Portal Extraction Based on an Opening Labeling for Ray Tracing

Laurent Noël(✉) and Venceslas Biri

LIGM, Université Paris Est, Creteil, France
Laurent.Noel@u-pem.fr

Abstract. Rendering photo-realistic images from a 3D scene description remains a challenging problem when processing complex geometry exposing many occlusions. In that case, simulating light propagation requires hours to produce a correct image. An opening map can be used to extract information from the geometry of the empty space of a scene, where light travels. It describes local thickness and allows to identify narrow regions that are difficult to traverse with ray tracing. We propose a new method to extract portals in order to improve rendering algorithms based on ray tracing. This method is based on the opening map, which is used to define a labeling of the empty space. Then portals - 2D surfaces embedded in empty space - are extracted from labeled regions. We demonstrate that those portals can be sampled in order to explore the scene efficiently with ray tracing.

Keywords: Ray tracing · Opening map · Labeling · Portals

1 Introduction

Light transport simulation refers to the process of rendering photo-realistic images from a 3D scene description. It remains one of the most challenging problem in computer graphics due to the complexity of scenes (mixture of non-homogeneous materials, complex geometry, non-uniform illumination).

Most state of the art methods rely on Path-Tracing and Monte-Carlo estimation to perform the simulation [1,2].

The color of each pixel can be expressed as an integral over the space of paths [3]. To compute this integral, many random paths traversing the pixel are sampled. A path is computed recursively using ray tracing. A first point x is sampled by tracing a random ray starting at the camera and traversing the pixel. Then the path is extended by sampling a random direction on the hemisphere of its last point. After many reflections, the path eventually reaches a light source and its energy can be computed. All these paths are used as samples for Monte-Carlo integration in order to estimate the integral expressing the color of the pixel. The stochastic aspect of this kind of simulation produces errors in the resulting image which appear as noise. Increasing the number of paths attenuates this noise and the result converges to a noise-free image. To improve the convergence rate of the simulation, one must choose a sampling

J.A. Benediktsson et al. (Eds.): ISMM 2015, LNCS 9082, pp. 27–38, 2015.
DOI: 10.1007/978-3-319-18720-4_3

strategy that samples more frequently paths carrying high energy. Building such a strategy can be challenging when dealing with complex geometry exposing many occluders. In that case, most of the sampled path does not reach a light source and carry no energy.

Digital Geometry and Mathematical Morphology [7] propose many tools to analyze the shape of discretized objects. In particular, for light transport simulation, the shape of the empty space (medium in which the light travels) can be analized to develop efficient rendering algorithms. Surprisingly, just few attempts have been made to use such tools in the rendering community [4,5,6].

In this paper we propose to investigate the use of the opening function [8] to analyze the shape of the empty space of a scene. This function expresses the local thickness of empty space surrounding each point and can be used to identify separations between large and narrow regions of the space. Our goal is to build a labeling on a voxelization of the empty space such that two neighbor regions, having a high difference in their thickness, are separated. This labeling allows us to extract portals, which are surfaces that separate different regions. Those portals can be sampled by a rendering algorithm in order to efficiently explore narrow regions that lead to light sources. We first analyze a simple labeling only based on the opening map of the empty space. Then we point out that it produces many regions that are false positives for our problem. We propose an other labeling built upon the propagation performed by the algorithm that compute the opening map. We show that this labeling defines better regions and enables the extraction of meaningful portals embedded in empty space. We present our results on different scenes and explore the limits of our method on the application to ray tracing.

2 Motivation

We first present a simple scene commonly used in computer graphics for algorithms that deal with complex occlusions and explain why the opening map can be of interest to us.

The Ajar Door scene (figure 1) is composed of two large rooms separated by a door slightly opened. For a configuration where a light source is placed in one room and the camera in the other room, the illumination becomes hard to sample efficiently. Indeed, blindly tracing random paths starting at the camera will end up with a lot of paths that does not reach the room containing the light source.

The opening map of the empty space E tells us for each voxel $\mathbf{x}$ the radius of the maximal ball inscribed in E that contains $\mathbf{x}$ (figure 2). We observe that the passage highlighted in red is characterized by lower values for this opening map than the two large rooms.

