Experiential Sampling in Multimedia Systems

Mohan S. Kankanhalli, Jun Wang, and Ramesh Jain, *Fellow, IEEE*

Abstract— **Multimedia systems must deal with multiple data streams. Each data stream usually contains significant volume of redundant noisy data. In many real-time applications, it is essential to focus the computing resources on a relevant subset of data streams at any given time instant and use it to build the model of the environment. In this paper, we formulate this problem as an experiential sampling problem and propose an approach to utilize computing resources efficiently on the most informative subset of data streams. We generalize the notion of static visual attention to multimedia data streams in a dynamical systems setting. The goal-driven generalized attention is maintained by a sampling representation that uses the current context and past experience for attention evolution. We have developed the theoretical background, algorithms and an evaluation measure for this technique. We have successfully applied this framework to the problems of traffic monitoring, face detection and monologue detection.**

Index Terms— **Dynamical Systems, Experiential Computing, Experiential Sampling, Sampling, Visual Attention**

I. Introduction

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

- It consists of a multiplicity of usually correlated data streams. Thus, 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.
- They possess a tremendous amount of redundancy.
- The data is dynamic with temporal variations with the resultant history.

However, many current approaches towards multimedia analysis do not fully consider the above attributes which lead to two main drawbacks – *lack of efficiency* 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. Hence speed-accuracy tradeoffs cannot properly be exploited. The lack of adaptability stems from the fact that the context of the data is often ignored. As a result, rigid computational procedures are employed for analyses that remain fixed when the environment itself is changing. Moreover, the context of multiple correlated data streams is not fully harnessed in order to perform the task at hand.

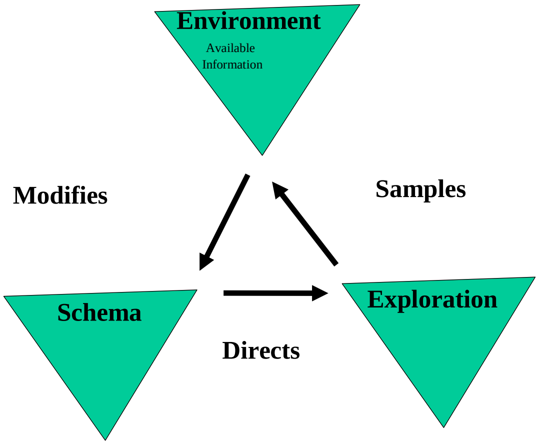


Figure 1. Neisser's Perceptual Cycle. (Based on Figure 2 in [44])

On the other hand, we have solid evidence that humans are superb at dealing with large volumes of disparate data using their sensors [6]. For instance, the human visual system is quite successful in understanding the surrounding environment at an appropriate accuracy quite efficiently. This is due to many factors [16]: the excellence of the physical visual sensing system, the richness of fusion information from perception, implicit understanding of every visual object, and the common understanding of how the world works. These attributes in the *experiential environments* [7] play an important role for the human visual perception to understand the visual scene accurately and quickly under fairly adverse conditions. The vision for experiential computing was introduced in [7], which envisages that multimedia analysis should also have the ability to process and assimilate sensor data like humans. Examples of such problems being currently tackled are speaker recognition [45], speech event detection [45], speaker change detection [45], monologue detection [46] and cross-modal information retrieval [47]. Many tasks like remote monitoring, understanding semantics and adaptive presentations also fall

M. S. Kankanhalli is with the School of Computing, the National University of Singapore. Kent Ridge, Singapore 119260 (Phone: (65) 6516-6738 Fax: (65) 6779-*4580* e-mail: mohan@comp.nus.edu.sg).

J. Wang is with the Faculty of Electrical Engineering, Mathematics and Computer Science, the Delft University of Technology, the Netherlands. (e-mail: j.wang@ewi.tudelft.nl).

R. Jain is with the Bren School of Information and Computer Sciences, the University of California, Irvine. (e-mail: jain@ics.uci.edu).

under this paradigm. Therefore, we would like to articulate the following goal for such multimedia systems:

"*In an experiential computing environment, the system should sense the data from the environment. Based on the observations and experiences, the system should collate the relevant data and information of interest related to the task. Thus, the system interacts naturally with all of the available data based on its interests in light of the past states in order to achieve its designed task.*"

It is apparent that many current multimedia systems approaches ignore their contextual environments and do not have the ability to adapt to the environment. They, instead of processing relevant data, perform pervasive non-focused computations. In this paper, we argue that like human perception, multimedia analysis should be placed in the context of its environment. It should have the following characteristics:
1. The ability to "focus" (have attention), i.e., to selectively process the data that it observes or gathers based on the context.
2. The ability to perform experiential exploration of all of the available data streams.

To formulate the problem precisely, we will define the scenario more formally in section III and IV. Our ideas are articulated using some important concepts that Neisser [44] introduced in 1976 in his work on the notion of perceptual cycle to model how people perceive the world. He presented the idea that a perceiver builds a model of the world by acquiring specific signals and information to accomplish certain tasks in the natural environment. The perceiver continuously builds a schema that is based on the signals that he has received so far. This schema represents the world as the perceiver sees it at that instant. The perceiver then decides to get more information to refine the schema for accomplishing the task that he has in mind. This sets up the cycle as shown in Figure 1. The perceiver gets signals from the environment, interprets them using the current schema, uses the results to modify the schema, uses the schema to decide to get more information, and continues the cycle until the task is done.

