Adaptive Picture-in-Picture Technology based on Visual Saliency

Shijian Lu*, Byung-Uck Kim*, Nicolas Lomenie†, and Joo-Hwee Lim*
*Institute for Infocomm Research, A*STAR, Singapore
E-mail: {slu,bukim,joohwee}@i2r.a-star.edu.sg
†Computer Science, Universit Paris Descartes, France
E-mail: nicolas.lomenie@mi.parisdescartes.fr

***Abstract*—Picture-in-picture (PiP) is a feature of some television receivers and video devices, which allows one main program to be displayed on the full screen while one or more sub-program displayed in inset windows. Currently most TV/video devices require users to specify where and how large to place the sub-program over the main program display. This process is however not user-friendly as it involves a manual process and once specified, the size and the location of the sub-program will be fixed even when they block some key visual information from the main program. We propose an automatic and adaptive PiP technology that makes use of computational modeling of visual saliency. For each frame of the main program, a saliency map is computed efficiently which quantifies how probable a display region of the main program contains useful information and will attract humans' attention/eyes. The sub-program can thus be adaptively resized and placed to the display region that contains the least useful information. Preliminary experiments show the effectiveness of the proposed technology.**

I. Introduction

Picture-in-picture (PiP) is a feature of some media devices that allows users to watch two programs at once. Typically one main program is displayed in full screen while one sub-program is concurrently played in inset window within the main program display. The sub-program usually has no sound, but allow users to keep track of what's happening on the second/more channels. In earlier days, the PiP technology was designed for some high-end television sets, where two independent tuners or signal sources are needed to supply a main program and a sub-program, respectively. In recent days, some new media such as Blu-ray Disc also include the PiP technology, allowing viewers to watch more than one media sources at the same time.

PiP technology has been studied for many years, ranging from the earlier hardware design [1], [2] and the recent low-level software processing such as transcoding [3], [4], [5]. On the other hand, the high-level design of intelligent PiP technology lags far behind. In particular, the current PiP technology usually provides an interface through which users can specify how large and where to place the sub-program. Once specified, the inset window will be at a fixed position with a fixed size. Such manual pre-specification of the inset window is instead not a good experience to users. More importantly, the inset window of fixed size and fixed position could block some key visual information from the main program from time to time. An alternative and more user-friendly approach is to remove the user-specification process and at the same time, the sub-program will adapt their size and positions so that useful information from the main program will not be blocked.

We propose an automatic and adaptive PiP technology that exploits computational modeling of visual saliency. For each frame of the main program, a saliency map is computed that tells how probable each region of a video frame contains important information and will attract viewer's attention/eyes. The sub-program can thus be resized and placed to the region that contains the least salient information. In particular, we use four corners of the display screen as candidate regions based on the guideline that the sub-program should not be placed at the center of the display screen. The display corner with the least salient information is thus selected to place the sub-program. Experiments show very promising results.

II. Adaptive PiP Technology

This section describes the proposed adaptive PiP technology. In particular, we will divide this section into three subsections, which deal with computational modeling of visual saliency, adaptive specification of the size and position of the sub-program, and discussion, respectively.

A. Computational Modeling of Visual Saliency

Computational modeling of visual saliency aims to build a saliency map for an image or video frame, and it has a wide range of applications in adaptive image/video compression, visual search, etc. [6]. It has been drawing increasing research interest in recent years thanks to the advances in eye-tracking devices, with which human fixations can be recorded while a subject is freely viewing a scene or image.

Quite a number of saliency models have been reported in recent years [9]. We adopt the saliency model by S. Lu et al. [8]. The model computes saliency from image co-occurrence histogram (ICH) and shows very good performance in efficiency and prediction accuracy. Consider a single-channel integer image I. Let $\mathbb{K} = \{1, 2, \cdots, k\}$ be a set of k possible image values within I (k is 256 for a 8-bit integer image). H, the ICH of the image I is defined as follows:

$$H = \big[h(m,n)\big], m, n \in \mathbb{K} \tag{1}$$

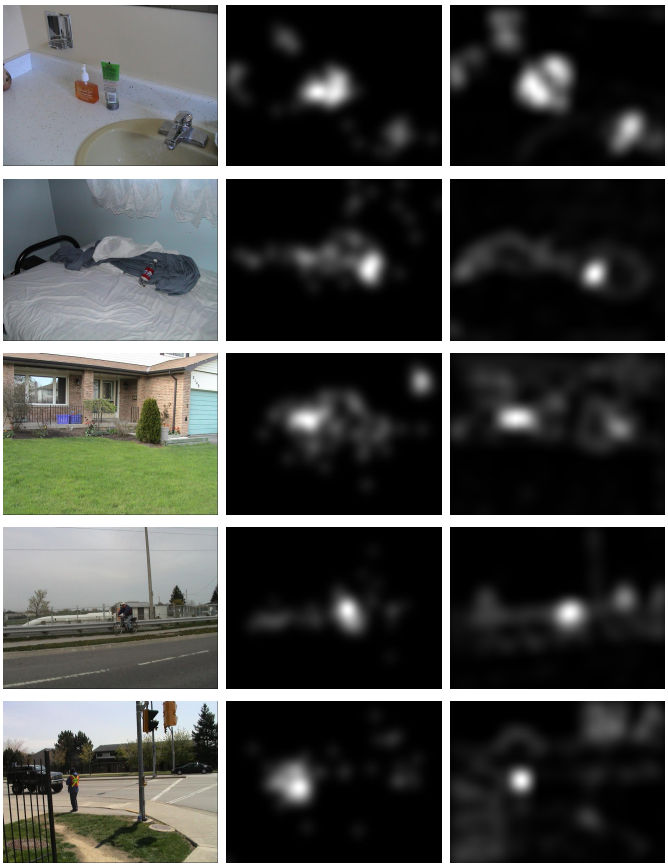

Fig. 1. The ICH based saliency model predicts the human fixations accurately. For the three images from the AIM dataset [7] in the first column, columns 2 and 3 show the corresponding fixational maps and saliency maps, respectively.

where H is a symmetric square matrix of size $k \times k$. An ICH element $h(m,n)$ is the co-occurrence frequency of image values m and n within a square neighborhood window of size z. H is constructed as follows. For each image pixel with a value of m, all image pixels within the local neighborhood window are examined one by one. If a neighboring pixel has a value of n, the ICH element $h(m,n)$ is increased by one. The ICH is built after all image pixels within I are examined as described above.

A probability mass function (PMF) P can then be determined through normalizing H by its sum. Since saliency is usually negatively correlated with occurrence/co-occurrence, an inverted PMF $\overline{P}$ is computed as follows:

$$\overline{p}(m,n) = \begin{cases} 0 & \text{if } p(m,n) = 0 \\ 0 & \text{if } p(m,n) > U \\ U - p(m,n) & \text{if } p(m,n) \leq U \end{cases} \quad (2)$$

where $p(m,n)$ denotes an element of P. As defined in Equation 2, elements of $\overline{P}$ are set to 0 when there are no corresponding pixel value pairs within the image or when the corresponding P elements are larger than a certain threshold (i.e. they are common and therefore inconspicuous). The threshold U denotes a uniform distribution whose value is the inverse of the average of non-zero elements within P [8].

Saliency can then be computed from $\overline{P}$. For each image pixel at location (i,j), the corresponding image saliency $S(i,j)$ is computed as follows:

$$S(i,j) = \sum_{i'=i-z}^{i+z} \sum_{j'=j-z}^{j+z} \overline{p}\big(x(i,j), x(i',j')\big) \quad (3)$$

where z denotes the size of the neighborhood window, which is the same as used for the ICH construction. The notations $x(i,j)$ and $x(i',j')$ denote image values at locations (i,j) and (i',j'), respectively and $\overline{p}\big(x(i,j), x(i',j')\big)$ is therefore the element of $\overline{P}$ indexed by $x(i,j)$ and $x(i',j')$.

Fig. 1 shows the saliency maps that are produced by the saliency model in [8]. For the sample images from the AIM dataset [7] shown in the first column, the second column shows the corresponding fixational maps that are built through smoothing of the eye fixations that are collected from 20 subjects for each image. The third column shows the computed saliency maps that predict the human eye fixations accurately. The adaptive sub-program specification will be done based on the saliency maps to be discussed in the next subsection.

B. Adaptive Sub-Program Specification

The sub-program is usually placed at one of the four corners of the main program so that it will have less effect on the overall picture of the main program. We follow this tradition and the problem is simplified by two sub-problems, namely, which corner to place the sub-program and how large the sub-program should be set within the main program.