Based on this observation, we present a new labeling method that partitions the empty space in regions based on the opening map. More importantly we extract *portals* which are surfaces that separate a region from its neighbor regions. By sampling those portals we are able to trace rays that leave a given region in order to efficiently explore the empty space.

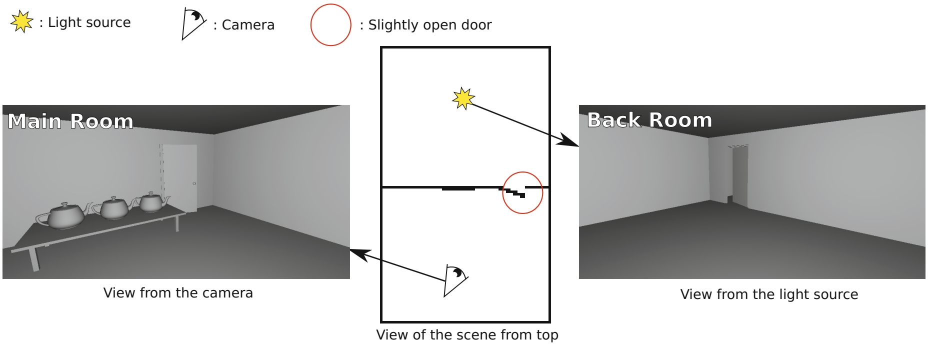

Fig. 1. The Ajar Door scene. The center image is a cut along the vertical axis of a voxelization of the scene. White pixels represent the empty space and black pixels represent the scene. The red circle encloses the small passage that rays must take to go from one room to the other.

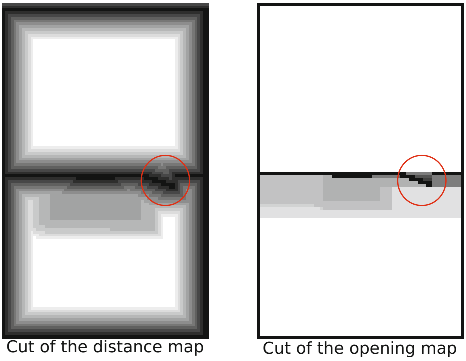

Fig. 2. By dilating the distance map (left), we obtain the opening map (right) of the empty space. This map enhances narrow regions which are difficult to explore by tracing random rays.

3 Opening Based Labeling

We first define the opening map of the empty space and the labeling of that map that allows us to identify connected regions of constant opening. We show on an example scene that this method is not efficient enough to get meaningful regions for our problem. In section 3.3 we propose another labeling method based on the same core of ideas. We show in section 4 that it defines better regions for our application to ray tracing.

3.1 Mathematical Background

Let $G = [\![0, w[\![\times [\![0, h[\![\times [\![0, d[\![\subset \mathbb{Z}^3$ a finite 3D grid of voxels of resolution (w, h, d).

Let $S \subset G$ be the set of voxels corresponding to the voxelized 3D scene. We use the 26-connectivity to define the adjacency of voxels of S.

The discrete empty space $E = G \setminus S$ is the complementary of S in the grid G. We use the 6-connectivity to define the adjacency of voxels of E.

Definition 1. *If $\mathbf{x} = (x_1, x_2, x_3)$, $\mathbf{y} = (y_1, y_2, y_3) \in \mathbb{Z}^3$ are two voxels, we denote by $d_\infty(\mathbf{x}, \mathbf{y}) = \max(|x_1 - y_1|, |x_2 - y_2|, |x_3 - y_3|)$ the 26-distance between the voxels $\mathbf{x}$ and $\mathbf{y}$.*

Definition 2. *The distance map $\mathcal{D}_\infty : G \rightarrow \mathbb{Z}^+$ is the function defined by:*

$$\mathcal{D}_\infty(\mathbf{x}) = \min_{\mathbf{y} \in S} d_\infty(\mathbf{x}, \mathbf{y}) \tag{1}$$

Note that we have $\mathbf{x} \in S \Leftrightarrow \mathcal{D}_\infty(\mathbf{x}) = 0$.