Our contributions in this paper are as follows. We introduce the experiential sampling framework to solve the problems of adaptation and efficiency when dealing with multiple data streams in a multimedia system. What it effectively achieves is the development of the idea of generalized attention. This key concept extends the notion of static visual attention to any type of multimedia data. Thus attention is generalized to data-streams such as video and also to other data-streams which need not even be perceived by humans. Moreover, it is modeled as a time-varying continuous function which is approximated by a sampling representation. Also, it explicitly recognizes the presence of multiple correlated data-streams. We have developed the theory and demonstrate its efficacy for several applications.

The paper is organized as follows. After surveying the related work in section II, we first start to define the problems and our solution: Experiential Sampling in section III.A. To effectively perform experiential sampling, we then propose a sampling-based dynamical attention driven analysis in the remaining part of the section. In section IV we extend our approach to handle multiple data streams. In section V we apply our framework to three applications and the corresponding experiments are described in section VI. Finally, we conclude our paper in section VII.

II. RELATED WORK

Since human perception is greatly aided by the ability to probe the environment through various sensors along with the use of the situated context, it has inspired context aware computing in the human computer interaction research community [15]. The basic idea there is to help the computer respond more intuitively to the human user based on the context. A comprehensive review of context aware computing can be found in [15, 16]. Our thrust is towards making multimedia analysis systems interact naturally with multiple data streams by considering the current context and past history.

The ability to "focus" the "consciousness" in human visual perception has inspired research in non-uniform representation of visual data. The basic idea is to do adaptive sampling which is basically the selection of the most informative samples in a data stream. Visual attention in human brains allows a small part of incoming visual information to reach the short-term memory and visual awareness, consequently providing the ability to investigate more closely. There is a growing interest in the study of the visual attention phenomenon by psychologists [6, 8]. The phenomenon of *inattentional blindness* is particularly interesting in which human subjects have been found not to observe major objects when paying attention to some other objects [8]. It has been found to be a useful aid in finding evidence for resolving the controversy between the conflicting spotlight and object models of visual attention. The spotlight model hypothesizes that visual attention is concentrated in a small contiguous region ("spotlight" or "zoom lens") which can move around in the field of vision. In contrast, the object model states that attention can be focused on spatially discontinuous objects (or a group of disparate objects). Experimental evidence seems to suggest that the human visual attention mechanism appears to be a combination of both models [6]. Computational modeling of visual attention has been investigated for potential usages in planning and motor control [14], video summarization [11] and object recognition [12]. The computational model of visual attention maintains a two-dimensional topographic saliency map by employing a bottom-up reasoning methodology [10]. Reference [13] attempts to model the influence of high-level task demands on the focal visual attention in humans. There is also the *foveation* technique [19, 21] for maintaining a high-resolution area of interest in an image. A uniform-resolution image can be foveated to transform into a spatially varying resolution image by either a log-polar [19] or a wavelet approach [20]. All these approaches recognize the need for doing adaptive sampling. But their approach is usually static. However, in humans, attention varies with the nature of task. In addition, visual attention is adaptive. This means it will vary depending on the visual environment and has a self-corrective mechanism utilizing experiences. Interestingly enough, psychologists have observed that unexpected objects have a lower probability of being observed when attending other objects [6]. This strongly suggests the human perceptual system has a concrete notion of history which is encoded as a priori probabilities. Thus, attention will vary over time. 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 sampling framework naturally expresses the dynamics of attention of a system. What is particularly appealing is that the attention states as well as the state-transitions are captured as a closed-loop feedback system. Moreover, the earlier adaptive sampling approaches consider only a single data stream. Our framework explicitly considers multimedia which consists of a multiplicity of correlated data streams. And these streams need not be audio or video – it can be any type of multimedia data including data not perceived by human sensors like infrared or motion sensors.

The Sampling Importance Resampling (SIR) method which can be used for modeling evolution of distributions was proposed in [26]. The dynamics aspects were developed in [27]. In a SIR filter, a set of particles, which move according to the state model, multiply or die depending on their “fitness” as determined by the likelihood function [41]. A general importance-sampling framework that elegantly unifies many of these methods has been developed in [25]. A special case of this framework has been used for the purpose of visual tracking in [18]. Though we also utilize the sampling method, we use it to maintain the generalized notion of attention. To the best of our knowledge, this is the first use of the sampling technique to maintain the dynamically evolving attention. Thus, unlike [18], 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 [7]. One of the key technical challenges in experiential computing is information assimilation, i.e., how to process in real time the disparate data received by multiple sensors. Our research in this paper aims to provide a sampling based dynamical framework to tackle this problem in the multimedia domain.