In our proposed technique, the aspect ratio of the sub-program is fixed so as to present the sub-program naturally. The size of the sub-program can thus be determined by a predefined threshold T_s as follows:

$$f_{max}(\frac{w_s}{w_m}, \frac{h_s}{h_m}) = T_s \quad (4)$$

where w_s, h_s, w_m, h_m denote the width and height of the sub-program and main program, respectively. The term $f_{max}()$ is a standard maximum function, which ensures that the sub-program will not have an ultra-large width or height when it has an ultra-large or ultra-small aspect ratio. T_s is the pre-defined threshold and we set it at 1/4 in our system.

Based on the principle that the image region with smaller saliency is less attractive and contains less useful information, the sub-program is placed to the main program corner that has the smallest saliency as follows:

$$CN = argmin(S_{cn}), cn = 1, \cdots, 4 \quad (5)$$

where S_{cn} is a four-element vector and each vector element is the mean saliency at one corner of the main program which is computed as follows:

$$S_{cn} = \frac{\sum_{i=1}^{h} \sum_{j=1}^{w} S(i,j)}{w \times h}, cn = 1, \cdots, 4 \quad (6)$$

where w and h denote the width and height of one corner of the main program. $S(i,j)$ denotes the saliency at (i,j) as computed in Equation 3.

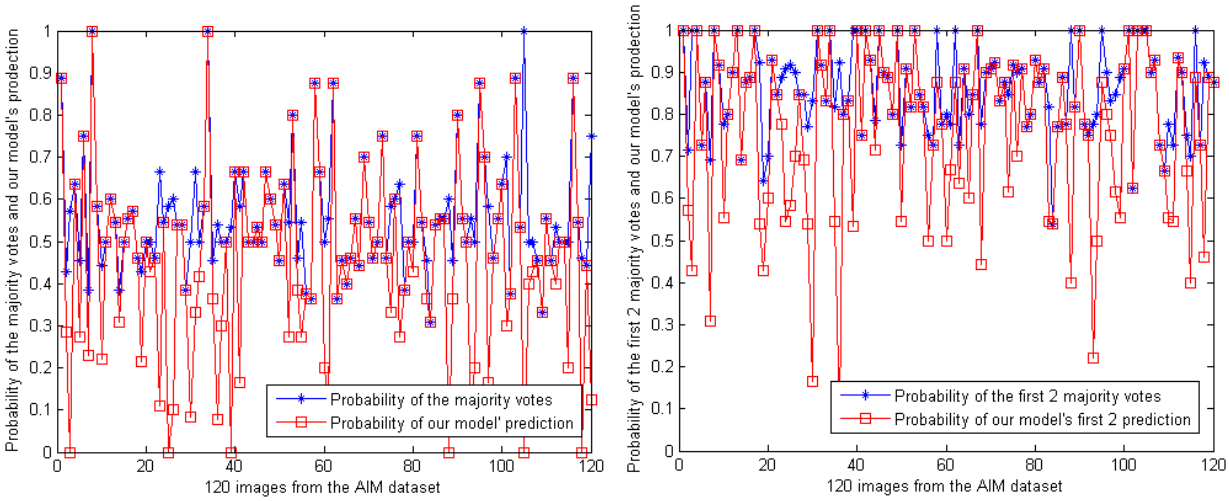

Fig. 2. Prediction accuracy of our proposed technique where the graph on the left shows the majority vote probability from the 8 subjects and the predicted probability by our proposed technique whereas the graph on the right shows the probability of the first 2 majority votes and the probability of the first two predicted corners (with the least saliency).

C. Discussion

The saliency of the main program needs to be computed as efficient as possible. There are two ways to speed up the computation process based on the saliency model in [8]. First, each frame of the main program can be down-scaled before the saliency computation. This has little effects on the computed saliency because the ICH based model is tolerant to the image scale variation. In particular, optimal saliency can often be computed when images are at 0.3-0.5 of the original image scale based on the public dataset in [7]. Second, the computation can be reduced significantly by setting a small neighborhood size z. Study in [8] shows that the variation of z also has little effects on the computed saliency.