Definition 3. *The maximal ball $\mathcal{B}_\infty(\mathbf{x})$ centered in $\mathbf{x}$ is the set:*

$$\mathcal{B}_\infty(\mathbf{x}) = \{\mathbf{y} \in G \mid d_\infty(\mathbf{x}, \mathbf{y}) < \mathcal{D}_\infty(\mathbf{x})\} \tag{2}$$

Definition 4. *The opening map $\Omega_\infty : G \rightarrow \mathbb{Z}^+$ is the function defined by:*

$$\Omega_\infty(\mathbf{x}) = \max_{\mathbf{x} \in \mathcal{B}_\infty(\mathbf{y})} \mathcal{D}_\infty(\mathbf{y}) \tag{3}$$

$\Omega_\infty(\mathbf{x})$ is the opening of the voxel $\mathbf{x}$.

For $\mathbf{x} \in E$, $\Omega_\infty(\mathbf{x})$ is the radius of the largest maximal ball of E containing the voxel $\mathbf{x}$. It gives us an information on the local thickness of empty space around $\mathbf{x}$. By looking at the variation of the opening map, we can identify narrow regions that connect large regions.

3.2 The Opening Labeling

Definition 5. *The opening region $\mathcal{R}(\mathbf{x}) \subset E$ of the voxel $\mathbf{x} \in E$ is the maximal 6-connected set of voxels containing $\mathbf{x}$ such that $\forall \mathbf{y} \in \mathcal{R}(\mathbf{x})$, $\Omega_\infty(\mathbf{x}) = \Omega_\infty(\mathbf{y})$.*

The region $\mathcal{R}(\mathbf{x})$ is a connected subset of E of constant opening. We denote by R_Ω the set of all opening regions.

Definition 6. *Let $L : R_\Omega \rightarrow \{1, ..., |R_\Omega|\}$ be a function that map each opening region to a label. The opening labeling of E for L is the map $L_\Omega : E \rightarrow \{1, ..., |R_\Omega|\}$ defined by:*

$$L_\Omega(\mathbf{x}) = L(\mathcal{R}(\mathbf{x})) \tag{4}$$

Limitations. We aim at tracing rays that travel the scene in order to reach regions containing a light source. In particular we want to be able to traverse efficiently narrow regions, such that sampled paths do not remain stuck in a region containing no light. As shown in figure 3, the labeling L_Ω produces false positive regions for our purpose. Regions like A are a problem because they lead directly on a wall. Ideally we want region A to be part of region B. Our experiments show that all voxels of A are contained in maximal balls of region B. We use this information in section 3.3 to define our new labeling.

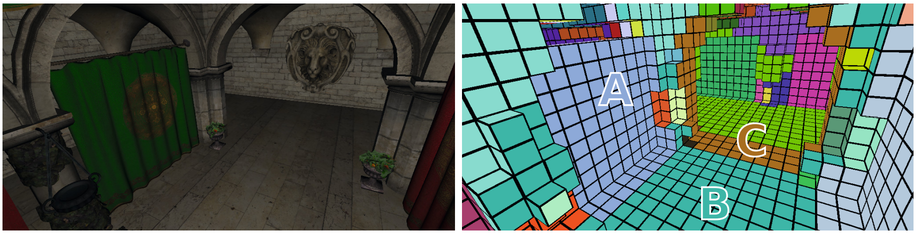

Fig. 3. The opening labeling of a scene. Region A is a false positive because going from B to A does not grant access to other regions of the scene and leads directly on a wall. Region C is interesting because it represents a passage to enter a corridor. It represents a hole in the 3D scene.

Finding a good merging criterion that meets our expectation would be difficult and would likely depend on parameters that vary from scene to scene. Instead of merging regions, we decided to develop another labeling method based on the maximal balls used to compute the opening map.

3.3 The Opening Forest Labeling

As we noted in section 3.2, maximal balls give us information on the similarity between two neighbor regions. When the union of maximal balls of a region A covers entirely a neighbor region B, it means that it is probably easy to access B from A by tracing random rays.

