

New Method for 4D Reconstruction of Medical Images

Lamyae MIARA, Said BENOMAR ELMDEGHRI, Mohammed Ouçamah CHERKAOUI MALKI

Department of Computer Science, University Sidi Mohammed Ben Abdellah
Faculty of Sciences Dhar El Mahraz, Fez, Morocco

Abstract—This paper proposes a new optimized method that is fast in rendering for 4D reconstruction from 2D medical images of human anatomy permitting their real-time refined visualization. This method uses the 3D reconstruction algorithm based on contour matching of medical image sequences and on the tessellation of recent GPU. In our framework, the construction of the low-resolution mesh that is based on contour extraction allows to create a 3D mesh without any ambiguity and exactly matches the real shape of the human anatomy. Such preliminary result is of great interest, since it permits to lead to other valuable realizations such as reducing the computation burden of basic meshes and displacement vectors. Moreover, one can achieve a very low storage memory, as well as one can ease the fast real-time 4D visualization with a high desired resolution. Hence, it is then straight forward that this study can contribute to easing the diagnosis and detection in real-time of human organs in motion damage and deterioration. Especially, 4D visualization technology that is still under development is highly important and needed for assessing some dangerously evaluative diseases, as in the case of lung diseases.

Keywords—2D medical image; 4D reconstruction; contour matching; recent GPU; 3D mesh

I. INTRODUCTION

From health and care perspectives, 4D visualization is the most effective method for assessing and preventing deadly diseases of human organs in motion. For instance, 4D visualization technology of lungs damaged can shed new light on the real possible dangers and eventual damage and complications on the patient health. As a real practical consequence, the use of computational methods in lung analysis may allow us to better understand and visualize what we cannot obtain from static 2D images, such as the respiratory behavior of muscles. To achieve this, the 4D reconstruction is an important and critical requirement.

More importantly, not to say that 4D reconstruction of the human organs in motion would be useful for visualization the correspondent mechanism, especially before surgery. For example, in patients suspected of having pulmonary diseases, the ability to visualize in 4D and observe the oxygen uptake capacity of the lungs would be useful for clinical diagnosis. So, a complete 4D reconstruction of the lungs may allow physicians to better understand the respiratory process (inspiration and expiration). Other disease states in which 4D modeling can potentially be useful for clinical decision making include pulmonary nodules, pneumothorax, and chronic diseases [1] characterized by irreversible decrease in bronchial caliber.

Unfortunately, the 4D reconstruction is a computationally intensive job, and therefore doctors use specific workstations [2].

In this paper, we propose a powerful and optimized algorithm to reconstruct a 4D image from segmented contours of 2D MRI images, to be able to do the 4D reconstruction in home computers.

Our proposed framework has the following advantages:

- Topological changes in 2D contours are automatically processed in both spatial and temporal dimensions.
- The result is a distortion-free mesh that corresponds to a good approximation that matches exactly the real human organs anatomy.
- The use of the tessellation of recent graphics cards makes the real-time display very fast.
- Memory usage is highly optimized by considering the large number of details in the 2D images.

The remainder of this paper is organized as follows. In section 2, an overview on the state of the art is presented and the limitations of some reported works are discussed. In section 3, our methodology is clearly exposed regarding to the previously reported studies. Section 4 presents some technical explanations and discussions on the proposed method and its obtained results. Finally, some necessary concluding remarks are provided in section 5.

II. RELATED WORK

Research studies on 4D reconstruction algorithms often focuses on both, improving the quality of 4D visualization from standard acquisition data and its fast execution. Among these algorithms is the Feldkamp-Davis-Kress (FDK) algorithm [3], which is a filtered back-projection algorithm widely used for 3D and 4D image reconstructions from cone-beam projections measured with a circular orbit of the x-ray source. There is also the Mc Kinnon-Bates (MKB) algorithm [4] in which a time-averaged 3D prior image is first reconstructed. It is then projected at the same angles as the original projection data, creating time-averaged re-projection that are next subtracted from the original (non-blurred) projections to create the well-known motion-coded differential projections. Such differential projections are reconstructed into PC differential images that are added to the well-sampled 3D image before creating a much higher quality 4D image.

Other distinguished algorithm is the ROOSTER algorithm [5] that iteratively computes volumes that minimize the sum of the least squares difference between simulated volume projections, real measured projections, and a spatiotemporal "total variation" term that favors distinct homogeneous regions with sharp edges. In addition, we also mention the Motion Compensated [6] FDK (MCFDK) algorithm [7] in which motion was estimated from 4DCT planning volumes by Deformable Image Registration (DIR) [8] between respiratory phase volumes using a B-spline method from the Elastix toolkit [9]. In addition, the 4DCT DIR produces Deformation Vector Fields (DVF) that are used as inputs to the MCFDK reconstruction. Note that the MCFDK algorithm can be interpreted as a variant of the FDK method where back-projection is performed along curved trajectories to account for motion. And lastly the Motion Compensated (MCMKB) algorithm which is an extension of [10], where 4DFDK reconstructions are used for DVF estimation. This study considers a much more under sampled acquisition than [10], which may explain why we were not able to produce convergent DVF estimates from the 4DFDK reconstructions.

Note that for the algorithms that use surface rendering, there is the classical method that uses interpolation between 3D images of to give a 4D result. These algorithms are effective alternatives for 4D reconstruction of medical images without the need for powerful machines, but they produce ghost artifacts that do not represent the exact anatomy of human organs and not achieve the determination of accurate motion trajectories. Another negative fact of using mentioned algorithms, is that it cannot lead to an instantaneous precise information on a particular organ and even to precise measurements on the organ to be reconstructed. In contrast, our proposed approach is based on the 4D surface rendering that allows a good visualization of the geometry and a precise topological shape associated to the three-dimensional structures of human organs and their spatial-temporal relationships. Moreover, such adapted procedure makes it possible for comparing the reconstructed data with the original data. For these reasons, it is stressed out that the surface rendering method eases the achievement of a fast real-time 4D visualization, thanks to the use of certain units of the graphic card originally conceived for 3D video games.

