DIRECT HAPTIC RENDERING FOR LARGE DATA SETS WITH HIGH GRADIENTS

Bob Ménélas Nicolas Fauvet Mehdi Ammi Patrick Bourdot
CNRS-LIMSI, University of Paris-Sud XI
BP 133, 91403 Orsay France
{bob.menelas, nicolas.fauvet, mehdi.ammi, patrick.bourdot}@limsi.fr

ABSTRACT
This paper addresses haptic rendering for large data sets resulting from CFD (Computational Fluid Dynamics) applications. We present a new haptic interaction technique for volume rendering of isosurfaces. This method is based on an existing process [1]. This contribution makes it possible to take into account 3D regions presenting high frequency data. In addition, with this technique we can explore and understand complex data fields without having any intermediate geometrical representation (such as the polyhedral mesh provided by Marching Cubes [14]), thus having a very fast haptic rendering loop. Lastly, we led several psychophysical studies to assess the interest of this method for isosurface haptization.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Haptic I/O; I.3.7 [Computer Graphics]: Virtual Reality; I.3.6 [Computer Graphics]: Interaction Techniques

General Terms
Algorithms, Performance, Design, Reliability, Experimentation

Keywords
Computational Fluid Dynamics (CFD), Haptic Volumic Rendering, High Spatial Frequency, Isosurface, Large Data set

1. INTRODUCTION
Due to recent advances in computing, computational models tend to supplement the laboratory experiments by simulating very complex systems. They give access, on one hand, to fields which were inaccessible in the laboratory (e.g. evolutions of the climate), and on the other hand to certain situations not easily under controlled in real life (e.g. temperatures close to the absolute zero). Comprehension of the studied phenomenon implies a meticulous analysis of the simulation results represented by a multidimensional matrix.

Various exploration methods were developed in order to provide a set of intuitive tools for exploring and understanding the data set. In CFD, various desktop applications are used to analyze data generated by numerical simulations. These solutions are centered on the visual rendering [6]. Virtual Reality (VR) immersive visualization approaches improve these solutions by adding a stereoscopic feedback in large scale display environments. Physicists can naturally observe the CFD data by navigating intuitively in the immersive environment. They can observe all the details of the scene by just approaching elements involved on the isosurface.

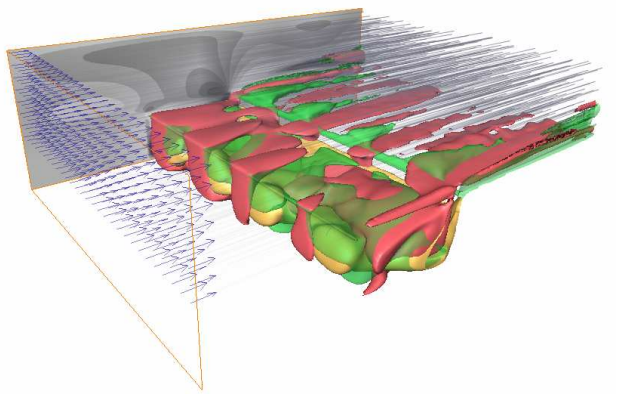

Figure 1. Visual representation of a CFD simulation

However, these methods remain limited. They do not exploit all the potential of human perception and interaction. This limits their effectiveness. In such approaches, based only on visual feedback, too much data are presented to the user through this sole channel. This leads to an overflow of this modality and may confuse the user [19]. Fig. 1 illustrates this point by presenting at the same time three isosurfaces, some streamlines and a 2D slice.

It is known that humans take advantage of all sensory channels to investigate and interpret their environment. For that reason we propose a multimodal VR approach which reinforces and supplements the virtual feedback by adding the haptic rendering for the data analysis. In medical data (computer tomography and magnetic resonance imaging) exploration, haptic feedbacks can enhance the feeling of presence and improve comprehension for a quicker task execution and a more accurate data analysis [5] [17] [27]. In addition, haptic feedback is very interesting for the

perception of local information such as curvatures or textures of objects.

Currently, there are various methods for haptic rendering of data sets [6]. However, these dedicated techniques remain not easily applicable to simulations such as CFD applications which display important data variability. Our work presents a new haptic volume rendering method. This method enables the exploration of data sets presenting regions with high gradients without any gestural constraint as regards specific directions that some algorithms usually imposed. This new method combines a haptic volume rendering method [1] with a collision detection technique [23].

This paper is structured in the following way. The next section describes our data. Section 3 summarizes a state of the art on haptic rendering of data sets and presents the problems related to our CFD data. Section 4 details this new approach. Section 5 is devoted to the evaluation of this technique. In section 6 we conclude the paper and introduce our future work.

2. DATA SET PRESENTATION

In order to justify our approach we need to describe the specific data set we are working with. These data result from a wind-tunnel simulation in a cavity. This simulation is initially composed of 3D rectilinear grids of velocities of size of 259x127x128 units. Based on these velocities, we compute several useful fields, such as vorticity. This one is the measure of the rotational speed around each axis. It plays an important part in the study of vortices in the cavity. The same goes for the *Q Criterion* field, which is another variable to approach vortices : from the vorticity, is extracted the shear component, so as to retain only the rotational part of vorticity. An isosurface of *Q Criterion* highlights specific tubular structures in the flow which are closely studied by physicists. The last velocity-based field we computed is the *Helicity* field. This consists in projecting the rotational speed on the velocity vector. It represents helices along the direction of the flow. The sign of this field depends on the rotation direction.

