3D model automatic exploration: Smooth and Intelligent Virtual Camera Control

Zaynab Habibi, Guillaume Caron, El Mustapha Mouaddib

University of Picardie Jules Verne, MIS Laboratory

Abstract. In a 3D dense point clouds model, virtual tour without assistance is a complex and difficult task discouraging users from doing so. The aim of this work is to achieve a virtual navigation support tool. It will help users to perform virtual tour to explore the 3D model. In particular, the tool will allow to guide the camera automatically. We assume that the user is attracted by rich information areas in the model. This important assumption will be modelled by entropy. Secondly, in order to achieve a realistic automatic navigation we must avoid obstacles, ensure a relevant camera orientation during its motion and regulate the visual movement in the produced image. In this paper, we propose a solution to this problem based on a hierarchical algorithm, which combines the main task to be achieved and the realistic constraints. We validate the system on different complex 3D models: lab, urban environment and a cathedral.

1 Introduction

The last few years, 3D virtual applications such as architectural walk-through and videogames are spreading on and receive much attention from researchers in computer animation. However, existing tools for 3D exploration of virtual environments show some inconveniences, particularly for uninitiated people: the difficulty of finding a relevant point of view and the synthetic appearance of the camera movement. Thus, controlling the virtual camera is a task of primary importance in order to transmit relevant information to the user, i.e. to ensure the appropriate visualization within a 3D model.

The aim of the paper is to address the visual content-based automated and relevant control of a virtual camera. Considering a 3D model, a result that we want to automatically obtain is illustrated in Fig. 1. The virtual camera moves in the 3D scene in order to visualize the most important information.

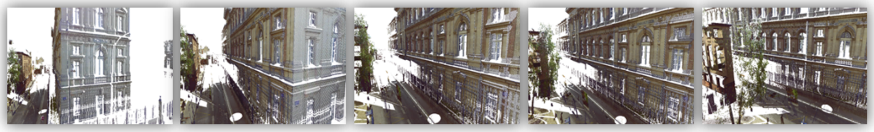


Fig. 1: Ideal camera path maximizing the relevant information in the image.

Existing automatic camera control approaches are either path planning methods or image-based ones.
For path planning methods, preliminarily to the planning stage itself, waypoints must be defined, either manually or automatically, by sampling the camera configuration space. The selection of good viewpoints, that is to visualize the maximum information of the scene, generally exploits information theory-based measures. Indeed, under the hypothesis that users are naturally pulled through areas of high information, the entropy is exploited to characterize the attraction of an area of the 3D model. In a static environment, the viewpoint quality can be evaluated by computing a geometric visual entropy [27]. Polygonal meshes are considered in the latter work. To compute the probability distribution for entropy expression, [27] use the relative area of 3D mesh projected faces over the sphere of directions centered in the viewpoint. The maximum entropy is reached when a certain viewpoint can see all the faces with the same projected area. All candidate views have been manually determined [27] and then validate on the entropy criterion.

In lighting context, illumination entropy has been exploited [15] to find the view that maximize the scene illumination. An adaptive search algorithm uses pixel brightness values (Y tristimulus value of the CIE 1931 standardized color model) to compute the probability distribution of the entropy.
In multi-room 3D model, [1] exploit the geometric visual entropy in order to find the most relevant viewpoints. Then build a path exploring all the cells using a backtracking algorithm.

Indeed, connecting between several viewpoints, whatever the technique to determine them (manually, based or not on geometric [27] or photometric [15] entropy), is the next step for path planning. Other path-planning techniques, as the A-star algorithm [29], the traveling salesman problem [25], roadmaps [20][24], spatial decomposition methods [5][18] and potential field-based approaches [6] [17] can be used for connecting viewpoints. And finally, the virtual camera path is built by the set of these viewpoints using an interpolation method (a spline curve, for instance [2]) to determine the intermediate camera positions.

The second family of virtual camera control is image-based control methods. The camera path is, thus, defined by optimizing a cost function, defined using some image properties. An iterative process is performed to update the six camera degrees of freedom. Within this context, Courty et al. [11] proposed a visual servoing based approach. This approach requires adding constraints on the virtual camera to define the positioning task in the environment. This technique allows, on one hand, following a particular object on the screen, and on the other hand, provides assistance to create a camera path using cinematographic primitives.

To have a wider view of camera control methods, one can refer to the detailed state of the art made by Christie et al. [10].

Previously mentioned path-planning methods exploit the visual entropy in a preliminary process to make use of the selected viewpoints later. These approaches can be accompanied by an obstacle avoidance constraint [11] or not.

Thus, only waypoints are relevant, not the camera motion, visually or dynamically speaking, to reach them (unknown problem for interactive methods because the user can manage speed). The problem comes from the fact that such virtual camera control proceeds following a set of different and, almost, disconnected stages. On the opposite, the virtual visual servoing framework, while not dealing with visual entropy, elegantly merges, in the same solver, all constraints simultaneously. Thus, we propose to address the entropy-based virtual camera control under the virtual visual servoing framework, with new constraints on the camera configuration, described below.