To recap, the main goal of this paper is mainly to propose a new algorithmic solution that is faster to compute based on accelerating rendering method and other efficient computational techniques. Moreover, the proposed methodology can achieve a high visual resolution by avoiding heavy drawbacks on memory storage and acquisition. Indeed, such improvement handles efficiently the limitations of the previously discussed algorithms that have been reported and recognized in the literature.

III. METHODOLOGY

This section describes the new method developed in this paper for an optimized and fast 4D visualization from standard 2D medical images. This method consists first of the 3D

reconstruction of a basic mesh from the 2D images of the C_m slices at time T_0 and the extraction of the lost data, in the form of displacement vectors, which will be stored in a displacement map. Thus, we perform the extraction of the displacement vectors of the 2D slices of each sequence from time T_1 to time T_{N-1} based on the mesh at time T_0 , i.e., without constructing the base mesh of times T_1 to T_{N-1} . In special cases, it is found that the number of contours at time T_i changes in time T_{i+1} , which makes it necessary to add additional parts in the base mesh. As a result, a high-resolution visualization can be achieved at the GPU map level by combining the different parts of the base mesh and the set of displacement maps.

The reminder of our contribution is as follows. In the first part given by section 3.1, we determine the input of our algorithm and explain the basic idea behind its conception. Next, the second and third and fourth parts considered in section 3.2 and 3.3 and 3.4 describe in more details the first steps of the proposed algorithm consisting of contours extraction and displacement maps. Some technical details are provided in section 3.5 to show how one can map the computed mesh at time T_i to the other at time T_{i+1} . Finally, it is shown how a high resolution 4D visualization can be handled in the last Section 3.6.

A. Overview

Our algorithm is based on a sequence of 2D medical images identified at successive time periods from time T_0 to time T_{N-1} . These images are subjected to a segmentation process to extract the target organ to be reconstructed and visualized in real-time. For this purpose, we apply the contour extraction algorithm on each slice C_k of each sequence S at each time T_i . In this way, we build the base mesh of the sequence at time T_0 and perform the extraction design of the corresponding displacement map, as well as for the other sequences (see Fig. 1). It is stressed out that, in the general, one may have a base mesh composed of other several bases mesh. The 4D reconstruction is then done with a single base mesh and $N+1$ displacement maps, which speeds up the rendering and optimizes memory usage.

In what follows, the illustration in Fig. 2 depicts the necessary steps to be executed to get the desired results for a fast and clear 4D visualization.

B. Extraction Des Contours

Based on the segmented slices, we need to extract the contours of the object to be reconstructed. For the detection of its contours, several methods can be possibly used. Such methods are grouped into several distinct classes. For instance, one can use those based on nonlinear filtering such as the median filter, or more recently the one in [11]. Other methods of interest are high-pass filtering, such as Prewitt, Sobel, and Canny detectors [12], or such that the multi-scale analysis developed with the wavelet theory [13] [14]; or the one based on the rare redundant dictionary approximation [15].

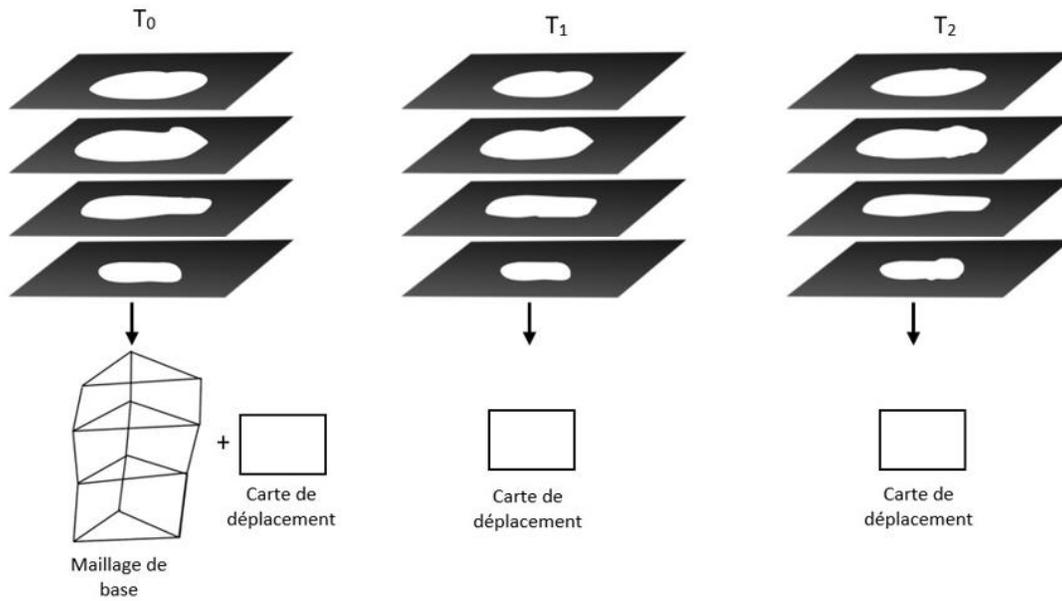


Fig. 1. Construction of Base Mesh and Displacement Map from a 2D Segmented Image Sequence.