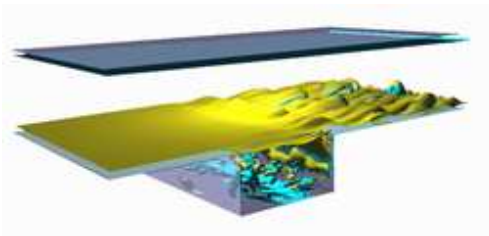

Figure 2. The cavity with a transversal vorticity isosurface.

The specificity of the studied data set is its non-regular sampling. Physicists don't need the same simulation precision in the entire cavity. Fig. 2 presents the cavity shape integrating an isosurface.

Over this data set, we chose to build a space partitioning tree to speed up computations. Since we are rendering a rectilinear 3D grid, an axis-aligned space partitioning scheme seems to perfectly fit the needs. The volumetric nature of the grid leads us toward an octree approach for spatial partitioning. This technique was adapted to the specific non-regular sampling (Fig. 3).

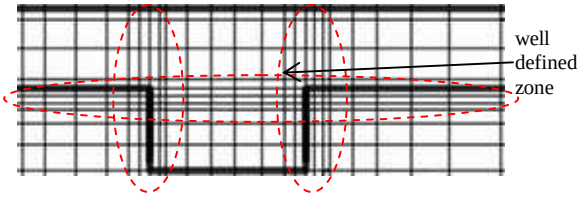

Figure 3. Grid partitioning

We recursively divide the volume space into eight axis-aligned sub spaces, until the point where we can only subdivide the volume over one or two dimensions, creating four or even two children instead of the classical eight ones we have on the upper levels of the tree. These cases arise when one dimension is smaller than the other on the full grid, even when the sizes are power-of-two. Every cell of this tree stores minimum and maximum values of the field existing in its sub-hierarchy.

The specific hierarchical structure built on top of our data is the root of many optimizations, in both visual and haptic rendering of the data set. We can quickly access a particular region of the data, and determine if an isosurface is going to go through a specific cell or not, thus enhancing the isosurface extraction and surface proximity for collision detection.

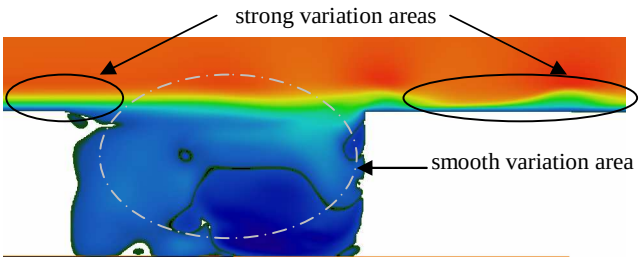

Figure 4. 2D slice representing values repartition

Lastly, we emphasize the high frequencies of data generated by the CFD simulation. On figure 4, we can appreciate a 2D slice representing the values distribution of one component of the velocity vector. On this slice, maximum values are represented in red while minimum values are blue. In the middle of the cavity, we notice a gradual variation of the color. In this zone we have a smooth variation of the data, while at both ends of the cavity, the abrupt passage of the red to blue evidences about strong variations of data present in these regions. Actually, these last zones contain high gradients.

In the next section we present a brief state of art about the haptization methods used in large data sets exploration.

3. RELATED WORK

Traditionally, haptic rendering is used, in VR applications, to simulate realistic contacts with virtual objects. It comprises generally two engines: a collision detection engine to avoid objects interpenetration and a dynamic engine to compute the force resulting from the interaction.

Within the framework of large data sets exploration, the aim is to transfer to the user a force in order to inform him about the local volumetric data. Indeed, the haptic feedback has the potential to improve the understanding of data sets [8] [27]. In this direction, various data type fields (scalars, vectors and torques) were

explored and analyzed using our sense of touch. In the following subsection, we present the related work in this domain. We classify these techniques in two main groups, surface and volume rendering techniques [11].

3.1 Haptic surface rendering

In surface rendering approaches, the haptic feedback is used to simulate the response of touching a virtual surface. The operator uses the epistemic function to touch a surface like in real world.

Several techniques were developed in this class of methods. The first approach consists to extract an explicit representation of the global surface. The *Marching Cube algorithm* [14] is the most common algorithm for the computation of this geometrical envelope. Once the surface is generated, the haptic feedback is calculated through a classic collision detection module coupled with penalty based method [17] [18]. The main limitation of this method is the relation between computation time and data volume, this limits its application to non complex data.

The local intermediate representation presents a solution to this constraint. This method used a local runtime surface estimation that required less memory than global representation. In [12] the voxel data in a neighborhood of the probe position is used to generate isosurfaces through points with similar density values. With this method, real time updating is possible because the surface is generated on the fly, what reduces substantially preprocessing time and memory used.

Chen et al. [4] proposed with the same approach a haptic rendering method for isosurfaces without an explicit isosurface extraction. In this algorithm, a virtual plane is used as an intermediate representation of the isosurface to compute the interaction force applied to the haptic interface.

