

# Improving On-Patient Medical Data Visualization in a Markerless Augmented Reality Environment by Volume Clipping

Márcio C. F. Macedo, Antônio L. Apolinário Jr.  
Department of Computer Science  
Federal University of Bahia (UFBA)  
Bahia, Brazil

Email: marciocfmacedo@gmail.com, apolinario@dcc.ufba.br

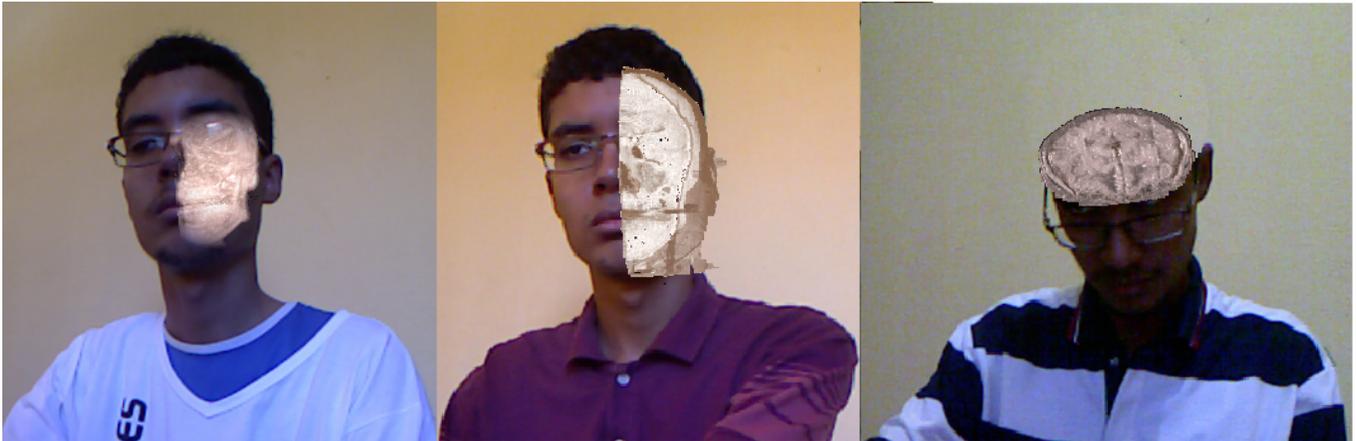


Fig. 1. Some results of the proposed methods: Focus+context visualization based on smooth contours (left) and visible background with Computed Tomography (CT) (middle) and Magnetic Resonance Imaging (MRI) data (right). Each one of these methods takes advantage from the volume clipping to improve the visual perception of the augmented reality scene.

**Abstract**—To improve the human perception of an augmented reality scene, its virtual and real entities can be rendered according to the focus+context visualization. This paradigm is specially important in the field of on-patient medical data visualization, as it provides insight to the physicians about the spatial relations between the patient's anatomy (focus region) and his entire body (context region). However, the current existing methods proposed in this field do not give special treatment to the effect of volume clipping, which can open new ways for physicians to explore and understand the entire scene. In this paper we introduce an on-patient focus+context medical data visualization based on volume clipping. It is proposed in a markerless augmented reality environment. From the estimated camera pose, the volumetric medical data can be displayed to a physician inside the patient's anatomy at the location of the real anatomy. To improve the visual quality of the final scene, three methods based on volume clipping are proposed to allow new focus+context visualizations. Moreover, the whole solution supports occlusion handling. From the evaluation of the proposed techniques, the results obtained highlight that these methods improve the visual quality of the final rendering. Furthermore, the application still runs in real-time.

**Keywords**-Volume Clipping; Focus+Context Visualization; Augmented Reality; Volume Rendering;

## I. INTRODUCTION

Patient's anatomical structures are commonly displayed as 2D images, corresponding to slices of the 3D volumetric data. In this case, the physician must analyze the images and mentally compose what is shown on the screen to the patient. Augmented Reality (AR) can be applied in this context to transfer this task of mental mapping to a computer. Therefore, the physician will be able to visualize, at the same time, the patient and a part of his own anatomy in the display. In general, this kind of application aims to improve surgical planning, training and operation, medical diagnosis and/or post-operative examination.

Augmented Reality is a technology in which the view of a real scene is augmented with additional virtual information. Accurate tracking, or camera pose estimation, realistic rendering of the AR scene and real-time interactivity are the most important technical challenges of AR applications.

Based on the statement mentioned above, the advantage of transferring the on-patient medical data visualization mental task to a computer depends on the quality of the composition between the virtual (i.e. medical data) and real (i.e. patient's image) entities in the AR environment. Instead of superimpos-

ing the virtual medical data onto the patient, a better solution is to show the patient's anatomy as a focus region in the context of the patient's body. This process is known as the focus+context (F+C) visualization paradigm [1] and it has already proven to be efficient in the field of medical data visualization [2], [3], as it improves the visual perception of the scene by the physician. In this context, one way to improve the understanding of the scene is by using volume clipping. Therefore, the effect of volume clipping added in the F+C visualization technique can open new ways for physicians to explore and understand the scene.

In this paper we present a serie of techniques to improve on-patient medical data visualization in a markerless AR environment (MAR) by using volume clipping. Three methods are described to enable F+C visualization based on volume clipping. Moreover, the proposed approach supports occlusion handling.

Although there is not currently a markerless medical AR environment with high accuracy (e.g. [4], [5]), the most recent real-time solutions use off-the-shelf hardware and provide a good composition between the real and virtual entities in the AR environment (e.g. [6], [7]). This way, they can be used to validate our F+C techniques.