The computation of the opening map is performed by successive dilations of the distance map on maximal balls of E. This process can be seen as a propagation that explores E by following maximal balls in decreasing order of their radius. Such propagation implicitly defines a forest that respects the properties of definition 7.

Definition 7. *An opening forest for E is a map $\Omega_f : E \rightarrow E$ such that if $\Omega_\infty(\mathbf{x}) = \mathcal{D}_\infty(\mathbf{x})$ then $\Omega_f(\mathbf{x}) = \mathbf{x}$. In that case $\mathbf{x}$ is a root of Ω_f. If $\Omega_\infty(\mathbf{x}) \neq \mathcal{D}_\infty(\mathbf{x})$ then $\Omega_f(\mathbf{x})$ is the center of a maximal ball of radius $\Omega_\infty(\mathbf{x})$ containing $\mathbf{x}$. In that case $\Omega_f(\mathbf{x})$ is the parent of $\mathbf{x}$.*

There exist many opening forest for an unique opening map since a voxel $\mathbf{x}$ can be contained in several maximal balls of radius $\Omega_\infty(\mathbf{x})$. Our goal is to build the regions of our new labeling from the trees of an opening forest. Since we want our regions to be 6-connected, we need to use at least an opening forest such that each tree is 6-connected.

Algorithm 1 computes both the opening map and an opening forest Ω_f that meets this criterion. Using a 26-distance map allows to compute the opening map by performing six scans of the 3D grid in each of the six directions (north, south, est, west, top and bottom). Each scan performs an independent dilation of each line of the 26-distance map and can be easily implemented in parallel.

Algorithm 1. Calculate the opening map and the opening forest

```
Input: The distance map D∞
Output: The opening map Ω∞ and the opening forest Ωf
  Ω∞ ← D∞
  for (i, j, k) ∈ ⟦0, w⟦×⟦0, h⟦×⟦0, d⟦ do
      Ωf(i, j, k) ← (i, j, k)
  end for
  for (j, k) ∈ ⟦0, h⟦×⟦0, d⟦ do
      dilateLine(Ω∞([0...w[, j, k), Ωf([0...w[, j, k))
      dilateLine(Ω∞(]w...0], j, k), Ωf(]w...0], j, k))
  end for
  for (i, k) ∈ ⟦0, w⟦×⟦0, d⟦ do
      dilateLine(Ω∞(i, [0...h[, k), Ωf(i, [0...h[, k))
      dilateLine((Ω∞(i, ]h...0], k), Ωf(i, ]h...0], k))
  end for
  for (i, j) ∈ ⟦0, w⟦×⟦0, h⟦ do
      dilateLine(Ω∞(i, j, [0...k[), Ωf(i, j, [0...k[))
      dilateLine(Ω∞(i, j, ]k...0]), Ωf(i, j, ]k...0]))
  end for                         ▷ Function dilateLine is defined in algorithm 2
```

We use the opening forest to build a new labeling of the empty space E. First we define the notion of *region root*.

Definition 8. *Let* $\mathbf{x} \in E$ *such that* $\Omega_f(\mathbf{x}) = \mathbf{x}$. *The region root* $\mathcal{R}_{root}(\mathbf{x}) \subset E$ *is the maximal 6-connected set of voxels containing* $\mathbf{x}$ *such that* $\forall \mathbf{y} \in \mathcal{R}_{root}(\mathbf{x})$ *we have* $\Omega_f(\mathbf{y}) = \mathbf{y}$ *and* $\Omega_\infty(\mathbf{x}) = \Omega_\infty(\mathbf{y})$.

All voxels of a region root have the same opening and are roots of the opening forest. The figure 4 illustrates different regions roots of a scene.

Let R_{root} be the set of all region roots of E. The *opening forest labeling* is built from the opening forest and region roots.

