Experience based Sampling Technique for Multimedia Analysis

Jun Wang
Faculty of Information Technology and Systems,
Delft University of Technology
2628 CD Delft, The Netherlands
Tel: (31) 15 278 3830
j.wang@ewi.tudelft.nl

Mohan S Kankanhalli
Department of Computer Science, School of Computing,
National University of Singapore
Singapore 117543
Tel: (65) 6874 6738
mohan@comp.nus.edu.sg

ABSTRACT

We present a novel *experience based sampling or experiential sampling* technique which has the ability to focus on the analysis's task by making use of the contextual information from the environment. In this technique, *sensor samples* are used to gather information about the current environment and *attention samples* are used to represent the current state of attention. The task-attended samples are inferred from experience and maintained by a sampling based dynamical system. The multimedia analysis task can then focus on the attention samples only. Moreover, past experiences and the current environment can be used to adaptively correct and tune the attention. Experimental results have been presented to demonstrate the efficacy of our technique.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis-*Color, Motion, Sensor fusion, Time-varying imagery*. I.6.5. [**Simulation and Modeling**]: Model Development - *Modeling methodologies.*

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Experiential Computing, Sampling, Dynamical Systems, Attention.

1. INTRODUCTION

Multimedia processing often deals with spatio-temporal data which have the following attributes:

- They possess a tremendous amount of redundancy
- The data is dynamic with temporal variations with the resultant history
- Some data can be live with real-time processing requirements
- It does not exist in isolation – it exists in its context with other data. For instance, visual data comes along with audio, music, text, etc.

Any analysis approaches, ignoring the above attributes, will have two main drawbacks – *inefficiency* and *lack of adaptability*. The inefficiency arises from the inability to filter out the relevant aspects of the data and thus considerable resources are expended on superfluous computations on redundant data. If the ambient experiential context is ignored, the approaches cannot adapt to the changing environment. Thus, the processing cannot adapt itself to the task at hand.

The vision for experiential computing was introduced in [2], which envisages that multimedia analysis should also have the ability to deal with above problems while processing and assimilating sensor data. Therefore, we would like to articulate the following goal for multimedia analysis:

"*In an experiential environment, analysis is based on sensing the data from the environment. Based on the observations and experiences, collate the relevant data and information of interest related to the task of the analysis. Thus, the analysis process interacts naturally with the data based on its interests in light of the past experiences of that analysis."*

In this paper, we argue that multimedia analysis should be placed in the context of its environment. It should have the following characteristics: 1. the ability to "focus", i.e., to selectively process the data that it observes or gathers based on the context. 2. experiential exploration of the data. Past analysis should help improve the future data assimilation.

In order to achieve this, we introduce a novel technique called *experience based (or experiential) sampling*. As shown in Figure 1(a), by sensing the contextual information in the environment, only the relevant samples survive for performing of the final task. These samples precisely capture the most important data. What is interesting is the past samples influence future sampling via feedback. This mechanism ensures that the analysis task benefits from past experience.

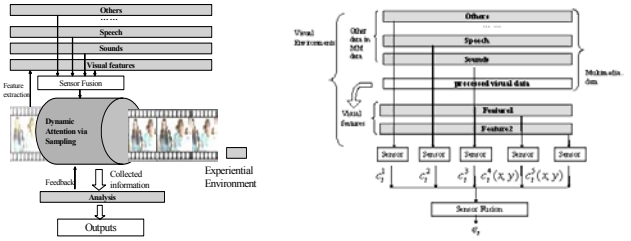
2. RELATED WORK

Since human perception is greatly aided by the ability to probe the environment through various sensors along with the use of the

MM'03, November 2-8, 2003, Berkeley, California, USA.

situated context, it has inspired context aware computing in the Human Computer Interaction research community [4].