III. Experiential Sampling

Let us assume that we are given $S_1, S_2 \ldots S_n$ synchronized data streams belong to the space of multimedia data streams $\mathcal{M}$. These data streams have K types of data in the form of image sequence, audio stream, motion detector, annotations, symbolic streams, and any other type that may be relevant and available. Also, metadata for each of the streams $MD_1, MD_2 \ldots, MD_n$ is available in the context of the environment. This metadata may include things like location and type of the sensor, viewpoint, angles, camera calibration parameters or any other similar parameters relevant to the data stream. Since a data stream is usually not directly very useful, some feature detectors must be applied to each data stream to obtain features that are relevant in the current environment. We assume that the multimedia system is a discrete time (or a sampled continuous time) dynamical system. When features are based on time intervals, they will be considered as detected at the end of interval, which is denoted as t, where $t=1,\ldots,T$.

Given the above data environment, there are now many very interesting problems that one faces, including the following that are directly relevant to the main theme that we wish to address in this paper:

- How to focus on the most relevant data in a particular data stream?
- How to focus on the most relevant data in multiple correlated data streams?
- For the given task, what is the minimum number of data streams required?
- How does one sample the data streams? How can one minimize sampling for maximizing the efficiency?
- Can one use alternate data streams to perform the same task with different costs?
- Given that M streams are necessary for a given task, how does one combine the information from the data streams?

We believe that this issue of determining which data streams are relevant and even among those streams which ones provide most relevant information at any given moment is a very important problem that needs to be addressed and has been ignored in the current literature. Current multimedia systems, usually start with the assumption that there is a given set of n data streams, unfortunately in most cases $n=1$ making it a signal analysis rather than a multimedia problem, and one must deduce or extract all information from there to build the schema representing the environment. There are other issues related to semantics and indexing that we do not wish to address here. Now we are ready to define what experiential sampling is and then address this in the remaining part of the paper.

A. Defining Experiential Sampling

Experience is defined as the accumulation of knowledge or skill that results from direct participation in events or activities [51]. Direct participation implies having access to the environment of the event in order to observe it using all potential sensory mechanisms available to the perceiver or the experiencer. In such an environment, the experiencer is driven by the goal of maximizing the efficacy of building the schema with minimal efforts to accomplish the most efficient mechanism to accumulate the knowledge. This task translates into selecting appropriate data streams at any given time, based on the current schema, for paying attention.

We define experiential sampling as the process of identifying the most relevant data stream among the available streams at a given instant to utilize for interpretation to refine the current model of the environment.

In this section, we introduce our experiential sampling technique. There are two major components in this technique. The first is how to sense and fuse experiences (contextual information) in the experiential environment. The second is how to build a dynamic attention model to select the data (or region) of interest.

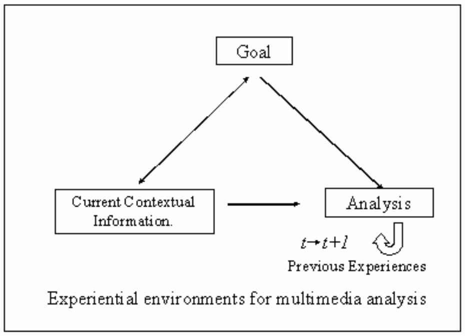

Goal
Current Contextual Information.
Analysis
$t \to t+1$
Previous Experiences
Experiential environments for multimedia analysis

Figure 2. Experiential Environment Relationships.

1) Experience

Our definition of experience is based on [7, 51].

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.*

Although experience and experiential environments are domain dependent and their components are not clear in general, we define three main components as follows:

Current contextual information: *is the current existing information about the environment that needs to be specified to characterize the current state of the multimedia system with respect to the current goal.*

Past experience: *is the accumulated experience of the multimedia analysis task performed in the past.*

Goal: *is the purpose of the current analysis task. It is used to define what the related experiences are, and what analysis technique should be employed to accomplish the task.*

There are some relationships among these components. The current contextual information can be characterized by features extracted from the visual scene and other accompanying multimedia data (audio, speech, text etc.). The current goal and prior knowledge provide a top-down approach to analysis. It also determines which features of the visual scene and other accompanying data type should be used to represent the environment. The past experiences encapsulate the experiences till the current state. The relationships are shown in Figure 2. These relationships can help us define the experiential environment when we perform multimedia analysis. More importantly, when we consider the experiential environment, the analysis process systematically integrates the top-down and bottom-up approaches by employing the context and history.

2) Goal oriented attention from experiential environments

As mentioned in the introduction, we allow the system to sense the data from the experiential environment. Based on the observations and experiences, it collates the relevant data and information of interest related to the task of the analysis and discards irrelevant information. In this regards, a central problem is the allocation of the goal oriented attention within the experiential environments. Note that attention is intimately related to the goal – generic attention does not make sense. We base this discussion on video which is a prototypical multimedia data type. Moreover, in this section, we will first concentrate on developing ideas for a single data stream. We will generalize this to multiple data streams in section IV.

In our framework, we allow the analysis task to guide the attention onto regions or data of interest from the entire spatio-temporal data. We first introduce a vector to represent the spatial position of the goal oriented attention in a given time t as:

$$a_G(t) = [x, y]'$$

where x=1,…,X and y=1,…,Y are spatial coordinates and t=1,…,T is temporal position. $a_G(t)$=$[x,y]'$ indicates the current a*ttended* position is $[x,y]'$ in a time slice t. ' denotes the transpose operator. Without loss of generality, the stream dimension {*1*,…,*n*} can be further added when multiple streams are considered, while the spatial coordinates *x, y* can be dropped when non-spatial streams are considered.