1.1 General strategy

We propose to implement the photometric visual entropy, computed from all image intensities, since it depends on the viewpoint but it is independent of the 3D model type (mesh or point cloud). The challenge is, then, to perform the visual servoing-based virtual camera control using the photometric entropy feature, maximizing the amount of perceived information in the image. Indeed, it will allow the determination of a relevant direction of the camera movement from any viewpoint without any planning stage or the need to connect between several viewpoints independently. Furthermore, to mechanically ensure a relevant camera orientation during its motion, five operational degrees of freedom (three translations, then pan and, then, tilt orientations) are considered for the camera, instead of the six Cartesian degrees of freedom, generally considered in the virtual visual servoing.

The virtual camera must avoid obstacles during its movement. A lot of work in camera control has been tackled to deal with the latter issue such as ray tracing methods [26][8][4], bounding volumes methods [11] [13] and image-based methods [22]. To avoid obstacles, Courty et al. [11] define a function that reaches an infinite value when the distance between the camera and the obstacle is null. We adapted this method that consists of maximizing the distance between the camera and all the closest 3D points of the 3D model.

The maximization of photometric entropy and the maximization of distances between the camera and obstacles are merged into the same problem as a hybrid control law [16]. This allows avoiding sudden and repetitive motion direction changes as soon as an obstacle is detected.

When the photometric entropy and obstacle avoidance based control law converges, the static relevance of the image is locally ensured. The images got during the camera motion have an increasing static relevance along the achieved camera path, as it gets closer to the optimum. This is ensured by the optimization method implemented by virtual visual servoing [11]. However, the dynamic relevance is not guaranteed as jumps between two successive images during the camera motion might be observed. To overcome this discontinuity issue, particularly in the image motion appearance, we propose the first optical flow-based control method to regulate or confine the camera motion. Indeed, nothing has been found on that topic in the state of the art of image-based methods.

To merge the entropy-obstacle hybrid control with the optical flow regulation, we choose to implement a hierarchical control in which the direction of motion in the five operational camera degrees of freedom space is obtained from the entropy-obstacle hybrid control law and its magnitude by the optical flow regulation law.

Obviously, once the photometric visual entropy reaches a local maximum in the 3D model, no camera motion will ever be observed again. Still in an automated approach, we implemented an alternation of entropy maximization and minimization: first, a photometric entropy maximization process is led in order to bring the camera to a rich information area and second, a photometric entropy minimization process is applied to cause a change of the area. The alternation between these two processes is performed to move away from the local maxima/minima and therefore allows the generation of a long camera path. During the entropy maximization process, since the system is driving the camera through richer and richer visual information, it is not a drawback if the camera speed is slightly slowing down. That is why optical flow confinement-based control law is mentioned above. However, during the entropy minimization process, slow motions are not relevant since the visual information is getting poorer and poorer. Thus, during the entropy minimization process, the optical flow is not confined but regulated in order to have a visual motion of a constant speed, in a mean, as can be observed in cultural heritage documentaries, for instance.

1.2 Contributions

The general strategy is to achieve an automatic navigation carrying out important criteria. The contributions are as follows:

- We use the entropy to enssure a relevant motion direction.
- We add constraints to get realistic movement.
- We present a hierarchical optimization method to build relevant camera path.
- We tested our algorithm in indoor and outdoor navigation on different 3D complex models of colored 3D point clouds.

The remainder of this paper presents, first, a description of the proposed method. Then, we present some experimental results, and a conclusion ends the paper.

2 Proposed method

To combine the principal task and the constraints, we could intuitively stack all of them in a single matrix and make an overall resolution. However, the result is falling below our expectations because in some particular situations one of the constraints is not fully taken into account. In image based approaches, the redundancy formalism is often used to combine several tasks. This technique is performed by assigning to each task a specific number of degrees of freedom of the camera. Courty et al. [11] have exploited this method to perform the main task of tracking objects, which does not require all the degrees of freedom of the camera, combined with a secondary task (occlusions management or lighting ...). In the redundancy formalism, it is not allowed for a degree of freedom to be operated by several tasks concurrently. This formalism can not be applied to our problem because the principal task that we consider (maximizing photometric entropy) requires all the degrees of freedom of the camera.

We propose a new algorithm to control automatically the camera while achieving the desired constraints. We choose a hierarchical approach done in two steps. As illustrated in Fig. 2, our method takes as input the 3D model and the camera pose **r**. During the first step, we compute the entropy-obstacle hybrid control law to estimate the motion direction. Then, during the second step, using the optical flow based constraint, we adjust the movement magnitude. A camera path is built in an iterative optimization process taking into account these two steps.

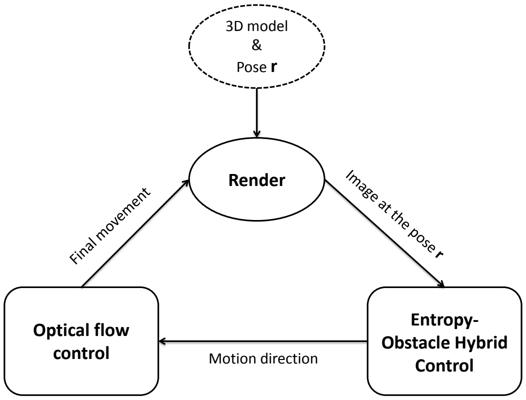


Fig. 2: The hierarchical strategy.