In addition, the main program corner with the smallest saliency could change frequently among the four corners depending on the contents of the main program frames. On the other hand, the position of the sub-program cannot be changed from one corner to another frequently because frequent change is visually disturbing to users/viewers. A threshold T_c needs to be set as follows that controls the change of the position of the sub-program:

$$\frac{S_{least}}{S_{current}} < T_c \tag{7}$$

where S_{least} and $S_{current}$ denote two elements of S_{cn} in Equation 6, which correspond to two main program corners that have the least average saliency and have the sub-program placed, respectively. S_{least} and $S_{current}$ will therefore be the same when the current main program corner (where the sub-program is placed) has the smallest average saliency. On the other hand, the sub-program will be placed to a new corner when S_{least} is much smaller than $S_{current}$ as controlled by the user-defined threshold T_c.

III. Experiments

A. Experiment Design

Preliminary experiments have been conducted based on the AIM dataset [7] that consists of 120 natural images of different characteristics. The evaluation is based on how accurate the proposed technique can predict the humans' judgment. In particular, each of the 120 natural images is first presented to eight naive subjects in sequence and for each subject, 1-2 corners of the presented image are voted for sub-program placement. As a result, 8-16 votes are collected for the four corners of the image under study, which results in a four-bin histogram h where each bin represents the number of votes at the corresponding image corner.

The prediction accuracy is evaluated based on our technique's predictions and humans' judgment as follows:

$$Acc_1 = \frac{\sum_{i=1}^{120}(h_i(p_i))}{\sum_{i=1}^{120}(f_{max}(h_i))} \tag{8}$$

where $f_{max}(x)$ denotes the maximum function and h_i denotes the voting histogram of the i-th image. The term p_i denotes the predicted corner by our proposed technique that has the least saliency. Therefore, Acc_1 evaluates how accurate our technique can predict the majority vote by the 8 subjects (over the 120 images). Considering that there are often more than one good corner for sub-program placement (each subject is actually allowed to vote two corners), another accuracy Acc_2 can be evaluated based on the first two predicted corners, i.e., the first two corners with the least saliency, as follows:

$$Acc_2 = \frac{\sum_{i=1}^{120}(h_i(p_i) + h_i(p_i'))}{\sum_{i=1}^{120}(f_{max}(h_i) + f_{smax}(h_i))} \tag{9}$$

where $f_{smax}(x)$ returns the second maximum element of the argument vector x and p_i' denotes the predicted main program corner that has the second least saliency.

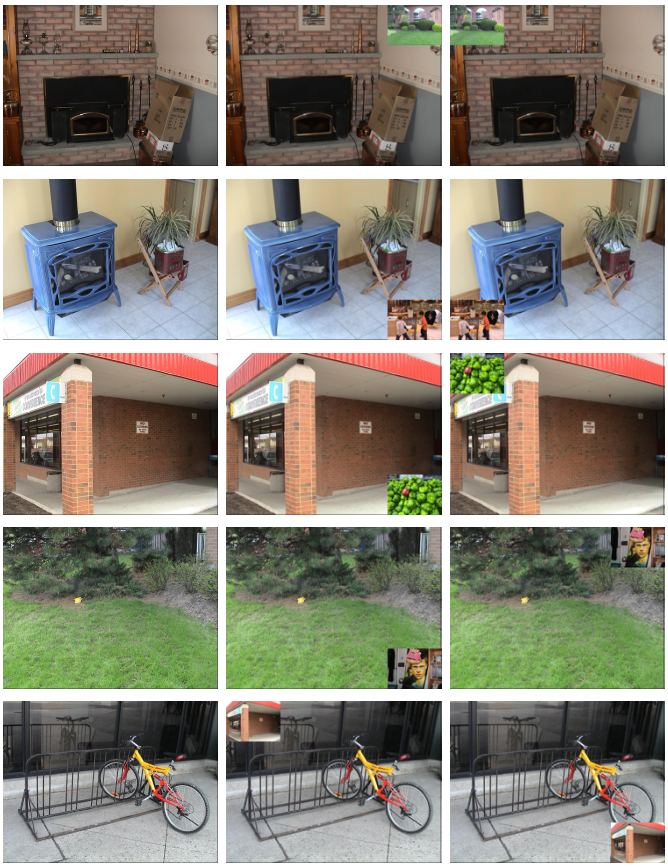

Fig. 3. Sub-program images are placed properly based on the visual saliency: For the five images in the first column from the AIM dataset [7], columns 2 and 3 show the corresponding composite images where sub-program images (randomly selected from the AIM dataset) are placed at the least and the most salient main program corners, respectively.