To infer the attention from the environment, we define the current contextual information with respect to the attention at the time t as:

$$e(t) = \{e(x, y, t) \mid x = 1, ..., X; y = 1, ..., Y\}$$

where again *x, y* are spatial coordinates and t is the temporal position. It includes any contextual information which could help in inferring the goal oriented attention (we will show later it is a combination of different feature cues). Therefore, it can also be considered as the measurement (e.g. motion, colors etc) of the attention with respect to the given spatial coordinates and time. For this, the values of the elements are required to be normalized to the range of [0,1). The sum total of accumulated contextual information for the attention is defined as $E(t)$=$\{e(1),\ldots,e(t)\}$.

In this paper, we attempt to infer the attention from the experiential environment. By employing probabilistic reasoning, we define the *a posteriori* probability $P(A_G(t)|E(t))$ with $A_G(t)$=$\{a_G(1),\ldots,a_G\ (t)\}$ as the goal oriented attention up to time t. For real time applications, we need to estimate $P(\mathrm{a}_G(t)|E(t))$ rather than $P(A_G(t)|E(t))$. Here we assume that the attention at each spatial position {*x, y*} is only dependent on the context measurement around the position {*x, y*}. Then we have the following equation:

$$P(a(t) = [x, y]' \mid E(t)) = P(a(t) = [x, y]' \mid E(x, y, t))$$

Note that this notion of attention is a generalization of visual attention [10] in the sense it can be applied to any multimedia stream which may be non-visual. For example, this definition subsumes the notion of aural attention which is also related to the cock-tail party effect in digital audio processing. And this generalized attention concept can be applied to non-visual, non-audio data as well. Also, it is a phenomenon which dynamically varies with time unlike the notion of static image attention dealt by the bulk of the visual attention literature. Moreover, attention is always goal-driven.

B. Goal oriented attention driven analysis

In this section, we formulate the goal oriented attention driven analysis by using the Bayesian framework.

1) Signal to symbol matching

The central problem of multimedia content analysis is the signal to symbol matching. Fundamentally, it involves mapping the relationships between the digitized spatial-temporal data and *semantic symbolic identity.* We define this mapping function as S_M. Many analysis approaches only unite the *local content intrinsic* features to perform content analysis. Here "local" and "intrinsic" refer to the fact that these features come from the information of the symbolic identity itself. By employing probabilistic reasoning, such analysis approaches, which we classify as *local feature centered approaches,* can be expressed as maximizing the *a posteriori* probability

$$SID = S_M(f_L) = \arg\max_H P(H \mid f_L) \tag{1}$$

where *SID* is the estimated true semantic symbolic identity, f_L denotes the local intrinsic features and H is the hypothesis of the symbolic identity. For instance, in face detection, the hypothesis is face region and non-face region. Note that in this section, since we only discuss the situation within a given time slice, we simply drop the entire notation related to time.

For instance, the local feature centered approach, which has been the dominant theme in computer vision for many years, exclusively uses object intrinsic features to represent the objects and to perform object detection/recognition tasks [2, 3, 4, 5, 33].

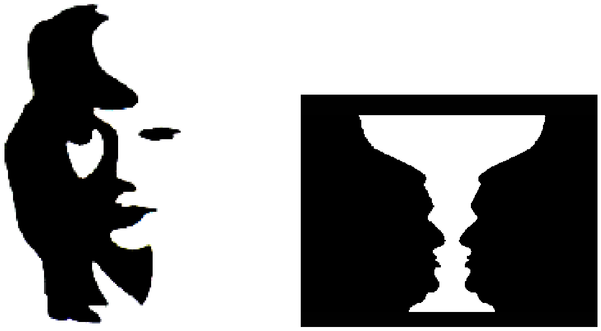


Figure 3. Attention helps analysis. (a) A woman's face or a saxophone player. (b) A vase or head to head?

2) A Bayesian framework for integrating attention

However, the symbolic identities physically exist in their environment and not in isolation. It is a well-known fact that focus of attention plays an important role in the human visual system to understand the visual scenes. It can selectively process the data that it observes or gathers based on the context. The illusions in Figure 3 shows that the role of goal oriented attention in top-down visual system increases in importance and can become indispensable when the viewing conditions deteriorate or when ambiguity exists. In Figure 3 (courtesy of http://members.lycos.co.uk/brisray/optill/othis.htm), if we look at the entire image (process all the data in the image), we maybe confused whether there is a saxophone player or a woman's face. However, if we just focus our attention on the dark region, we instantly identify that there is a saxophone player. Contrarily, if we focus our attention on the white region towards the right, it could convince us that it is a woman face. Similar ambiguity exists in the second illustration of Figure 3 as well.

In some respects, in the visual scene, the object intrinsic features and their differences with respect to the global environment features make the object distinct from the environment. In the early vision of human brain, by making use of these features, goal driven focus of attention allows human visual perception to quickly become aware of objects of interest from large volumes of visual data in the visual environment [10, 12, 13, 34]. Recently, Jordan et al. in [32] have stated that contextual information plays an important role to make reliable inferences in situations where the measurements produce ambiguous interpretations. Torralba [31] mainly interpreted the scene information as context and developed contextual priors for object detection.

