CEIG - Spanish Computer Graphics Conference (2016)
Alejandro García-Alonso and Belen Masia (Editors)

Spatial opacity maps for direct volume rendering of regions of interest

A. Rodríguez-Aguilera[1] and A. León[1]

[1]Virtual reality lab, University of Granada, Spain

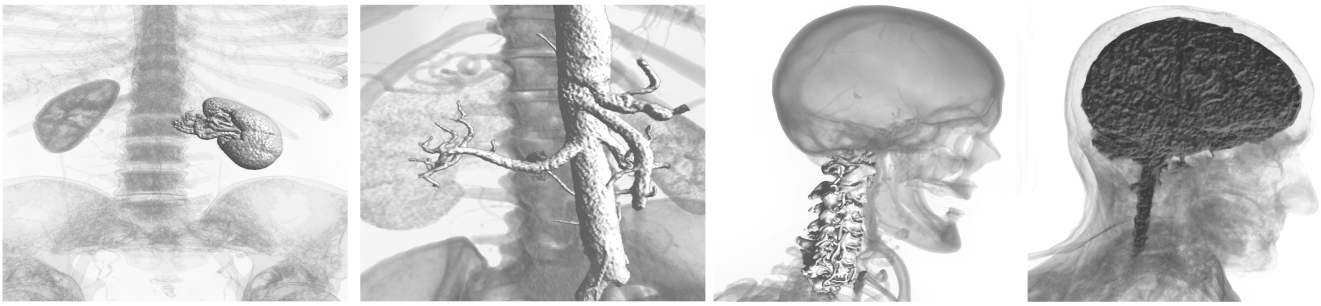

Figure 1: *A single point selection on the 2D slices of a medical dataset is the sole input for our method. Our spatial opacity maps highlight the selected feature and a simple automatic transfer function generation algorithm reveals relevant contextual information.*

Abstract

Despite the mature state of the volume rendering field, its adoption in medical applications is hindered by its complex parametrization and control, and 2D slice based tools are still preferred for clinical workflows.
In this paper, we introduce the concept of spatial opacity maps as an interactive tool for exploring volumetric data focusing on the rendering of features of interest. In a region growing fashion, the maps are dynamically created from a user input on the 2D slices, taking into account not only the density values of the structure but also the topology. Using this approach, an inexperienced user is able to generate meaningful 3D renderings with no need to tweak non-intuitive visualization parameters. The spatial opacity maps are independent of the current visualization parameters and they can be easily plugged into the volume rendering integral and combined with other approaches for region of interest (ROI) visualization. We combine our approach with a simple automatic transfer function generation algorithm to improve the visualization of the contextual data.

1. Introduction

During the last decades, the volume rendering pipeline has been gradually improved to a great level of quality and maturity. State-of-the-art techniques for volume rendering [EHK*06, BRGIG*14] allow to interactively generate 3D visualizations of volumetric data including advanced illumination, material and shadowing techniques that provide rich information regarding depth and spatial relationships in the data. However, as stated in recent works [BKKG08, WVFH12], its wide adoption for clinical applications is very limited, mainly due to the excessive level of expertise required to control the visualization parameters that, in the end lead the clinicians to work with classical, 2D based approaches, which are closer to their training knowledge and are thus more practical.

Despite this fact, volume rendering is undeniably useful in many contexts as it is able to provide additional information regarding spatial properties of features of interest and contextualized relations between different features in the data, which provide meaningful information. Therefore, during the last years, there is a great effort to minimize the complexity of controlling the visualization of features of interest.

One way to achieve this goal that has proven useful is to combine the classic 2D pipeline with the 3D volume rendering [WVFH12,

DOI: 10.2312/ceig.20161310

KBKK07], connecting them unidirectionally or bidirectionally in order to achieve a complementary control of both metaphors.

In this work, we present the spatial opacity map as a novel tool to easily highlight features of interest in volumetric data using the slice-based selection metaphor as sole input for the method. The opacity maps account for the spatial topology of the selected feature, which allows to preserve context information while focusing on the visualization of the region of interest, and they are seamlessly integrated into the volume rendering integral to highlight the feature of interest with no need to modify visualization parameters. Moreover, since the opacity maps are defined as a separated, easy-to-plug component for the volume rendering pipeline, they can be easily combined with existing volume exploration techniques and state-of-the-art rendering techniques.

The opacity maps are interactively generated through a parallel GPU computation on the volume data as we explain in this work. In our examples we show that even in a complete absence of manual or assisted preprocessing, they are able to easily provide a meaningful visualization regarding a selected structure of interest, but they can also benefit from complimentary techniques, such as automatic transfer function generation methods, as the one proposed in this work.

2. Related work

The study of techniques to ease the 3D exploration of volume data and the highlighting of regions of interest (ROIs) has spanned many approaches focusing on different parts of the control interface and volume rendering pipeline.

