3D 가상환경에서의 관심 추적

이성길[0] 김정현
포항공과대학교 컴퓨터공학과 가상현실 연구실
(yskill, gkim)@postech.ac.kr

Pseudo-Attention Tracking in 3D Virtual Environments

Sungkil Lee[0] Gerard Jounghyun Kim
Virtual Reality Laboratory
Department of Computer Science and Engineering
Pohang University of Science and Technology

요 약

We propose and demonstrate a methodology to algorithmically track regions that are believed to be visually attentive in dynamic and interactive virtual environments. The " attention map" at each frame is constructed by considering both low level and raw visual features and the other high level knowledge about the scene. In particular, as for the raw bottom-up visual features, our algorithm incorporates dynamic 3D features such as depth, object motion, and feature alterations, in addition to the usually considered features such as color, intensity, and orientation, to make the tracking applicable to dynamic 3D virtual environments. As for the top down knowledge, user's motion and object familiarity are used to modulate contributions from the low level features. These measures are used to filter the scene images and generate visually salient regions from which salient objects are extracted and tracked. The salient object tracking is further stabilized using the Kalman filter. The resulting " pseudo" attention tracking can be used effectively, without employing an expensive eye tracker, for perceptually based rendering (such as giving depth of field effects), and video compression.

1. Introduction

With today's ever improving processing and graphic rendering power (relative to their cost), perceptually based rendering can be employed and possibly improve the task performance or spatial presence. For example, in a complex virtual environments, users are visually overwhelmed (consciously or not), without any focused objects (every objects are clear and focused), which is not the way humans operate in the real world. Naturally, finding the focused area of the scene during interaction or highlighting the spatially important features throughout the scene can impact the user's task performance and sense of spatial presence. While an eye tracking sensor can be used for this purpose, it is still very expensive and difficult to use (e.g. heavy, cumbersome).

Instead, we can apply principles learned in the human perception and attempt to computationally guess where the user might be looking and focusing at. This is called the " pseudo" attention tracking (in the sense that it is not the actual human attention that is being tracked). For instance, current research findings on the attentional

mechanism have identified two major components: (1) bottom-up image features and local contrast, and (2) top-down properties such as prior knowledge, memories, goals or task [1, 2]. Numerous computational methods based on biological vision mechanism have been proposed [3, 4, 5, 6, 7, 8]. Among them, the saliency model of Koch and Ullman gave rise to the numerous software and hardware implementations [7, 9]. However, all these models were purely based on the static bottom-up features and did not need the top-down knowledge about the scene. Moreover, these frameworks mainly targeted still images only. In this paper, we propose and demonstrate a methodology to algorithmically track regions that are believed to be visually attentive in dynamic and interactive virtual environments. The " attention map" at each frame is constructed by considering both low level and raw visual features and the other high level knowledge about the scene. In particular, we extend the previous approaches by additionally incorporating motion, depth, and temporal feature alternation as for the raw bottom-up visual features to make the tracking applicable to dynamic 3D virtual environments. Our framework also incorporates top down knowledge, such as object familiarity and user motion, for a more realistic attention tracking. Salient regions (pixel level) are calculated at every simulation frame and mapped to the corresponding objects. Then, we compensate for the abrupt changes the attention levels of the objects using the linear Kalman filtering [10]. Such a " pseudo" attention tracking can be used effectively, without employing an expensive eye tracker, for perceptually based rendering (such as giving depth of field effects), and video compression.

2. Bottom-up Feature based Saliency Map

A computational framework for extracting human visual attention has originally been proposed by Itti et al. [7]. Their framework consists of three main processes, (1) extracting bottom-up feature maps, (2) constructing conspicuity maps by multi-scale center-surround operations, and (3) integration of the conspicuity maps into a saliency map.

In virtual environments, model space preattentive features such as depth, motion, shape, and 3D orientations are available for a more robust prediction of human attention. We adopted two model space features, namely, depth and motion. It is difficult to derive a general measure for " shapes," and considering 3D orientation is redundant because 2D orientation is already included as an image space feature. A temporal feature change is also used as a dynamic attentional feature. For each feature map, a conspicuity map is constructed by a center-surround operation. A multi-scale center-surround operation represents the subtraction of fine center $c \in \{2,3,4\}$ and coarse surround $s = c + \delta, \delta \in \{3,4\}$ at different levels of the image pyramid. Practically this process could be simplified by using a DoG (Difference of Gaussian) filter. Lastly, the conspicuity maps are linearly combined into one saliency map.

The following sections describe how each feature maps were created. We omit the explanation of the static image space feature extraction as we have followed the similar measures used by Itti et al. [7].

