined from Normal Vectors.
  - 2.2 Centerline Tracking from Image Accumulation.
  - 2.3 Optimisation of the center line position.
- 3. Reconstruction and Geometric Analysis
  - 3.1 Reconstruction.
  - 3.2 Arc/line detection.
- 4. Conclusion

1. Introduction

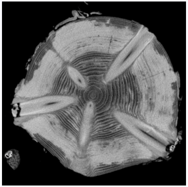
Tubular Objects

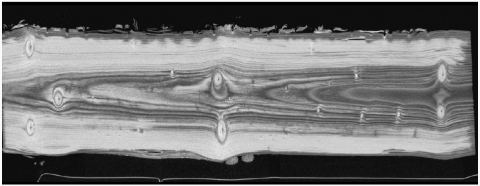
- 3D medical imagery (ex. aneurysm artery cerebral).

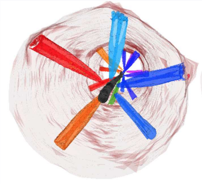
1. Introduction

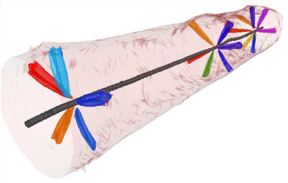
Tubular Objects

- 3D medical imagery (ex. aneurysm artery cerebral).
- Volumic images of wood trunk (to detect knots).





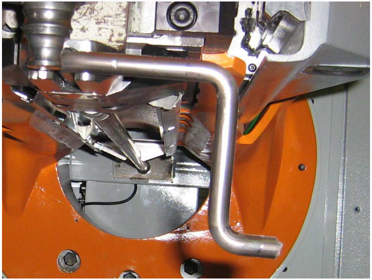




1. Introduction

Tubular Objects

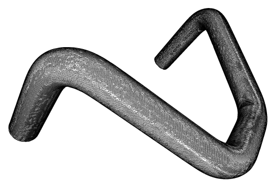
- 3D medical imagery (ex. aneurysm artery cerebral).
- Volumic images of wood trunk (to detect knots).
- Industrial context : metallic tube with bending machine.



1.2 Objective of geometric measure

Analyze of 3D tube

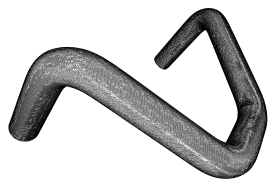
- Motivation : (retro-engineering) recover the parametric model of the tube (LRA model).



1.2 Objective of geometric measure

Analyze of 3D tube

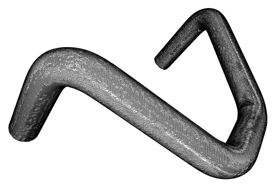
- Motivation : (retro-engineering) recover the parametric model of the tube (LRA model).
- Data points obtained from laser scan.



1.2 Objective of geometric measure

Analyze of 3D tube

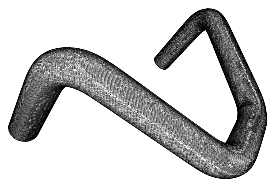
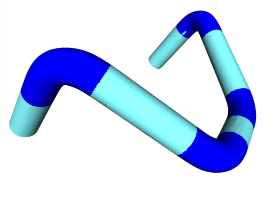
- Motivation : (retro-engineering) recover the parametric model of the tube (LRA model).
- Data points obtained from laser scan.



1.2 Objective of geometric measure

Analyze of 3D tube

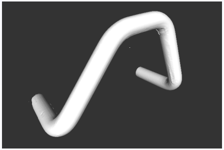
- Motivation : (retro-engineering) recover the parametric model of the tube (LRA model).
- Data points obtained from laser scan.

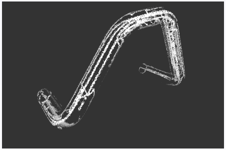




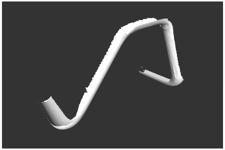
1.2 Objective of geometric measure

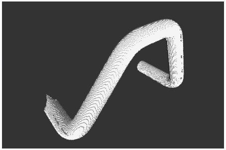
Analyze of 3D tube

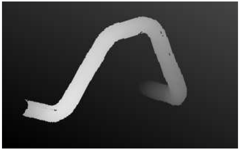
- Motivation : (retro-engineering) recover the parametric model of the tube (LRA model).
- Data points obtained from laser scan.
- Method independent of the type of data : set of point, 3D volume , 2.5D (height map).


full scan


partial scan


partial scan (direction)



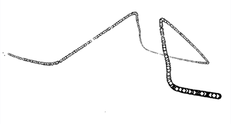


1.3 Existing Works

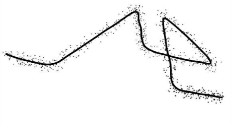
Existing method dealing with a set of points

- Minimal cover and thinning [Lee 00].

point source (1000)



thinned set of point



interpolation by B Spline

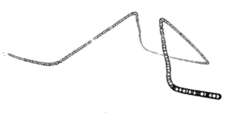
Results extracted from the article [Lee 00]

1.3 Existing Works

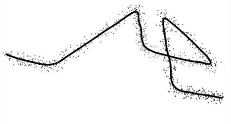
Existing method dealing with a set of points

- Minimal cover and thinning [Lee 00].
- Improvement with tube with non constant diameter [Lee & Kim 04].

point source (1000)



thinned set of point



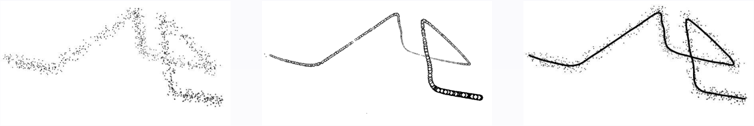
interpolation by B Spline

Results extracted from the article [Lee 00]

1.3 Existing Works

Existing method dealing with a set of points

- Minimal cover and thinning [Lee 00].
- Improvement with tube with non constant diameter [Lee & Kim 04].
- Missing precision for geometric analysis [Bauer & Polthier 09].



point source (1000) thinned set of point interpolation by B Spline

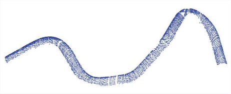
Results extracted from the article [Lee 00]

1.3 Existing Works

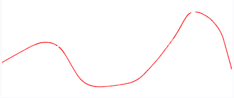
Existing method dealing with a set of points

- Minimal cover and thinning [Lee 00].
- Improvement with tube with non constant diameter [Lee & Kim 04].
- Missing precision for geometric analysis [Bauer & Polthier 09].
- Bauer and Polthier propose a method to recover the parametric model [Bauer & Polthier 09].