B. Experimental Results

Experiments show that the proposed technique predicts the humans' judgment accurately. For the 120 images tested, the average prediction accuracy reaches up to 81% for Acc_1 and 89% for Acc_2, respectively. Figs. 2a and 2b show the prediction accuracy Acc_1 and Acc_2 of the 120 individual images, respectively. As Fig. 2 shows, the prediction by our technique matches perfectly with the majority votes for many of 120 image studied, especially for the case of Acc_2. At the same time, there are also around 10-20 images that are not predicted accurately based on the proposed technique.

Fig. 3 shows examples where the first column show five sample images from the AIM dataset [7]. The second and the third columns show the composite images where the sub-program images (randomly picked from the AIM dataset) are placed at the least salient and the most salient main program corners, respectively. In particular, the first three images have perfect matching between our technique's prediction and the subjects' judgment. The last image has the worst matching because all 8 subjects voted the bottom-left corner as the best corner for sub-program placement. The fourth image lies between where most votes put the bottom-left corner as the best placement corner but 2 votes choose the bottom-right corner. As Fig. 3 shows, overall sub-program placed at the least salient corner (shown in the second column) blocks less useful information and presents a more natural view compared with sub-program placement at the most salient corner (shown in the third column).

C. Discussion

Some work will be further exploited. One aspect is to test the proposed technique over a larger dataset and at the same time recruit more subjects for user study. The correlation between our technique's prediction and the subjects' voting results over a larger scale will give a more comprehensive picture. Another aspect is to test the proposed technique on real video data. Some new evaluation metrics need to be designed because it is an extremely time-consuming process to carry out subject voting on video frames. We will study these topics in our future work.

IV. Conclusion

This paper presents an automatic and adaptive PiP technology that makes use of computational modeling of visual saliency. For each frame of the main program, a saliency map is computed efficiently which quantifies how probable a display region of the main program contains useful information and will attract humans' attention/eyes. The sub-program can thus be adaptively resized and placed to the display region that contains the least useful information. Preliminary experiments show the effectiveness of the proposed technology. Experiments on a public dataset show that the prediction accuracy reaches up to 88% compared with human subjects' voting.

References

[1] M. Honzawa and M. Koyama and T. Hibino and H. Miyashita and Y. Shiine, New picture in picture LSI enhanced functionality for high picture quality, IEEE Transactions on Consumer Electronics, vol. 36, no. 3, pp. 387–394, 1990.

[2] M. Brett and D. Wendel, High performance picture-in-picture (PIP) IC using embedded DRAM technology, IEEE Transactions on Consumer Electronics, vol. 45, no. 3, pp. 698–705, 1999.

[3] C. H. Li and C. N. Wang and T. Chiang, A low complexity picture-in-picture transcoder for video-on-demand, International Conference on Wireless Networks, Communications and Mobile Computing, vol. 2, pp. 1382–1387, 2005.

[4] C. H. Li and H. Lin and C. N. Wang and T. Chiang, A fast H.264-based picture-in-picture (PIP) transcoder, IEEE International Conference on Multimedia and Expo, vol. 3, pp. 1691–1694, 2004.

[5] L. Wang and Y. Dong and H. Bai and W. Liu and K. Tao, A word-based approach for duplicate picture in picture sequence detection, IEEE International Conference on Broadband Network and Multimedia Technology, pp. 286–290, 2011.

[6] J. Tsotsos, "Analyzing vision at the complexity level," *Behav. Brain. Sci.,* vol. 13, no. 3, pp. 423–445, 1990.

[7] N. Bruce and J. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *J. Vis.,* vol. 9, no. 3, pp. 1–24, 2009.

[8] S. Lu and J. H. Lim, "Saliency Modeling from Image Histograms," *European Conference on Computer Vision,* pp. 321–332, 2012.

[9] A. Borji, D.N. Sihite, L. Itti, Quantitative Analysis of Human-Model Agreement in Visual Saliency Modeling: A Comparative Study, *IEEE Transactions on Image Processing,* vol. 22, no. 1, pp. 55-69, 2012.