2.1 Extracting Motion (3D/Dynamic Feature)

Motion is also one of the most important attentional features in dynamic environments. The previous approaches [11] has used pixel space approaches and considered only translational motions to create motion maps. However, in virtual environments we

can easily take advantage of the object level motion information. The translational motion velocity, M_t, is defines as:

$$M_t = \frac{1}{z}\frac{\sqrt{\delta x^2 + \delta y^2 + \delta z^2}}{\delta t} \quad (1)$$

where δx, δy and δz indicates the distance of center of objects between the previous frame and current frame in 3D model space and δt is the simulation time. Rotational motion should be particularly treated in the model space, not in the image space, because the center of object might be used as the axis of rotation. We use angular velocity as the value of rotational motion, thus all the pixels in an object can be treated equally. Rotational motion, M_r, is defined as the following:

$$M_r = |\omega| = \frac{|\delta\theta|}{\delta t} \quad (2)$$

where $\delta\theta$ is the changed rotation angle measured by rotation matrix. Figure 1 shows an example of a motion feature map.

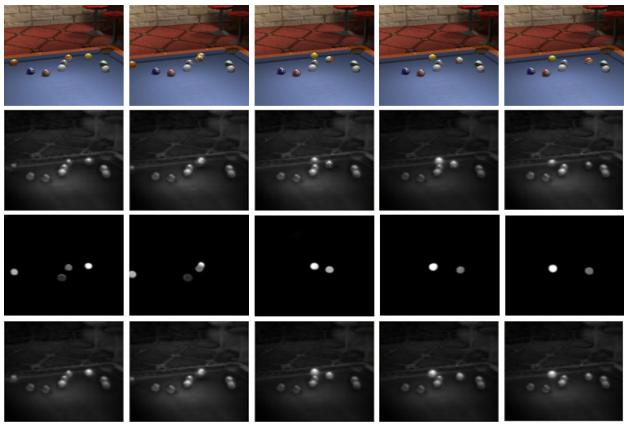
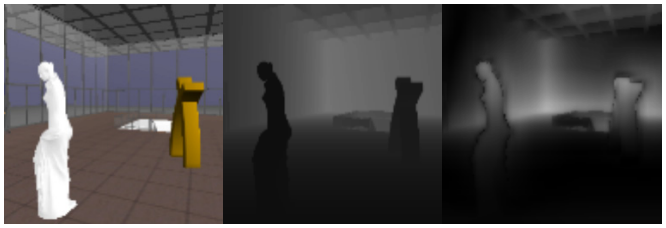


Fig. 1. Extracting motion information: a motion sequence (row 1), saliency map without the motion feature (row 2, observe that a static ball is selected as salient object), a motion feature map (row 3), a saliency map with motion feature added (row 4, moving balls are more salient).

2.2 Extracting Depth (3D Feature)

Humans focus and attend objects at different depth ranges, and thus depth information can be used as effective attentional cue. Depth can be computed easily from the depth buffer (See Figure 2).



(a) (b) (c)

Fig. 2. An example of a depth feature map and its conspicuity map. (a) original image, (b) depth feature map, (c) depth conspicuity map. Note that bright pixels represent more conspicuous region in the image with respect to the given feature.

2.3 Temporal Feature Change (Dynamic Feature)

With the static bottom-up feature based saliency maps, temporal change in various visual features can hardly be captured (e.g. flickering red/blue box in red background, deformable objects). One such dynamics is represented in motion, and this has been addressed above (Section 2.1). To capture other types of changing features, we compare the previous and current feature maps and build temporal difference maps. To reduce computation, the features may be designated ahead of time. (e.g. human joints in animation).

3. Adding Top-down Contextual Features

Bottom-up saliency map is only useful for situations where the user is looking at a scene with no particular goal oriented cognitive activity. This is one reason that several previous works that only consider bottom-up visual features resulted in significant discrepancy to the actual saliency map measured with eye-tracking sensors. [12, 13].

Most virtual environments require a task for the user and this will greatly influence the region of attention at different times in addition to the raw visual features. While such situational information can be input manually given the interactive scenario of a virtual environment, we seek to partially automate this process by defining two heuristic measures to reflect the user's intention and thus estimate one's focused object/area.

3.1 User Motion (Spatial Context)

One way to deduce where the user will look at is to use the direction of the user's motion and where the user's head is directed at is moving. A spatial context map $SC(u,v)$ is defined as the following:

$$SC(u,v) = \frac{1}{2}(SC_d(u,v) + SC_r(u,v)) \quad (3)$$

where $SC_d(u,v)$ and $SC_r(u,v)$ each denote navigational direction (given by user input) and head rotation (given by a sensor commonly used in virtual environments).

$$SC_d(u,v) = \hat{d}_n \cdot \hat{d}(u,v) - k_{sc} e^{-\frac{x^2+y^2}{2\sigma_{sc}^2}} \quad (4)$$

where $\hat{d}_n$ and $\hat{d}(u,v)$ represent the projected and normalized direction of navigation and direction from center at pixel (u,v), and k_{sc} and σ_{sc} represents the constant and standard deviation necessary to the Gaussian degradation of center.