2.1 Task and constraints specification

Photometric entropy: We consider $I(\mathbf{r})$ the image at the pose $\mathbf{r} = (t_X, t_Y, t_Z, \theta u_X, \theta u_Y, \theta u_Z)^T$, where the three translations are $[t_X, t_Y, t_Z]$ and the rotation is represented by an angle θ and an unit vector (axis of rotation) $\mathbf{u} = [u_X, u_Y, u_Z]$. The photometric entropy of this image is given by:

$$E(\mathbf{r}) = -\sum_i P_i(\mathbf{r}) log_2(P_i(\mathbf{r})) \tag{1}$$

where i is in the range of image grey levels of a pixel in the image, $i \in [0; 255]$ and $P_i(\mathbf{r})$ is the probability of the luminance i to exist in image I. For clarity, in the following $P_i(\mathbf{r})$ is replaced by P_i.

The probability distribution is obtained using the normalized histogram of intensity value. Inspired from Dame et al. in [12] for mutual information, in a different context. We approximate the normalized histogram using a second order B-spline [21] ϕ that makes the photometric entropy twice differentiable, so the second order optimization can be achieved.

$$P_i = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \phi(i - I(\mathbf{u}, \mathbf{r})) \tag{2}$$

where $\mathbf{u}(u, v)$ are the coordinates of a pixel $\mathbf{u}$, $I(\mathbf{u}, \mathbf{r})$ is the intensity at the current pose $\mathbf{r}$ and $N_{\mathbf{u}}$ is the number of image pixels.

Obstacles avoidance cost function: All the vertices within a distance below one meter from the camera will be considered. The used cost function defined in [11], tends to infinity when the distance between the camera and the obstacle is null. Each vertex is considered as an obstacle and we maximize the distance between all the considered vertices and the camera which is equivalent to minimize:

$$O(\mathbf{r}) = \sum_{k} \frac{1}{2||{}^{w}\mathbf{p}_c -^{w} \mathbf{p}_{o_k}||^2} \tag{3}$$

Such that k is the index of a vertex. ${}^{w}\mathbf{p}_c = (x_c, y_c, z_c)^T$ is the camera position and ${}^{w}\mathbf{p}_{o_k} = (X_{o_k}, Y_{o_k}, Z_{o_k})^T$ is the position of one vertex in the world frame $\Re_w$.

Optical flow description: The goal of this constraint is to ensure a smooth movement in the image space. Our approach consists in fixing an optical flow magnitude between images. We choose a magnitude for the optical flow $\triangle\mathbf{u}_{des} = \left[\triangle u_{des}, \triangle v_{des}\right]^T$ and use the intensity conservation equation in order to link $\triangle\mathbf{u}_{des}$ and the camera pose $\mathbf{r}$:

$$I(\mathbf{u}, \mathbf{r}) = I(\mathbf{u} + \triangle\mathbf{u_{des}}, \mathbf{r} + \triangle\mathbf{r}) \tag{4}$$

2.2 Solving process

The principal task and constraints are nonlinear equations w.r.t the camera pose $\mathbf{r}$. So, we choose an iterative nonlinear optimization scheme. The linearization will be achieved using the Taylor expansion. We present first, the camera control for the six Cartesian camera degrees of freedom and the transformation to five operational camera degrees of freedom will be explained at the end of this section.The iterative scheme consists in estimating the pose increment $\triangle\mathbf{r}$ to modify the current pose $\mathbf{r}_k$ (step k). :

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \triangle\mathbf{r} \tag{5}$$

Photometric entropy criterion: To maximize the photometric entropy is equivalent to minimize its opposite $-E(\mathbf{r})$. Applying the first-order Taylor expansion leads to:

$$\frac{\partial(-E(\mathbf{r_0}))}{\partial \mathbf{r}} + \left(\frac{\partial^2(-E(\mathbf{r_0}))}{\partial \mathbf{r}^2}\right) \triangle \mathbf{r} = 0 \tag{6}$$

To solve this system, we need to compute all the derivatives of the photometric entropy. Starting by the first derivative of the entropy and using Eq. (1), we obtain:

$$\frac{\partial}{\partial \mathbf{r}}(-E(I(\mathbf{r}_0)) = \sum_i \frac{\partial P_i}{\partial \mathbf{r}}(1 + log(P_i)) \tag{7}$$

Using the expression of the probability expressed in Eq. (2) leads to:

$$\frac{\partial P_i}{\partial \mathbf{r}} = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \left(-\frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}))}{\partial(i - I(\mathbf{u}, \mathbf{r}))} \nabla I \, \mathbf{L_u} \right) \tag{8}$$

Where $\nabla I = (\nabla_u I, \nabla_v I)$ is the gradient of the image $I(\mathbf{r})$ and $\mathbf{L_u}$ the interaction matrix at $\mathbf{u} = (u, v)$.

$$L_{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} L_{\mathbf{x}} \tag{9}$$

$\mathbf{x} = (x, y)$, where:

$$\begin{cases} u = & \alpha_u x + u_0 \\ v & \alpha_v y + v_0 \end{cases} \tag{10}$$

with $(\alpha_u, \alpha_v, u_0, v_0)$ are the intrinsic camera parameters and $L_{\mathbf{u}}$ is the interaction maxtrix defined in [13].