A large body of approaches provides new tools and models for the volume rendering stage. Viola et al. [VKG04] proposed an importance-driven approach to perform focus+context rendering by using pre-segmented object information to compose the final image. Correa et al. [CSC06] proposed a set of manipulation operators applied on the rendering stage, also relying on pre-segmented data. Bruckner et al. [BGKG06] proposed a rendering model, providing a substitute for the conventional clipping techniques by preserving some of the context information during the interaction. Since the most time-consuming manual task during 3D volume rendering is the search for a good transfer function, many works propose interactive techniques to ease this process. Good examples are the work of Kniss et al. [KKH01], where they proposed several manipulation widgets for an assisted specification of multidimensional transfer functions, and the work of Correa and Ma [CM11], introducing the concept of visibility-driven transfer functions as a semi-automatic method for easing the process of generating meaningful transfer functions to maximize the visibility of regions of interest. Roettger et al. [RBS05] proposed a feature selection tool introducing the spatialized transfer functions, which are automatically generated after classifying the different regions of the model to provide different colors to the different identified feature classes and allows the user to highlight the identified classes by selecting them on the transfer function. The main drawback of this approach is the non-intuitive selection mechanism, since the generated spatialized transfer functions lack a real physical or visual meaning w.r.t. the real feature and thus the selection process becomes difficult. Moreover, the method requires an initial manual setup of the *feature radius* parameter and similar features may become merged into the same class, thus impeding the selection of the individual features. These and other approaches provide very powerful tools for the direct volume rendering stage, but they either rely on pre-processed segmentation or assisted trial and error exploration procedures directly over the 3D visualization.

Other works rely on the 2D selection of features using the 2D slices of the model to generate 3D visualizations, exploiting the fact that this is the standard exploration approach used in clinical workflows. Kohlmann et al. [KBKK07, KBKG08] proposed a system to generate meaningful volumetric views by 2D-selecting features and automatically generating the camera position, zoom factor and clipping plane. However, their approach also relies in a good pre-defined transfer function for the model.

Our approach is also based on the selection of features on the 2D slices to generate the spatial opacity maps, however, it also avoids other inputs such as a manually tuned transfer function or any other manual pre-processing of the data. Sherbondy et al. [SHN03] proposed a similar interactive 2D-based exploration approach by obtaining the selected feature using a region growing approach. However, they apply the obtained binary segmentation as a mask to the rendering, impeding the visualization of any kind of contextual information aside from the highlighted feature. Our spatial opacity maps can be seamlessly plugged into the standard volume rendering pipeline, thus they can take advantage of any defined transfer function to display contextual information as we demonstrate in our results, and they can also be combined with existing focus+context techniques, such as the ones previously referred in this section. Huang and Ma [HM03] also exploit the region growing approach by generating a transfer function based on the selected feature, but generally, a lack of context information is observed since the transfer function is defined to only reveal the opacity values present in the feature, and undesired features with similar opacities to the selected feature may occlude the region of interest. Our opacity maps avoid these two issues because they highlight only the selected feature instead of all the features with similar opacity and, as already mentioned, their compatibility with the standard volume rendering pipeline offers the possibility to expose contextual data using different focus+context techniques.

3. Spatial opacity map

The principle of our spatial opacity maps and their generation procedure are founded on the well known seeded region growing segmentation approach [AB94]. We define a homogeneity criterion between neighboring voxels based on their properties regarding a seed voxel. In the original region growing approach this criterion is used to determine the membership of the new voxel to the selected region attending to the value of the voxels, generating a binary membership classification as output. Instead, we apply it to compute an opacity value for the new voxel based on the opacity value of its neighbor and the homogeneity factor, thus the output of our approach is a 3D scalar field of opacity values over the volume.

The generation process starts when the user selects a point in a 2D slice. The voxel containing the point is considered as the *seed*

voxel, its value on the opacity field is set to the maximum opacity value and the standard deviation of voxel values regarding its 1-ring neighborhood is computed and stored. After that, the opacity field is iteratively expanded by updating the 6 direct neighbors of the voxels that had their opacity modified in the previous iteration (in the first iteration, the 6 direct neighbors of the seed voxel are candidates for updating).

Candidate voxels are updated using an opacity extinction function. Let d_s be the density value of the *seed voxel* and let σ_s be its stored standard deviation, let v be a candidate voxel with density value d_v and current opacity o_v that needs to be updated due to a previously modified neighboring voxel w with current opacity o_w, a new candidate opacity is computed as

$$o_v^* = o_w - E_v, \tag{1}$$

with

$$E_v = \frac{|d_s - d_v| - \sigma_s}{\lambda \sigma_s}. \tag{2}$$

E_v automatically adapts the extinction behavior to the presence of noise in the data of the selected feature, and can be further tuned by setting the λ value. We have found however that with $\lambda = 30$ we obtain good results in all the scenarios tested in this paper, thus we have simply set it by default and the sole input in all the examples shown in this work is a 2D selection in the slices.

Since a voxel can be visited several times by several neighbors, its opacity value is only updated if $o_v^* > o_v$. If that is the case, the voxel opacity is updated, thus its neighbors will require an update on the next iteration.