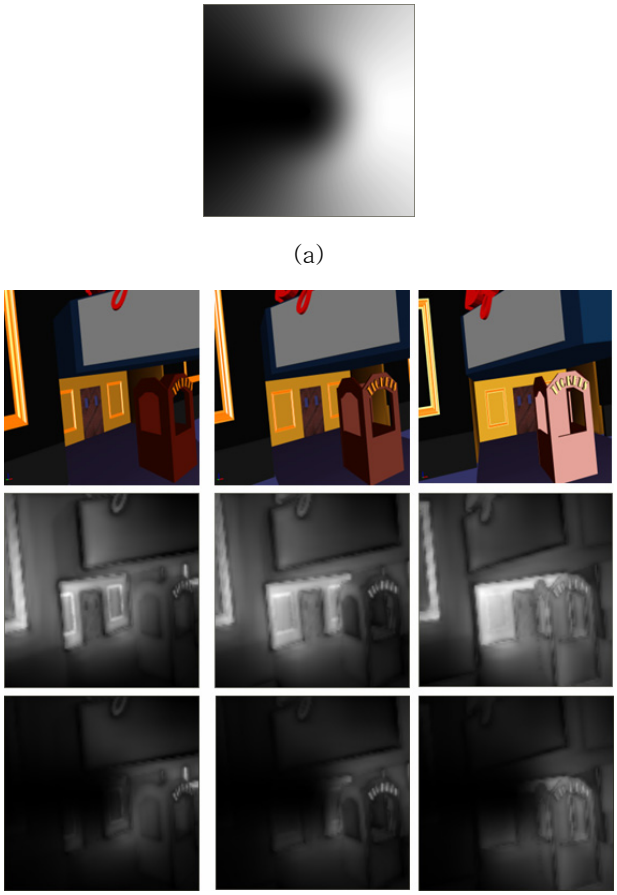
$$\begin{aligned} d_n(u,v) &= M_{proj}(loc(x,y,z,t) - loc(x,y,z,t-1)) \\ d(u,v) &= (u - u_{center}, v - v_{center}) \end{aligned} \quad (5)$$

where M_{proj} denotes the projection matrix from the 3D space to the image space, $loc(x,y,z,t)$, the view position at current simulation tick, (u_{center}, v_{center}), the center of image screen space. A simple example of a spatial context map to the navigational direction $\hat{d}_n = (1,0)$ is shown in the Figure 3.

Similarly to the navigational direction map, head rotational map is formed as the following:

$$SC_r(u,v) = \hat{r}_h \cdot \hat{d}(u,v) - k_{sc} e^{-\frac{x^2+y^2}{2\sigma_{sc}^2}} \quad (6)$$

where $\hat{r}_h$ represents the normalized direction of head rotation.



(a)

(b)

Fig. 3. An example of a navigational direction map for $\hat{d}_n = (1,0)$, (a). The right region is assumed to be more salient, because user is moving to the right (b). First row: Original image sequence, Second row: Saliency maps without the navigational feature, Third row: Modified saliency map considering the navigation direction.

3.2 Familiarity Context

Familiarity context represents the degree of familiarity with objects in a given virtual environments. We first use a heuristic not to select backdrop objects as salient ones. Even though one might actually attend to an object in the far back, it most probably will not bear any significance in terms of improving task performance. The second heuristic involves the interaction history. For simple navigation, it is not likely that users will attend to already seen objects. On the other hand, for task-oriented virtual environments, it is more likely that a particular set of objects will be used more and thus attended more. To implement this heuristic, during the simulation, the system records the interaction time (with a particular object) and uses it as a probability for the next interaction (and saliency). Familiarity context map FC_{int} for the interaction history is defined as the following:

$$FC_{\text{int}}(u,v) = \frac{t_{\text{int}}(object(u,v))}{T_{\text{int}}} \qquad (7)$$

where t_{int} and T_{int} represent the individual object and total interaction time, and $object(u,v)$, the object corresponding to the pixel position (u,v).

4. Pseudo-Attention Tracking

In the previous sections, we described how to saliency maps with respect to a number of different image, model, and top-down information features. The goal of the pseudo-attention tracking is to generate a smooth tracking of focused objects using the consolidated (by linear combination) saliency map computed at each simulation tick.