*Contributions:* In summary, the contributions of this paper are:

- F+C visualization based on smooth contours: An extension of the F+C visualization technique proposed in [8] to smooth the transition between the image resulting from the volume clipping and the real scene;
- F+C visualization based on visible background: Two methods for F+C visualization based on volume clipping which take advantage from the specific features of Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) data.

The paper is organized as follows. Section 2 provides a review on the related work on medical AR and F+C visualization. Section 3 presents the MAR environment used in this paper. Section 4 introduces the on-patient medical data visualization based on volume clipping. Section 5 discusses the experimental results. Finally, in Section 6, a summary and discussion of future work are presented.

## II. RELATED WORK

The field of medical AR can be conveniently divided into two basic groups: marker-based and markerless.

Over the past decades, many relevant approaches have been proposed for marker-based medical AR, such as [2], [3], [9], [10]. Fiducial markers provide fast and accurate tracking. However, they are still intrusive, because they are not part of the original scene. Furthermore, the hardware of the optical tracking system in some applications is too expensive [2]. Therefore, a markerless solution would be desirable in this context.

A few recent works have been proposed in the field of markerless medical AR. Some of them do not run in real-time, such as [4], [5]. From the best of our knowledge, there

are only two exceptions: the *miracle* proposed in [6] and the semi-automatic MAR environment proposed in [7].

*Miracle* is a magic mirror for teaching anatomy. It runs in real-time because the pose of the user's body is tracked with a real-time markerless tracking algorithm: the NITE skeleton tracking<sup>1</sup> [6].

The semi-automatic MAR environment uses the Kinect to reconstruct and track a region of interest in the patient. From a 3D reference model reconstructed previously, the virtual volumetric medical data can be positioned into the scene at the location of the patient's real anatomy. The techniques proposed in this work run in real-time by exploiting the parallelism provided by the graphics processing unit (GPU) [7].

To validate our techniques, we use a markerless tracking solution because it requires low-cost hardware components and runs in real-time. The problem associated with the current solutions in this field is accuracy. The techniques, in general, were conceived to help physicians during surgeries or for diagnosis, however their accuracy allows their use only for education purposes, which do not require high accuracy in the AR solution. Despite this scenario, a MAR environment can be used to validate our approach as we are main interested in evaluate performance and visual quality of the F+C visualization techniques. Therefore, the semi-automatic MAR environment proposed in [7] was chosen because it can be used for a specific region of interest in the patient, in opposition to the solution proposed in the *miracle*, which works on the patient's full body.

An application for on-patient medical data visualization demands a special attention to the composition between the virtual and real entities of the AR environment. Many approaches have been proposed in recent years to dynamically define how the composition will be done. Some of them are based on importance maps (e.g. [11], [12], [13]) that work purely on the color space. These methods capture the features of the image, however they are not accurate enough to be used for medical applications.

In the field of on-patient medical data visualization, the Contextual Anatomic Mimesis proposed in [8] controls the visualization by using three parameters: curvature of the patient's skin surface, angle of incidence factor (i.e. angle between the normal on the skin surface and a vector pointing from the position of the surface and the eye) and distance falloff (i.e. distance between each point on the surface and the intersection point of the line of sight and the skin surface). This is one of the first and most important methods proposed specifically for the field of on-patient medical data visualization, as it provides improved perception of the 3D medical data in the scene. However, it does not include the effect of volume clipping into the F+C visualization technique.

## III. MARKERLESS AUGMENTED REALITY ENVIRONMENT

The MAR environment used in this work is based on the one proposed in [7]. It will be briefly described in this section.

<sup>1</sup><http://www.openni.com>

To track the volumetric medical data in the AR environment without markers, a 3D reference model of the region of interest in the patient is generated. In this work, the region of interest consists in the patient’s face. To segment the face from the scene, a Viola-Jones face detector [14] is applied in the color image provided by the Kinect sensor. Once the face is detected, the region that contains the face is fixed. Then, the user is constrained to move his face in this fixed region. As long as the color and depth sensors of the Kinect are calibrated, this segmented region of the image can be transposed to the depth image. By denoising the depth map using a bilateral filter [15] and by converting the filtered depth map into a vertex and a normal map, the KinectFusion algorithm [16] is used to reconstruct the 3D reference model in real-time. The KinectFusion is an algorithm that integrates raw depth data captured from an RGB-D sensor into a 3D grid to produce a high-quality 3D reconstruction of the object/scene of interest. The grid stores for each voxel the distance to the closest surface and a weight that indicates uncertainty of the surface measurement. These volumetric representation and integration are based on the VRIP algorithm [17]. The 3D reconstructed model is extracted by detecting zero-crossings on the grid through a ray caster. This representation of the KinectFusion is specially important for the F+C visualization based on visible background on MRI data, where the ray casting is used to clip the 3D reference model directly from the KinectFusion’s grid. All of the steps described above run in GPU.

As evaluated in [18], the KinectFusion algorithm has accuracy of 10mm. Therefore, it is assumed that its reconstructed models are suitable to be used as reference for tracking in AR applications.

The 3D reference model reconstruction is done only once and it is the basis for the markerless AR live tracking. A semi-automatic registration method is used to place the medical data into the scene and align it to the 3D reference model in terms of scale, positioning and orientation. The result can be fine adjusted by the user.