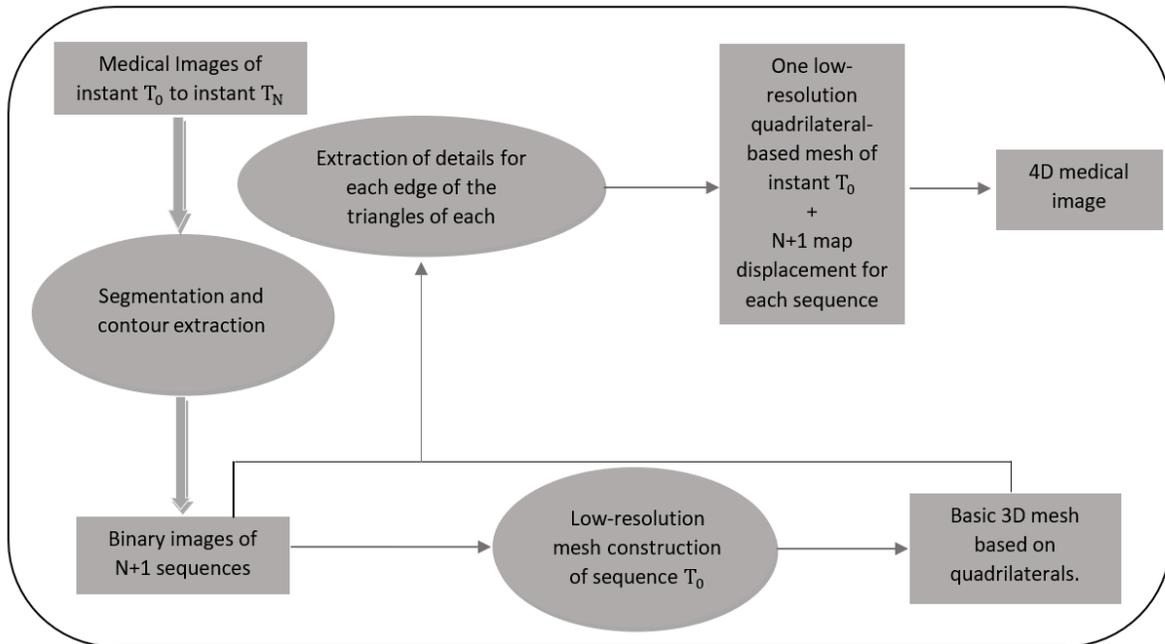


Fig. 2. Diagram Representing the Different Steps of our Algorithm

Note that the key role of the previously mentioned methods of contour traversal consists in defining the abrupt changes of pixel intensity. In fact, the real rising issue to overcome is that we should obtain a chained list of contour points, respecting a fixed order. So, the strategy we used is to extract a point from this contour, and then we must extract the other points in an ordered way following the path of these contours as it has been performed in [16].

To construct the displacement map, it is emphasized that the extraction of the contours shall be carried out for all the slices of all the sequences at the time periods T_i (see Fig. 3).

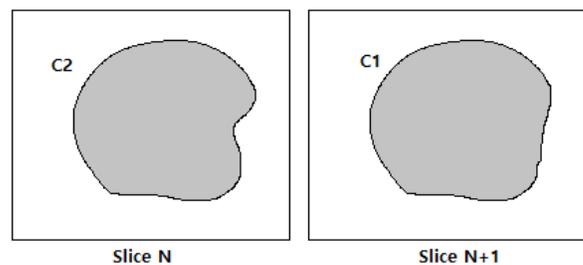


Fig. 3. Two Contours C_1 and C_2 of Two Successive Slices: the N Slice and the N+1 Slice at Time t .

C. Construction of the basic Mesh

The basic mesh is the first structure to be reconstructed without refined details. Note that in our design approach, the mesh construction consists in mapping the contours represented by triangles of each contour of each slice n with those of the slice $n+1$. As a matter of fact, drawing faces from the contours is not at all an easy task because this depends on the resolution of the following three hard problems [17]:

- Matching problem: How to connect the contours in slice n with a contour in slice $n+1$?
- Tiling problem: How to connect the points of contour C_n in slice n with the points of contour C_{n+1} in slice $n+1$?
- Connection problem: How to divide the contour C_i in slice n that corresponds to the contours C'_{n+1} and C''_{n+1} in slice $n+1$?

The illustration shown in Fig. 4 highlights the possibilities for solving the matching problem associated to the case when the number of contours in adjacent slices is not the same (the contours can be split or merged).

It is worth mentioning that for the determination of the relationship between the contours of two consecutive slices (see Fig. 5), we have used the correspondence factor method, see for instance [16].

D. Displacement Vectors

As it has been revealed in previous illustrations, the first constructed mesh is a basic mesh which does not reflect the true shape of the anatomy in question. It is then necessary to add to it the lost details in the construction of the base mesh during the real-time visualization. For this task, we shall then extract the displacement vectors using the available contour data and the edges of the quadrilaterals obtained after joining the triangles that represent the contours [18].

The strategy used to extract the displacement vectors is based on the discretization of the concerned contours (see Fig. 6). This approach has a great interest for two essential issues. On one hand, the generation of such discretized displacement vectors allows more precision for the construction of the real shape of the organ anatomy (see Fig. 7). On the second hand, through the discretization of the anatomy contour with a fixed point, this makes it easy to incorporate the suitable level of details in all polygons of the base mesh.

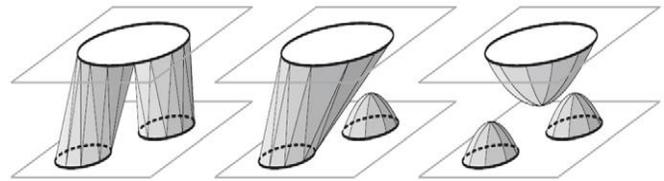


Fig. 4. Three Possible Solutions to the Topology Change Matching Problem.