Therefore, it is absolutely necessary to build in the attention phenomenon into the multimedia analysis process. Based on this, we extend the signal to symbol mapping function formulated in equation (1) by adding the attention A. Therefore, the multimedia analysis problem as shown in equation (1) essentially becomes maximizing the symbolic identity's posterior probability $P(H|f_L,a)$. That is the probability of identity H, given the current intrinsic feature F_L and the current attention a. According to this model, we will use a Bayesian reasoning framework to embed the attention and experiential environment E into the multimedia analysis tasks. Bayes' theorem can be used to factorize the probability $P(H|\, f_L,a)$.

$$P(H \mid f_L, a) = \frac{P(f_L \mid H, a)}{P(f_L \mid a)} P(H \mid a) \tag{2}$$

The identity feature is directly affected mainly by the identity. There is very little influence coming from the attention. Here we assume that the local feature f_L is independent of the attention A. Therefore the equation (2) can be rewritten as:

$$P(H \mid f_L, a) = \frac{P(f_L \mid H)}{P(f_L)} P(H \mid a) \tag{3}$$

Therefore, the probability of the hypothesis H given local feature f_L and the attention a is factorized into two components. The first component is the effect from the local feature f_L on hypothesis H. The second component is the attention oriented priors on the hypothesis.

It can also be further factorized. Therefore the above equation becomes,

$$\begin{aligned} P(H \mid f_L, a) &= \frac{P(f_L \mid H)}{P(f_L)} \cdot \frac{P(a \mid H) P(H)}{P(a)} \\ &= \frac{P(f_L \mid H) P(H)}{P(f_L)} \cdot \frac{P(a \mid H)}{P(a)} \end{aligned} \tag{4}$$

In the end, we have the final equation,

$$P(H \mid f_L, a) = P(H \mid f_L) \cdot \frac{P(a \mid H, E)}{P(a \mid E)} \qquad (5)$$

where we treat attention in the experiential environment E. Therefore we add the dependence of E in the probability of the attention. The numerator of the second component in equation (5) is the attention aroused by both the symbolic identity and its experiential environment. The denominator of the second component is the attention aroused by the experiential environment only. By this denominator, the attention aroused by the environment is inhibited. Therefore, we can see that these arousing and inhibiting attentions can contribute to the multimedia analysis task. We call this attention *goal-driven attention*. From section III.A.1, our experiential environment E includes the goal. It means the goal about obtaining the symbolic identity *SID* has been considered in this framework. Therefore we denote

$$P(a_G \mid E) = \frac{P(a \mid H, E)}{P(a \mid E)} \qquad (6)$$

We can now rewrite equation (1) as.

$$\begin{aligned} SID &= S_M(f_L, a_G) \\ &= \arg\max_H P(H \mid f_L, a_G) \\ &= \arg\max_H P(H \mid f_L) \cdot \frac{P(a \;\mid H, E)}{P(a \mid E)} \\ &= \arg\max_H P(H \mid f_L) \cdot P(a_G \mid E) \end{aligned} \qquad (7)$$

From the above equation, we can see that the final posterior probability has two components. The first component is the local posterior probability which can be inferred from the symbolic identity's local features. In general, local feature centered approaches exclusively concentrate on obtaining this probability. The second component is the impact coming from the goal-driven attention. This part serves as an amplification factor on the identity centered approach of the first component.

C. *Sampling based dynamical attention driven analysis*

From above analysis, we can see that the attention helps the multimedia analysis task. Given our task in this paper is identifying the most relevant data stream among the available streams at a given instant, based on the above discussion, we treat the information which makes the term $P(a|E)$ (For the sake of simplicity, we will drop the subscript G later on. However, $a(t)$ and $A(t)$ will *always* denote goal oriented attention.) smaller as the *irrelevant information*. We discard it since we would not like to do the time-consuming processing (shown in equation (1)) on the irrelevant information which give a lower value for $P(a|E)$. Contrarily, we treat the information which gives higher value on $P(a|E)$ as the *relevant information* and perform detailed processing (to obtain $P(H|f_L)$) on it.

There are two steps involved in performing this attention driven analysis. Firstly, we use samples and their weights to dynamically maintain the attention with respect to the experiential environment. Secondly, we propose the use of a re-sampling approach to obtain relevant information captured in the samples, which is employed to perform the multimedia analysis task based on the attention. The (visual or otherwise) 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. However, not making any assumption about this distribution leads to intractability of computation.

All the past work on extraction of visual attention uses the saliency map representation to denote the visual attention in an image [9, 10, 12, 13]. The saliency map is built by either linear combination of features or by training [28]. There are two weaknesses of these approaches. First, most of the methods perform bottom-up computation which does not take into account the past experiences of the system [10]. Secondly, the temporal variation of attention is not modeled.