After the placement of the medical data into the scene, the markerless tracking is started. The live tracking is done in two steps: during the reconstruction of the 3D reference model, to integrate the different viewpoints into a single reference model, and during the MAR with the patient and the medical data. A real-time variant of the Iterative Closest Point (ICP) algorithm [19] is used to estimate the transformation that aligns the current depth frame captured by the Kinect sensor with the previous one represented by the 3D reference model. As stated in [16], the use of the 3D reference model by the ICP allows a more consistent rigid registration with less incremental error. However, in presence of fast rigid motion, the ICP may fail. Taking advantage from the fact that the region of interest in this paper is a head, to minimize this problem, a real-time head pose estimation [20] is used to give a new initial guess to the ICP to compute correctly the current transformation [21].

#### IV. ON-PATIENT MEDICAL DATA VISUALIZATION BASED ON VOLUME CLIPPING

##### A. Volume Rendering

Volume rendering is concerned with techniques for generating images from volume data [22]. These images can be generated by solving the volume rendering integral based on a emission-absorption optical model, as shown in Equation 1.

$$I(D) = I_0 e^{-\int_{s_0}^D k(t)dt} + \int_{s_0}^D q(s) e^{-\int_s^D k(t)dt} ds. \quad (1)$$

where  $I(D)$  is the resulting radiance energy of integrating from the entry point into the volume ( $s = s_0$ ) to the exit point toward the camera ( $s = D$ ). The absorbed energy and emission components are represented by the absorption and emission coefficients  $k$  and  $q$  respectively.  $I_0$  is the radiance in the entry point  $s_0$ .

The volume rendering integral cannot be evaluated analytically. Therefore, the volume is typically rendered according a compositing scheme known as front-to-back direct volume rendering (DVR), the numerical computation of the Equation 1. The front-to-back DVR is defined by:

$$C_{dst} = C_{dst} + (1 - \sigma_{dst})C_{src} \quad (2)$$

$$\sigma_{dst} = \sigma_{dst} + (1 - \sigma_{dst})\sigma_{src} \quad (3)$$

where  $C_{dst} = c_{i+1}$ ,  $C_{src} = c_i$ ,  $\sigma_{dst} = 1 - T_{i+1}$ ,  $\sigma_{src} = \sigma_i$ , given the voxel  $i$  being traversed.  $C$  and  $c$  represent the color contribution,  $T$  the transparency and  $\sigma$  the opacity of the voxel.

To render the medical data based on DVR, a single rendering pass ray casting is applied based on the bounding box of the volume [23]. To render high quality images from the medical data in real-time, several techniques are employed: stochastic jittering to reduce sampling artifacts, fast GPU-based tri-cubic filtering [24], [25] to reduce filtering artifacts, empty-space leaping to skip non-visible voxels [26], early ray termination if the opacity accumulated is sufficiently high, pre-integrated transfer functions [27] to capture the high frequencies introduced in the transfer functions with low sampling rates and Blinn-Phong illumination with on-the-fly gradient computation to add realism in the final rendering.

Sometimes, the goal of the volume rendering is to extract and emphasize important parts of a volume. This issue can be solved by using volume clipping, which cuts away unimportant regions of the volume [22]. In this work, the volume is clipped according to six planes parallel to the faces of the volume bounding box. Nevertheless, we emphasize that the F+C visualization techniques are general in the sense that can be used independent of the technique applied to clip the volume.

After the medical volume rendering, the color frame buffer of the volume is loaded and sent to a shader to blend it with the RGB data coming from the Kinect sensor. The blending is done by the following linear interpolation:

$$I_{final} = \beta * I_{real} + (1 - \beta) * I_{medical} \quad (4)$$

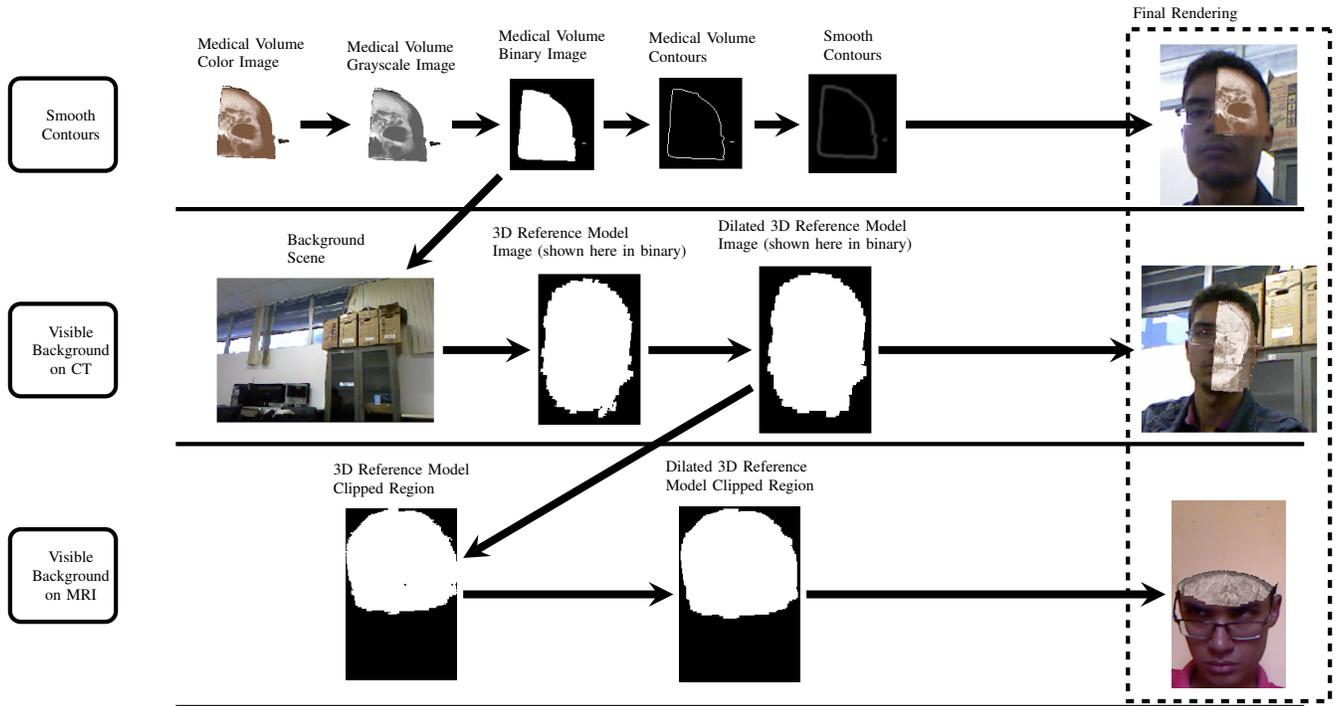


Fig. 2. A schematic view of the proposed techniques. F+C visualization based on smooth contours (top layer): From the binary image of the medical volume, contours are extracted and blurred to be used as a mask that smoothes the transition between the medical data and the real scene on the final rendering. F+C visualization based on visible background on CT data (middle layer): From the binary image of the medical volume, the dilated image of the 3D reference model and the background scene, the soft tissue of the medical data can be displayed merged with the background, while the bone is normally visualized. F+C visualization based on visible background on MRI data (bottom layer): By adding the clipping on the 3D reference model, the organs of the medical data can be displayed in the context of the patient’s region of interest.

where  $I_{real}$  is the image captured by the sensor,  $I_{medical}$  is the image corresponding to the medical volume, and  $I_{final}$  is the resulting image. The contribution of each image ( $\beta$ ) is defined dynamically by using the F+C visualization, which will be described in the next subsection.

Incorrect occlusion of virtual and real objects in an augmented scene is one of the fundamental problems in AR. To solve it for a specific viewpoint, the depth images of the 3D reference object  $D_{ref}$  and the 3D object coming from the sensor’s live stream  $D_{live}$  are used. If the depth from  $D_{live}$  is lower than  $D_{ref}$ , the object captured by the sensor is in front of the reference object and the volume is the occludee, otherwise, the volume is the occluder.

### B. Focus + Context Visualization

In this paper we introduce three methods for F+C visualization based on volume clipping. In a naive implementation, when the volume is clipped and its image is rendered in an AR environment, we obtain a result similar to the one presented in the Fig. 3(a), where there is no occlusion between the internal region of the volume visible after the clipping and the patient. If desirable, this effect can be removed in the single-pass ray casting. We check if the first hit position of the ray in the volume is in the clipped region. If it is, the ray is discarded from rendering, otherwise, the ray traverses the volume as normally done in the standard ray casting algorithm. The result of the application of this algorithm can be seen in Fig. 3(b).

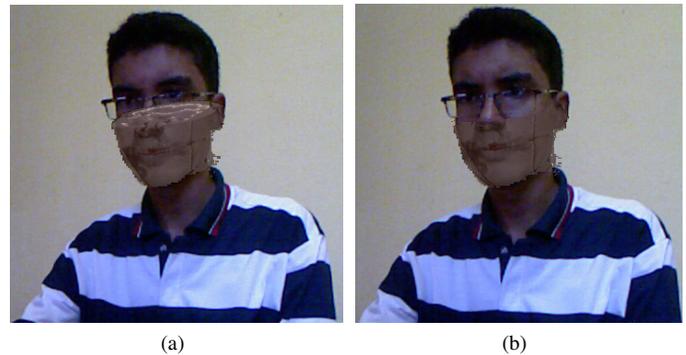


Fig. 3. Occlusion between volume’s internal structures and the patient’s region of interest. (a) Direct volume rendering with clipping. (b) Volume clipped rendered according to the proposed algorithm.

One of the issues related to the integration of the volume clipping into the AR environment is the presence of the visible edge located in the intersection between the volume and the clipping plane. As discussed in [8], it is desirable a smooth transition between the volume in the focus region and the rest of the AR scene. In order to achieve this goal, it is proposed here a new method for F+C visualization based on **smooth contours**, an algorithm that adds a smooth transition between the image resulting from the volume clipping and the context region. An overview of this algorithm can be seen in Fig. 2,

top layer.

The image of the volume is converted into a gray-scale image and then is binarized with the threshold computed from the Otsu’s method [28]. The contours of the binarized image are extracted by the method proposed in [29]. The contours are blurred by  $m$  iterations of a  $n \times n$  gaussian mask and sent to the shader. From empirical tests,  $m = 5$  and  $n = 3$  produced the best results. In the shader, the resulting image  $I_{smoothContours}$  is a mask that gives a weight to the blending between the volume contours and the scene. Also, a factor  $w_c$  is dynamically defined by the user to adjust the level of smoothing in the contours. Therefore, with this algorithm,  $\beta$  can be defined for the entire scene as:

$$\beta = I_{smoothContours} * w_c \quad (5)$$

The influence of the smooth contours can be seen in Fig. 4. It is visible that the proposed algorithm adds a smooth transition between the volume and the real scene.