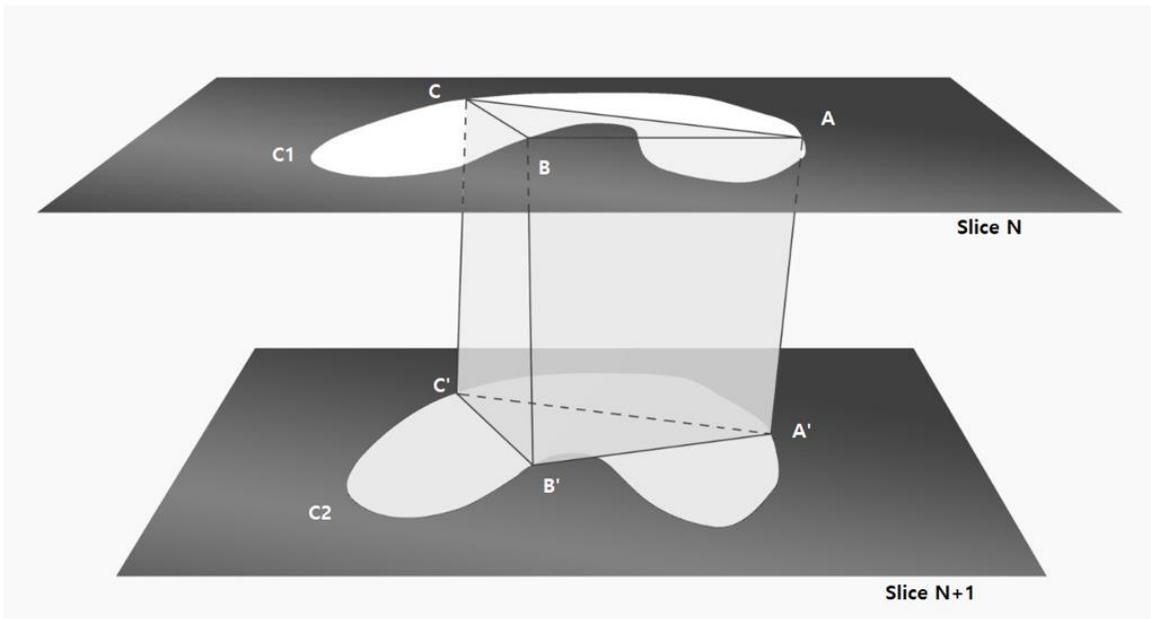


Fig. 5. Base Mesh Construction for Two Contours of Two Consecutive Slices.

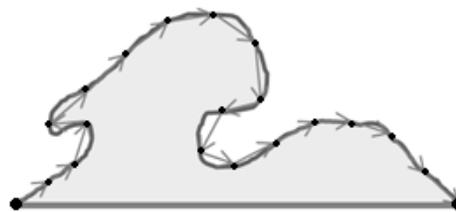


Fig. 6. Extraction of Contours with Displacement Vectors.

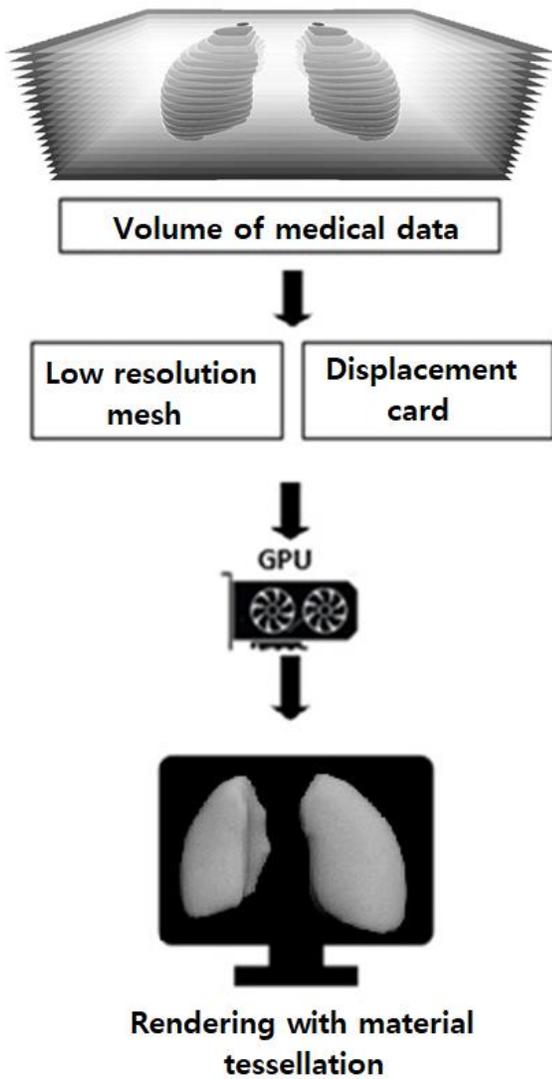


Fig. 7. Rendering with Hardware Tessellation to Generate the High-resolution Mesh.

Indeed, the key role of these constructed displacement vectors is that they can automatically generate the displacement map representing an image determined by its moving distances and directions from the points on the contour surface during the real-time display.

E. Reconstruction of the 4D Mesh

- Creation of sub-meshes and displacement vectors

Here, we are now able to characterize the 4D construction of the mesh. At the beginning of such construction, we start then from an already defined set of n sequences of slices from an initial time T_0 to a final time T_{n-1} . For each sequence we have m slices from C_0 to C_{m-1} . These slices are already segmented to extract only the object to be reconstructed (see Fig. 8).

This set of slices will be divided into $m-2$ sub-assemblies, each sub-assembly is represented by 2 consecutive slices of the n sequences. In particular, the first sub-assembly 0 will be represented by slice 0 and slice 1 of all the sequences.

Next, each subset must undergo some additional treatments as described in what follows:

For each sequence i we need to determine the different correspondences existing between the contours of 2 consecutive slices [16]. Contours that do not have a correspondent in the other slice will be neglected.

In Fig. 9, we have a correspondence between contour 1 and 2 of slice 1 and contour 1 and 2 of slice 2 respectively; contour 3 of slice 1 does not correspond to any contour in slice 2, so it will be deleted (see Fig. 10). Each correspondence gives us a base sub-mesh and displacement vectors that will be added to this mesh at time T_i .

The next step is to find for each sub-mesh its correspondent in the following defined sequences, for which we say that the correspondence of a sub-mesh of the sequence i to the sub-mesh of the sequence $i+1$ is satisfied if the contours of these two sub-meshes correspond to each other at each slice and with the same number (see Fig. 11).

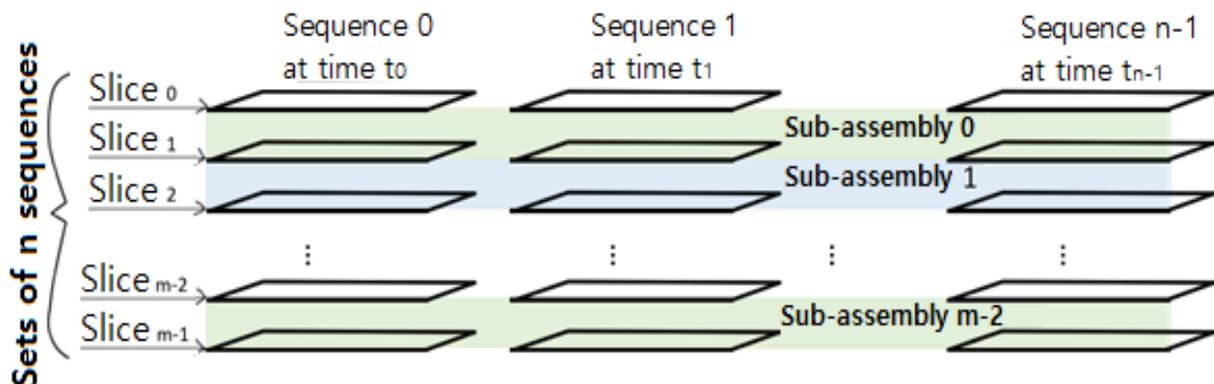


Fig. 8. The Input to this Algorithm is in the Form of a Sequence of 2D Slices for each Instant.