The development below represents the second derivative of the photometric entropy:

$$\frac{\partial^2(-E(\mathbf{r_0}))}{\partial \mathbf{r}^2} = \sum_i \frac{\partial P_i}{\partial \mathbf{r}} \frac{\partial P_i^T}{\partial \mathbf{r}} \frac{1}{P_i} + \frac{\partial^2 P_i}{\partial \mathbf{r}^2}(1 + log(P_i)) \tag{11}$$

Given the equation of $\frac{\partial P_i}{\partial \mathbf{r}}$, the second derivative of the probability is given by:

$$\frac{\partial^2 P_i}{\partial \mathbf{r}^2} = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \left(\frac{\partial^2 \phi(i - I(\mathbf{u}, \mathbf{r}))}{\partial(i - I(\mathbf{u}, \mathbf{r}))^2} \mathbf{L_I^T} \mathbf{L}_I - \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}))}{\partial(i - I(\mathbf{u}, \mathbf{r}))} \mathbf{H}_I \right) \tag{12}$$

In this last equation, $\mathbf{L}_I = \nabla I \, \mathbf{L_u}$ and $H_I = \mathbf{L_u}^T \nabla^2 I \, \mathbf{L_u} + \nabla_u I \, \mathbf{H}_u + \nabla_v I \, \mathbf{H}_v$ where $\nabla^2 I = \begin{pmatrix} \nabla I_{uu} & \nabla I_{uv} \\ \nabla I_{vu} & \nabla I_{vv} \end{pmatrix}$ is the gradient of image gradient. $\mathbf{H}_u$ and $\mathbf{H}_v$ are the hessian of size 6 × 6 of the two coordinates of the point $\mathbf{u}$ [19].

Obstacles avoidance constraint: The goal is to minimize Eq. (3). Using the Taylor expansion, we have:

$$O(\mathbf{r}) \simeq O(\mathbf{r_0}) + \left(\frac{\partial(O(\mathbf{r_0}))}{\partial \mathbf{r}}\right) \triangle \mathbf{r} \tag{13}$$

As we want to avoid that the camera crosses an obstacle, only the position and not the orientation is considered because we suppose that our camera is a point (optical center). In the camera reference frame $(\Re_c)$ ${}^c\mathbf{p}_c = (0,0,0)^T$, then [11]:

$$\frac{\partial\left(O(\mathbf{r}_0)\right)}{\partial \mathbf{r}} = \sum_k \frac{1}{||{}^cP_{o_k}||^2}\left({}^cX_{o_k}, {}^cY_{o_k}, {}^cZ_{o_k}, 0, 0, 0\right) \tag{14}$$

Optical flow control constraint: From the Eq. (4), and using again the Taylor expansion, we obtain:

$$I(\mathbf{u} + \triangle_{\mathbf{u}_{des}}, \mathbf{r} + \triangle \mathbf{r}_f) = I(\mathbf{u}, \mathbf{r}) + \frac{\partial I}{\partial \mathbf{u}} \triangle \mathbf{u}_{des} + \frac{\partial I}{\partial \mathbf{r}} \triangle \mathbf{r}_f \tag{15}$$

Substituting Eq. (4) in Eq. (15) gives:

$$\frac{\partial I}{\partial \mathbf{u}} \triangle \mathbf{u}_{des} + \frac{\partial I}{\partial \mathbf{u}} \mathbf{L_u} \triangle \mathbf{r}_f = 0 \tag{16}$$

This last equation will be used later Eq. (20) to compute the final camera displacement.

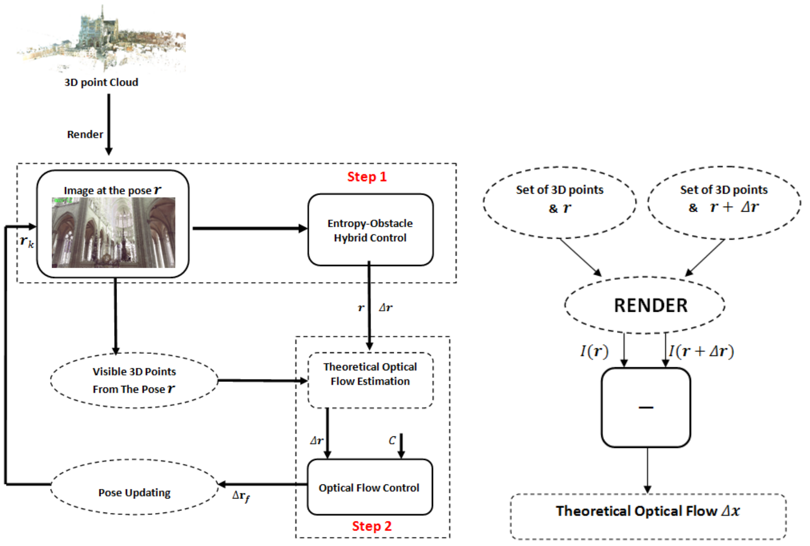
Hierarchical Algorithm: In order to satisfy the principal task and the two constraints presented in the previous section, we propose a hierarchical strategy. It consists in combining photometric entropy optimization (max/min) with the obstacles avoidance by stacking them into the same matrix to estimate $\triangle \mathbf{r}$. This will give us the motion direction. Then, we estimate the "amplitude" (to regulate the movement) of $\triangle \mathbf{r}$ by keeping the same direction previously calculated. This "amplitude" must respect a globally constant optical flow to have a visual regular velocity. Figure 3a describes the adopted camera control algorithm.