Definition 9. *Let* $L : R_{root} \to \{1, ..., |R_{root}|\}$ *be a function that maps each region root to a label. The opening forest label* $L_f(\mathbf{x})$ *of the voxel* $\mathbf{x}$ *is recursively defined by:*

Algorithm 2. Dilate a line of N maximal balls represented by their radius and center

```
Input: R: a line of N radius value, C: a line of N voxel centers
Output: R and C are dilated according to the radius values of R
  function dilateLine(R[N], C[N])
    maxballQueue ← EmptyQueue
    for i ← 0 to N − 1 do
      currentBall ← { index: i, radius: R[i], center: C[i], end: i + R[i] }
      if currentBall.radius = 0 then
        maxballQueue ← EmptyQueue
      else if maxBallQueue not empty and i = maxballQueue.front().end then
        maxballQueue.pop_front()
      end if
      if maxBallQueue is empty then
        maxballQueue.push_back(currentBall)
      else
        if currentBall.radius ≥ maxballQueue.front().radius then
          maxballQueue ← EmptyQueue
          maxballQueue.push_back(currentBall)
        else
          while currentBall.radius ≥maxballQueue.back().radius do
            maxballQueue.pop_back()
          end while
          if maxballQueue.back().end < currentBall.end then
            maxballQueue.push_back(currentBall)
          end if
        end if
      end if
      R[i] ← maxballQueue.front().radius
      C[i] ← maxballQueue.front().center
    end for
  end function
```

$$\begin{cases} L_f(\mathbf{x}) = L(\mathcal{R}_{root}(\mathbf{x})) \;\; \textit{if}\; \Omega_f(\mathbf{x}) = \mathbf{x} \\ L_f(\mathbf{x}) = L_f(\Omega_f(\mathbf{x})) \;\; \textit{otherwise.} \end{cases} \tag{5}$$

The opening forest labeling is a propagation of the label of a region root to the trees rooted in that region.

The opening forest region $\mathcal{R}_f(\mathbf{x})$ of a voxel $\mathbf{x}$ is the maximal set of voxels such that $\forall \mathbf{y} \in \mathcal{R}_f(\mathbf{x}), L_f(\mathbf{x}) = L_f(\mathbf{y})$. This set is connected.

We denote by R_f the set of all opening forest regions of E.

3.4 Opening Portals

We now define *opening portals*, which are 2D-surfaces that separate opening forest regions. Since opening portals are not composed of voxels, we need to define them as set of 2-faces, which are elements of the *voxel complex framework* [9].

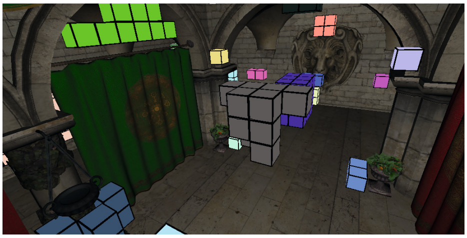

Fig. 4. Some region roots of a scene. Each region is a connected set of centered voxels of the same opening.

Let $X, Y \in R_f$ be two opening forest regions of E. We say that X and Y are neighbor regions if $\exists \mathbf{x} \in X, \exists \mathbf{y} \in Y$ such that $\mathbf{x}$ and $\mathbf{y}$ are 6-neighbors.

Definition 10. *Let $\mathbf{x} \in \mathbb{Z}^3$ be a voxel. The 3-face associated to $\mathbf{x} = (x_1, x_2, x_3)$ is the set $F(\mathbf{x}) = \{(x_1, x_2, x_3), (x_1+1, x_2, x_3), (x_1, x_2+1, x_3), (x_1, x_2, x_3+1), (x_1+1, x_2+1, x_3), (x_1, x_2+1, x_3+1), (x_1+1, x_2, x_3+1), (x_1+1, x_2+1, x_3+1)\}$.*

The 3-face of a voxel corresponds to its eight corners. This notion allows to define the two dimensional face separating two neighbor voxels.