In another aspect, the ability to "focus" the "consciousness" in human visual perception has inspired research in non-uniform representation of visual data. The computational modeling of visual attention [3] has been investigated. There is also the *foveation* technique [6] for maintaining a high resolution area of interest in an image. Unfortunately, the above saliency map based visual attention models and foveation approaches are image based that do not provide a mechanism to evolve and adapt attention dynamically. Contrastingly, our technique presented here naturally expresses the dynamics of attention of a system.



(a) **(b)**

Figure 1. (a) Experiential Sampling for multimedia processing. (b)The framework of sensing environments.

The *Sampling Importance Resampling* (SIR) method which can be used for modeling evolution of distributions was tackled in [7]. A special case of this framework has been used for the purpose of visual tracking in [5]. To the best of our knowledge, this is the first use of the sampling technique to maintain the dynamically evolving attention. Thus, unlike [5], the number of samples dynamically changes for the purpose of adaptively representing the temporal visual attention. This is in tune with the growing realization that computing systems will increasingly need to move from processing information and communication to the next step: dealing with insight and experience [2].

3. EXPERIENTIAL SAMPLING

We base our experience based sampling technique on video which is a prototypical multimedia data type.

3.1 Definitions

Our definition of experience is based on [2].

Experience in Multimedia Analysis: *is any information that needs to be specified to characterize the current state of the multimedia system. It includes the current environment, a priori knowledge of the system domain, current goals and the past states.*

In our framework, we allow the analysis to guide the attention on to regions or data of interest from the entire spatio-temporal data. The experience based sampling technique is defined as follows:

Experience Based Sampling (ES): *The current environment is first sensed by uniform random sensor samples and based on experiences so far, compute the attention samples to discard the irrelevant data. Spatially, higher attended samples will be given more weight and temporally, attention is controlled by the total number of attention samples.*

3.2 Sensor Sampling

In ES, we represent the environment e_t at time t as

$$e_t = \{S(t), A(t)\} \tag{1}$$

The environment e_t comprises of *sensor samples S(t)* and the *attention samples A(t)*. The sensor samples (SS) are basically uniform random samples at any time t which constantly sense the environment. The attention samples (AS) are the dynamically changing samples which essentially represent the data of interest at time t. The AS is actually derived dynamically and adaptively at each time instance from the SS through sensor fusion and the assimilation of the past experience. Once we have the AS, the multimedia analysis task at hand can work only with these samples instead of the entire multimedia data. Figure 1(b) shows our framework for doing sensor fusion within the experiential environments. It should be understood that our data assimilation process is sampling based. Not all data need to be processed.

In our framework, $S(t)$ is a set of $N_S(t)$ sensor samples at time t which estimates the state of the environment. These sensor samples are randomly and uniformly generated. Since we do not change the number of the sensor samples with time, we will drop the time parameter and N_S denotes the number of sensor samples at any point in time. $S(t)$ is then defined as:

$$S(t) = \left\{ s(t); \Pi^S(t) \right\} \tag{2}$$

where $s(t)$ depends on the type of multimedia data. For spatial data, $s(t) = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{N_S}, y_{N_S})\}$ at time t, this is the set of spatial coordinates of the sensor samples. These coordinates are generated randomly and uniformly at every time instance. $\Pi^S(t)$ is the associated weight or the importance of each sample which is represented as $\Pi^S(t) = \{\pi_1^S(t), \pi_2^S(t), \ldots, \pi_{N_S}^S(t)\}$. Now each $\pi_i^S(t)$ is obtained by performing sensor fusion of the q spatial cues $C(t)$ available from the multimedia data (like color, motion, texture etc.). Thus, the set of cues is given by $C(t)=\{c\_sp_1(t), c\_sp_2(t), \ldots, c\_sp_q(t)\}$ where each individual spatial cue $c\_sp_i(t)$ is given by $c\_sp_i(t) = \{(x_i^1, y_i^1, w\_sp_i^1), \ldots, (x_i^{N_S}, y_i^{N_S}, w\_sp_i^{N_S})\}$ Note that the coordinates x and y refer to the spatial coordinates of the sensor samples and w_sp_i refers to the weight of that particular cue at that sample coordinate. Now it can be easily seen that