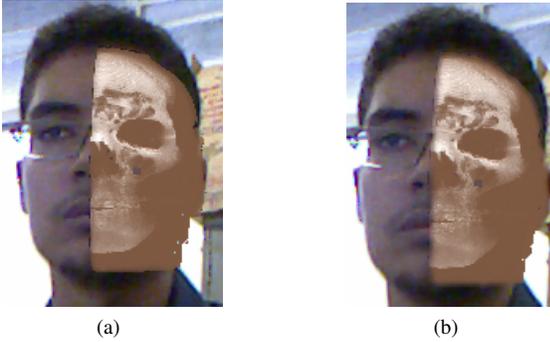


Fig. 4. Influence of the smooth contours in the final rendering. (a) Direct volume rendering with clipping. (b) Volume clipped rendered according to the proposed algorithm.

This method can be easily integrated with the F+C method proposed by Bichlmeier et al. [8], where  $\beta$  will be the maximum between the value obtained from our method and the method proposed in [8]. An example of the result of the proposed integration can be seen in Fig. 5.

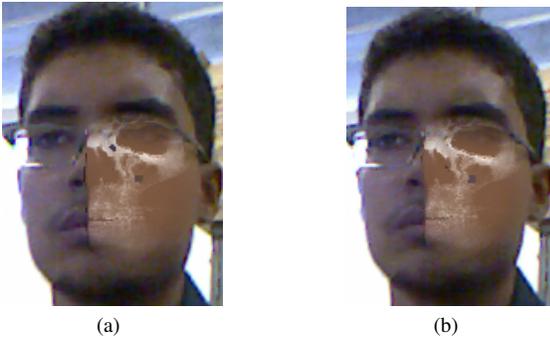


Fig. 5. F+C visualization based on smooth contours (a) and its extension with the distance falloff parameter (b).

Taking advantage from the clipping effect, we propose another two methods for F+C visualization, now based on

**visible background.** In these methods, we take advantage from the type of scanning (CT or MRI) to enable new ways for physicians to explore the medical data on the patient.

In volume rendering, CT data can be used to enable the visualization of internal structures such as bones. In the case of a head, this structure can be the patient’s skull. By designing an appropriate transfer function, the skull can be visualized apart from the soft tissue of the volume. In this situation, it is desirable to see the real background scene when looking at the region between the skull and the soft tissue or the soft tissue itself, because both are context regions of the volume and the virtual background can be seen, depending on the transfer function chosen. An overview of the proposed approach to enable this kind of visualization can be seen in Fig. 2, middle layer.

The background scene is captured and stored in the memory. Next, the image of the volume after clipping is binarized and sent to the shader as a foreground subtraction mask  $I_{subtraction}$ . This mask identifies the region where the background can be visualized based on the gray intensity of the volume. In our case,  $D_{ref}$ , the depth image of the 3D reference model, does not overlap perfectly the patient’s region of interest. To solve this problem,  $D_{ref}$  is dilated only on its contours to preserve the original depth of the 3D reference model and sent to the shader to represent the patient’s region of interest. The pseudocode of the shader to render the F+C visualization based on visible background on CT data can be seen in the **Algorithm 1**, lines 1-15, 22-24.

The color image captured from the Kinect sensor is rendered in the region that does not represent the patient’s region of interest (i.e. where the depth of the 3D reference object is 0, as it was not reconstructed) (lines 2-4). The captured color image is also rendered when the volume is occluded and the occludee has depth (e.g. it is not in a hole region) (lines 5-7). Next, if the fragment is in the subtraction mask region, the volume or the background scene are rendered. Otherwise, the fragment is in the clipped region and the real color image is rendered (lines 23-24). The gray intensity is computed from the volume (by the *gray* function) and assigned to  $\beta$ . Considering that the bone is rendered with a gray level greater than the soft tissue’s and than a user-defined threshold  $w_{grayLevel}$ , it is rendered without the background scene. As the bone and the soft tissue have different gray intensities,  $w_{grayLevel}$  can be adjusted to render the bone with its real color and the soft tissue linearly interpolated with the background scene (lines 8-15). If the bone is rendered with a gray level lower than the one present in the soft tissue, the condition in line 12 can be adjusted from less than (<) to greater than (>) to separate better these regions.

In volume rendering, MRI data can be used to enable the visualization of internal structures in the body such as organs. In an AR environment, the best way to visualize these data is clipping not only the volume but also the corresponding region on the patient’s image. In this situation, it is desirable to see the background scene in the region clipped. An overview of the proposed approach to visualize this kind of scenario can

---

**Algorithm 1** Shader for F+C visualization based on visible background

---

```
1: for each fragment on the shader do
2:   if  $D_{ref} == 0.0$  then
3:     return  $I_{real}$ ;
4:   end if
5:   if  $D_{live} < D_{ref}$  and  $D_{live} \neq 0.0$  then
6:     return  $I_{real}$ ;
7:   end if
8:   if  $I_{subtraction} == 1.0$  then
9:     if CT data then
10:       $grayLevel \leftarrow gray(I_{medical})$ ;
11:       $\beta \leftarrow grayLevel$ ;
12:      if  $grayLevel < w_{grayLevel}$  then
13:        return  $\beta * I_{background} + (1 - \beta) * I_{medical}$ ;
14:      end if
15:      return  $I_{medical}$ ;
16:    else
17:      if  $I_{medical} == 0.0$  then
18:        return  $I_{background}$ ;
19:      end if
20:      return  $I_{medical}$ ;
21:    end if
22:  end if
23:  return  $I_{real}$ ;
24: end for
```

---

be seen in Fig 2, bottom layer.