On the other hand, based on the Sequential Importance Sampling (*SIS*) algorithm [25, 29, 34], we use *attention samples* to represent the probability of attention $P(a|E)$. For example, in the one dimensional case, the probability of attention $P(a|E)$ is maintained by N attention samples $AS(t)=[as^1(t),\ldots,as^N(t)]$ as well as their weights $\Pi(t) = [\pi^1(t),\ldots,\pi^N(t)]$ as shown in Figure 4. It provides a flexible representation of the probability with minimal assumptions. The number of samples employed can be adjusted to achieve a balance between the accuracy of the approximation and the computation load. Moreover, it is easy to incorporate this representation within a dynamical system which can model the temporal continuity of attention if we consider each sample as a particle and each particle having its own dynamics.

In this sampling representation, the location of samples and their associated weights are employed to represent the attention probability $P(a|E)$. This means that for a particular region (say around a certain position x in Figure 4), the more samples fall into this region and the higher their weights are, the higher is the probability of attention in this region. Apparently, the probability distribution is not fully represented by the distribution of the samples. It also relies on the weights of the samples. However, since we use attention to get the relevant information, we would like the probability of the attention be fully represented by the distribution of the attention samples, not partially on their weights. That is the highly attended regions should have more samples and vice versa. A re-sampling method is therefore introduced to let only the distribution of samples reflect the distribution of attention. In addition, since the attention is inferred from experience (which will be discussed in section III.C.3) and experience itself encapsulates the goal and environment, our sampling based dynamical attention model systematically integrates the top-down and bottom-up approaches.

The entire probabilistic notation used in this section is shown in Figure 5. In the remaining part of this section, we first provide the solution to the static case in section III.C.1. We then extend the solution to the dynamic case in section III.C.2. We treat attention as a Bayesian inference problem and develop an approach to obtain relevant information from the approximated dynamical attention probability. In section III.C.3, 4 and 5, a sampling based approach is introduced to maintain the probability of the dynamic attention. Important concepts like *environment sampling*, *sensor sampling*, *attention sampling*, as well as *attention saturation* are described in section III.C. 3, 4, and 5, respectively.

1) Static attention driven analysis

In our sampling technique, the second factor in the equation (7), called *goal driven attention*, is represented by samples and their associated weights. Those samples which have higher weights can survive as the samples in the next time slice. Therefore, samples represent the higher task driven attention of equation (7). In contrast, other regions having less attention value have less impact on equation (7). Based on this, we perform the multimedia analysis task (indicated by $P(H|\ f_L\ )$) only on these samples and treat other data as the irrelevant data which is to be discarded from the analysis point of view.

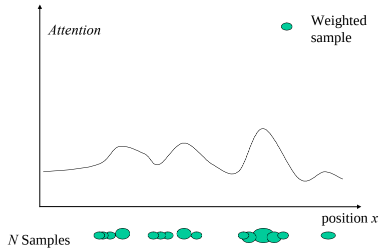


Figure 4. The multi-modal attention probability can be represented by *N* samples *AS(t)*=[a*s*^{1}*(t)*,…,*as*^{N}*(t)*] and their weights $\prod(t)$ =[$\pi^1(t)$,…, $\pi^N(t)$].

The entire algorithm including the dynamics will be discussed in the next section. But, first let us consider the simple static case. Here we assume that we know $P(a|E)$ and we are able to simulate N *i.i.d.* (independently and identically distributed) random samples $\{a_1, a_2, a_3, \ldots, a_N\}$ according to $P(a|E)$. For instance, in a spatial case, they are a set of spatial coordinates. Their associated weights $\{w_1, w_2, w_3, \ldots, w_N\}$ can be obtained by $w_i = P(a|E)$. So the weight w_i is directly proportional to attention probability $P(a|E)$ such that the sum of the N weights is equal to the total attention at that time.

$P(A(t)\,|\,E(t))$: The *a posteriori* probability of attention given the contextual information up to now.

$P(a(t)\,|\,E(t))$: The *a posteriori* probability of attention at time *t* given the contextual information up to now.

$P(e(t)\,|\,a(t))$: The likelihood of the attention at time *t* with respect to the current contextual information.

$P(a(t)\,|\,a(t-1))$:The dynamics of the evolution of attention.

Figure 5. The probability distributions.

In this case, the distribution of the samples actually reflects the distribution of the attention probability. Differing from the classical perfect Monte Carlo sampling [25] which uses samples to approximate the distribution and consequently get its expectation, we use the sampling method to maintain our attention probability and consequently collect relevant information while discarding irrelevant information. By choosing a proper number of samples, the samples will only exist in the higher attended regions as shown in Figure 6(b) since the high attention data is given by the distribution $P(a|E)$. These samples intuitively represent the relevant information to be processed. Note that selection of the number of random samples N depends on current overall attention (measured by the *attention saturation* which will be introduced later) as well as the trade off between the computation load and the representation accuracy.