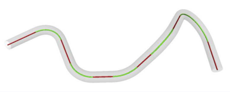
source



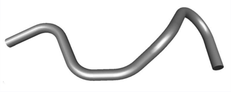
scan partiel



projection



decomposition in arcs/segments



reconstruction

Results extracted from the article [Bauer & Polthier 09]

1.3 Existing Works (2)

Extraction methods from median axis

- Method given from volumic data (see review [Cornea & Silver 07]).
- Sensible small defects.

(a) thinning
1 min 4s

(b) geometric
0.5 s

(c) potential field
3min

(d) proposed
6s+3s

1.3 Existing Works (3)

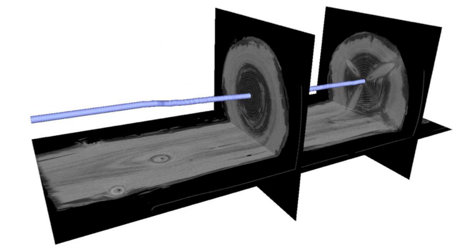
Other methods

- Method based on GVF [Xu & Prince 98] to process volumetric discrete object [Bauer & Bischof 08].
- Method to process incomplete set of points [Tagliasacchi *et al.* 09].

2. Extraction of Centerline from Tubular Objects

Main idea :

- Exploit an accumulation image defined from normal vectors.
- Idea inspired from the pith detection of wood trunk :
  $\Rightarrow$ PhD thesis of Adrien Krähenbühl
- Simple dependance through the surface normals.



2. Extraction of Centerline from Tubular Objects

Main idea :

- Exploit an accumulation image defined from normal vectors.
- Idea inspired from the pith detection of wood trunk :
  $\Rightarrow$ PhD thesis of Adrien Krähenbühl
- Simple dependance through the surface normals.

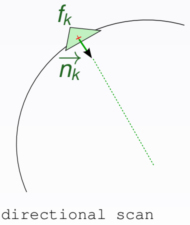
Method defined by 3 main steps :

- Step 1 : accumulation of the surface normals.
- Step 2 : tracking of the local maxima accumulation.
- Step 3 : optimisation of the position of the digital points of previous step.

2.1 Accumulation Image defined from Normal Vectors

Accumulation according the surface normals

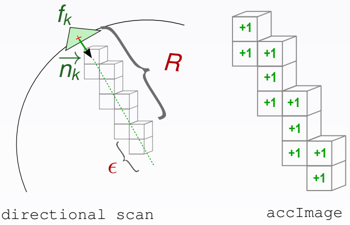
- From the input mesh faces f_k, in the normal direction $\overrightarrow{n_k}$.



2.1 Accumulation Image defined from Normal Vectors

Accumulation according the surface normals

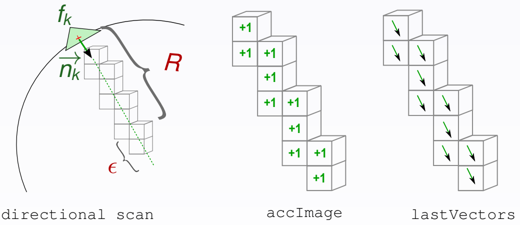
- From the input mesh faces f_k, in the normal direction $\overrightarrow{n_k}$.
- Increase by 1 all voxels intersecting the line defined by $\overrightarrow{n_k}$.
  $\Rightarrow$ on a maximal distance $R + \epsilon$.



2.1 Accumulation Image defined from Normal Vectors

Accumulation according the surface normals

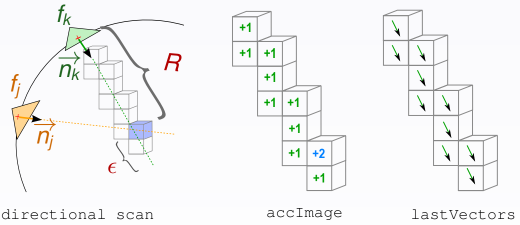
- From the input mesh faces f_k, in the normal direction $\overrightarrow{n_k}$.
- Increase by 1 all voxels intersecting the line defined by $\overrightarrow{n_k}$.
  $\Rightarrow$ on a maximal distance $R + \epsilon$.
- Retain the direction of the vector contributing to the accumulation.



2.1 Accumulation Image defined from Normal Vectors

Accumulation according the surface normals

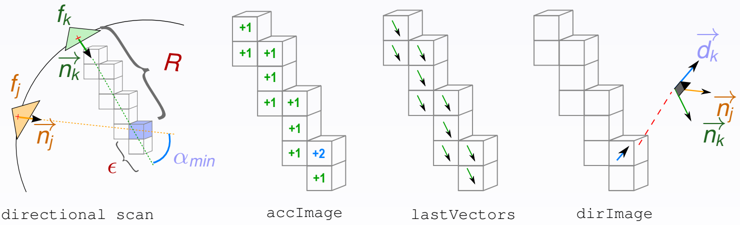
- From the input mesh faces f_k, in the normal direction $\overrightarrow{n_k}$.
- Increase by 1 all voxels intersecting the line defined by $\overrightarrow{n_k}$.
  $\Rightarrow$ on a maximal distance $R + \epsilon$.
- Retain the direction of the vector contributing to the accumulation.