The background scene is saved. Next, taking advantage from the volumetric representation of the KinectFusion’s grid, in which the 3D reference model is stored, the patient’s region of interest is clipped. To do that, the algorithm ray casts the KinectFusion’s grid and when the ray traverses a zero-crossing position and it is in the clipped region, the voxel’s corresponding pixel is rendered in the output image. The output image from this algorithm is  $I_{subtraction}$ , as it will be used to indicate whether the background image must be rendered. The medical volume is clipped separately and sent to the shader. Both  $I_{subtraction}$  and  $D_{ref}$  are dilated because of the problem of overlapping described before. The pseudocode of the shader to render the F+C visualization based on visible background on MRI data can be seen in the **Algorithm 1**, lines 1-8, 16-24.

The color image captured from the Kinect sensor is rendered in the same way as described in visible background on CT data. The main difference here is that if the subtraction mask is active (i.e. the patient’s region of interest is clipped) and if there is medical data to be visualized, it is rendered. Otherwise, the background image is rendered.

## V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section we analyze the performance for each one of the main methods for F+C visualization discussed in this paper.

For all tests we used an Intel(R) Core(TM) i7-3770K CPU @3.50GHz 8GB RAM and a NVIDIA GeForce GTX 660. We used the open source C++ implementation of the KinectFusion<sup>2</sup> released by the PCL project [30].

The medical dataset used is a CT volumetric data of a head from the Visible Human Project<sup>3</sup> of resolution  $128 \times 256 \times 256$  and a MRI volumetric data of a head from the MRI Head available in Volume Library<sup>4</sup> of resolution  $256^3$ . Both are of typical resolution of a head medical volume and therefore do not affect our performance evaluation. The reference human head is reconstructed with the KinectFusion using a grid with volume size of  $70cm \times 70cm \times 140cm$  and resolution of  $512^3$ .

We evaluate the performance and visual quality of the proposed techniques in a scenario where the patient’s head is augmented with a volumetric dataset of a head. The use of a generic volume does not affect our visual quality evaluation since the volume is scaled and positioned semi-automatically by the user. In terms of accuracy, the reconstruction has accuracy  $\approx 10mm$ , live tracking has a not incremental accuracy  $\approx 3mm$  and the accuracy of the registration between the medical data and the reference model depends on the quality of the user’s fine adjustment because of the use of a generic volume. However, in this work we are main interested in evaluate the visual quality and performance of the final rendering of the scene.

### A. Performance Evaluation

In our preprocessing computation, the 3D reference model is reconstructed at 30 frames per second (FPS). In general, the user takes less than 10 seconds to place and adjust the volume into the scene. The markerless live tracking and the volume rendering techniques run at 30 FPS. The performance of our application for each F+C visualization technique can be seen in the Fig. 6.

The F+C visualization proposed in [8] runs in full real-time because it operates directly on the shader by changing the three parameters described in the Section II.

The F+C visualization based on smooth contours decreases the overall performance because of the reading of the frame-buffer to get the image of the volume rendered, which takes most of the time to be executed in this mode. This operation is needed because the algorithm to find the contours on the volume can not be used directly on the shader.

The F+C visualization based on visible background on CT data is faster than the one based on smooth contours because the binarization of the image of the volume is performed on the shader and the image does not need to be readback by the CPU. The dilation applied on  $D_{ref}$  is negligible in terms of performance.

The F+C visualization based on visible background on MRI data is slightly slower than the one based on CT data because of the ray casting performed on the KinectFusion’s grid to render the clipped region. When it is desirable high-quality

<sup>2</sup><http://svn.pointclouds.org/pcl/trunk/gpu/kinfu/>

<sup>3</sup><http://www.nlm.nih.gov/research/visible/>

<sup>4</sup><http://www9.informatik.uni-erlangen.de/External/vollib/>

rendering, the step size of the ray casted can be decreased, at the cost of loss in performance.

In an application where the performance is a critical factor, one solution to improve it is by changing the volume size of the KinectFusion’s grid from  $512^3$  to  $256^3$ . The drawback of this changing is the presence of more artifacts in the final composition, as the 3D reference model will be reconstructed with a lower quality and it has a direct relation with the proposed techniques for two reasons: it is the reference for live tracking and, from it, the region of interest is dilated or clipped.

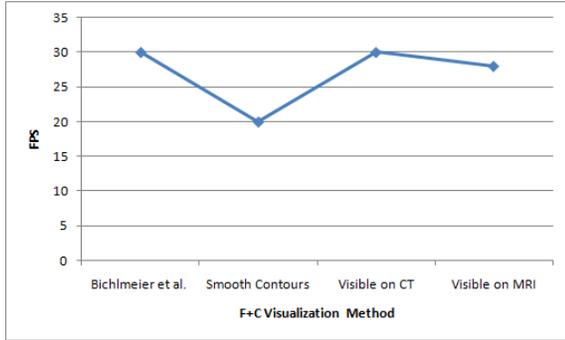


Fig. 6. Performance results measured in frames per second (FPS) for each one of the F+C visualization techniques discussed in this paper. Bichlmeier et al. is the technique proposed in [8].

### B. Visual Quality Evaluation

By the use of the shader proposed in the **Algorithm 1**, our approach handles occlusion as can be seen in Fig. 7.



Fig. 7. Occlusion support.

The influence of the parameter  $w_c$  in the F+C visualization based on smooth contours can be seen in Fig. 8. As  $w_c$  increases, the transition between the volume and the real scene becomes smoother. At the same time, the volume contours become less visible.