Step 1: Motion direction estimation

- It is obtained directly by stacking Eq. (6) and Eq. (13) and using the pseudo inverse $()^+$ of this system of seven equations. The estimation of $\triangle \mathbf{r}$ is given by:

$$\triangle \mathbf{r} = -\begin{pmatrix} \frac{\partial^2(-E(\mathbf{r}_0))}{\partial \mathbf{r}} \\ \frac{\partial(O(\mathbf{r}_0))}{\partial \mathbf{r}} \end{pmatrix}^+ \begin{pmatrix} \frac{\partial}{\partial \mathbf{r}}(-E(\mathbf{r})) \\ O(\mathbf{r}) \end{pmatrix} = -\mathbf{H}^+\mathbf{g} \tag{17}$$

Hence, from the displacement $\triangle \mathbf{r}$, we will use the exponential map of the SE(3) to update the current pose $\mathbf{r}$ and thus obtain the transformation matrix at this pose.



(a) Camera control process

(b) Computing theoretical optical flow

Fig. 3: Overall view of the algorithm.

Step 2: Final pose computation In this second step we regulate the "amplitude" of the movement.

- Compute the theoretical optical flow as in Fig. 3b, where $I(\mathbf{r})$ and $I(\mathbf{r} + \triangle\mathbf{r})$ are the generated synthesized images.
- We will fix a constant value C as a desired amplitude of the global optical flow.

The desired flow is computed as follows:

$$\triangle u_{des} = \frac{C.\triangle u}{\sqrt{\triangle u^2 + \triangle v^2}} \tag{18}$$

$$\triangle v_{des} = \frac{C.\triangle v}{\sqrt{\triangle u^2 + \triangle v^2}} \tag{19}$$

such that C is the constant corresponding to the optical flow mean amplitude and $\triangle u$ and $\triangle v$ are the u and v components of the flow.

From the Eq. (16) we compute the final pose displacement:

$$\triangle\mathbf{r}_f = -\mathbf{L}_{\mathbf{u}}^{+}\triangle\mathbf{u}_{des} \tag{20}$$

Where $\mathbf{L}_{\mathbf{u}}^{+}$ is the pseudo inverse of $\mathbf{L}_{\mathbf{u}}$. All matching pixels between the two images are used and $\mathbf{L}_{\mathbf{u}}$ is over-determined.

Managing rotations: The automatic navigation of the camera in the 3D model provides unrealistic rotations around the optical axis of the camera ”Roll”. Therefore, we consider that the camera has only five operational degrees of freedom, three translations (t_X, t_Y, t_Z) and two rotations (θ_p, θ_t) corresponding to the ”pan” and the ”tilt” instead of the six Cartesian degrees of freedom generally considered in the virtual visual servoing.
The camera is simulated as a virtual robot with only five degrees of freedom, where the three translations take place on the cartesian frame (world frame $\Re_w$) and the two rotations on the end-effector (camera frame $\Re_c$). We note by $\mathbf{q}_k$ the position of the robot at the step k.
We start by computing the robot jacobian on the $\Re_c$ frame ${}^c\mathbf{J}_c$:

$$ {}^c\mathbf{J}_c =^c \mathbf{V}_w^w\mathbf{J}_w \tag{21}$$

where, ${}^w\mathbf{J}_w$ is the robot jacobian expressed in the $\Re_w$ frame and ${}^c\mathbf{V}_w$ a twist transformation matrix from $\Re_w$ to $\Re_c$.
Thus the Eq. (17) becomes:

$$\dot{\mathbf{q}}_k = (\mathbf{H}{}^c\mathbf{J}_c)^+\mathbf{g} \tag{22}$$

The same steps are performed a second time for Eq. (20):

$$\dot{\mathbf{q}}_{kf} = (\mathbf{L}_\mathbf{u}^c\mathbf{J}_c)^+\triangle_{\mathbf{u}_{des}} \tag{23}$$

$\dot{\mathbf{q}}_{kf}$ represents the five operational degrees of freedom increment. The iterative scheme consists to modify the current pose $\mathbf{q}_k$ using:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \dot{\mathbf{q}}_{kf} \tag{24}$$

The transformation matrix is built from the translation vector ${}^c\mathbf{t}_w$ and the rotation matrix ${}^c\mathbf{R}_w$, where ${}^c\mathbf{R}_w = R_X(\theta_t)R_Z(\theta_p)$ and the translation vector ${}^c\mathbf{t}_w = -{}^c\mathbf{R}_w^c\mathbf{t}_w$ such as ${}^c\mathbf{t}_w = (\mathbf{q}_{k+1}[0], \mathbf{q}_{k+1}[1], \mathbf{q}_{k+1}[2])^T$.

3 Experimental results

We tested our approach on three different 3D models composed from 3D colored point clouds. Our algorithm is designed to run on extremely large data sets. An example of such data, is a set of N merged scans of a cathedral scan and urban environments. Scans are obtained by a FARO laser scanner. Environments have not been scanned totally, this is why we notice some holes, that have a white color on the images. There are now about 100 million of 3D colored points in each model. For the visualization of the 3D model we use the OGRE 3D graphics engine. On the 3D model of the cathedral we perform an indoor and an outdoor navigation. However, on the urban environments 3D models, we just perform an outdoor navigation.
This section demonstrates our results on this type of data. We applied the proposed approach to perform an automatic navigation in these 3D models based on the principal task and the two constraints previously described.