2.1 Accumulation Image defined from Normal Vectors

Accumulation according the surface normals

- From the input mesh faces f_k, in the normal direction $\overrightarrow{n_k}$.
- Increase by 1 all voxels intersecting the line defined by $\overrightarrow{n_k}$.
  $\Rightarrow$ on a maximal distance $R + \epsilon$.
- Retain the direction of the vector contributing to the accumulation.
- Compute the principal direction of an accumulation voxel : $\overrightarrow{d_k} = \overrightarrow{d_j} + (\overrightarrow{n_k} \wedge \overrightarrow{n_j})$



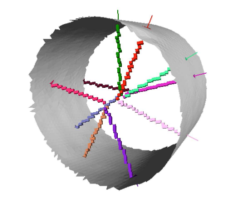
2.1 Image d'accumulation définie à partir des vecteurs normaux (2)

```
Input: mesh // Triangular mesh of a tube
accRadius // Accumulation length from center of faces
minNorm = 0.1 // Minimum norm value
Ouput: accImage // Accumulation of normal vector number passing through a coordinate
dirImage // Cross product of all normals passing through a coordinate
maxAcc // Maximum number of normals passing through an (x,y,z) coordinate
maxPt // maxAcc coordinates
Variable: lastVectors // The last considered normal for each (x,y,z) coordinate
mainAxis // Vector contributing to the cross product of a directional vector
lastVectors = Image3D(mesh.dimensions())
maxAcc = 0
foreach face in mesh do
    currentPt = face.center
    normalVector = face.normalVector().normalized()
    while distance(currentPt, face.center) < accRadius do
        if accImage[currentPt] != 0 then
            mainAxis = lastVectors[currentPt] × normalVector
            if norm(mainAxis) > minNorm then
                dirImage[currentPt] += mainAxis*sign(mainAxis • dirImage[currentPt])
        lastVectors[currentPt] = normalVector
        accImage[currentPt]++
        if accImage[currentPt] > maxAcc then
            maxAcc = accImage[currentPt]
            maxPt = currentPt
        currentPt += normalVector
```

2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

Illustration of the construction

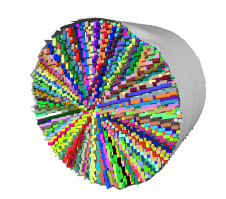
- Accumumation from normals.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

Illustration of the construction

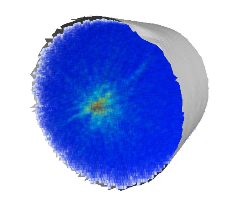
- Accumumation from normals.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

Illustration of the construction

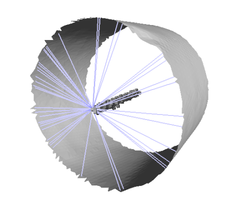
- Accumumation from normals.
- Image of accumulation.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

Illustration of the construction

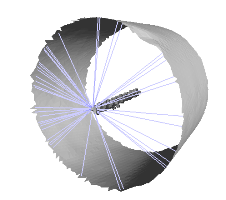
- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

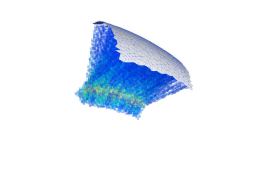
Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



Robustess towards different types of data

- partial mesh,



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

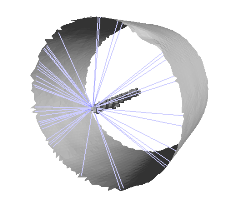
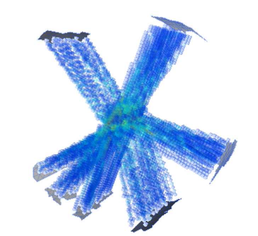


Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.

Robustess towards different types of data

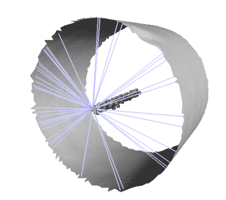
- partial mesh,



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

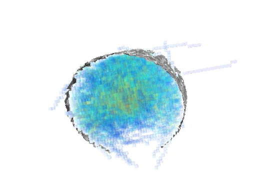
Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



Robustess towards different types of data

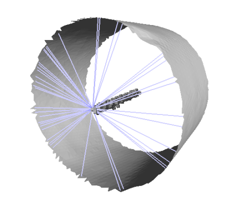
- partial mesh,
- noise,



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

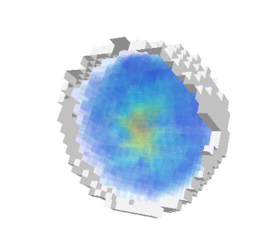
Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



Robustess towards different types of data

- partial mesh,
- noise,
- digital surface,



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

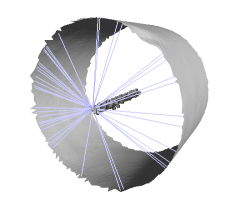
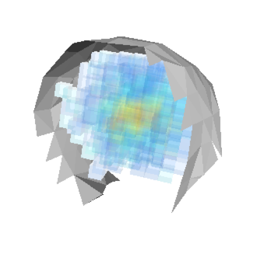


Illustration of the construction

- Accumamation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.

Robustess towards different types of data

- partial mesh,
- noise,
- digital surface,
- low resolution mesh.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

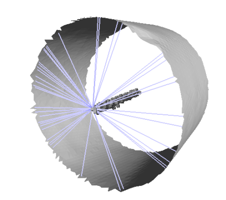
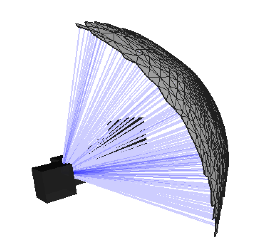


Illustration of the construction

- Accumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.

Robustess towards different types of data

- partial mesh,
- noise,
- digital surface,
- low resolution mesh.



2.1 Image d’accumulation définie à partir des vecteurs normaux (3)

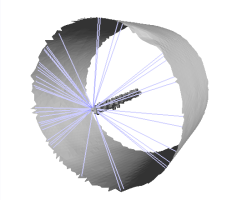
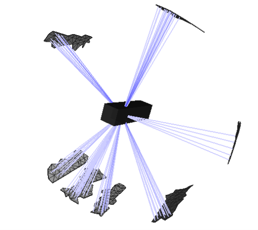


Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



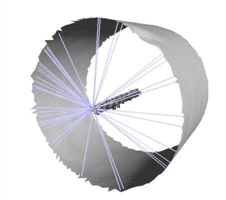
Robustess towards different types of data

- partial mesh,
- noise,
- digital surface,
- low resolution mesh.