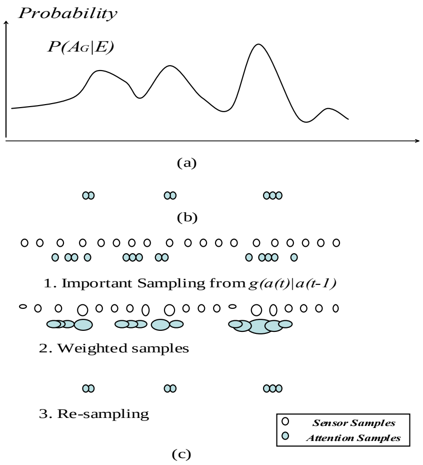


Figure 6. A sampling based dynamical attention model. (a) Attention (b) Samples as relevant information (static case) (c) Samples as relevant information (dynamic case).

```
The algorithm Static_SBADA (P(a|E))
Results= {}
begin
1.  Draw N number of random i.i.d. samples {a_1, a_2, a_3 ,…, a_N} with respect to P(a|E). The associated weights {w_1, w_2, w_3 ,…, w_N } are calculated by the equation w_i = P(a|E).
2.  for i = 1 to N do /* for each random sample*/
        begin
3.      if  w_i >w_T  then do
            begin
4.          F_L  ← feature extraction from the location of a_i in the multimedia data.
5.          Compute P(H=1|f_L)
6.          Calculate P(H=1|f_L, a)= P(H=1|f_L).w_i
7.          SID = 1? 0: P(H=1|f_L, a)>0.5
8.          Results=Results + {a_i, SID}
            end
        end
end
output(Results)
```

Figure 7. The Static SBADA algorithm.

When we know the probability $P(\mathrm{a}|E)$, the algorithm for this *Static Sampling Based Attention Driven Analysis (Static SBADA)* approach is shown in Figure 7. For the sake of simplicity, in this illustration, we consider the two-class classification problem and assume we have two hypothesis *H=1* and *H=0* (e.g. face and non-face) and we know the probability *P*(a|*E*) for *H=1*(Note that it can be easily extended for the multiple-class classification problems).

It is clear that our method allows performing of the actual multimedia analysis task on the *N* samples. For example, if the task is face detection, it can be performed on the regions of the thresholded samples (by the threshold w_T in Figure 7). But when dealing with spatio-temporal multimedia data, the focus of attention dynamically varies not only along the spatial axes but also along the temporal axis. A dynamic attention model needs to be investigated in order to achieve effective and efficient spatio-temporal data analysis.

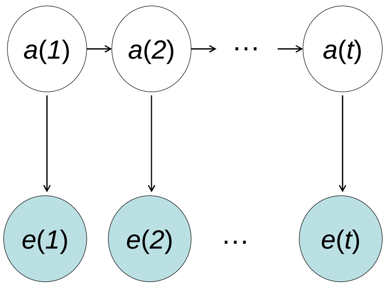


Figure 8. State-space model for attention.

2) Dynamical attention driven analysis

In this section, we aim to infer the current attention *a(t)* from the contextual information *E(t)* until the time *t*, i.e. calculate the *a posteriori* probability *P*(*a(t)*|*E(t)*).

a) Dynamical evolution of attention

The attention is inferred from the observed experiences coming from the environment. 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*)). Note that *E*(*t*) consists of all the observed experiences until time *t* which means *E*(*t*)={*e*(1),...,*e*(*t*)}, and *a*(*t*) is the "attention" in the scene. Attention has temporal continuity which can practically be modeled by a first-order Markov process state-space model [29] as shown in Figure 8. The value of *a*(*t*) may not be observed though the experience *e*(*t*), which influences the attention *a*(*t*), is observable. In this model, the new state depends only on the immediately preceding state, independent of the earlier history. This still allows quite general dynamics, including stochastic difference equations of arbitrary order. Therefore,

$$P(a(t)\,|\,a(t-1),...,a(0)) = P(a(t)\,|\,a(t-1)) \qquad (8)$$

Our target, the posterior probability *P*(*a*(*t*)|*E*(*t*)), can be factorized by using the Bayes' rule. The formalization is shown in equation (9).

$$\begin{aligned} &P(a(t)\,|\,E(t)) \qquad (9)\\ &= P(a(t)\,|\,e(t), E(t-1))\\ &= \frac{P(e(t)\,|\,a(t), E(t-1))P(a(t)\,|\,E(t-1))}{P(E(t)\,|\,E(t-1))}\\ &= \frac{P(e(t)\,|\,a(t))P(a(t)\,|\,E(t-1))}{P(E(t)\,|\,E(t-1))}\end{aligned}$$

where $(P(e(t)\,|\,a(t), E(t-1)) = P(e(t)\,|\,a(t)))$

$= kP(e(t)\,|\,a(t))P(a(t)\,|\,E(t-1))$

where $\left(k = \frac{1}{P(E(t)\,|\,E(t-1))} \right)$

Since we are interested in the attention *a(t)*, *k* becomes a normalization factor which does not depend on the attention.

The prior probability *P*(*a*(*t*)|*E*(*t*-1)) in the equation (9) can be further formulated as follows (a detailed explanation can be found in [18]).

$$\begin{aligned} &P(a(t)\,|\,E(t-1)) \qquad (10)\\ &= \int_{a(t-1)} P(a(t)\,|\,a(t-1), E(t-1))P(a(t-1)\,|\,E(t-1))da(t-1)\\ &= \int_{a(t-1)} P(a(t)\,|\,a(t-1))P(a(t-1)\,|\,E(t-1))da(t-1)\end{aligned}$$

According to equation (10), *P*(*a*(t)|*E*(*t-1*)) is dependent on the probability *P*(*a*(*t*)|*a*(*t-1*)) and *P*(*a*(*t-1*)|*E*(*t-1*)).