In [9] Itkis et al. used a directional constraint to provide intuitive exploration modes for volumetric data sets; this algorithm restricts the user motion in specified directions. For example, to guide the user in a vector field data, the proxy can be constrained along a streamline. In such model, any effort to move the haptic interface in a direction perpendicular to the current orientation of the field results in a strong opposing force.

To produce natural haptic feedback with solid contents (Computed Tomography data), Lundin et al. [16] constrain perpendicularly a proxy according to the gradient vector. Furthermore, the displacement of the proxy (virtual effector) relative to the probe (real effector) is used to simulate the surface contact. Thanks to this method, material properties like friction, stiffness and surface penetrability can thus simulate interactions with soft and hard tissue.

Finally, in this class of approaches, the haptic representation can be generated with implicitly defined surfaces, like with volumetric description, as suggested in [11] [24], or with NURBS (Non-Uniform Rational B-Spline) as described in [26]. Thus, on the basis of this representation, Salisbury et al. [24] exploit the tangent plane to the surface at the contact point to compute the haptic feedback resulting from the interaction to a surface defined by equation $S(p)=0$.

3.2 Haptic volume rendering

In volume rendering approaches, there is a direct relation between the data and the computed force. Here, the haptic feedback is estimated through a vector-valued function (transfer function) of the volumetric data, computed at the end effector position (x_{probe}). It is also possible to consider the end effector velocity (v_{probe}), in the transfer function, to calculate the final force (Eq. 1.).

$$\vec{F}_{fb} = F(\vec{x}_{probe}, \vec{v}_{probe}) \qquad (1)$$

The mapping of the field value on a viscosity as illustrated in Eq. 2., provides a simple way to correlate the field value on the end effector velocity (*viscosity mapping*).

$$\vec{F}_{fb} = -T(\varphi(\vec{x}_{probe}).\vec{v}_{probe} \qquad (2)$$

$\varphi(x_{probe})$ represents the field value at end effector position. With this transfer function the field value is translated as a viscosity to the user. The generated feedback is thus proportional to the exploration speed. In [3] Bartz et al. translate a pre-computed distance field to the user according to this metaphor.

The *viscosity mapping* is suitable to inform about the value distribution in the volume but is not relevant for the analysis of a specific point. This limitation was highlighted by Aviles and Ranta in [2]. Since the force feedback is directly proportional to the probe velocity (v_{probe}.), the low speed (x_{probe}~0) required by point analyze will produce a force feedback close to zero (F_{fb}~0).

The gradient scalar is also used in some works in order to inform about the relative distribution of the scalars in the volume.

$$\vec{F}_{fb} = T(\varphi(\vec{x}_{probe}).\vec{\nabla}\varphi(\vec{x}_{probe}) \qquad (3)$$

During the year 1993, Iwata et al. in [10] suggested a technique for the understanding of a scalar field variation. During the field exploration, a constant direction force proportional to the field value at the probe position is transmitted to the user. With this metaphor the haptic interface is attracted or repulsed (according to the sign of the transfer function) by regions having high scalars value. The generated feedback is very suitable for volumes having low frequency data, producing a soft push towards regions of interest. However, if the magnitude of the attractive force is too high, or in the case of high frequency data, unstable behavior can occur in the form of vibrations.

While projecting the end effector velocity onto the gradient vector, Pao et al. in [22] produce a viscosity feedback only in the direction of the gradient

$$\vec{\mathrm{F}}_{\mathrm{fb}} = -T(\phi(\vec{x}_{\mathrm{probe}}).\vec{v}_{\mathrm{probe}} \frac{\vec{\nabla}\phi(\vec{x}_{\mathrm{probe}})}{\left|\vec{\nabla}\phi(\vec{x}_{\mathrm{probe}})\right|} \qquad (4)$$

The viscosity feedback generated with this function (4) is linked to the scalar distribution in the volume. Due to the viscosity component which absorbs the haptic interaction energy, this technique provides a better stability than the gradient model proposed in Eq. 3. However, the same problems evoked in (Eq. 2) still remained since the feedback depends on the velocity of the users movements. In ([13] [21] [22] they illustrated this model with various data type fields The evaluations of Reimersdahl et al. in [25] let to compare various Pao‘s methods. This work concludes that within a scalar field framework, the *Gravity Scalars* method is best suitable to identify a chosen value, while the *Viscosity Scalar* method is best to identify zones containing a range of values.

Lastly, let us quote Avila's work [1]. With this method, it is possible to haptically examine all the data volume or a part of it like an isosurface. The force resulting from the volume exploration is perceived like a retarding force and stiffness proportional to the field intensity. The force is expressed in the following way:

$$\vec{F} = \vec{S} + \vec{R} \tag{5}$$

In this equation S is related to the stiffness and is consequently directed according to the volume gradient at the probe position, while R expresses the environment viscosity opposed to the haptic interface motion (Fig. 5). For isosurface rendering, this method expresses the retarding and the stiffness forces directly from the volume data. The penetration distance is thus expressed by an approximation of the distance to an isosurface computed via field value differences. To render the collision with an isosurface, they consider the upper d_i and the lower d_j values representing the virtual thickness of the surface. Then, the rendered force is computed, only when the value at the probe position is between the values d_i and d_j, as expressed in (6) (7) (8).

$$\vec{S} = f_s(d) \times \vec{N} \tag{6}$$

$$\vec{R} = f_r(d) \times \vec{V} \tag{7}$$

$$\begin{cases} f_s(d) = \begin{cases} C1 \bullet \dfrac{d - d_i}{d_j - d_i} & \text{if } (di<d<dj) \\ 0 \quad else \end{cases} \\ f_r(d) = \begin{cases} C2 \bullet (d - d_i) + C2 & \text{if } (di<d<dj) \\ 0 \quad else \end{cases} \end{cases} \tag{8}$$

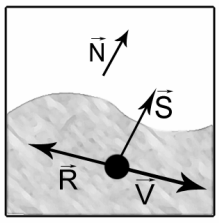

Figure 5. Representation of various components of the force F.