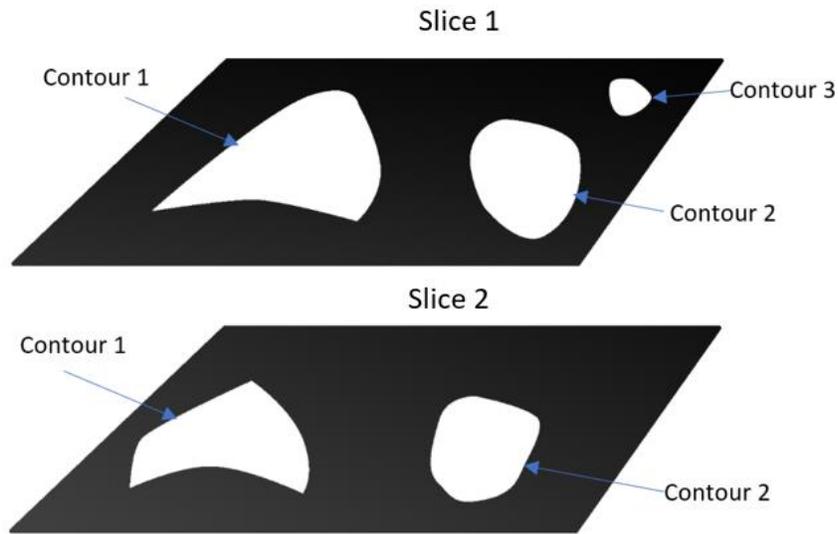


Fig. 9. Correspondence between the Contours of a Sequence.

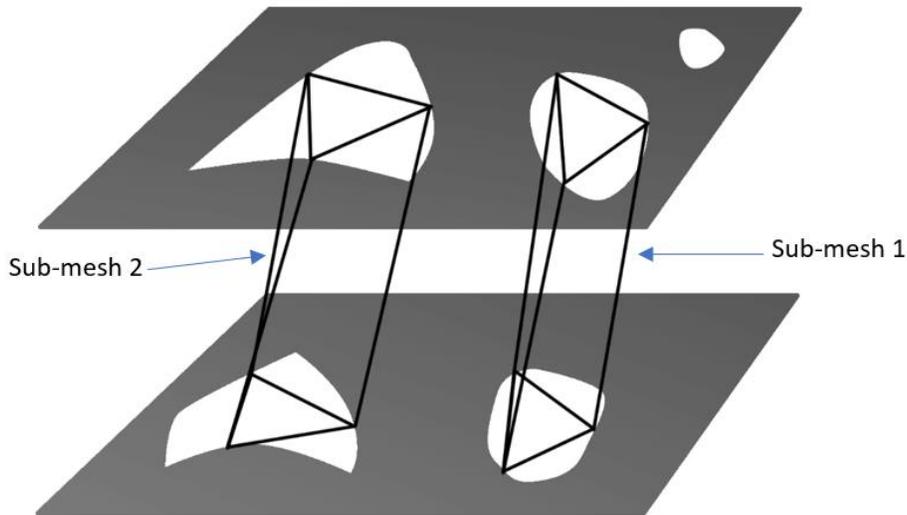


Fig. 10. Representation of the Sub-meshes which Represent the Correspondences between the Contours of the same Sequence.

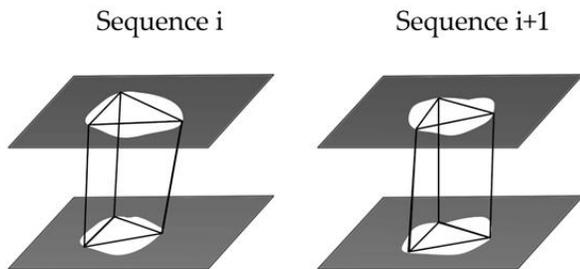


Fig. 11. The Sub-mesh of Sequence i Corresponds to the Sub-mesh of Sequence $i+1$.

If we fail to find the correspondence of a sub-mesh of a sequence i in the following sequence, the appearance of a new sub-mesh is obligatory, and it will be from the instant T_{i+1} (see Fig. 12).

Note that we can also have a new sub-mesh in a sequence i if we fail to find a correspondence between its contours and the contours of the sequence of the instant T_{i-1} (see Fig. 13).

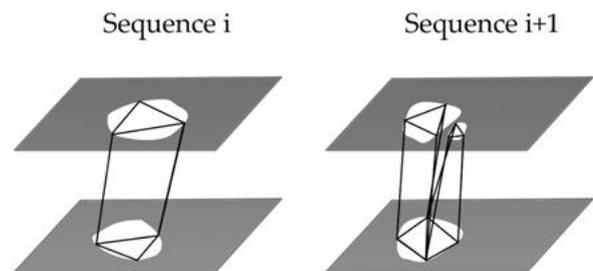


Fig. 12. No Correspondence between These Two Sub-meshes (the Number of Contours in Sequence i is different from the Number of Contours in Sequence $i+1$).

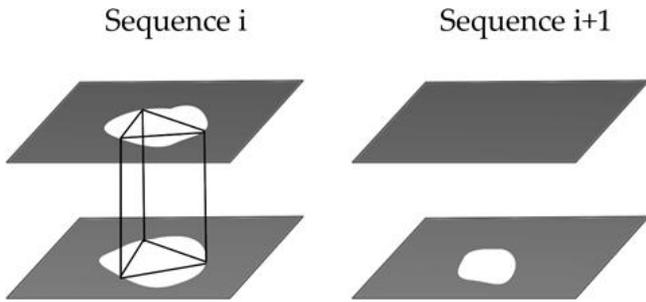


Fig. 13. This Sub-mesh does not have a Correspondent in the following Sequence.