The influence of the parameter  $w_{grayLevel}$  in the F+C visualization based on visible background on CT data can be seen in Fig. 9. In this figure, it is visible that by changing the parameter, the volume can be rendered almost completely invisible, with the soft tissue linearly interpolated with the background scene or with the volume rendered without the background scene.

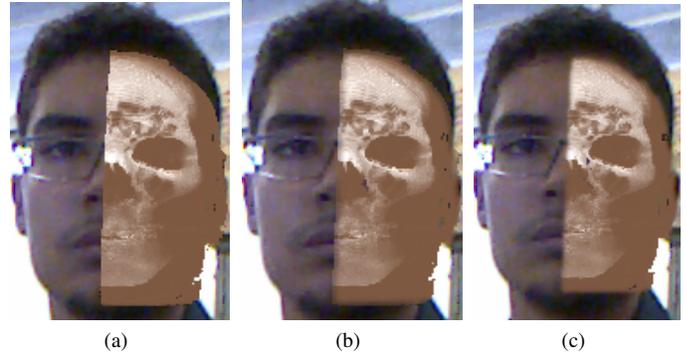


Fig. 8. Influence of the parameter  $w_c$  in smooth contours. (a)  $w_c = 0$ . (b)  $w_c = 2$ . (c)  $w_c = 4$ .

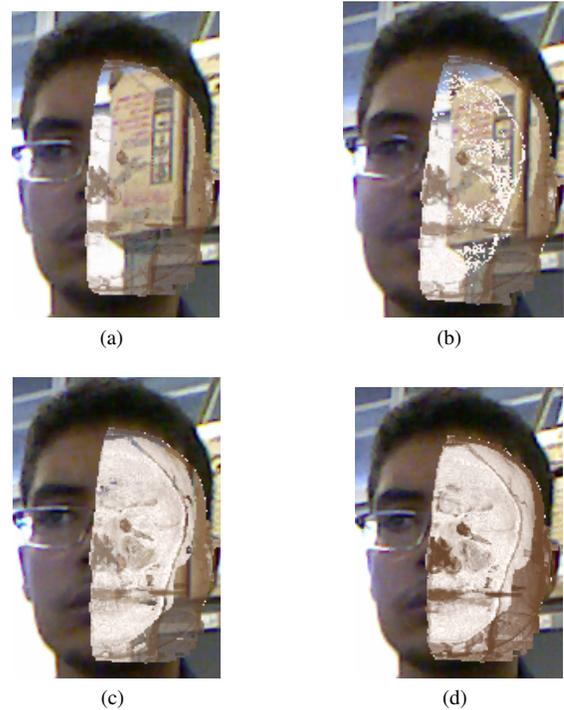


Fig. 9. Influence of the parameter  $w_{grayLevel}$  in visible background on CT data. (a)  $w_{grayLevel} = 0$ . (b)  $w_{grayLevel} = 0.5$  (c)  $w_{grayLevel} = 0.75$ . (d)  $w_{grayLevel} = 1$ .

In Fig. 10, we can see more examples of interactions with the F+C visualization based on visible background with MRI data. In the upper clipping, there is a good composition of the medical data in the scene. However, in the lateral clipping, it is visible the 3D reference model clipping and the background rendering because of the shadow cutting present on the final scene.

## VI. CONCLUSION AND FUTURE WORK

We have presented some methods to improve on-patient medical data visualization in an MAR environment by using volume clipping. We have evaluated the performance and the visual quality of the proposed approaches and shown that they

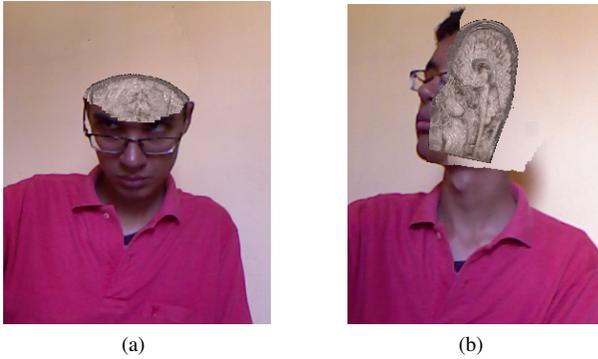


Fig. 10. Examples of interactions with the F+C visualization based on visible background on MRI Data. (a) Upper clipping. (b) Lateral clipping.

are capable to run in real-time. Moreover, the visualization methods improve the visual perception of the final scene by enabling smooth transition between the volume and the real scene and by adding new ways to understand the scene taking advantage from the volume clipping. In addition, our approach supports occlusion.

Encouraged by the field of image-based lighting, for future work we intend to integrate the real local and global illumination effects into the volume rendering. Also, one of the current limitations of the proposed approach is that the patient must maintain the region of interest as-rigid-as-possible during tracking and reconstruction. In real situations, the deformation must be taken into consideration to deal with these changes in the reference model. In terms of tracking, our approach does not support relocalization of the patient's region of interest if the markerless tracking fails during the AR step. Moreover, the tracking fails if the region of interest is not visible in the scene. Further improvements can be applied in this sense to create a more robust algorithm.

#### ACKNOWLEDGMENTS

We are grateful to the PCL project for providing the open-source implementation of the KinectFusion algorithm. We are also grateful to Gabriele Fanelli for providing the open-source implementation of her real-time head pose estimation algorithm. Finally, we would like to thank the anonymous reviewers for their valuable comments and suggestions. This research is financially supported by FAPESB and CAPES.