2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

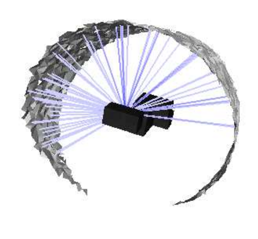
Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



Robustess towards different types of data

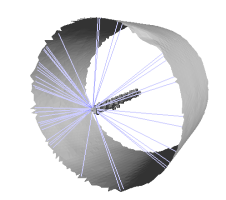
- partial mesh,
- noise,
- digital surface,
- low resolution mesh.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

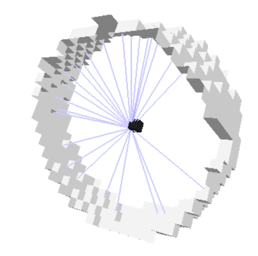
Illustration of the construction

- Accumumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



Robustess towards different types of data

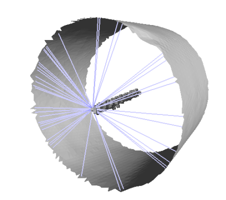
- partial mesh,
- noise,
- digital surface,
- low resolution mesh.



2.1 Image d'accumulation définie à partir des vecteurs normaux (3)

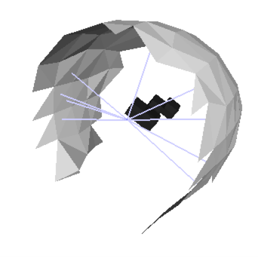
Illustration of the construction

- Accumation from normals.
- Image of accumulation.
- Recover the source voxel associated to a voxel.



Robustess towards different types of data

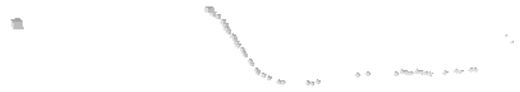
- partial mesh,
- noise,
- digital surface,
- low resolution mesh.



2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.



$t = 150$

2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

$t = 100$

2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

$t = 50$

2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

$t = 25$

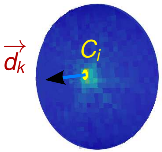
2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

Tracking algorithm

- Start from maximal accumulation point C_0.
- Define patch image I^i_{patch} for the current point C_i .
- Exploit main tubular direction $\overrightarrow{d_k}$ from I^i_{patch}.



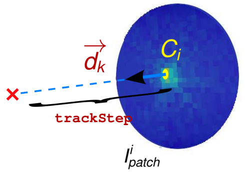
2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

Tracking algorithm

- Start from maximal accumulation point C_0.
- Define patch image I^i_{patch} for the current point C_i .
- Exploit main tubular direction $\overrightarrow{d_k}$ from I^i_{patch}.



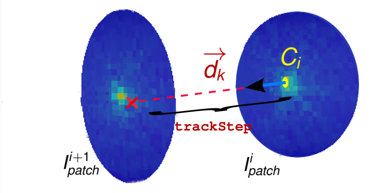
2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

Tracking algorithm

- Start from maximal accumulation point C_0.
- Define patch image I^i_{patch} for the current point C_i .
- Exploit main tubular direction $\overrightarrow{d_k}$ from I^i_{patch}.



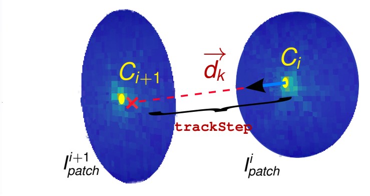
2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

Tracking algorithm

- Start from maximal accumulation point C_0.
- Define patch image I^i_{patch} for the current point C_i .
- Exploit main tubular direction $\overrightarrow{d_k}$ from I^i_{patch}.



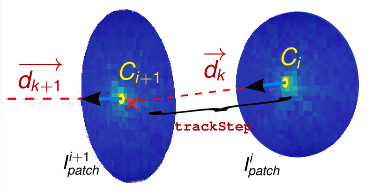
2.2 Centerline Tracking from Image Accumulation

Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

Tracking algorithm

- Start from maximal accumulation point C_0.
- Define patch image I^i_{patch} for the current point C_i .
- Exploit main tubular direction $\overrightarrow{d_k}$ from I^i_{patch}.



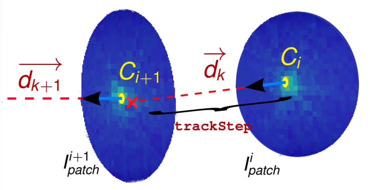
2.2 Centerline Tracking from Image Accumulation

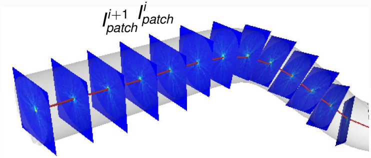
Basic extraction

- Single threshold t not direct.
- Choice of parameter t.
- Not necessary connected.

Tracking algorithm

- Start from maximal accumulation point C_0.
- Define patch image I^i_{patch} for the current point C_i .
- Exploit main tubular direction $\overrightarrow{d_k}$ from I^i_{patch}.





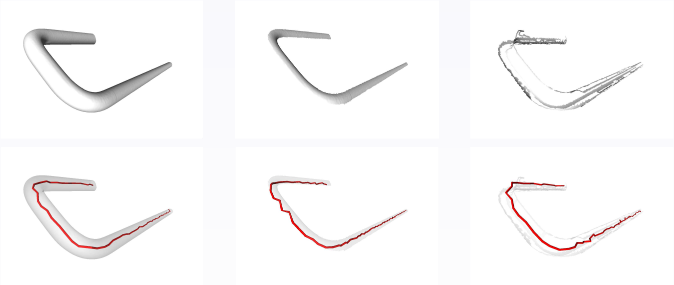
2.2 Centerline Tracking from Image Accumulation (2)

```
patchImageSize = 2 * aRadius ;
centerline = emptySet()
continueTracking = true
patchSize = 2 * accRadius
currentPt = startPt
lastVect = trackInFront ? dirImage( startPt ) : - dirImage( startPt )
previousPt = startPt - lastVect * trackStep
while continueTracking do
    centerline.append( currentPt )
    dirVect = dirImage[currentPt].normalized()
    if lastVect.dot(dirVect) < 0 then
        dirVect = -dirVect
    continueTracking = isInsideTube( accImage, currentPt, previousPt, trackStep, π/3 )
    previousPt = currentPt
    // Defined the next image patch center point
    centerPatch = currentPt + ( dirVect * trackStep )
    if not accImage.domain().contains( centerPatch ) then
        break
    // Extract a 2D image of size 2 * accRadius from the 3D image accImage, centered on
    // centerPatch and directed along dirVect
    patchImage = extractPatch( accImage, centerPatch, dirVect, 2 * accRadius )
    maxCoords = getMaxCoords( patchImage )
    lastVect = dirVect
    previousPt = currentPt
    currentPt = patchSpaceToAccImageSpace( maxCoords)
return centerline
```