3.1 Photometric entropy maximization

This first set of experiment aims to show the behaviour of photometric visual entropy maximization. First, it is applied on a simple and flat 3D point cloud: a lab room with white wall and posters. Applying the entropy maximization algorithm, the path followed by the virtual camera tends to maximize the presence of the posters in the camera field of view Fig. 4. This is a first experimental validation of the entropy maximization to drive the virtual camera to interesting areas. The entropy maximization is applied to a virtual camera that is outside of a cathedral huge point cloud. Starting from a pose from which the cathedral is not filling the camera field of view Fig. 5, the entropy based control law drives it to maximize the amount of information in the image. Figure **??** shows the photometric visual entropy evolution over frames.

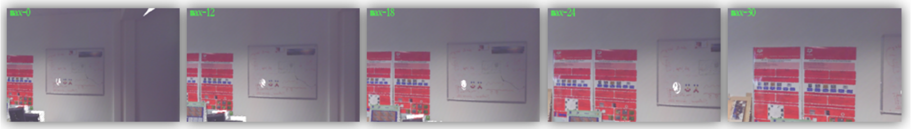


Fig. 4: Resulting visual camera path maximizing the relevant information in the image.

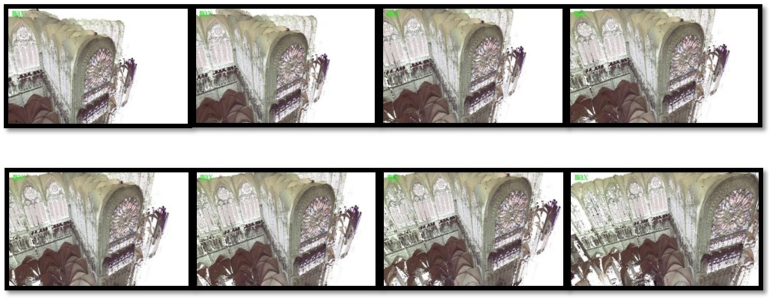
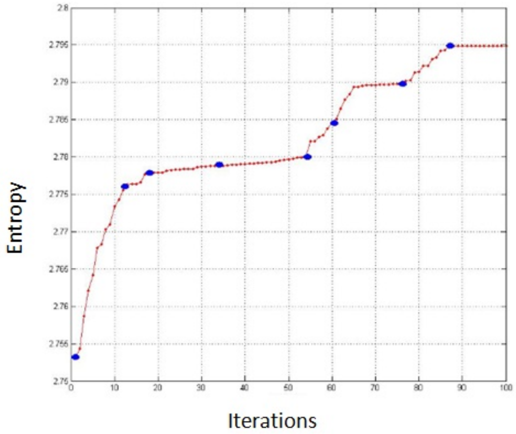


Fig. 5: Automatic navigation with photometric entropy maximization: view from outside.

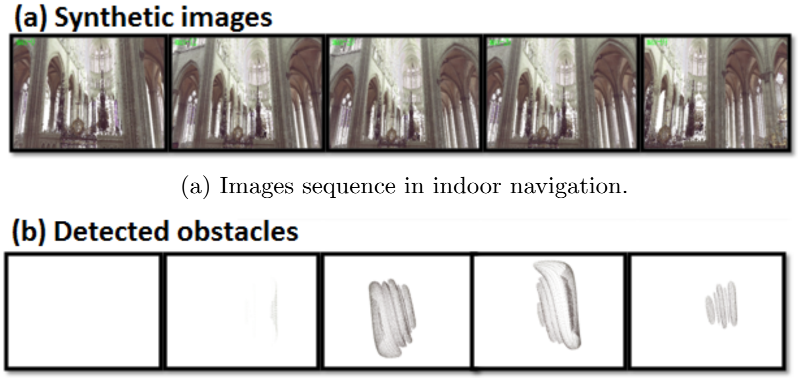
3.2 Obstacles avoidance

While experiments of section 3.1 did not drive the camera through obstacles, this situation may be encountred inside the 3D point cloud of the cathedral Fig. 6 with chairs, pilars, and so on. Figure 6 demonstrates the behaviour of the camera for the entropy and obstacles hybrid control law Eq. (17). We observe the synthetic images Fig. 6a captured at each camera viewpoint and corresponding



captionPhotometric entropy evolution. Blue dots correspond to images of Fig. 5 (from left to right and top to bottom).

images that contain obstacles (in our case vertices) surrounding the viewpoint below one-meter depth Fig. 6b. The equirectangular camera was implemented through GPU shaders programming. We notice that starting from the fourth image the camera moves away from obstacles, since the obstacles (vertices) number on the equirectangular image decreases.

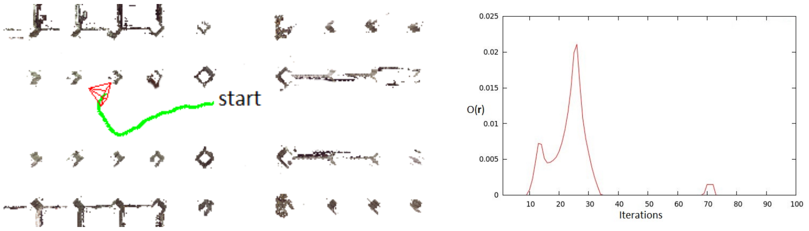


(a) Images sequence in indoor navigation.

(b) Detected obstacles.

Fig. 6: Obstacles avoidance constraint behaviour.

In Fig. 7a, we observe in green the camera trajectory projected on a 2D plane of the 3D model of the cathedral which passes between two pilliars and the Fig. 7b shows the evolution of the cost function during the generated camera movement.