$$\pi_i^S(t) = \sum_{j=1}^{q} \alpha_j \cdot w\_sp_j^i \tag{3}$$

where α_j is the importance of the j^{th} cue. So we basically employ the linear combination as the sensor fusion strategy. Note that if the cue is not spatial, then instead of the spatial coordinates, an appropriate reference (e.g. time) can be used for that cue. Usually, spatial cues are obtained from visual features. For instance, the motion cue is a spatial cue since it varies according to its spatial position. It can be simply defined as

$$w\_mot(x, y) = \left| i_t(x, y) - i_{t-1}(x, y) \right| \tag{4}$$

Here the weight is the absolute difference of corresponding pixel intensity values of two neighboring frames.

3.3 Attention Sampling

Attention in a scene can be represented by a multi-modal probability density function. Any assumptions about the form of this distribution would be limiting. We use a sample-based method to represent the attention. For example, in the one dimensional case, the attention is maintained by N samples $a(t)=[s^1(t),...,s^N(t)]$ and their weights $\Pi(t)=[\pi^1(t),..., \pi^N(t)]$. It provides a flexible representation with minimal assumptions. The number of samples can be adjusted to achieve a balance between the accuracy of the approximation and the computation load.

We represent the dynamically varying $N_A(t)$ number of attention samples $A(t)$ using:

$$A(t) = \{a(t); \Pi^A(t)\} \tag{5}$$

where $a(t)$ again depends on the type of multimedia data. For spatial data, $a(t) = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{N_A(t)}, y_{N_A(t)})\}$, is the set of spatial coordinates of the attention samples. $\Pi^A(t)$ is the associated weight or the importance of each sample which is represented as $\Pi^A(t) = \{\pi_1^A(t), \pi_2^A(t), \ldots, \pi_{N_A(t)}^A(t)\}$. Again, each of the $\pi_i^A(t)$ value is obtained by performing sensor fusion of the q cues $C(t)$ available from the multimedia data.

Note that for sensor samples, the number of samples was fixed a priori at N_S and these samples are generated uniformly and randomly at every time instant. But the number of attention samples $N_A(t)$ varies with time. We determine $N_A(t)$ based on the current state and the past experiences. The current state is essentially captured by the sensor samples. Therefore, the influence of the current state on the number of attention samples is by using the means of fusion of the sensor sample data:

$$N_A(t) = N_{Max} \frac{1}{n} \sum_{i=[t-n,t]} \frac{1-\exp(-\beta \cdot \mathrm{T}(t))}{1+\exp(-\beta \cdot \mathrm{T}(t))}, \qquad \mathrm{T}(t) = \frac{1}{N_S} \sum_{i=1}^{N_S} \pi_i^S(t) \tag{6}$$

where N_{Max} is the maximum number of samples the system can handle. The value n is the temporal neighborhood. The aim of averaging n number of recent temporal attention epochs is to suppress noise and to maintain temporal continuity. β is a scaling factor for the squashing function to normalize the relationship.

3.3.1 Dynamical Evolution of Attention Samples

Attention is inferred from the observed experiences coming from the experiential environments. That is, we try to estimate the probability density of the attention (which is the state variable of the system) at time t using $P(a_t|E_t)$. E_t consists of all the observed experiences until time t which is $E_t=\{e_1,...,e_t\}$, a_t is the "attention" in the scene and $a(t)$ is the sampled representation of a_t. We adopt first-order Markov process state-space model [7] to model the temporal continuity of attention samples.

Based on the state space model, the *a posteriori* density $P(a_t|E_t)$ can iteratively be obtained by knowing the observations (likelihood) $P(e_t|a_t)$, the temporal continuity (dynamics) $P(a_t|a_{t-1})$ and the previous state density $P(a_{t-1}|E_{t-1})$.

3.4 Experiential Sampling Technique

We are now ready to fully describe the technique based on the background developed so far.