2.3 Optimisation of the center line position (1)

Main drawback

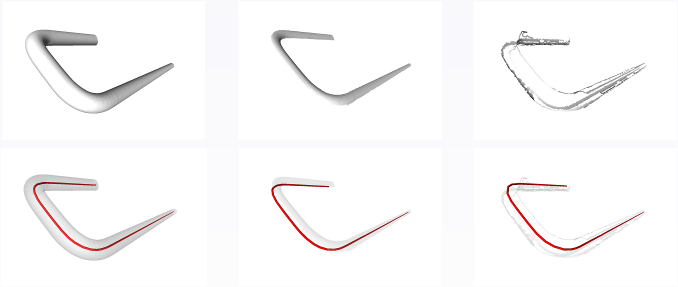
- Due to digital space, the center line is not perfectly smooth/centered.
- Maximal accumulation not always perfectly centered in case of noise.



2.3 Optimisation of the center line position (1)

Main drawback

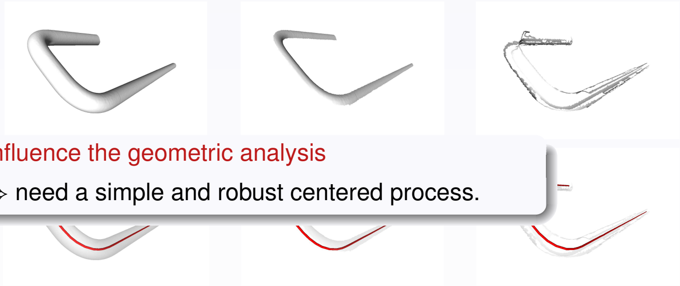
- Due to digital space, the center line is not perfectly smooth/centered.
- Maximal accumulation not always perfectly centered in case of noise.



2.3 Optimisation of the center line position (1)

Main drawback

- Due to digital space, the center line is not perfectly smooth/centered.
- Maximal accumulation not always perfectly centered in case of noise.



Influence the geometric analysis

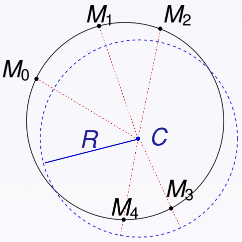
⇒ need a simple and robust centered process.

2.3 Optimisation of the center line position (2)

Main idea of the optimisation process

- Minimize the fitting error $E_s(C)$ between the center line position C and the tube of radius R.

$$E_s(C) = \sum_{i=0}^{N-1} (\|\overrightarrow{CM_i}\| - R)^2. \qquad (1)$$

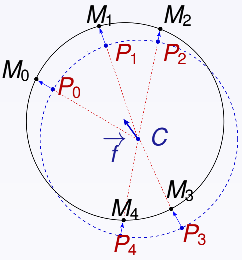


2.3 Optimisation of the center line position (2)

Main idea of the optimisation process

- Minimize the fitting error $E_s(C)$ between the center line position C and the tube of radius R.

$$E_s(C) = \sum_{i=0}^{N-1} (\|\overrightarrow{CM_i}\| - R)^2. \quad (1)$$



Minimisation from simple gradient descent

- By derivation :

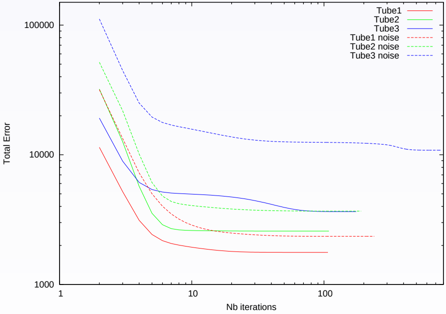
$$\nabla E_s(C) = 2 \sum_{i=0}^{N-1} \frac{\overrightarrow{CM_i}}{\|\overrightarrow{CM_i}\|} (R - \|\overrightarrow{CM_i}\|). \quad (2)$$

- Can be interpreted as elastic forces, minimized by applying on C at each step : $\overrightarrow{f} = \sum_{i=0}^{N} \overrightarrow{P_i M_i}$.

2.3 Optimisation of the center line position (3)

Simple optimisation

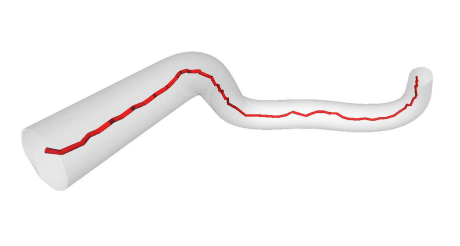
- Only one parameter ϵ_o for convergence.
- Direct to implement and fast convergence.
- Comparable process used to fit 3D curves on unorganized data points [Fang *et al.* 1992].



2.3 Optimisation of the center line position (3)

Simple optimisation

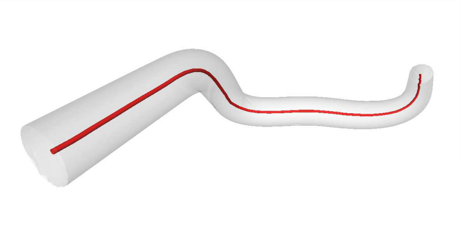
- Only one parameter ϵ_o for convergence.
- Direct to implement and fast convergence.
- Comparable process used to fit 3D curves on unorganized data points [Fang *et al.* 1992].