(a) Camera trajectory on the 2D map of the 3D model of the cathedral

(b) Obstacles cost function evolution

Fig. 7: Camera path.

3.3 Optical flow control

In this third experiment, the camera is guided by the proposed hierarchical control law.

Starting from the same initial pose of the camera, two images sequences were generated. The first sequence considers the constraint of optical flow control and the second doesn't. The two curves of Fig. 8 show the evolution of the average optical flow norm calculated for both sequences. The red curve clearly shows that we keep almost regular flow compared to the green curve where the evolution of the flow is disturbed during the whole sequence. In order to generate a regular camera movements avoiding significant jumps over its trajectory in the 3D model, this constraint is necessary and yields to the desired result. The video accompanying this paper visualy illustrates the difference between the two camera paths obtained before and after the flow control.

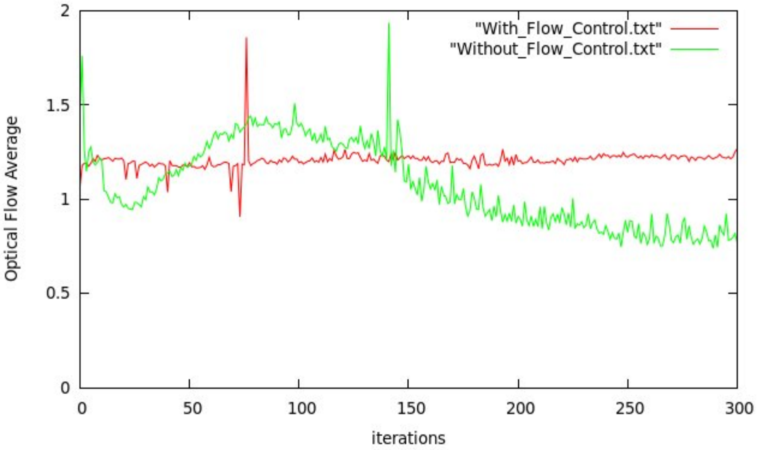


Fig. 8: Difference between the average of the optical flow norm for an images sequence considering the optical flow control constraint (red), and the other not considering it(green).

3.4 ”Realistic” navigation

In this last experiment, the optical flow control depends on the entropy maximization or minimization. Indeed, maximization and minimization of the entropy is alterning to reach and, then, leave local maxima. The optical flow is confined in order to spend more time in rich information parts of the scene. However, during the entropy minimization process, slow motions are not relevant since the visual information is getting poorer and poorer. Thus, during the entropy minimization process, the optical flow is not confined but regulated in order to have a visual motion of a constant speed.
Figure 9 shows some images extracted from the video generated by applying the proposed algorithm. Starting from a small facade, the algorithm moves the camera to include the entire facade of the most interesting building of the 3D model.

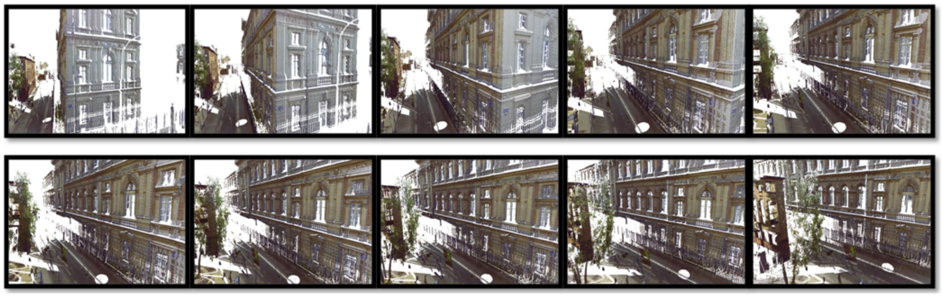


Fig. 9: Automatic navigation.

4 Conclusion and future works

In this paper we presented a new approach to achieve automatic navigation in a 3D dense colored point cloud model. The proposed method generates smooth and realistic camera trajectories. The method implements a hierarchical algorithm performed in two steps. The first step concerns the motion direction estimation by stacking the principal task (photometric entropy maximization) and one of our constraints (obstacles avoidance).We show the benefits of using a photometric entropy to guide the camera to interesting areas. The second step controls the optical flow in the image in order to avoid sudden jumps of the camera. We have demonstrated, particularly in Fig. 9, that from a starting camera viewpoint where the image is poor, the algorithm moves the camera to a rich viewpoint by generating a smooth and realistic movement. The method is general and can integrate other criteria to optimize.
One of our future perspectives is to achieve a framework that introduce interactivity to this automatic method by adding other constraints in order to assist the user during the exploration of the 3D model. Then, we plan to carry out a study based on user opinion to evaluate the system, and maybe even improve it.