Finally, after applying this additional treatment to all the subassemblies, we obtained series of sub-meshes, each sub-mesh has the following information:

- Start sequence number (S_d).
- End sequence number (S_f).
- Displacement vectors for each edge in the different sequences from the first sequence to the final sequence.

Note that each sequence has its own displacement vectors that refine the corresponding base sub-mesh. Another fact shown the above table in Fig. 14, that sequences from 0 to i , sequences from $i+1$ to j , sequences from $j+1$ to k , as well as sequences from $k+1$ to $n-1$ do not share the same basic sub-mesh in a sub-assembly.

- Construction of the 4D mesh and interpolation

Following the previous development for the obtained results from N given sequences, we have by now all the necessary ingredients to form 3D meshes for all sequences in the form of a single compound base mesh and N displacement maps. Thus, the next treatment is the visualization of the dynamic 3D mesh using all the available data and the gathered

information. However, the success of this crucial procedure results from the number of given sequences and the required quality of the visualization in terms of display smoothness. In fact, if the number of sequences is not sufficient or the required quality is very high; an extra interpolation is then obligated to perform for additional renderings to maintain the desired display quality between each consecutive sequences. Especially, the interpolation will be applied to the displacement vectors. In the normal case, where there is a correspondence between the meshes of sequence i and those of sequence $i+1$, the position of these interpolated meshes between the two sequences is deduced by creating intermediate displacement maps based on the already computed displacement maps in sequence i and sequence $i+1$. In the opposite case of no existing correspondence between the meshes associated to consecutive sequences (see Fig. 15); then, the calculation of the intermediate mesh can be carried out after finding the suitable correspondence between the contours.

In this case, for example, where there is no correspondence between the mesh of sequence i and the mesh of sequence $i+1$, we must create a new sub-mesh at time i .

This sub-mesh will have the sequence i as the starting sequence. Thus, its elements must correspond to the elements of the sub-mesh of sequence $i+1$. To do so, we will use the correspondence factor between the contours of slice j of time i and the contours of slice j of time $i+1$ [16], to create a correspondence between the contours of the same slice, the correspondence direction, and the correspondence percentage.

According to the calculated correspondence factor of the contour C_1 with the contours C_2 and C_3 of the sequence $i+1$, we find that the green part of the contour C_1 corresponds to the contour C_3 (see Fig. 16), and the orange part corresponds to the contour C_2 . Thus, we can build the sub-mesh below which will be the sub-mesh used from the instant $i+1$:

	Seq. 0	Seq. 1	...	Seq. i	Seq. i+1	...	Seq. j	Seq. j+1	...	Seq. k	Seq. k+1	...	Seq. n-1
S. -M. ₁	Start			end									
S. -M. ₂					start		end						
S. -M. ₃								start		end			
S. -M. ₄											Start		End

These sequences use the same basic sub-mesh for a sub-assembly i

Fig. 14. This Table shows that each Sub-mesh has a Start and an end Number.

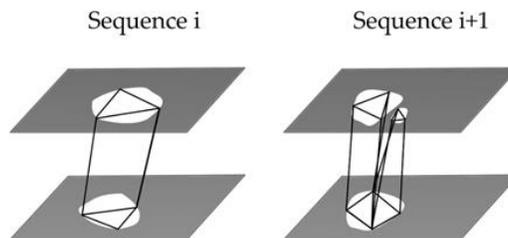


Fig. 15. No Correspondence between the Meshes of Two Consecutive Sequences.

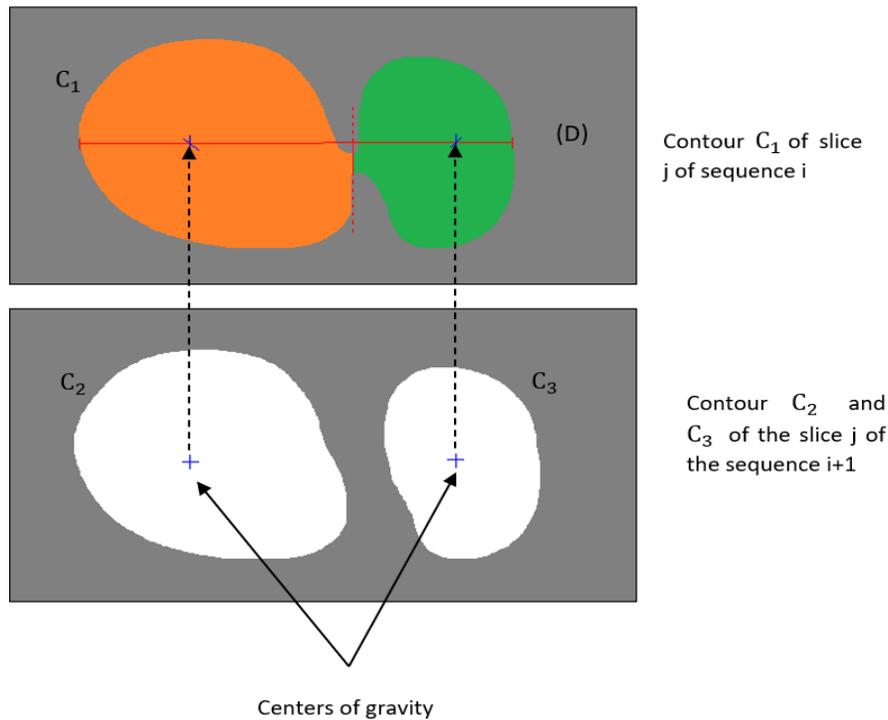


Fig. 16. Create a Correspondence between the Contour C_1 of the Cut j of the Sequence i and the Contours C_2 and C_3 of the Same Cut of the Sequence $i+1$.