Definition 11. *Let $\mathbf{x}$, $\mathbf{y}$ be two 6-adjacent voxels. The set $F_s(\mathbf{x}, \mathbf{y}) = F(\mathbf{x}) \cap F(\mathbf{y})$ is the 2-face separating $\mathbf{x}$ and $\mathbf{y}$. It is composed of four points.*

Definition 12. *Let $X, Y \in R_f$ be two neighbor opening forest regions of E. The opening portal $P_{X,Y}$ separating X and Y is the set defined by:*

$$f \in P_{X,Y} \Leftrightarrow \exists \mathbf{x} \in X, \exists \mathbf{y} \in Y \text{ such that } f = F_s(\mathbf{x}, \mathbf{y}) \tag{6}$$

We can build rays going from X to Y by sampling points on the 2-faces of $P_{X,Y}$.

4 Results and Discussion

We present the result of our *opening forest labeling* on different scenes and for different resolutions of the voxelization. To illustrate the regions, we display the voxelization of the scene such that the color of a face f of each voxel $\mathbf{x} \in S$ depends on the label of the empty-space voxel $\mathbf{y} \in E$ which is adjacent to $\mathbf{x}$ for the face f ($f = F_s(\mathbf{x}, \mathbf{y})$). We also expose opening portals separating different regions and we compare a local ray sampling strategy (uniform sampling of directions on the hemisphere of the origin point) to the a strategy that sample rays passing through our opening portals.

The application of our sampling strategy to a rendering algorithm is left for a future article. We discuss it in more depth in the conclusion.

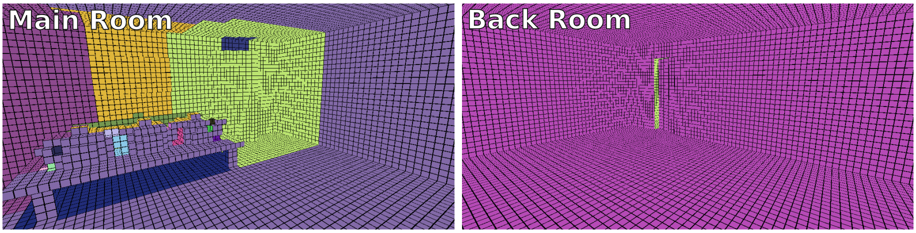

Fig. 5. Our opening forest labeling for the Ajar Door scene for a grid resolution of (72, 118, 28). We observe that the back room is composed of one large region (right) separated from the main room by the narrow door entrance, as expected. The main room (left) is split in several regions allowing to travel the scene efficiently by sampling portals (see figure 6).

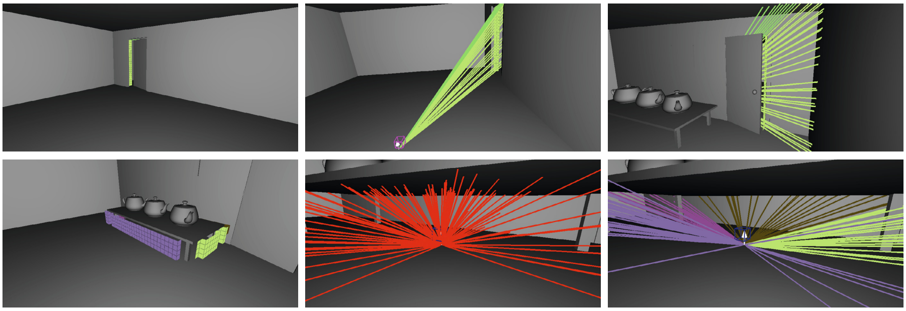

Fig. 6. *Top-left*: the green portal separates the back room from the main room. *Top-center*: 128 rays are sampled through the portal, starting at a point of the back room. *Top-right*: we observe that all rays reach the main room. *Bottom-left*: portals separating the region under the table from neighbor regions. *Bottom-center*: rays sampled with the local strategy. Many of them hit the back of the table. *Bottom-right*: with our strategy, all sampled rays leave the region.

Ajar Door Scene. Figures 5 and 6 demonstrate the application of our method to the Ajar door scene presented in section 2. Using our portals, we observe that we are able to sample rays efficiently to travel from region to region. In particular it makes it easy to pass through the door entrance or to leave the region under the table.