Algorithm (Experience based sampling technique)

t=0:

Experiential environment sensing

1. Initialization: Set N_s , N_{Max} and set $N_A(0) = 0$
2. Sense the surveillance environment:
   2.1. Randomly create N_s number of sensor samples
   2.2. for each i, compute w_sp_i sensed from the environment.
   2.3. Perform sensor fusion using equation (3) for each cue.
   (If $N_A > 0$, the attention samples' weights also need to be updated by the same method as shown in step 2.2 and.2.3.)
3. Compute N_A from equation (6).
   3.2. If $N_A = 0$, set $t = t+1$ and go to step 2. Otherwise go to step 5.

Building the attention model using sampling

4. Create or resampling $N_A(t)$ number of attention samples $a(t) = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{N_A(t)}, y_{N_A(t)})\}$ and their weights
5. Attention driven analysis: Perform the actual video analysis task on the $N_A(t)$ attention samples. For example, if the task is face detection, do it on the regions of the attention samples. If the task is successful (e.g. a face is detected), increase the weights of the relevant attention samples and update the cue extraction model. This is the accumulating past experience.
6. Treat sensor samples N_S and attention samples $N_A(t)$ as a whole and normalize their weights $\pi^S(t)$ and $\pi^A(t)$ to make $\sum \pi^A(t) + \sum \pi^S(t) = 1$
7. Create a new set of attention samples for time $t+1$ by propagating the attention samples $N_A(t)$ by dynamical evolution.
8. $t = t+1$; go to step 2.

3.5 An Example: Face Detection

In the face detection application, the attended regions are those where the probability of finding a face is high. The face attention information can be inferred from the contextual information (cues from skin color, motion and speech) in the experiential environment. We use the *adaboost* face detector [1] as the multimedia analysis task. Face detection is only performed on the attention samples to achieve robust real time processing. Past face detection results serve as experiences to adaptively correct the attention samples and the skin color model in the attention inference stage. This adapts the face detector to cope with a variety of changing visual environments. The methods of obtaining the cues are now described:

Skin color cue: we use the 1-D histogram H_t (at time t) of hues (color) channel from HSV color system to represent skin color [8]. It could dynamically change according to the current environment. H_t is calculated from the face regions obtained by face detection in time $t-1$. Based on this loop, the skin color model is dynamically updated.

We define $w\_sp^i_{Skin}$ as the weight for the skin color cue. The stored skin color histogram H_t in time t is employed as a lookup table [8] to calculate the weight $w\_sp^i_{Skin}$ as shown below.

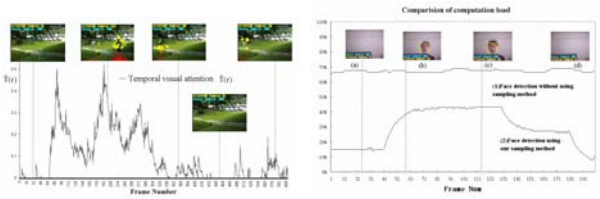
$$w\_sp^i_{Skin}(x, y) = lookup\ (x, y, H_t) \tag{7}$$

Motion cue: The weight of motion cue is defined as $w\_sp^i_{MT}$ which is obtained by equation (4).

Speech cue: Speech implies the possible existence of the face in the visual data. The face detector should work on more samples during the time slice when speech exists. In our method, we dynamically increase the number of attention samples when speech exists.

4. EXPERIMENTS

In this section, we present experiment results. The reader can see all the results with actual videos at our website [9].



(a) (b)

Figure 2. (a) Temporal motion attention in the MPEG 7 clip. (b) Comparison of computation load.

The evolution of temporal attention (represented by the number of attention samples) is shown in Figure 2(a). It shows that the temporal motion attention, calculated from the equation (6), evolves according to the motion activity in a pedestrian sequence.