3.1. Parallel implementation

The expansion process of a spatial opacity map is easily computed using the GPU by processing all the voxels requiring an update in parallel in each iteration. After setting the seed voxel, for each expansion iteration a kernel is invoked, launching one thread per voxel. The algorithm is shown in Algorithm 1.

Algorithm 1 Opacity expansion kernel

Input: Voxel v
$o_v \leftarrow v.opacity$
$o_w \leftarrow 0$
$N \leftarrow v.neighbors$
for each n **in** N **do**
  $o_w \leftarrow max(o_w, n.opacity)$
end for
$E_v \leftarrow$Equation 2
$o_v^* \leftarrow o_v - E_v$
$o_v \leftarrow max(o_v, o_v^*)$

Each thread loads the current opacity value of its assigned voxel and the highest opacity value of its neighbors. Then, the candidate opacity is computed using (1). If it is higher than the voxel's current opacity, it is updated.

The expansion reaches elements directly connected to the last influenced voxels, thus on the n^{th} iteration the kernel is launched on a $(n+1)^3$ region centered on the seed voxel, ensuring its reach to all the candidate voxels.

3.2. The volume integral

In order to visualize a generated spatial opacity map, we plug it into the volume rendering integral.

Let us first recall the base volume emission-absorption model [EHK*06]. The radiant energy C reaching the eye from a given direction is defined as an integral along the direction ray

$$C = \int_0^\infty c(t) \cdot e^{-\int_0^t \kappa(\hat{t}) d\hat{t}} dt,$$

with $c(s)$ and $\kappa(s)$ the emission and absorption coefficients, defined as scalar fields over the volume.

In practice, a numerical approximation of the integral is used, first approximating the emissions and absorptions along the ray as $C_i = c(i \cdot \Delta t)\Delta t$ and $A_i = 1 - e^{-\kappa(i \cdot \Delta t)\Delta t}$, now in the forms of colors and opacities respectively, to finally yield

$$\tilde{C} = \sum_{i=0}^{n} C_i \prod_{j=0}^{i-1} (1 - A_j), \tag{3}$$

with n the number of samples.

Our opacity maps are in fact scalar fields $o(s)$ that can be integrated along with the existing volume properties. Thus, we redefine the opacity component as $A_i^* = (1 - e^{-\kappa(i \cdot \Delta t)\Delta t}) o(i)$.

Substituting this new opacity component in (3) yields a modified volume rendering equation

$$\tilde{C} = \sum_{i=0}^{n} C_i \prod_{j=0}^{i-1} (1 - A_j^*).$$

With this formulation, we seamlessly integrate the opacity maps within the standard volume rendering pipeline, and thus we are able to apply state-of-the-art rendering along with other existing approaches applied to the standard pipeline.

4. Interactive visualization

The scalar field of a spatial opacity map can be configured to produce opacity values within the range $[o_{min}, o_{max}]$, with $0 \leq o_{min} < o_{max} \leq 1$. Thus when a seed voxel is selected, its opacity value is set to o_{max} and the opacity value of the rest of the voxels is set to o_{min} before the iterative expansion starts.

Opacity maps encompass enough information to produce meaningful renderings without modifying other visual parameters, but their true potential shows up when they are combined with other existing tools, both automatic and assisted. Moreover, the interactive control of the expansion of the selected feature also proves useful for further understanding of the anatomy, since it allows to gradually explore complex topologies, which is of interest for features such as vascular networks. In this section we explore these different options.