References

1. Andújar Carlos, Vázquez P., Fairén Marta: Way-Finder: guided tours through complex walkthrough models. Computer Graphics Forum, Wiley Online Library, **23** (2004) 499–508
2. Bares William, McDermott Scott, Boudreaux Christina, Thainimit Somying: Virtual 3D camera composition from frame constraints. Proceedings of the eighth ACM international conference on Multimedia, ACM, (2000) 177–186
3. Bares William H., Thainimit Somying, McDermott Scott, Boudreaux C.: A model for constraint-based camera planning. Proceedings of the 3rd international conference on Intelligent user interfaces, Smart Graphics, Papers from the 2000 AAAI Spring Symposium, AAAI Spring Symposium, (2000) 84–91
4. Bares William H., Zettlemoyer Luke S., Rodriguez Dennis W., Lester James C.: Task-sensitive cinematography interfaces for interactive 3D learning environments. Proceedings of the 3rd international conference on Intelligent user interfaces, ACM, (1998) 81–88
5. Bandi Srikanth, Thalmann Daniel: Space discretization for efficient human navigation. Computer Graphics Forum, Wiley Online Library, **17** 3 (1998) 195–206
6. Beckhaus Steffi, Ritter Felix, Strothotte Thomas: Cubicalpath-dynamic potential fields for guided exploration in virtual environments. Computer Graphics and Applications, Proceedings in The Eighth Pacific Conference, IEEE, (2000) 387–459
7. Blinn Jim: Where am I? What am I looking at?(cinematography). Computer Graphics and Applications, IEEE, **8**, 4 (1988) 76–81
8. Bourne Owen, Sattar Abdul, Goodwin Scott: A constraint-based autonomous 3d camera system. Constraints, Springer, **13**, 1-2 (2008) 180–205
9. Chaumette Francois: Potential problems of stability and convergence in image-based and position-based visual servoing. The confluence of vision and control, Springer (1998) 66–78
10. Christie Marc, Machap Rumesh, Normand Jean-Marie, Olivier Patrick, Pickering Jonathan: Virtual camera planning: A survey. Smart Graphics, Springer, (2005) 40–52
11. Courty Nicolas, Marchand Eric: Computer animation: A new application for image-based visual servoing. Robotics and Automation. Proceedings ICRA. IEEE International Conference, IEEE, **1** (2001) 223–228
12. Dame Amaury, Marchand Eric: Entropy-based visual servoing Robotics and Automation, 2009. ICRA'09. IEEE International Conference , (2009), 707–713
13. Feddema John T., Mitchell Owen R.: Vision-guided servoing with feature-based trajectory generation for robot. Robotics and Automation, IEEE Transactions on, **5**, 5 (1989) 691–700
14. Gleicher Michael, Witkin Andrew Through-the-lens camera control. ACM SIGGRAPH Computer Graphics, **26**, 2 (1992) 331–340
15. Gumhold Stefan: Maximum entropy light source placement. Visualization, VIS, IEEE, (2002) 275–282
16. Hager Gregory D.: A modular system for robust positioning using feedback from stereo vision. Robotics and Automation, IEEE Transactions on, IEEE, **13**, 4 (1997) 582–595
17. Halper Nicolas, Helbing Ralf, Strothotte Thomas: A Camera Engine for Computer Games: Managing the Trade-Off Between Constraint Satisfaction and Frame Coherence. Computer Graphics Forum, Wiley Online Library, **20**, 3 (2001) 174–183

18. Lamarche Fabrice: Topoplan: a topological path planner for real time human navigation under floor and ceiling constraints. Computer Graphics Forum Wiley Online Library **28**, 2 (2009) 649–658
19. Lapresté Jean-Thierry, Mezouar Youcef: A Hessian approach to visual servoing. Intelligent Robots and Systems IROS, IEEE **1**, (2004) 998–1003
20. Li Tsai-Yen, Cheng Chung-Chiang: Real-time camera planning for navigation in virtual environments. Smart Graphics, Springer (2008) 118–129
21. Maes Frederik, Collignon Andre, Vandermeulen Dirk, Marchal Guy, Suetens Paul: Multimodality image registration by maximization of mutual information. Medical Imaging, IEEE Transactions on, **16**, 2 (1997) 187–198
22. Marchand Eric Hager Gregory D.: Dynamic sensor planning in visual servoing. Robotics and Automation, Proceedings International Conference on, IEEE **3**, (1998) 1988–1993
23. Ozak Maya, Gobeawan Like, Kitaoka Shinya, Hamazaki Hirofumi, Kitamura Yoshifumi, Lindeman Robert W.: Camera movement for chasing a subject with unknown behavior based on real-time viewpoint goodness evaluation. The Visual Computer, Springer, **26**, 6-8 (2010) 629–638
24. Salomon Brian, Garber Maxim, Lin Ming C., Manocha Dinesh: Interactive navigation in complex environments using path planning. Proceedings of the 2003 symposium on Interactive 3D graphics, ACM, (2003) 41–50
25. Serin Ekrem, Hasan Adali, Serdar Balcisoy Selim: Automatic path generation for terrain navigation. Computers & Graphics, Elsevier, **36**, 8, (2012) 1013–1024
26. Tomlinson Bill, Blumberg Bruce, Nain Delphine: Expressive autonomous cinematography for interactive virtual environments. Proceedings of the fourth international conference on Autonomous agents, ACM, (2000) 317–324
27. Vázquez Pere-Pau, Feixas Miquel, Sbert Mateu, Heidrich Wolfgang: Viewpoint Selection using Viewpoint Entropy. VMV, **1**, (2201) 273–280
28. ware1990exploration, Ware Colin, Osborne Steven: Exploration and virtual camera control in virtual three dimensional environments. ACM SIGGRAPH Computer Graphics **24**, 2 (1990) 175–183
29. Yeh I., Lin Chao-Hung, Chien Hung-Jen, Lee Tong-Yee: Efficient camera path planning algorithm for human motion overview. Computer Animation and Virtual Worlds, Wiley Online Library **22**, 2-3 (2011) 239–250