4. CONTRIBUTION

An isosurface is the 3D generalization of an isocontour. This is the geometric representation of all the points having a defined value (*the isosurface threshold*). Isosurfaces are used to explore the internal structures of volumic data in numerous disciplines. In CFD applications, this representation allows engineers to analyze data presenting the same features (pressure, velocity, etc.). Considering the importance of isosurfaces in various scientific and industrial domains, and the fact that haptic is very suitable for local information rendering, our contribution aims at the utilization of the haptic modality to enhance the perception of these objects. Hence, we propose in this paper a new approach for the haptization of isosurface because related works analyzed above are not adapted to the CFD specifications. In addition to the real time constraint (in order to insure interactive surface exploration), the haptic computation must be robust and stable (to not disturb users in their analysis of the data set). Moreover, the haptic interaction should not constrain the user's gesture in a specific direction, (one must be capable of exploring all parts of the explored object). Unfortunately, as evoked previously, the majority of existing methods [1] [13] [21] [22] are not adapted to high frequency data variation (spatial frequency). Furthermore, explicit surface based techniques [14] take considerable computing time; some other techniques are constraint based [5] [15] [9] or dedicated to implicit surface representation [11] [24]. In the next subsection we detail the solution proposed for the haptic rendering of high gradient data.

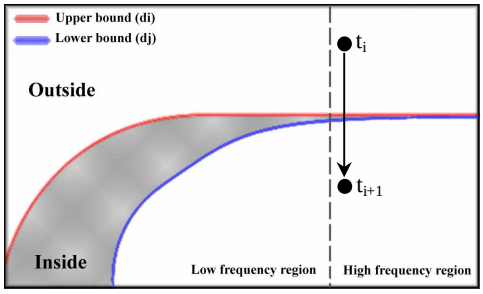

Figure 6. Movement from t_i to t_{i+1}

4.1 Problematic

As already mentioned, the known methods do not present at the same time the stability and the speed required in our framework. For this purpose we directed our work towards the adaptation of the Avila method. Indeed, without the stability problem with isosurfaces having high gradients data, this method presents the best haptic feeling and the fastest computing time [1]. Thus, we concentrated our work on the stability improvement of the Avila method.

The Avila method can be summarized as follows

```
Load Field
Compute Gradients on Field
For-each step of the haptic loop
  Get isovalue at current position : d
  If( di < d < dj)
    Get gradient at current position
    Get speed at current position
    Compute S  and  R
    F= S + R
  Else
    F= 0
End-loop
```

Let us underline that, the Avila method is based on the test carried out on the isovalue at the probe position. Let d denote the isovalue at the probe position; the isosurface limited by the values d_i (upper bound) and d_j (lower bound) is detected if we have:

$$d_i < d < d_j \tag{9}$$

However, we can note that, during the motion between two iterations t_i and t_{i+1}, the crossing from one side to the other of the isosurface (Fig. 6), results in the following condition:

$$\begin{cases} d(t_i) < d_i \\ d(t_{i+1}) > d_j \end{cases} \tag{10}$$

The temporal discretization of the previous algorithm does not take into account this case because it does not handle the time interval between two consecutive time steps, but only focuses on the current iteration. Hence when the upper and the lower bound isosurfaces are too close one to each other (i.e. case of high frequency regions see Fig 6), the Avila algorithm may not detect the intersection. This method is thus not directly applicable to data coming from CFD applications because they usually present regions containing high gradients (high spatial frequency).

4.2 Proposed approach

The proposed approach is to overcome the time discretization by taking into account the time step between t_i and t_{i+1}. Here, we describe a new algorithm which combines the Avila's haptic volume rendering method, with a collision detection technique based on *the proxy method for polyhedral representations* [23]. This new method will enable us to explore high frequency data. It is based on the following metaphor:

Let d denote an isosurface threshold (d_i<d< d_j); during displacements in a volume, we generate a force F each time we cross the isosurface in a direct direction (cross). This force is cancelled, when we cross this isosurface in the opposite direction (uncross)

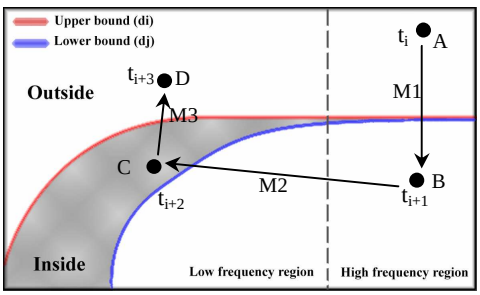

Figure 7. Movements in different regions (high and low frequency) of the data volume

Let us consider the diagram (Fig. 7) representing the displacements of a user in a data volume between the instants t_i and t_{i+3}. From t_i to t_{i+1}, the user's hand travels from A to B. According to the proposed metaphor there will be a not null force F_1 at position B, because at this moment we crossed the isosurfa-ce (*cross*). At position C, there will be a not null F_2 force, because we are still in the lower part of the isosurface.