#### REFERENCES

- [1] S. K. Card, J. D. Mackinlay, and B. Shneiderman, Eds., *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
- [2] O. Kutter, A. Aichert, C. Bichlmeier, J. Traub, S. M. Heining, B. Ockert, E. Euler, and N. Navab, "Real-time Volume Rendering for High Quality Visualization in Augmented Reality," in *AMI-ARCS 2008*. New York, USA: MICCAI Society, Sept. 2008.
- [3] M. Wiczorek, A. Aichert, O. Kutter, C. Bichlmeier, J. Landes, S. M. Heining, E. Euler, and N. Navab, "GPU-accelerated Rendering for Medical Augmented Reality in Minimally-Invasive Procedures," in *BVM 2010*. Springer, Mar. 2010.
- [4] L. Maier-Hein, A. M. Franz, M. Fangerau, M. Schmidt, A. Seitel, S. Mersmann, T. Kilgus, A. Groch, K. Yung, T. R. dos Santos, and H.-P. Meinzer, "Towards mobile augmented reality for on-patient visualization of medical images," in *Bildverarbeitung für die Medizin*, ser. Informatik Aktuell. Springer, 2011, pp. 389–393.

- [5] H. Suenaga, H. Hoang Tran, H. Liao, K. Masamune, T. Dohi, K. Hoshi, Y. Mori, and T. Takato, "Real-time in situ three-dimensional integral videography and surgical navigation using augmented reality: a pilot study," *International Journal of Oral Science*, no. 2, p. 98–102, 2013.
- [6] T. Blum, V. Kleeberger, C. Bichlmeier, and N. Navab, "miracle: Augmented reality in-situ visualization of human anatomy using a magic mirror," in *VR, 2012 IEEE*, 2012, pp. 169–170.
- [7] M. Macedo, A. Apolinario, A. C. Souza, and G. A. Giraldi, "A Semi-Automatic Markerless Augmented Reality Approach for On-Patient Volumetric Medical Data Visualization," in *SVR, Brazil*, 2014.
- [8] C. Bichlmeier, F. Wimmer, S. M. Heining, and N. Navab, "Contextual anatomic mimesis hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality," ser. ISMAR '07. Washington, DC, USA: IEEE, 2007, pp. 1–10.
- [9] H. G. Debarba, J. Grandi, A. Maciel, and D. Zanchet, "Anatomic hepatectomy planning through mobile display visualization and interaction," in *MMVR*, vol. 173. IOS Press, 2012, pp. 111–115.
- [10] J.-D. Lee, C.-H. Huang, T.-C. Huang, H.-Y. Hsieh, and S.-T. Lee, "Medical augmented reality using a markerless registration framework," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5286–5294, 2012.
- [11] E. Mendez, S. Feiner, and D. Schmalstieg, "Focus and context in mixed reality by modulating first order salient features," ser. SG'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 232–243.
- [12] C. Sandor, A. Cunningham, A. Dey, and V.-V. Mattila, "An augmented reality x-ray system based on visual saliency," in *ISMAR*. IEEE Computer Society, 2010, pp. 27–36.
- [13] D. Kalkofen, E. Veas, S. Zollmann, M. Steinberger, and D. Schmalstieg, "Adaptive ghosted views for augmented reality," in *ISMAR*. Adelaide, SA, Australia: IEEE Computer Society, Oct. 2013.
- [14] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004.
- [15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, jan 1998, pp. 839–846.
- [16] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," ser. UIST '11. USA: ACM, 2011, pp. 559–568.
- [17] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 303–312.
- [18] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, and D. Kondermann, "When can we use kinectfusion for ground truth acquisition?" in *IROS*. IEEE Computer Society, 2012.
- [19] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *3DIM*, 2001, pp. 145–152.
- [20] G. Fanelli, T. Weise, J. Gall, and L. V. Gool, "Real time head pose estimation from consumer depth cameras," in *DAGM'11*, Sept. 2011.
- [21] M. Macedo, A. Apolinario, and A. C. Souza, "A Robust Real-Time Face Tracking using Head Pose Estimation for a Markerless AR System," in *SVR, Brazil*, May 2013.
- [22] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel, *Real-time Volume Graphics*. USA: A. K. Peters, Ltd., 2006.
- [23] M. Hadwiger, P. Ljung, C. R. Salama, and T. Ropinski, "Advanced illumination techniques for gpu-based volume raycasting," ser. SIGGRAPH '09. New York, NY, USA: ACM, 2009, pp. 2:1–2:166.
- [24] C. Sigg and M. Hadwiger, "Fast third-order texture filtering," in *GPU Gems 2*, M. Pharr, Ed. Addison-Wesley, 2005, pp. 313–329.
- [25] D. Ruijters, B. M. ter Haar Romeny, and P. Suetens, "Efficient gpu-based texture interpolation using uniform b-splines," *J. Graphics Tools*, vol. 13, no. 4, pp. 61–69, 2008.
- [26] W. Li, K. Mueller, and A. Kaufman, "Empty space skipping and occlusion clipping for texture-based volume rendering," in *VIS 2003*. IEEE Computer Society, 2003, pp. 317–324.
- [27] K. Engel, M. Kraus, and T. Ertl, "High-quality pre-integrated volume rendering using hardware-accelerated pixel shading," ser. HWWS '01. New York, NY, USA: ACM, 2001, pp. 9–16.
- [28] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," *SMC*, vol. 9, no. 1, pp. 62–66, 1979.
- [29] S. Suzuki, "Topological structural analysis of digitized binary images by border following," *CVGIP*, vol. 30, no. 1, pp. 32–46, 1985.
- [30] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *ICRA*. IEEE Computer Society, may 2011, pp. 1–4.