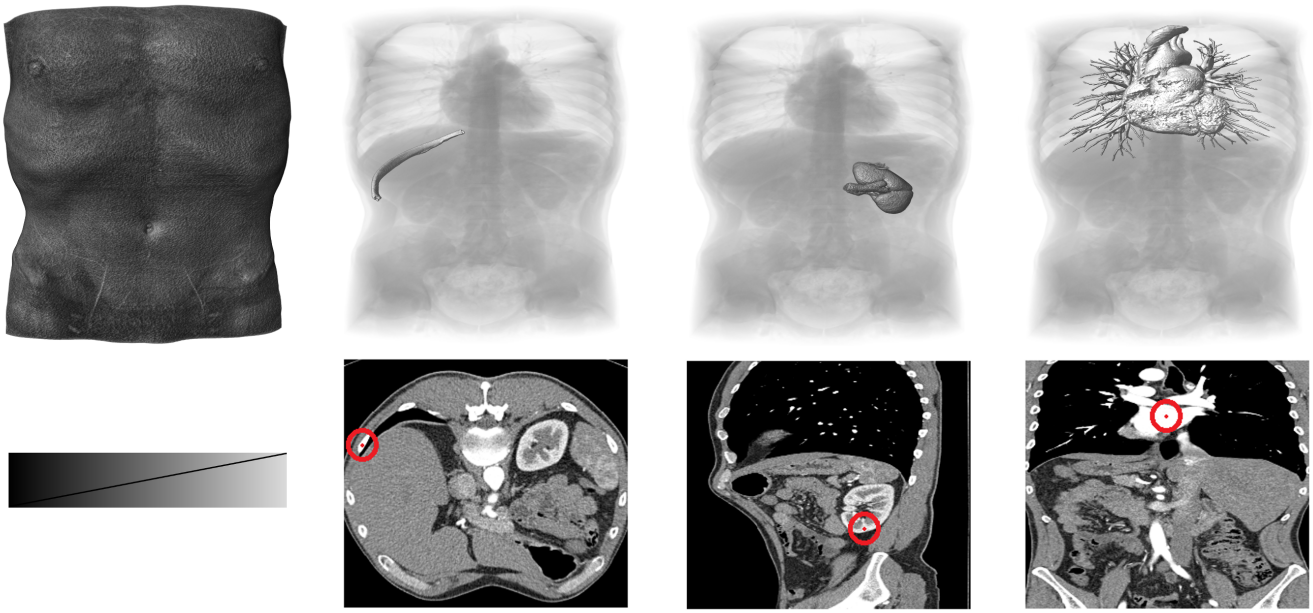

Figure 2: *The initial volume rendering state using the default transfer function (leftmost column) is combined with our opacity maps when the user selects and expands the desired feature. The final visualization is shown in the top row, and the selection of the data slices is shown below each case.*

4.1. Standalone visualization

The opacity maps make a clear distinction between ROI and contextual information, and thus a standalone visualization only accounting for this information should emphasize this difference. We test this by setting $o_{min} = 0.005$ (highly transparent) and $o_{min} = 1$ (opaque), so that the selected feature has the maximum visibility while preserving the contextual data, and we leave the default transfer function (Fig. 2, bottom left) untouched. Of course, a context-free visualization of the feature of interest is also easily obtained by setting $o_{min} = 0$.

Fig. 2 shows several examples of the achieved renderings, obtained simply by selecting the desired feature in the 2D slices. Several features can be highlighted at the same time by a simple combination of their opacity maps, as shown in Fig. 3.

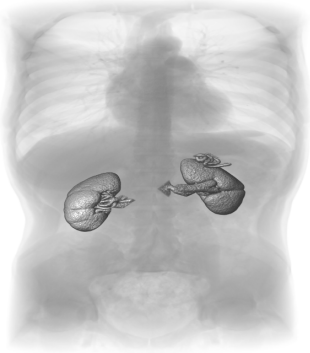

Figure 3: *Several opacity maps can be combined before performing the rendering stage. In this case, the two kidneys of a patient are highlighted.*

4.2. Enhanced contextual visualization

Although the standalone visualization allows a clear distinction of the ROI, all the contextual information is displayed homogeneously and thus makes it difficult to distinguish certain relations between features.

To provide a better visualization of the contextual information while maintaining the parameter-free interaction, an automatic transfer function generation method can be applied.

We exploit the data obtained from the seed voxel to generate a simple transfer function to distinguish contextual data with similar density values to those of the selected feature, referred as *primary* context information, from contextual data with density values far from those of the selected feature, referred as *secondary* context information:

We compute a Gaussian function with the form

$$g(x) = e^{-\frac{(x-\mu_s)^2}{2\sigma_s^2}},$$

with μ_s the mean density value of the 1-ring neighborhood of the seed voxel.