From t_{i+2} to t_{i+3}, the user's hand travels from *C* to *D*. At this configuration the force F_3 is null (equal to 0) because we crossed the surface in the opposite direction (*uncross*). We have to notice that results would be quite different with traditional volumic rendering methods [1] [10]. Indeed, none of them would translate the crossing of the isosurface during the displacement between *A* and *B*. This lack of robustness is related to tightening thickness of the isosurface in this area containing high spatial frequency data.

The following algorithm summarizes the suggested approach:

```
Load Field
Compute Gradients on Field
For-each step of the haptic loop
 For each point in the path
  get isovalue at current position: d
  old_ isovalue =  new_isovalue
  new_isovalue = d
  If (new_isovalue > d0)
    F= 0
    Cross =0
  Exit
  If (new_isovalue < d0 < old_ isovalue )
    Cross=1
    P1 =Current position
    P0 = starting position
    I = Intersection (P1P0 and Isosurface)
    N gradient at I
  Exit
  If (cross = 1)
    Compute T (project R on isosurface)
    V = intersection (RT and isosurface)
    F=S+R
 End loop
End-Loop
```

We can structure this algorithm into two principal steps. In the first one we check whether we crossed the isosurface or not, and compute the intersection point with the isosurface. The second stage consists in determining the proxy position in order to compute the force resulting from the interaction.

4.2.1 Intersection test

To prevent the missing of intersection points with the isosurface, we will not only check the value at the current position but at all points of our grid defining the movement. Hence for a displacement from *D* (Departure point) to *E* (End point), we have to approximate all points *P* defining the segment *[ED]* and for each point *P* test whether the value at this point: *d(P)* is equal or not to the isovalue threshold:

$$d(P) = d_0 \tag{11}$$

To approximate all these points, we may use the Bresenham algorithm adapted to the 3D [7]. Since we want to find all the points of the grid, the step increment of the algorithm is the minimum distance between two points of the grid. To know whether an intersection is a "*cross*" or an "*uncross*", we have to test the isovalue at the position just before the intersection point. Let *B* denotes this point; for a "*cross*" case we must have:

$$d(B) > d_0 \tag{12}$$

If this condition is not verified, we consider that we are in the opposite case (13).

$$d(B) < d_0 \quad (13)$$

When an intersection with the surface is detected, we compute the intersection point as follows:

Let *I* denote this intersection point, and *d(i)* the value of the filed at I since *I* intersects the isosurface, we have:

$$d_i < d(I) < d_j \quad (14)$$

At the point *I*, the value *d(I)* is between d_i and d_j. To find this point *I*, we have to explore the displacement segment and find all the points verifying equation (14). Then we look for the closest one to d_j in order to generate the strongest force feedback to the user. At the intersection point *I* we save the gradient *N* which will be used in the next step.

4.2.2 Proxy position computation

In this stage we have to compute the proxy position. The isovalue at this point must be the most suitable information related to the probe position. The proxy position is the point *V* (virtual), where would be the probe if the isosurface was constrained to move on the isosurface (Fig 8). This point *V* corresponds to the projection of probe position *R* (real effector) on the isosurface according to the gradient vector found in the previous step. To find this point *V*, at first we search a point *T* located at the other side of the surface in the following way:

$$T = R + \alpha \cdot N \quad (\alpha > 0) \quad (15)$$

The point V that we are looking for is the intersection of the segment *[RT]* with the isosurface. To compute the intersection of *[RT]* with the isosurface we use the same algorithm defined in [7].

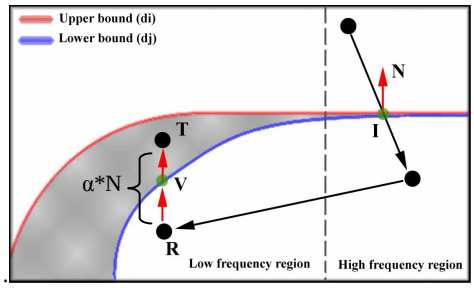

Figure 8. Intersection point computation

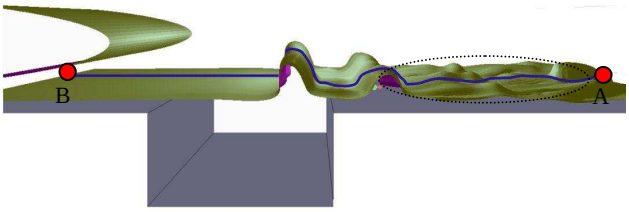

Figure 9. The isosurface presented to users

5. EXPERIMENTATION AND EVALUATION

To evaluate the contribution of the proposed improvement for the CFD data analysis, we carry out a series of psychophysical experiments. These evaluations consist in comparing the Avila method (M1) with the proposed method (M2). In the following subsections we present the experimental protocol used for this evaluation. Thereafter, we discuss the results obtained with these two methods.