(a.1) $N_A = 0$ (a.2) $N_A = 110$ (a.3) $N_A = 743$ (a.4) $N_A = 479$

(b.1) (b.2) (b.3) (b.4)

(c.1) Frame 2 (c.2) Frame 10 (c.3) Frame 42 (c.4) Frame 100

Figure 3. (a) Face attention (b) Skin color histogram H_t updated by feedback from past detection.(c) Speech experience (c.1) speech off N_A=0.(c.2) speech on. N_A becomes 1000. face detected.(c.3) speech off. N_A becomes 711(feedback from previous face detection). Face is detected (c.4) speech on. face detected. N_A becomes 1000.

It should be understood that all the results are obtained by only processing a few samples in the visual data. There is no need to processing the entire data. We use our sampling technique to solve the face detection problem. Sensor samples are employed to obtain the current visual environment from the skin color, motion and speech cues. The face attention is maintained by the attention samples. The face detector is only executed on the attention samples which indicate the most probable face data.

Figure 3(a) shows that depending on how much the attention is, the number of attention samples is different. Figure 3(b.1) is a face under normal light. Figure 3(b.2) shows its skin color saliency map calculated by the equation (7). Figure 3(b.3&4) are a shadowed face and its skin color saliency map. Figure 3(b.2&4) show that the feedback of face detector can update the skin color model H_t and make it more adaptive to the visual environment. Figure 3(c) shows that speech cue can help to create the attention samples when there is not any motion attention initially. As shown in Figure 3(c.3), even speech is off, relevant attention samples still survive by being giving higher weights from the feedback of the previous face detection.

We use a USB web camera to perform the real time face detection on a Pentium III 600MHz laptop. The graph of the computation load in this real time scenario is shown in Figure 2(b). In this experiment, curve 1 shows the computation load of the *adaboost* face detection while curve 2 indicates the computation load of our experience based sampling with *adaboost* face detector. This figure shows that by using our experience based sampling technique, computation complexity can be significantly reduced. In addition, the computation complexity also varies. When there is no face attention (see frame (a) and (d)), the only process is sensing by employing sensor samples. When the face comes (see frame (b) and (c)), the process includes attention samples and its load goes up.

5. CONCLUSIONS

In this paper, we describe a novel sampling based framework for multimedia analysis called experience based sampling. Inferring from this environment, the analysis procedure can select its data of interest while immediately discarding the irrelevant data. The results establish the efficacy of the sampling based technique. In the future, other applications like adaptive streaming and surveillance and more sources of different modalities will be further investigated.

6. ACKNOWLEDGEMENT

We are grateful to Ramesh Jain of GeorgiaTech for his discussions on Experiential Computing during his visit to NUS in October 2002. This work has originated from those discussions.

7. REFERENCES

[1] Viola, P., and Jones, M. J. Robust Real-time Object Detection. Tech. Rep. CRL 2001/01, Compaq Cambridge Research Laboratory, Cambridge, MA, 2001.

[2] Jain, R. Experiential Computing.2003. In Comm. of ACM July 2003.

[3] Itti, L., and Koch, C. Computational Modeling of Visual Attention. Nature Reviews Neuroscience, Vol. 2, No. 3, pp. 194-203, Mar 2001.

[4] Lieberman, H., and Selker, T. Out of Context: Computer Systems That Adapt to, and Learn From, Context. IBM Systems Journal 39, Nos. 3&4, 617-632 (2000, this issue).

[5] Isard,M., and Blake, A. Condensation-conditional Density Propagation for Visual Tracking. International Journal on Computer Vision, 29(1):5-28,1998.

[6] Chang, E-C., Mallat, S., and Yap, C. Wavelet Foveation. J. Applied and Computational Harmonic Analysis, volume 9, number 3, pages 312-335, October 2000.

[7] Doucet, A., Godsill,S.J., and Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. Statist. Comp., 10:197-208, 2000.

[8] Bradski, G. R. Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technique Journal, 2nd quarter'98.

[9] http://www.comp.nus.edu.sg/~mohan/ebs/ExpSampling.htm