2.3 Optimisation of the center line position (3)

Simple optimisation

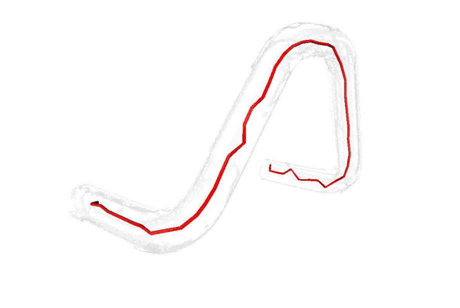
- Only one parameter ϵ_o for convergence.
- Direct to implement and fast convergence.
- Comparable process used to fit 3D curves on unorganized data points [Fang *et al.* 1992].



2.3 Optimisation of the center line position (3)

Simple optimisation

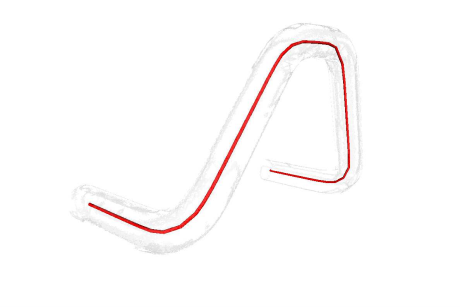
- Only one parameter ϵ_o for convergence.
- Direct to implement and fast convergence.
- Comparable process used to fit 3D curves on unorganized data points [Fang *et al.* 1992].



2.3 Optimisation of the center line position (3)

Simple optimisation

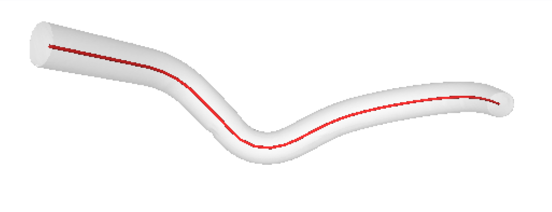
- Only one parameter ϵ_o for convergence.
- Direct to implement and fast convergence.
- Comparable process used to fit 3D curves on unorganized data points [Fang *et al.* 1992].



3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh,

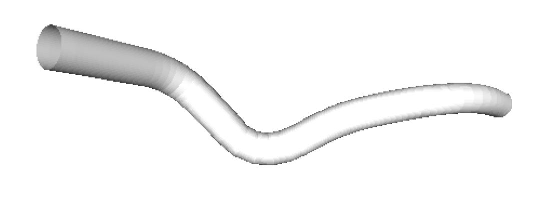


R = 6, t=6.23s, 151 444 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh,

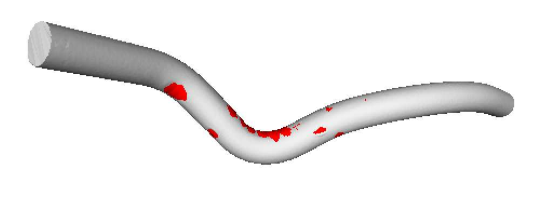


R = 6, t=6.23s, 151 444 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh,

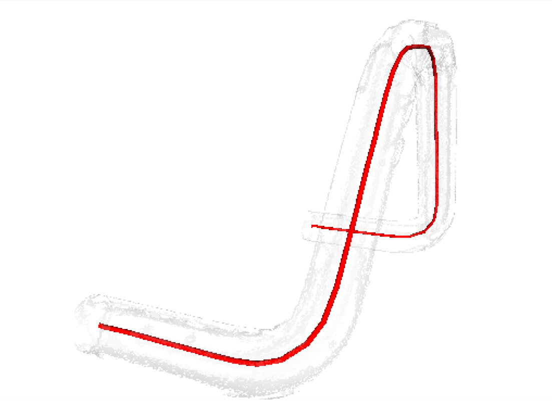


R = 6, t=6.23s, 151 444 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh,

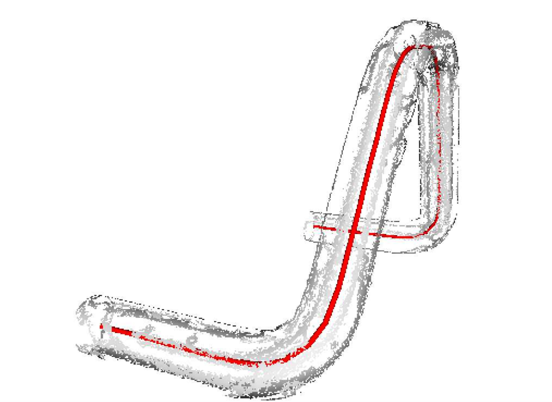


R = 6, t=2.45s, 37 527 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh,

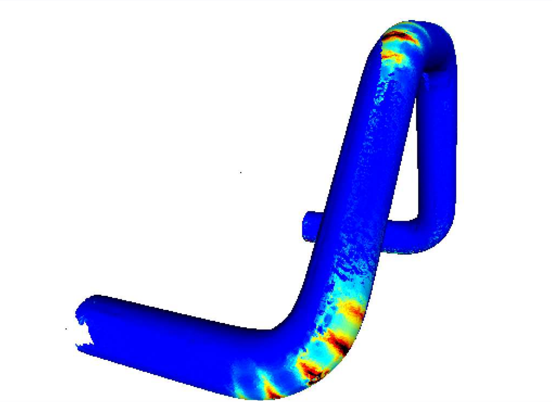


R = 6, t=2.45s, 37 527 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh,

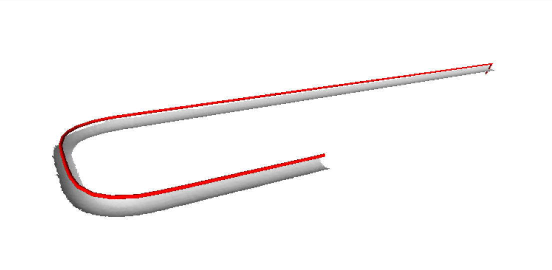


R = 6, t=2.45s, 37 527 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area,

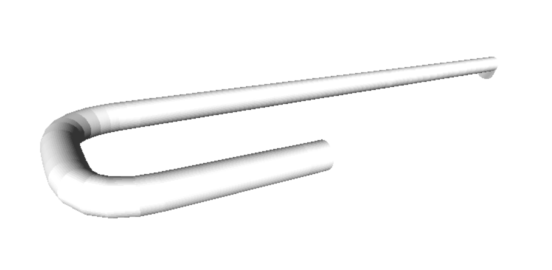


R = 6, t=4.36s, 52 914 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area,