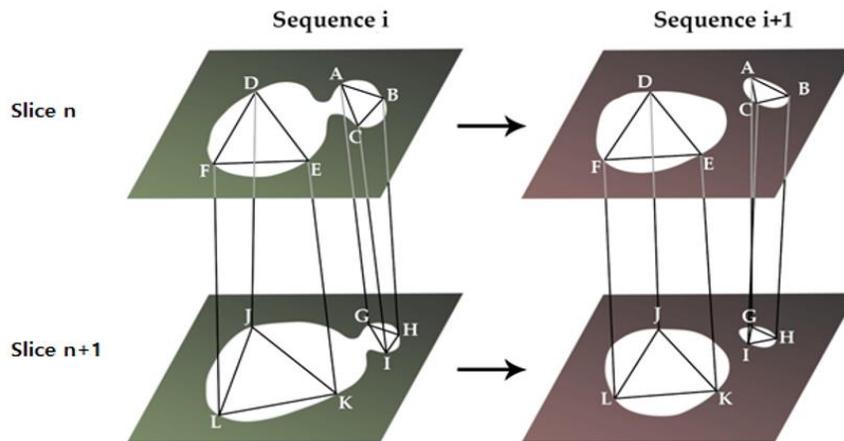


Fig. 17. Construction of the Sub-mesh which will be the Sub-mesh used from Time $i+1$.

After this operation, we can easily see that we have obtained a complete mesh associated to all the sequences (see Fig. 17). Based on such computed mesh, we can determine for each vertex at time T_i its corresponding in the next time T_{i+1} . As it can be deduced, this procedure allows the possibility to interpolate the meshes between any consecutive times.

IV. RESULTS AND DISCUSSION

The implementation of the proposed method of 4D mesh reconstruction from 2D medical images is based on the following steps:

- Detection and extraction of contours from a 2D medical image sequence.

- Construction of the basic 3D mesh of the T_0 sequence and of all the sequences that do not have some correspondence with the previous sequences.
- Extraction of displacement vectors for any sequence from the 2D images based on the corresponding 3D mesh.
- Interpolation of the 3D mesh between sequences based on the interpolation of the displacement vectors to have a smooth visualization.
- Construction of the high resolution 4D mesh from the obtained base mesh and the displacement vectors.

The first numerical result concerns the testing of our 4D reconstruction method in terms of the quality of the obtained 4D mesh. Effectively, the construction of the 4D mesh used by some classical methods is often based on the Marching Cubes algorithm combined with the interpolation method. It is worth mentioning that this approach causes ambiguity cases in some configurations [19]. In other negative situation, one may be confronted with complex areas causing an unexpected deformation during the mesh reconstruction [19]. In contrast, our new method leads to a low-resolution mesh based on contour extraction that makes it possible to create a 3D mesh without any ambiguity, matching exactly the actual shape of the anatomy. It is also noticed that the computed 4D mesh incorporate easily all the gathered anatomy information. Indeed, this shows that our method can overcome the staircase problem obtained with the classical method (see Fig. 18) (Marching Cubes with very small elementary cubes).

The second testified numerical result is related to the low storage level. For this purpose, we have compared the amount of information stored using the new adopted structure with the structure used in the conventional storage method (see Table I). As a result, by simplifying the calculations, we have applied both methods to a medical data volume of 512 x 512 x 38. This gives a size of the high-resolution mesh with the conventional method of about 2.48MB, which must be

multiplied by m (the number of given sequences). This represents a volume of about 50MB, while if we store the base mesh with the displacement vectors, we reach only 150KB. This is then a good storage reduction that is very important in practice.

The last important numerical result achieved by our algorithm is witnessed by the rendering speed. In the related numerical test, we have considered a set of 10 medical image sequences (T_0 to T_9), and in each sequence we have 38 slices. As a result, the table below shows the necessary rendering execution time for both methods (see Table II).

It is noticed that in the classical method we sent the high-resolution mesh of each time T_i directly to the GPU, while in our method we only sent to the GPU a set of basic sub-meshes with the displacement maps at each time T_n .

In another additional test, the same display is considered in both methods, by disabling in our method the interpolation between T_n time periods. The obtained comparison shows that the rendering time with the classical method is about 120 MS (8 frames per second) which is very slow about our method that achieved a rendering time equal to 19 MS only (52 frames per second). Roughly speaking, the obtained enhancement in speeding up the rendering is at least 6 times more.

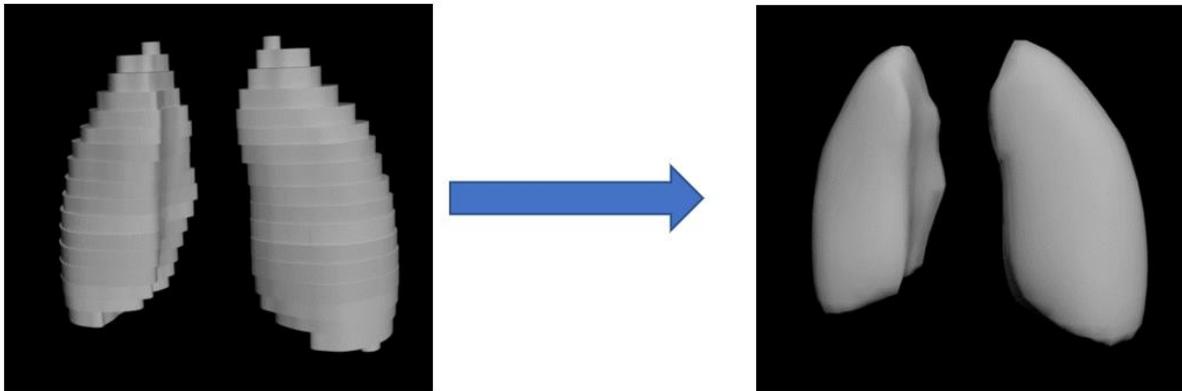


Fig. 18. Our Method Allows to Overcome the Staircase Problem Obtained with the Classical Method.