5.1 Experimental protocol

For our study, we went through two procedures. In the first one, we evaluate each algorithm's performances, computing the memory space used and the haptic loop frequency. In the second one, we evaluate the accuracy (follow-up's precision) and the haptic rendering quality (users' preference) of each method. Fig. 9 shows the isosurface and the trajectory propose for these experiments.

For the users' preference, we ask people about their feelings for each haptic method tested, about the differences they could feel, the quality of the sensation.

The follow-up precision is obtained as follows: we computed an error percentage over the total time of the experiment. The error

For the users' preference, we ask people about their feelings for each haptic method tested, about the differences they could feel, the quality of the sensation.

The follow-up precision is obtained as follows: we computed an error percentage over the total time of the experiment. The error represents the difference between the value at the user position and the isosurface threshold. Fig. 10 illustrates the way we generally define the error in such a task.

Let *e* be a measure of our error. Along the path function *L*, we can compute the path integral like

$$e = \int_L F(z).dz \quad (16)$$

In this equation, *F* is the volume function, returning field values, and *z* being a 3D position.

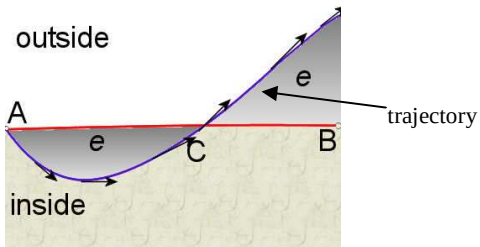

Figure 10. The error "*e*" between the curve *(C)* described by the user and the isosurface

Between the beginning *a* and the end *b* of the path followed during the experiment, the function *L(t)* gives the 3D position of the probe along the curve at *t*, leading to (17)

$$e = \int_a^b F(L(t)).L'(t).dt \qquad (17)$$

Since we want to measure our error relative to a given field value and not only relative to 0, we can input a constant function *G* representing the field value of the isosurface we want to follow. This results in (18)

$$e = \int_a^b \left| F(L(t)) - G(.L(t)) \right| .L'(t)dt \qquad (18)$$

Now, to make a percentage of error out of this, we need to divide that integral by the maximum-error integral. Let the function *H* be a constant function returning the maximum absolute field value relative to the value selected for the isosurface. Final equation for *e* is (19)

$$e = \frac{\int_a^b \left| F(L(t)) - G(.L(t)) \right| .L'(t)dt}{\int_a^b H(L(t)).L'(t).dt} \qquad (19)$$

Turning that equation to a numerical form results in (20), where at each time step we sum the product of the isovalue differences by the path ran, divided by the maximum error area.

$$e = \frac{\sum_{k=1}^{k=n} \left| F(L(t_k)) - G(.L(t_k)) \right| . \left\| L(t_k) - L(t_{k-1}) \right\| dt}{\sum_{k=1}^{k=n} \left| H(L(t_k)) \right| . \left\| L(t_k) - L(t_{k-1}) \right\| dt} \qquad (20)$$

When *e* is equal or close to zero, the error is really small. On the contrary, when *e* is close to one, the user has stayed really far away from the surface he was supposed to follow for the whole duration of the experiment. In addition to that error percentage computation, we recorded the errors over the time to highlight the kind of errors that users did during these experiments. The percentage calculation is a global measure of error, whereas the successive errors give a meaning to that percentage.

The experiments were carried out with five participants (Fig. 13). These users are coming from various professional fields; among them we count two mechanics, two robotics engineers and the last coming from Computer Aided Design (CAD) field. Since we don't want to favour any method, they were randomly divided into two groups G1 and G2. The members of the G1 group began the experimentation with the M1, and finished it with M2 (M1, M2). The second group was evaluated on the opposite configuration (M2, M1).

We lead our experiments in a Virtual Reality immersive context. This system is divided into two parts. The first module is a head tracking stereoscopic display system where as the second ensures the haptic rendering. Our experimental setup is composed of the following hardware and software components:

- Two retro-projected large screens (4m²), each one controlled by mainstream PCs with Xeon™ 2.80GHz CPUs and Nvidia QuadroFX 3000G videocards.
- Immersion CrystalEye stereoscopic shutter glasses synchronised with BARCO projectors,
- An ARTrack non-invasive infrared tracking system for head and 3D Mouse tracking,
- A Haption Virtuose™ 3 DoF haptic interface
- An haptic software module, running on a separate computer having the same configuration as the computers managing the visual immersive feedback, with which it interacts through a simple information exchange UDP protocol,
- A Gigabyte network connection between all the computers of the cluster,
- A Mercury AMIRA Software, computing the graphic rendering, and using the tracking of user head, 3D mouse and haptic position via homemade plug-ins.

Fig 11 illustrates the distribution of the different software modules for these experiments as well as the connections between the many computer units in the immersive room.

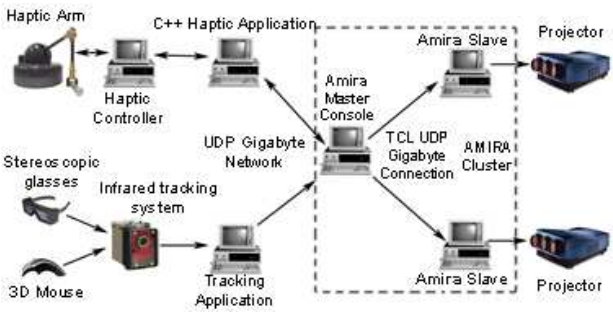

Figure 11. Hardware and software setup of the VR environment for experiments.