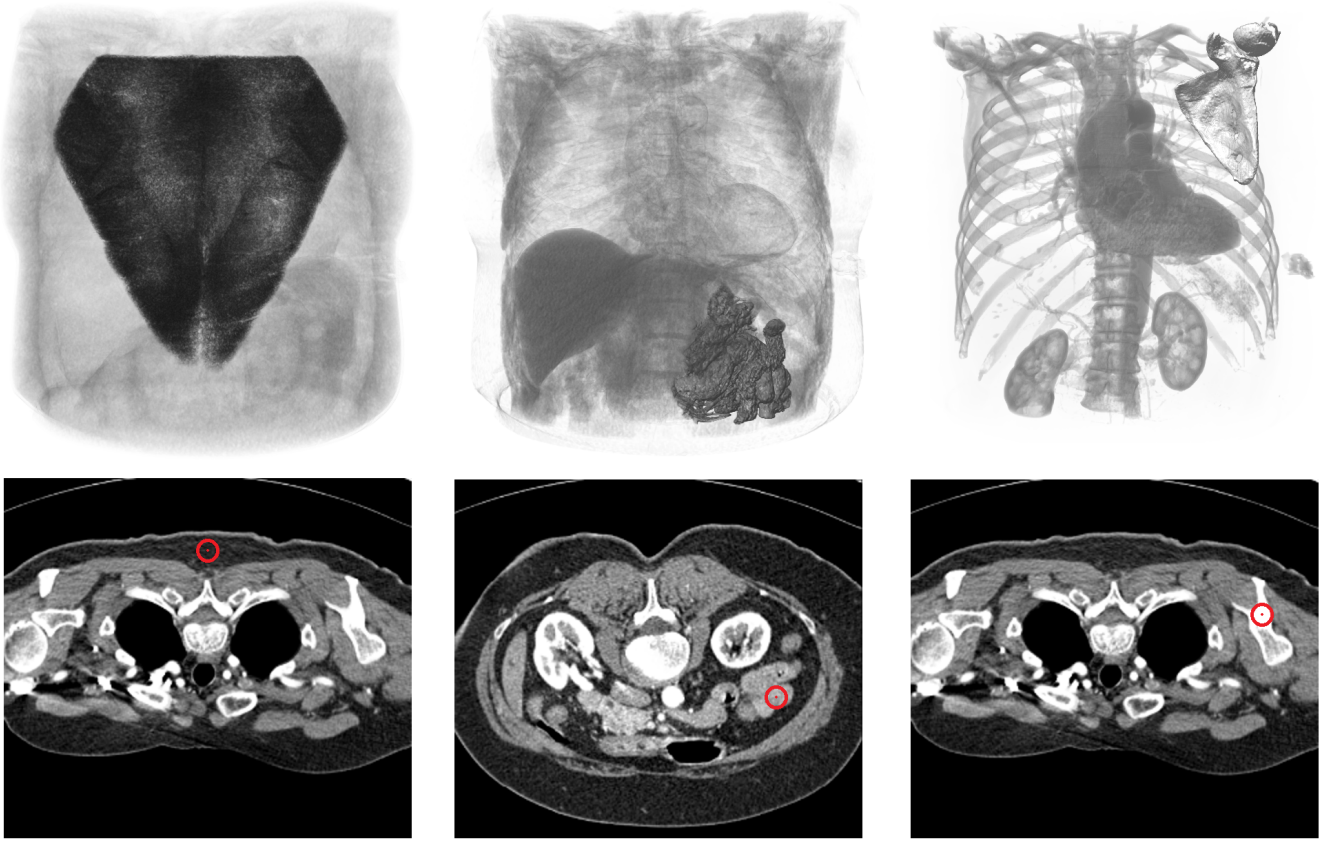

Figure 4: *From left to right, the fat layer of the back, the intestines and the scapula are selected. In each case, the transfer function is automatically updated to reveal related contextual structures.*

This function is then composed with the default transfer function $t(x)$ to produce a final transfer function with the form $f(x) = t(x) \cdot (a + b \cdot g(x))$, with a and b parameters that control the contrast between the *primary* and *secondary* context information. These parameters can be adjusted, ensuring $a \geq 0$, $b \geq 0$ and $a + b = 1$.

We combine our opacity maps with these automatic transfer functions by setting $a = 0.01$, $b = 0.99$, $o_{min} = 0.03$ and $o_{max} = 1$.

Fig. 4 shows the results using this combination on a CT scan. If the fat layer on the back is selected, the automatic transfer function hides the internal organs and the skeleton. If the intestine is selected instead, the transfer function automatically adapts to emphasize other internal organs while the intestine remains clearly highlighted by the opacity map. When the scapula is selected, it also becomes clearly highlighted, but in this case, the bones and other dense tissues stand out over the previous internal organs.

The proposed configuration also adapts to the different properties of the dataset, for instance, a CT scan with much lower contrast in Fig. 5. When the leg muscles, the kidney or the stent of the aorta are highlighted, the internal soft tissues or skeleton are properly provided as main contextual structures. Other results are shown in Fig. 1.

4.3. Interactive expansion

The expansion process of the opacity map can be performed automatically from the user-selected 2D point. However, enabling a direct control over the expansion process can be in fact regarded as an additional exploration tool that is specially useful when exploring features with a complex topology. Additionally, this active control allows the user to prevent the expansion from invading neighboring structures, which may happen if their density ranges overlap.

Therefore, we let the user to freely perform expansion steps at will, with the possibility to manipulate the 3D rendering at any intermediate point. This allows to better visualize the intricacies of complex topology, such as the vessels in a vascular network shown in Fig. 2 (rightmost column).

5. Results and discussion

We have implemented our method of parallel generation of the opacity maps using OpenCL and applied it using a variety of medical datasets. Along with the results shown throughout this work, we have applied our method to several datasets with different contrast and noise conditions to test the robustness of our algorithm.