TABLE I. COMPARISON BETWEEN THE AMOUNT OF INFORMATION STORED USING THE NEW STRUCTURE AND THE CONVENTIONAL STORAGE METHOD

	Basic mesh (bytes)	Number of sequences	Displacement vectors	Total (bytes):
The original method (MC)	2,480,892	20	0	=49,617,840
The new method	126,148 (T_0) 1,009 (T_5) 1,129 (T_{12}) 1,276 (T_{16})	The other sequences are based on these 4 basics meshes already sent	500*20 vectors *2 char (x,y) *1 octet	=149,562

TABLE II. COMPARISON BETWEEN THE RENDERING SPEED USING THE NEW METHOD AND THE CONVENTIONAL METHOD

	Data sent to the GPU	Size of the sent data	Average rendering time	Number of images displayed per second
The original method (MC)	High resolution meshes 10 of size 50 MB	50 Mo	120 ms	8,33
The new method	Basic sub-mesh set and 10 displacement cards	150 Ko	19 ms	52,63

V. CONCLUSION

In this paper, we have provided an enhanced algorithm that allows the reconstruction of a 4D mesh from a set of medical image sequences, using a base mesh and displacement vectors for each sequence. The use of displacement vectors to add extra details to the basic mesh has been shown to reduce the amount of information stored in the final 4D mesh. Another established fact is that the use of the tessellation unit of the graphics card allows speeding up the rendering time; since, the proposed method create the 4D mesh, based on the 3D reconstruction method that uses the GPU tessellation unit in the rendering. This modification aims to eliminate the cases of ambiguity, especially, in certain types of objects that cannot be treated easily with the reported conventional Marching Cubes method. This is also helpful for obtaining a low-resolution mesh that can be directly used in the rendering without going through a correction step.

Another particularity of the proposed method is that one can extract the contour of the anatomy to be reconstructed from the 2D images, and then build a basic mesh within a low computational time and with a reduced storage memory. By doing so, this basic mesh construction step is done for each sequence that does not have the same contours as the previous sequence, which leads to a very small number of basic meshes. Moreover, we generate displacement vectors for each sequence by discretizing the contours according to the desired level of details.

As a result, our framework has been proven to be efficiently suitable and adaptable for the following main technical issues:

- The construction of a low-resolution base mesh using the contour extraction and matching method overcomes the staircase problem as well as the ambiguity problems and generates a mesh that exactly matches the real anatomy.
- The computation of basic meshes and displacement vectors has a very low storage burden.
- The transmission of the information to the GPU allows accelerating the rendering time.

Given the important advantages of this 4D reconstruction method, it will be very useful to apply it to the vital organ that is the heart. Thus, we intend to do detailed research and analysis to treat all its particularities.

REFERENCES

- [1] P. J. Barnes, J. Baker and L. E. Donnelly, "Cellular senescence as a mechanism and target in chronic lung diseases," *Am. J. Respir. Crit. Care Med*, vol. 200, no. 5, p. 556–564, 2019.
- [2] E. G. M. Kanaga, J. Anitha and D. S. Juliet, "4D medical image analysis: a systematic study on applications, challenges, and future research directions," *Advanced Machine Vision Paradigms for Medical Image Analysis*, pp. 97-130, 2021.
- [3] S. Zhang, G. Geng and J. Zhao, "Fast parallel image reconstruction for cone-beam FDK algorithm," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 10, p. e4697, 2019.
- [4] G. McKinnon and R. Bates, "Towards imaging the beating heart usefully with a conventional CT scanner," *IEEE Trans Biomed Eng*, vol. 28, pp. 123-127, 1981.
- [5] C. Mory, *Cardiac c-arm computed tomography*, Université Claude Bernard Lyon, 2014.
- [6] S. Capostagno, A. Sisniega, J. W. Stayman, T. Ehtiati, C. R. Weiss and J. H. Siewerdsen, "Deformable motion compensation for interventional cone-beam CT," *Physics in Medicine & Biology*, vol. 66, no. 5, p. 055010, 2021.
- [7] S. Rit, J. W. Wolthaus, M. van Herk and J. J. Sonke, "On-the-fly motion-compensated cone-beam CT using an a priori model of the respiratory motion," *Medical physics*, vol. 36, no. 6Part1, pp. 2283-2296, 2009.
- [8] S. Oh and S. Kim, "Deformable image registration in radiation therapy," *Radiation oncology journal*, vol. 35, no. 2, p. 101, 2017.
- [9] G. Haskins, U. Kruger and P. Yan, "Deep learning in medical image registration: a survey," *Machine Vision and Applications*, vol. 31, no. 1, pp. 1-18, 2020.
- [10] M. J. Riblett, G. E. Christensen, E. Weiss and G. D. Hugo, "Data-driven respiratory motion compensation for four-dimensional cone-beam computed tomography (4D-CBCT) using groupwise deformable registration," *Medical physics*, vol. 45, no. 10, pp. 4471-4482, 2018.
- [11] O. Laligant and F. Truchetet, "A nonlinear derivative scheme applied to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 242-257, 2008.
- [12] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, November 1986.
- [13] S. Yi, D. Labate, G. R. Easley and H. Krim, "A shearlet approach to edge analysis and detection," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 929-941, 2009.
- [14] W. Gao, D. Fu, Y. Li and Y. Song, "An Image Edge Detection Method Based on Wavelet Transform Modulus Maximum," *Acta Microscopica*, vol. 28, no. 6, 2019.
- [15] J. Mairal, M. Leordeanu, F. Bach, M. Hebert and J. Ponce, "Discriminative sparse image models for class specific edge detection and image interpretation," in *European conference on Computer Vision*, Berlin, Heidelberg, 2008.
- [16] L. Miara, S. B. El Mdeghri and M. O. C. Malki, "Enhanced algorithm for reconstruction of three-Dimensional mesh from medical images using tessellation of recent graphics cards," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, 2020.
- [17] D. Meyers, S. Skinner and A. K. Sloan, "Surfaces from contours," *ACM Transactions on Graphics*, vol. 11, no. 3, pp. 228-258, 1992.
- [18] B. E. M. Said, C. M. M. Ouçamah and K. & Abdelhak, "Improvement of the generation of displacement vectors in the reconstruction of a 3D mesh for medical images," *International Journal of Medical Engineering and Informatics*, vol. 9, no. 1, pp. 20-29, 2017.
- [19] S. B. E. Mdeghri, M. M. O. Cherkaoui, A. Kaddari and A. Elloub, "Reconstruction of a 3D mesh with displacement vectors for medical images," *International Journal of Medical Engineering and Informatics*, vol. 7, no. 3, pp. 209-221, 2015.