Sibenik Scene. Figure 7 and 8 illustrate our method on the Sibenik scene, which represents a church. This scene is less affected by occlusions than the two others. However, we demonstrate that we can use our sampling strategy to pass through a specific portal in order to reach a particular area of the scene. Being able to achieve this is useful to reach a specific light source by performing less reflections.

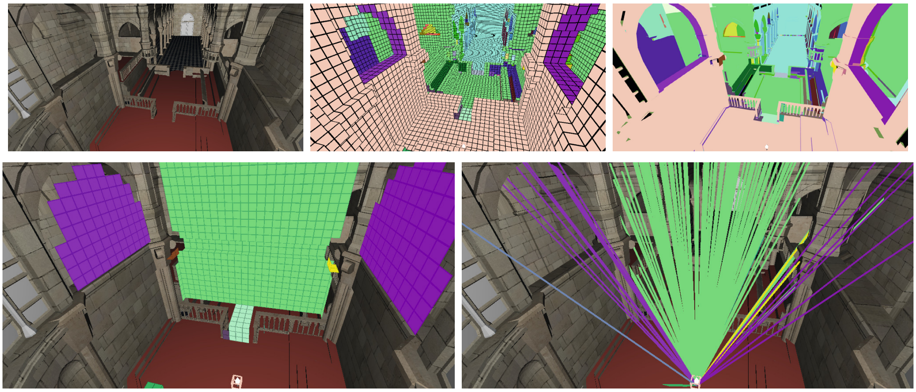

Fig. 7. Top row shows the labeling of the Sibenik scene for a grid resolution of (138, 104, 58). The top-right image is the labeling shown directly on the surfaces of the scene. The bottom-left image exposes three major portals of a region. The bottom-right image demonstrates the sampling of rays leaving that region by selecting portals based on their area (more rays are sampled on large portals).

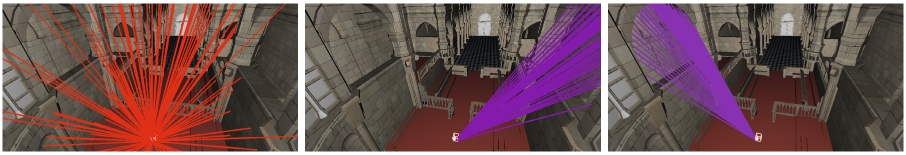

Fig. 8. Left image shows the local sampling of rays which is not efficient to pass through the left and right portal. Using our method, it is straightforward to sample rays through a specific portal as pointed out in the middle and right image.

Sponza Scene. Figure 9 and 10 show results of our method on the Sponza scene. This scene is composed of several corridors occluded from the main part of the scene by drapes. Starting from a corridor, reaching the main part is hard due to narrow exits. Our sampling strategy enables to do it efficiently. Figure 10 demonstrates the robustness of our method regarding the resolution of the voxel grid. Opening portals remain stable and fit better to the scene as we increase the resolution.

Limitations. Our new labeling still produces neighbor regions with close opening and highly connected by their maximal balls. Such regions are separated because their region roots are not connected. However, these regions exposes better coherency with their opening: a region having a region root with high opening has a high volume and allows to access all of its narrow neighbors using small portals.