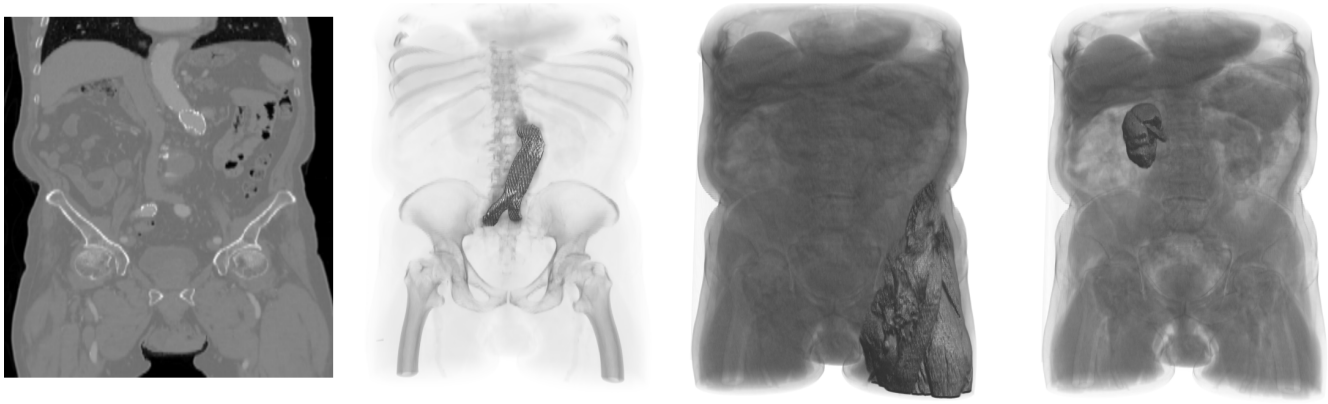

Figure 5: *In the presence of data with low contrast (leftmost), our method is also able to provide a clearly highlighted feature with contextual information. When the stent of the aorta is selected (left-center), the bone structures are revealed. If the leg muscles or the kidney are selected instead, the bones and stent are hidden, and other soft tissues are revealed.*

In this section we analyze and discuss the properties of our method to evaluate its behavior under different contrast and noise conditions. We have also conducted a performance test to measure the efficiency of our parallel implementation.

5.1. Method robustness

The main motivation for developing our method is to provide the user with an intuitive, parameter-free tool to explore medical datasets, and thus it should provide a proper response independently of the noise and contrast conditions of the data. For this reason, we have fixed the λ value for all the tested datasets (as explained in Sec. 3).

The datasets shown in Fig. 2 and Fig. 4 exhibit high contrast and a low amount of noise on the bone and vascular tissues. For these regions our method provides a very clear visualization of the selected features, which was expected since this is the ideal case for segmentation algorithms. The same outcome is observed for areas where low contrast and low amount of noise, such as the kidney and leg muscle areas shown in Fig. 5.

For datasets with a higher amount of noise and high contrast, such as the abdominal organs of the CT scan shown in Fig. 2, we also obtain a proper distinction of the desired features. For instance, we are able to clearly differentiate several parts of the kidney as shown in Fig. 6.

We find that our method fails for input datasets with low contrast and moderate or high noise. For instance we are unable to produce a clear view of the vascular network of the lungs in the dataset shown in Fig. 7. We believe that under these conditions, a region growing approach alone is not enough to capture features of interest, and other approaches should be explored.

We also observe malfunction of the method when the selected seed voxel belongs to a boundary between features, as seen in Fig. 8. We estimate the noise of the data through the computation of the standard deviation on the neighborhood around the seed voxel, thus we assume this neighborhood to belong to the same feature. This leads to a poor noise estimation for seed voxels located on boundary regions. A possible workaround for this issue was proposed by Huang and Ma [HM03], but we want to further explore solutions avoiding manual parameter tuning.

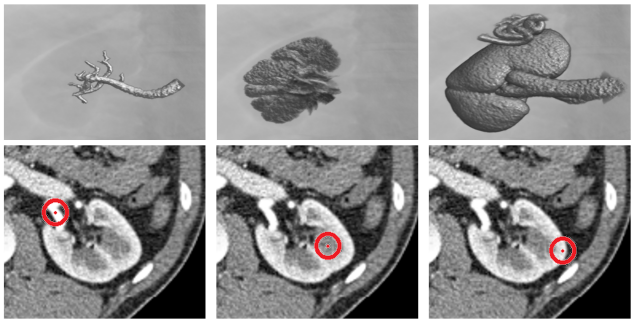

Figure 6: *From left to right, the renal vessels, the medulla and the cortex of the kidney can be highlighted by selecting the different structures in the 2D slices, shown below each case.*

5.2. Performance

We have evaluated the performance of our algorithm using an Intel Core i5-3570 machine with 8 GB RAM equipped with an AMD Radeon R9-270X.

Due to the simple GPU scheduling mechanism proposed in Sec. 3.1, the performance of the algorithm is independent of the input dataset, and the computational burden of an expansion iteration depends only on the size of the reached cubic region.