The consolidated saliency map supplies saliency information at the pixel level, and in order to extract the corresponding objects, we use the item buffer [14] which contains the object ID in each pixel position. If saliency is computed and tracked at the object level, the tracking can exhibit "popping" effects with abrupt changes in the current salient objects. While humans may change their focus very abruptly, the change of focus occurs in a continuous and smooth manner. To emulate this, we employ the well known Kalman Filtering [10]. For lack of space, we only illustrate the tracking results in Figure 4.

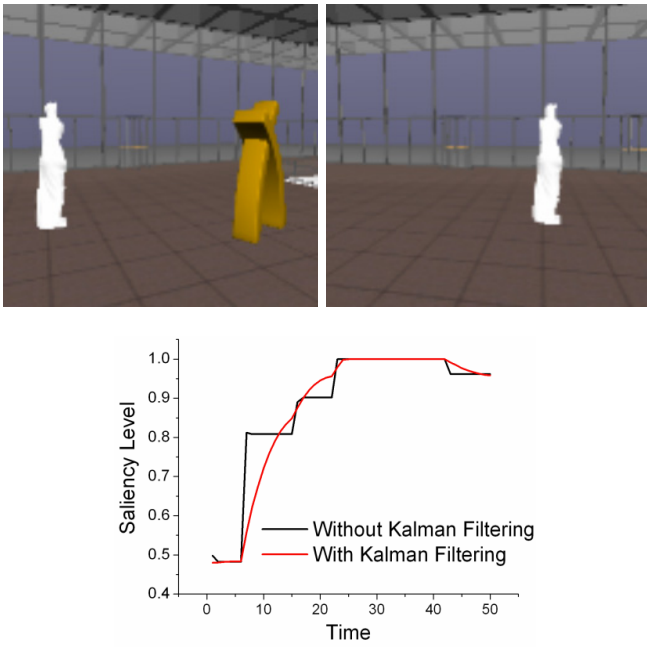


Fig. 4. Tracking saliency in a dynamic scene (Left) Initial view, time=0 (Center) Last view, time=50 (Right) Saliency level of Venus statue without and with Kalman tracking.

5. Conclusions

Previous computational approaches to attention tracking have only considered static visual features and are not suited for dynamic virtual environments. Our framework has incorporated dynamic and top down information features so that the resulting tracking framework can be applied to the perceptually based rendering of virtual environments. Perceptually based rendering of virtual environments is expected to improve the user task performance by channeling and focusing

user’s attention to important objects in the scene. It can be even beneficial for spatial presence (feeling of being in the scene) by highlighting important spatial features and assist the user in constructing a more concrete internal spatial model of the environment.

Fig. 5. Application of pseudo-attention tracking to “multiple” depth of field rendering. (Left) normal rendering (Right) "multiple" depth of field rendering

참고문헌

[1] Henderson, J. M. Human gaze control in real-world scene perception. Trends in Cognitive Sciences, 7:498-504, 2003.

[2] Loftus, G. R., Mackworth, N. H. Cognitive determinants of fixation location during picture viewing. Journal of Experimental Psychology: Human Perception and Performance 4: 565-572, 1978.

[3] Koch, C., Ullman, S. Shifts in selective visual attention. Human Neurobiology, 4: 219-227, 1985.

[4] Julesz, B., Bergen, R. Textons, the fundamental elements in preattentive vision and perception of textures. Bell System Tech., 62(6):1619-1645, 1983.

[5] Ahmad, S. VISIT: An efficient computational model of human visual attention. Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1991.

[6] Culhane, S. M., Tsotsos, J. K. An attentional prototype for early vision. The second European conference on computer vision, pp. 551-560, 1992.

[7] Itti, L., Koch, C. A Model of Saliency-based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11):1254-1259, 1998.

[8] Backer, G., Mertsching, B., Bollmann, M. Data- and model-driven gaze control for an active-vision system. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(12):1415-1429, 2001.

[9] Ouerhani, N., Hügli, H. Computing visual attention from scene depth. ICPR'00, pp. 375-378, 2000.

[10] Kalman, R. E. A new approach to linear filtering and predictive problems. Transactions ASME, Journal of basic engineering, 82:34-45, 1960.

[11] Longhurst, P., Debattista, K., Chalmers, A. A GPU based saliency map for high-fidelity selective rendering. The fourth international conference on computer graphics, virtual reality, visualization and interaction in Africa, pp.21-29, 2006.

[12] Ouerhani, N., Wartburg, R. von, Hügli, H. Empirical validation of the saliency based model of visual attention. Electronics letters on Computer Vision and Image Analysis, 3(1):13-24, 2004.

[13] Santella, A., DeCarlo, D.Visual interest and NPR: an evaluation and manifesto. NPAR'04, pp. 71-150, 1994.

[14] Weghorst, H., Hooper, G., Greenberg, D. P.: Improved Computational Methods for Ray Tracking. ACM Transactions on Graphics, 3(1):52-69, 1984.