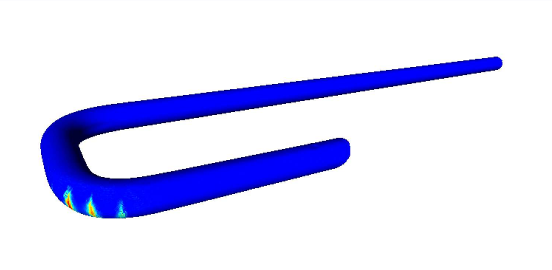


R = 6, t=4.36s, 52 914 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area,

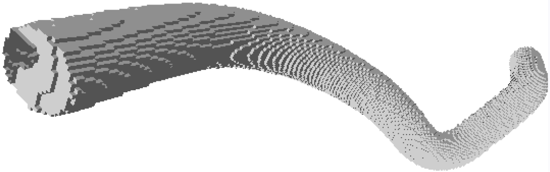


R = 6, t=4.36s, 52 914 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area, digital object,

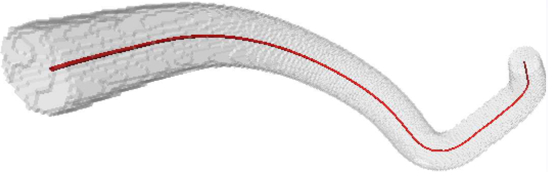


R = 4.9, t=6.91s, 60 768 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area, digital object,

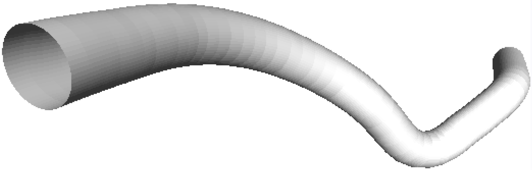


R = 4.9, t=6.91s, 60 768 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area, digital object,

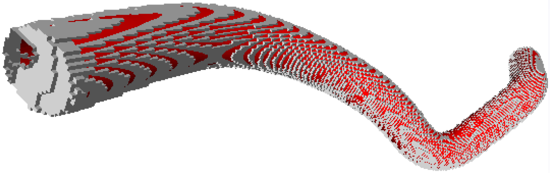


R = 4.9, t=6.91s, 60 768 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area, digital object,

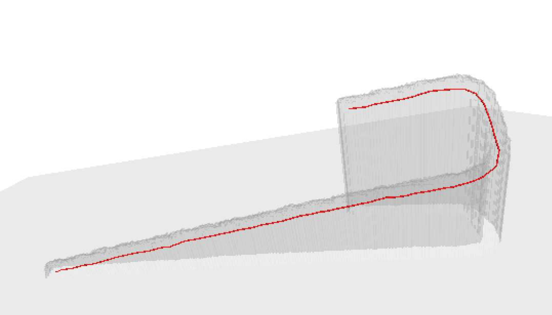


R = 4.9, t=6.91s, 60 768 faces

3.1 Reconstruction

Reconstruction on various types of data :

⇒ full mesh, partial mesh, reduced scaned area, digital object, Height map

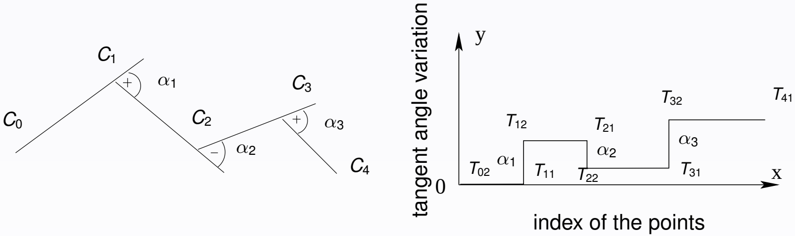


R = 8,t=22.33s, 645 450 faces

3.2 Arc/line detection

Main idea

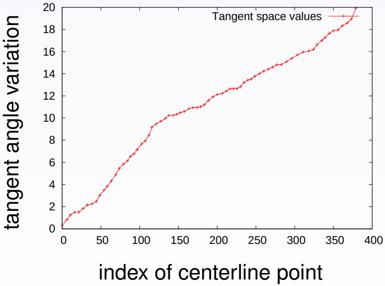
- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.

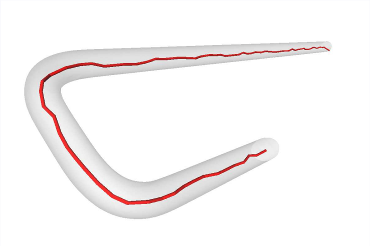


3.2 Arc/line detection

Main idea

- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.

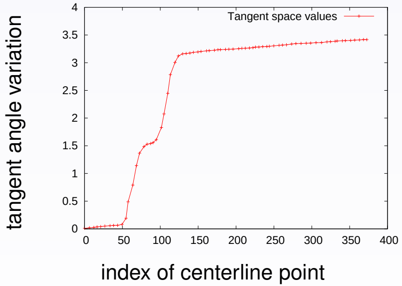


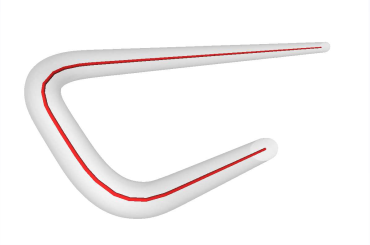


3.2 Arc/line detection

Main idea

- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.

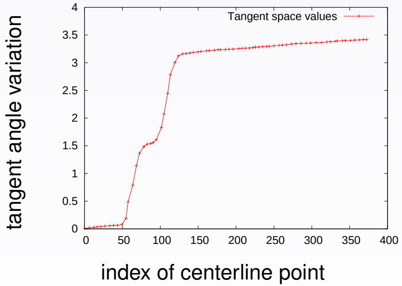


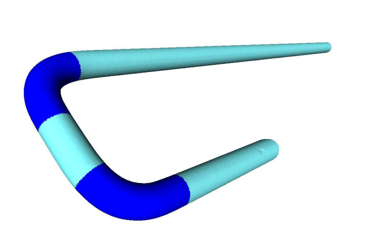


3.2 Arc/line detection

Main idea

- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.