Fig. 9 shows the computation time for an expansion iteration w.r.t. the cubic region size. Our implementation is able to provide

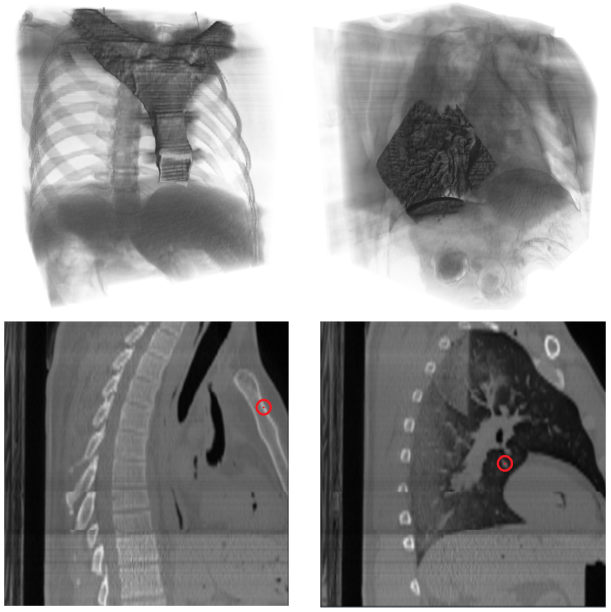

Figure 7: *In the presence of noise in data with low contrast, our algorithm is unable to capture the desired feature. The breastbone (left) is captured, but its boundary becomes blurred. In the case of the bronchi (right), neighboring structures are rapidly invaded during the expansion process.*

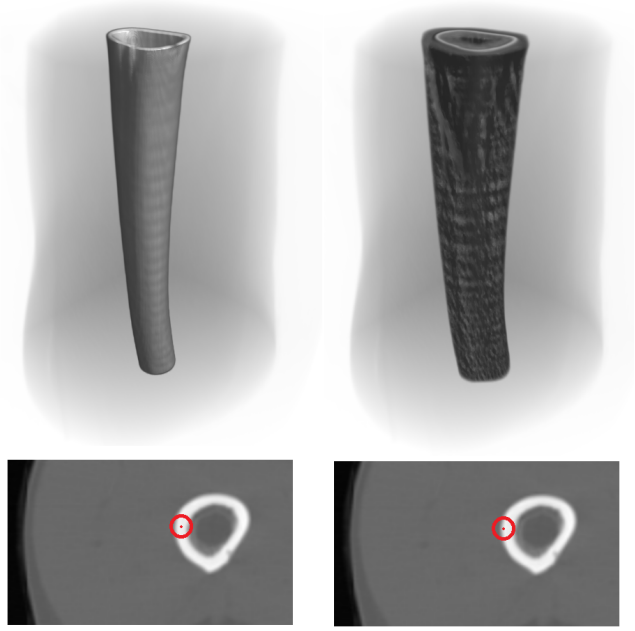

Figure 8: *Our method assumes an input point inside the feature of interest (left). If the selected point belongs to the boundary between structures, the change of structure is regarded as data noise and the method fails to capture the desired feature (right).*

real-time responses for small and medium size features and interactive responses for large features covering several million of voxels. For reference: the kidney in Fig. 2, bounded by a cubic region with 7.4 million voxels, required a total expansion time of 64 ms; the vascular network in Fig. 2, bounded by a cubic region with 89.3 million voxels, required a total expansion time of 714 ms; the stent in Fig. 5, bounded by a cubic region with 7.1 million voxels, required a total expansion time of 62 ms; the leg muscles in Fig. 5, bounded by a cubic region with 52.7 million voxels, required a total expansion time of 375 ms.

We have also measured the percentage of launched threads actually performing useful computation for all our examples and, although this amount varies depending on the topology of the selected feature, it ranges between a 0.7% and a 12%. This is due to the fact that the simplistic approach used for thread launching does not cope with the sparse nature of the expansion approach, since most of the launched threads perform computation on voxels that are not on the current frontier of the expanding feature and thus none of their neighbors had its opacity value updated.

Although our current implementation provides interactive responses, the performance would be greatly improved by using a sparse-aware scheduling approach ([RLAM15, Sæt13]).

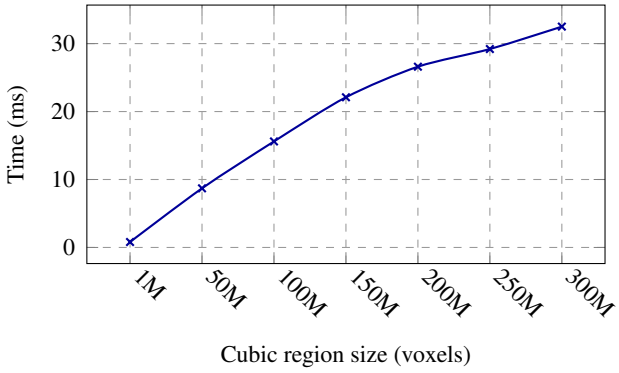

Figure 9: *Time required to perform an expansion iteration depending on the current size of the cubic region.*

6. Conclusion and future work

We have proposed the spatial opacity maps as a novel tool for interactive exploration of medical volume data. Our approach follows a parameter-free interaction paradigm and proves robust for many types of input datasets.

Although the spatial opacity maps as standalone tool are able to provide meaningful visualizations of the desired features, the display of the contextual information is rather poor, but we have shown

that improved visualizations can be achieved by combining our approach with other tools for medical image exploration. We have provided a method to seamlessly integrate the opacity maps into the standard volume rendering pipeline, and thus their combination with existing methods is achieved with almost no extra effort.