5.2 Results and discussions

In this section, we discuss about the results obtained in the two experiments. The algorithms performances as well as the results of the accuracy experimentation are commented in the following section.

The performance evaluations of the two algorithms show that the suggested approach, as well as the basic one offers good performances in term of computing time. The two methods have a very high haptic loop frequency about 1.4 KHz; this one is much higher than the minimal haptic frequency required about 300Hz for kinaesthetic and 1 kHz for tactile [20]. Moreover we can note that the data volume does not really affect the speed of the haptic loop. We can appreciate this in the Fig. 12; the haptic frequency hardly decreases whereas the data volume is at each step multiplied by eight (at each iteration there is twice more data on each axe). Nevertheless, we did not note any particular difference regarding the use of memory resources; this is related to the fact that the octree is the only consequent structure used by the two algorithms.

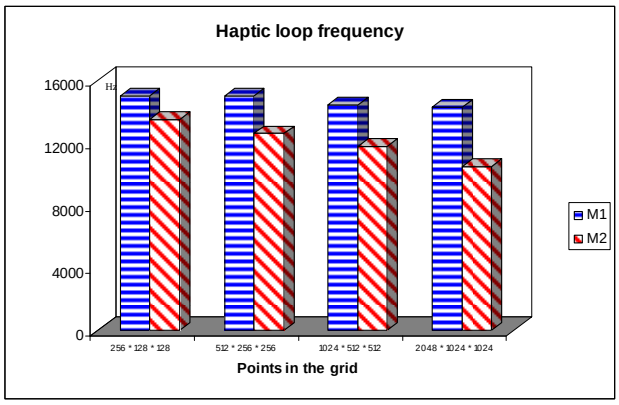

Figure 12. Speed of the algorithms

All the participants of the study carried out their task with the two methods without any specific difficulty; however they preferred M2 method (proposed approach) (Fig 14). They highlighted the fact that they have a better haptic feedback with M2. At the opposite of the M1 method, M2 method makes it possible to the user to perceive all the isosurface details, even the weak undulations (Fig. 9: doted line).

Participants' comments are confirmed by the error computation (Fig. 15). We note that errors oscillate around 0.05 % with the method M2; the bigger one is about 0.15 %. However, in M1 method there are various peaks on the user's route. The minimum error is about 1.18%. These results underline the weakness of this volumic approach (M1) which misses robustness with the high spatial frequency data. In the proposed route (Fig 9), the users have great difficulties to feel the isosurface, the peaks (positive and negative) on the error curve (Fig. 15.) reflects efforts deployed by the users to stay on the surface.

Figure 13. User during the evaluation session

These two experiments carried out, performance evaluation and measurement of the errors, show that our works did not penalize the execution speed of the basic method, but assume significant improvements in term of robustness and haptic rendering feeling, thus allowing a better analysis of data set having strong gradient.

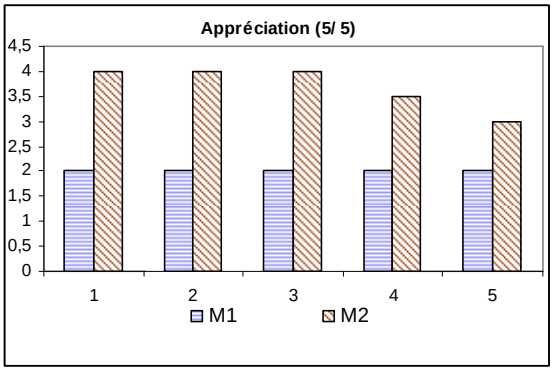

Figure 14. Users' appreciation

6. CONCLUSION

Our work aims at studying "Haptic Rendering" for large data sets such as data resulting from CFD applications. These data are characterized by a high spatial frequency that limits the efficiency of standard techniques. Our work described a new volume haptic rendering method for isosurface, in large data set, without any geometrical or intermediate representation. This method also allows surface exploration without constraints. Our approach uses a virtual proxy to compute the information contained in the isosurface through a local searching process. Our technical and psychophysical experiments confirm the efficiency of this method for CFD data analysis, in term of computation time and perception quality.

The method proposed in this paper utilizes the gradient direction to find the new proxy position. We supposed that may induce errors in specific contexts. We plan to investigate this through some experiments with other data sets.

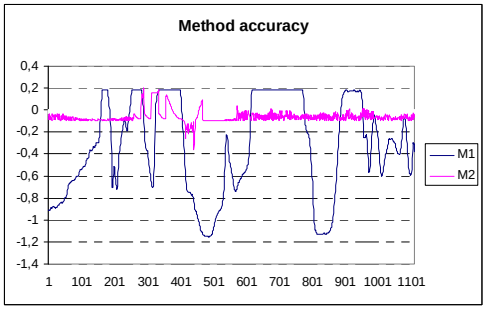

Figure 15. Two methods accuracy

7. REFERENCES

[1] Avila, R. S., and Sobierajski, L. M., 1996. A haptic interaction method for volume visualization, Proceedings of the 7th conference on Visualization '96, , San Francisco, California, United States, 197--204.,

[2] Aviles, W.A., and Ranta, J.F., 1999. Haptic Interaction with Geoscientific Data. In: Fourth PHANToMS User Group (PUG) Meeting. Cambridge, MA.