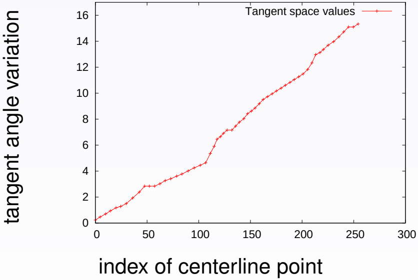


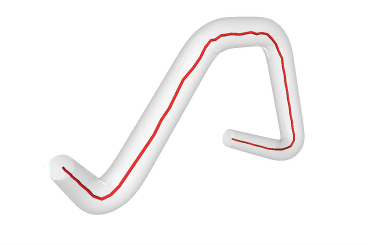


3.2 Arc/line detection

Main idea

- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.

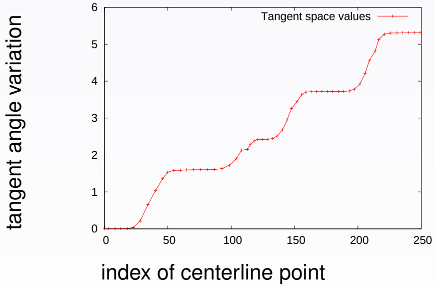


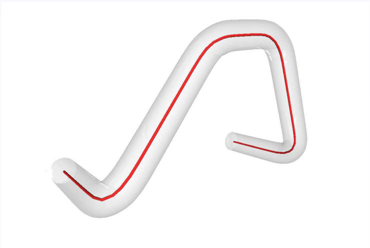


3.2 Arc/line detection

Main idea

- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.

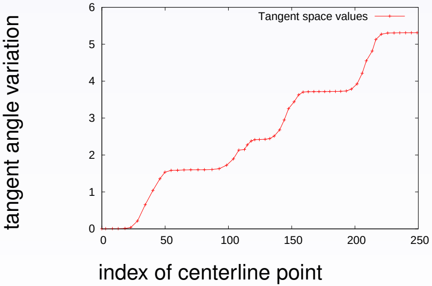


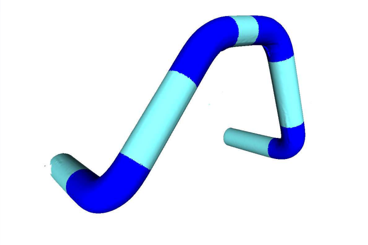


3.2 Arc/line detection

Main idea

- 3d arcs detection based on tangent space [Latecki & Lakamper 00].
- Extend to 3D the previous work of 2D arc detection [Nguyen & Debled 11].
- Detect co-linear sequence of points in the tangent space.





Implementation and demonstration online

Implementation

- Implemented using the DGtal Library [DGtal].
- All 3D illustrations applied from the DGtalTools package.
- Will be included as new tools in DGtalTools.

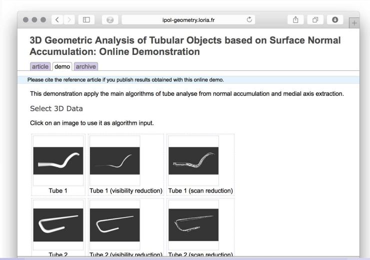
Implementation and demonstration online

Implementation

- Implemented using the DGtal Library [DGtal].
- All 3D illustrations applied from the DGtalTools package.
- Will be included as new tools in DGtalTools.

Online demonstration

- Online demonstration construction is in progress.
- Will be available in the ”workshop” section of the IPOL journal (`www.ipol.im`).



4. Conclusion

- Simple method to extract the centerline of tubular object.
- Fast with only few parameters.
- Independant of the type of input data (mesh, voxels, heightmap).

Perspectives

- Extend in gray level images.
- Improve the tracking with the procesing of joint.
- Adaptation to deal with non constant radius (need in the optimisation)

Merci de votre attention !

[Lee 00] Lee, I.K.
Curve reconstruction from unorganized points.
Computer aided geometric design 17(2), 161–177 (2000)

[Lee & Kim 04] Lee, I.K. and Kim, K.J.
Shrinking : Another method for surface reconstruction.
In : Geometric Modeling and Processing, 2004. Proceedings. pp. 259–266. IEEE (2004)

[Bauer & Polthier 09] Bauer, U., Polthier, K.
Generating parametric models of tubes from laser scans.
Computer-Aided Design 41(10), 719–729 (Oct 2009)

[Cornea & Silver 07] Cornea, N.D., Silver, D.
Curve-skeleton properties, applications, and algorithms.
IEEE Transactions on Visualization and Computer Graphics 13, 530–548 (2007)

[Xu & Prince 98] Xu, C., Prince, J.
Snakes, shapes, and gradient vector flow.
IEEE Transactions on Image Processing 7(3), 359–369 (Mar 1998)

[Bauer & Bischof 08] Bauer, C., Bischof, H.
Extracting Curve Skeletons from Gray Value Images for Virtual Endoscopy
Medical Imaging and Augmented Reality, pp. 393–402. No. 5128 in LNCS (2008)

[Tagliasacchi *et al.* 09] Tagliasacchi, A., Zhang, H., Cohen-Or, D.
Curve skeleton extraction from incomplete point cloud.
ACM Transactions on Graphics (TOG) 28(3), 71 (2009)

[Kerautret *et al.*] B Kerautret, A. Krähenbühl, I. Debled-Rennesson and J.-O. Lachaud
3D Geometric Analysis of Tubular Objects based on Surface Normal Accumulation
To appears in Proc ICIAP 2015

[Fang *et al.* 1992] Fang, Lian and Gossard, David C.
Fitting 3D Curves to Unorganized Data Points Using Deformable Curves
Visual Computing, 535–543 1992

[Latecki & Lakamper 00] Latecki, L., Lakamper, R.
Shape similarity measure based on correspondence of visual parts
IEEE Transactions on PAMI 22, 1185–1190 (2000)

[Nguyen & Debled 11] Nguyen, T.P., Debled-Rennesson, I.
Arc segmentation in linear time
14th International Conference, CAIP, Seville, Spain. LNCS, vol. 6854, pp. 84–92 (2011)

[DGtal] DGtal : Digital Geometry tools and algorithms library,
`http://libdgtal.org`