We have also proposed a simple automatic transfer function generation procedure, achieving a notable improvement in the visualization of contextual data, and we think that further combinations will lead to better visualizations while maintaining the parameter-free paradigm. For instance, we strongly believe that our method could be easily combined with the importance-driven volume rendering approach [VKG04] to serve as input for the latter and to improve the focus+context contrast, and with other automatic function generation methods, such as the one proposed by Zhou and Takatsuka [ZT09]. We would like to explore these and other combinations as future work.

The experiments carried out show that our implementation delivers interactive responses, although we will address the improvement of the GPU scheduling mechanism to account for the sparse nature of the computation in a way similar to the blocking method proposed in [RLAM15]. We believe that this will enable response times close to real-time even for large features.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. This work is supported by the University of Granada, under the "Formación de Profesorado Universitario, Plan Propio de Investigación 2012" program. This work is also supported by the project TIN2014-60956-R of the Spanish Ministry of Economy and Competitiveness with FEDER funds. The datasets used in this work were obtained from The Volume Library (lgdv.cs.fau.de/External/vollib/) and the Osirix repository (osirix-viewer.com/datasets/).

References

[AB94] ADAMS R., BISCHOF L.: Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 16*, 6 (1994), 641–647. 2

[BGKG06] BRUCKNER S., GRIMM S., KANITSAR A., GROLLER M. E.: Illustrative context-preserving exploration of volume data. *Visualization and Computer Graphics, IEEE Transactions on 12*, 6 (2006), 1559–1569. 2

[BKKG08] BRUCKNER S., KOHLMANN P., KANITSAR A., GROLLER M. E.: Integrating volume visualization techniques into medical applications. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on* (2008), IEEE, pp. 820–823. 1

[BRGIG*14] BALSA RODRÍGUEZ M., GOBBETTI E., IGLESIAS GUITIÁN J., MAKHINYA M., MARTON F., PAJAROLA R., SUTER S. K.: State-of-the-art in compressed gpu-based direct volume rendering. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 77–100. 1

[CM11] CORREA C. D., MA K.-L.: Visibility histograms and visibility-driven transfer functions. *Visualization and Computer Graphics, IEEE Transactions on 17*, 2 (2011), 192–204. 2

[CSC06] CORREA C. D., SILVER D., CHEN M.: Feature aligned volume manipulation for illustration and visualization. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (2006), 1069–1076. 2

[EHK*06] ENGEL K., HADWIGER M., KNISS J., REZK-SALAMA C., WEISKOPF D.: *Real-time volume graphics*. CRC Press, 2006. 1, 3

[HM03] HUANG R., MA K.-L.: Rgvis: Region growing based techniques for volume visualization. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on* (2003), IEEE, pp. 355–363. 2, 6

[KBKG08] KOHLMANN P., BRUCKNER S., KANITSAR A., GRÖLLER M. E.: Livesync++: Enhancements of an interaction metaphor. In *Proceedings of Graphics Interface 2008* (2008), Canadian Information Processing Society, pp. 81–88. 2

[KBKK07] KOHLMANN P., BRUCKNER S., KANITSAR A., KANITSAR A.: Livesync: Deformed viewing spheres for knowledge-based navigation. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (2007), 1544–1551. 1, 2

[KKH01] KNISS J., KINDLMANN G., HANSEN C.: Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the conference on Visualization'01* (2001), IEEE Computer Society, pp. 255–262. 2

[RBS05] ROETTGER S., BAUER M., STAMMINGER M.: Spatialized transfer functions. In *EuroVis* (2005), Citeseer, pp. 271–278. 2

[RLAM15] RODRÍGUEZ A., LEÓN A., ARROYO G., MANTAS J. M.: Sp-chainmail: a gpu-based sparse parallel chainmail algorithm for deforming medical volumes. *The Journal of Supercomputing 71*, 9 (2015), 3482–3499. 7, 8

[Sæt13] SÆTRA M. L.: Shallow water simulation on gpus for sparse domains. In *Numerical Mathematics and Advanced Applications 2011*. Springer, 2013, pp. 673–680. doi:10.1007/978-3-642-33134-3_71. 7

[SHN03] SHERBONDY A., HOUSTON M., NAPEL S.: Fast volume segmentation with simultaneous visualization using programmable graphics hardware. In *Visualization, 2003. VIS 2003. IEEE* (2003), IEEE, pp. 171–176. 2

[VKG04] VIOLA I., KANITSAR A., GROLLER M. E.: Importance-driven volume rendering. In *Proceedings of the conference on Visualization'04* (2004), IEEE Computer Society, pp. 139–146. 2, 8

[WVFH12] WIEBEL A., VOS F. M., FOERSTER D., HEGE H.-C.: Wysiwyp: what you see is what you pick. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (2012), 2236–2244. 1

[ZT09] ZHOU J., TAKATSUKA M.: Automatic transfer function generation using contour tree controlled residue flow model and color harmonics. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 1481–1488. 8