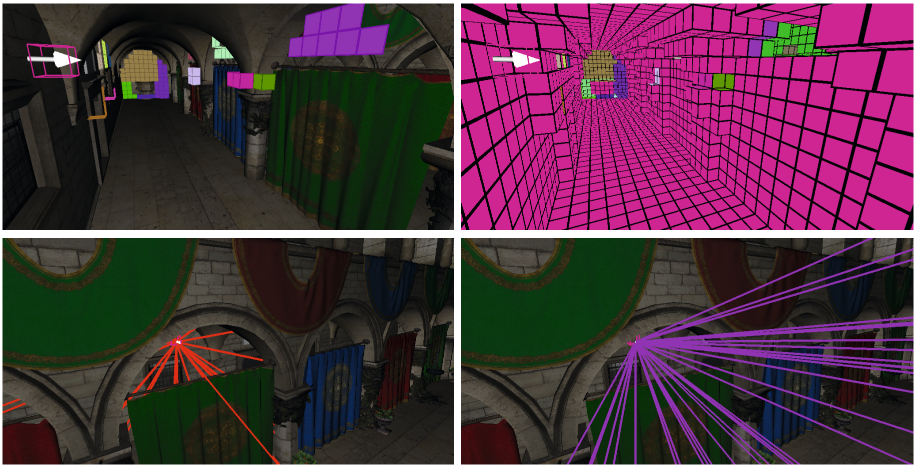

Fig. 9. Top row: portals and labeling of a corridor of the Sponza scene for a resolution of (118, 50, 72). Bottom row: comparison of local sampling (left) and sampling through a chosen portal (right) which allow to leave efficiently the corridor.

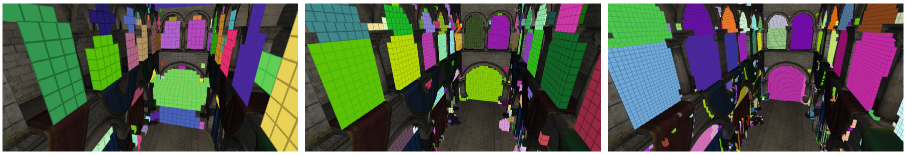

Fig. 10. Illustration of portals when we increase the resolution of the voxel grid. We observe that the separation between regions becomes more precise as we increase the resolution of the voxel grid.

A more important issue for our sampling procedure is the presence of highly concave regions. In such a region and without more information, we could sample a portal which is not visible from the origin of the ray. To apply our method to light transport algorithms, it might be unavoidable to use a convex decomposition algorithm or to improve our algorithm to guaranty the convexity of regions.

5 Conclusion and Future Works

We presented a new labeling method to partition the empty space of a 3D scene according to opening and maximal balls. This labeling is used to extract portals between 3D regions that can be efficiently sampled to go from one region to another when tracing rays in the 3D scene.

On a future work, we would want to explore other labeling methods in order to compare them with our on the application to ray tracing. As an example, our region roots can be used as seeds for a watershed propagation on the opening map. More importantly, we aim at enhancing our method to obtain approximately convex regions. As mentioned in section 4, not dealing with convex regions can be a problem because sampling points on portals does not guaranty their visibility from a point of the region.

The purpose of our method is to use our portals to improve sampling strategies of light transport algorithms. The idea is to select portals to sample in order to reach light sources efficiently. Our future work will be focused on this application and the investigation of methods to manage discontinuities that can be introduced by partitioning the scene in discrete regions.

References

1. Georgiev, I., Křivánek, J., Davidovič, T., Slusallek, P.: Light transport simulation with vertex connection and merging. ACM Trans. Graph (SIGGRAPH Asia 2012) 31, 6 (2012)
2. J. Vorba, O. Karlík, M. Šik, T. Ritschel, Jaroslav Křivánek: On-line Learning of Parametric Mixture Models for Light Transport Simulation. ACM Trans. Graph., SIGGRAPH (2014)
3. E. Veach: Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Standford Univeristy (1997)
4. Biri, V., Chaussard, J.: Skeleton based importance sampling for path tracing. In: Proceedings of Eurographics (2012)
5. J. Chaussard, L. Noël, V. Biri, M. Couprie: A 3D Curvilinear Skeletonization Algorithm with Application to Path Tracing Discrete Geometry for Computer Imagery, Proceedings (2013)
6. Laurent Noël, Venceslas Biri: Real-Time Global Illumination using Topological Information. Journal on Computing (2014)
7. Azriel Rosenfeld, Reinhard Klette: Digital Geometry Geometric Methods for Digital Picture Analysis. Morgan Kaufmann Series in Computer Graphics and Geometric Modeling (2004)
8. Coeurjolly, D.: Fast and Accurate Approximation of Digital Shape Thickness Distribution in Arbitrary Dimension. Computer Vision and Image Understanding (CVIU) 116(12), 1159–1167 (2012)
9. Bertrand, G., Couprie, M.: Powerful Parallel and Symmetric 3D Thinning Schemes Based on Critical Kernels. Journal of Mathematical Imaging and Vision 48(1), 134–148 (2014)