[3] Bartz, D., and Gürvit, Ö., 2000. Haptic navigation in volumetric data sets. Proc. of PHANToM User Research Symposium, 2000.

[4] Chen, K.-W., Heng, P-A., and Sun. H., 2000. Direct haptic rendering of isosurface by intermediate representation. In ACM Symposium on Virtual Reality Software and Technology VRST 2000.

[5] Durlachn,. I., and Mavora, S. 1995. Virtual reality: Scientific and technological challenges. Washington, DC: National Academy Press

[6] Fauvet, N., Ammi. M., and Bourdot, P. 2007. Experiments of Haptic Perception Techniques for Computional Fluid Dynamics, In proceeding of IEEE. In Cyber world 2007, Hannover Germany, 322--329

[7] Figueiredo, O. and Reveillès, J.-P. 1996. New Results about 3D Digital Lines, Vision Geometry V, Robert A. Melter, Angela Y. Wu, Longin Latecki, Editors, Proc. SPIE 2826, Aug 96,Denver CO, 98--108

[8] Gibson, S., Samosky, J., Mor, A., Fyock, C. and et al. 1997 Simulating arthroscopic knee surgery using volumetric object representations, object representa-tions, real-time volume rendering and haptic feedback. 1997. In First Joint Conference on Computer Vision, Virtual Reality, and Robotics in Medicine and Medical Robotics and Computer Assisted Surgery, 369--378

[9] Ikits, M., Brederson, J. D., Hansen, C. D., and Johnson, C. R. 2003. A constraint-based technique for haptic volume exploration." In Proceedings of IEEE Visualization '03, 263--269

[10] Iwata, H., Noma, H. 1993. "Volume haptization", 1993. Proceedings, IEEE 1993 Symposium on Research Frontiers in Virtual Reality, 16--23.

[11] Kim, L., Kyrikou, A., Sukhatme, G. S., and Desbrun, M. 2002. An implicit-based haptic rendering technique. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems

[12] Körner, O., Schill, M., Wagner, C., Bender, H.-J., and Männer, R., 1999. Haptic volume rendering with an intermediate local representation. In Proceedings of the 1st International Workshop on the Haptic Devices in Medical Applications, 79--84.

[13] Lawrence, D.A., Lee, C. D., Pao, L. Y., and Novoselov, R.Y. 2000. Shock and vortex visualization using a combined visual/Haptic interface, Proceedings of the conference on Visualization '00, Salt Lake City, Utah, United States, 131--137,

[14] Lorensen, W. E., and Cline, H. E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In Maureen C. Stone, editor, Proceedings of Computer Graphics (SIGGRAPH '87 Proceedings), volume 21, 163--169.

[15] Lundin, K., Sillen, K. E., Cooper, M., and Ynnerman, M. D. 2005. A Haptic visualization of computational fluid dynamics data using reactive forces

[16] Lundin, K., Ynnerman, A., and Gudmundsson, B. 2002. Proxy-based haptic feedback from volumetric density data." In Proceedings at Eurohaptic 2002, University of Edinburgh, United Kingdom, 104--109

[17] Mark, W., Randolph, S., Finch, M., Verth, J. V., and Taylor, R. M. 1996, Adding force feedback to graphics systems: Issues and solutions. In SIGGRAPH 96. Conference Proceedings, 447--452

[18] Massie, T.H., and Salisbury, J. K., 1994, The PHANToM haptic interface: A device for probing virtual object. In Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, volume 1, 295--301.

[19] Mousavi, S., Low, R., and Sweller, J. 1995. "Reducing cognitive load by mixing auditory and visual presentation modes". Journal of Educational Psychology 87, volume 2, 319--334.

[20] K.V. Nesbitt, Designing Multi-sensory Displays for Abstract Data, PhD thesis, University of Sidney, 2003

[21] Novoselov, R. Y., Lawrence, D.A., and Pao, L.Y. 2002. Haptic Rendering of Data on Unstructured Tetrahedral Grids. In Proc. IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Orlando, FL, 193--200,

[22] Pao, L.Y., and Lawrence, D.A. 1998. Synergistic Visual/-Haptic Computer Interfaces. Proc. Japan/USA/Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering, Hanoi, Vietnam, 155--162.

[23] Ruspini, D. C., Kolarov, K., and Khatib. O. 1997 The haptic display of complex graphical environments. Computer Graphics, 31(Annual Conference Series), 345--352.

[24] Salisbury, K. and Tarr, C. 1997. Haptic rendering of surfaces defined by implicit functions. In Proceeding of the ASME 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.

[25] Van Reimersdahl, T., Bley, V.F., Kuhlen, T. and Bischof, C. 2003. Haptic Rendering Techniques for the Interactive Exploration of CFD Data sets in Virtual Environments.

[26] Van. Thompson II, T., Johnson, D.E., and Cohen, E. 1997 Direct haptic rendering of sculptured models. In Proceedings of Symposium on Interactive 3D Graphics, Providence.

[27] Wall, S., and Harwin, W. 2000 Quantification of the effects of haptic feedback during a motor skills task in a simulated environment