From the above two equations, we know that the posterior density *P*(*a*(*t*)|*E*(*t*)) can be iteratively obtained by knowing the observation (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*)) . This procedure is succinctly captured in Figure 9. Initially we assume that the *P*(*a*(*1*)|*E*(*1*)) is zero.

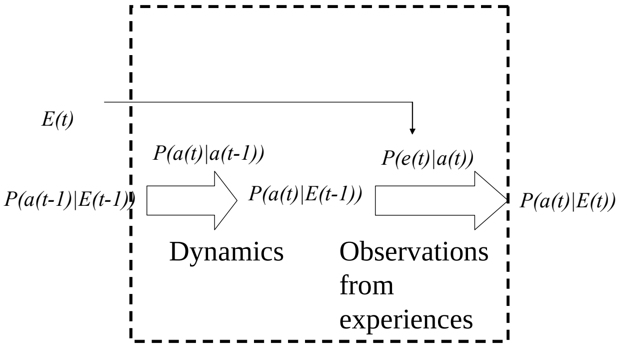


Figure 9. Iteration of calculating attention state density *P(a(t)|E(t))* in the state-space model. By knowing the previous state density *P(a(t-1)|E(t-1)* and current experience *e(t)* , *P(a(t)|E(t))* can be approximated by a sampling method in the form of samples.

During each iteration, the three probabilities for obtaining the posterior density *P*(*a*(*t*)|*E*(*t*)) are calculated as follows:

***P*(*a*(*t*)|*a*(*t-1*)):** Since we have assumed a Markov state-space model, the dynamics of attention evolution is described by a stochastic differential equation where the deterministic part models the system knowledge and the stochastic part models the uncertainties. Thus the dynamics *P*(*a*(*t*)|*a*(*t-1*)) can be obtained by:

$$\begin{aligned} &P(a(t)\,|\,a(t-1)) \qquad (11)\\ &= (2\pi)^{-k/2}\,|\,Q\,|\exp(-\frac{1}{2}[a(t)-\Phi a(t-1)]'Q^{-1}[a(t)-\Phi a(t-1)])\end{aligned}$$

where *Q* is the covariance matrix of the random noise and the term Φ is basically the deterministic part which is the state transition matrix. This formulation is same as that of the Kalman filter. The problem of parameter estimation has been explored in [24].

***P*(*e*(*t*)|*a*(*t*)):** As mentioned before, since we use the current contextual information e(*t*) to infer the goal oriented attention *a(t)*, we select the contextual information regarding the attention. In another words, the contextual information can be considered as the measurements of the attention coming from the experiential environment. If we assume that the context measurement is independent on each other, we then can define the likelihood of attention in each position to follow the Gaussian distribution.

$$P(e(t)\,|\,a(t)) = \prod_{x=1,y=y}^{x=X,y=Y} P(e(x,y,t)\,|\,a(t)) \qquad (12)$$

$$P(e(x,y,t)=1\,|\,a(t)=[x_a,y_b]') = L\exp\{-\frac{(x-x_a)(y-y_b)}{\delta^2}\}$$

where δ^2 is the constant which is used to control the randomness level and *L* is the normalizing constant.

When the situation that the measurement $e(x,y,t)$ is not binary, the above equation can be modified as follows:

$$P(e(x,y,t)=1\,|\,a(t)=[x_a,y_b]') = L\cdot e(x,y,t)\cdot\exp\{-\frac{(x-x_a)(y-y_b)}{\delta^2}\}$$

***P*(*a*(*t-1*)|*E*(*t-1*)):** This is the posterior probability of attention during time *t-1*.

b) Sequential simulation-based solutions

Instead of using Kalman filters, the sequential simulation method (sequential importance sampling *(SIS)*) [25, 29, 34] can be invoked to generate a numerical solution for dynamically approximating the density $\pi(a(t)) =$ *P*(*a*(*t*)|*E*(*t*)). The approach has an advantage in terms of the capacity for generalization.

Let *S*(*t-1*)={*s*$_1$(*t-1*), *s*$_2$(*t-1*), …, *s*$_N$(*t-1*)} denote *N* random draws that are properly weighted by the set of weights *W*(*t-1*) ={*w*$_1$(*t-1*), *w*$_2$(*t-1*),…, *w*$_N$(*t-1*)} with respect to $\pi(a(t-1))$.

At time *t*, firstly, a set of samples *S*(*t*) is drawn from a so-called *importance function* *g*(*a*(*t*)|*a*(*t-1*))[25, 29, 34] (as shown in Figure6 (c).1).The importance function is defined depending on the application. Secondly, their associated weights are obtained by:

$$w_i(t) = w_i(t-1)\frac{P(e(t)\,|\,a(t))P(a(t)\,|\,a(t-1))}{g(a(t)\,|\,a(t-1))} \qquad (13)$$

where *i*=1,…,*N* and the definitions of *P*(*e*(*t*)|*a*(*t*) and *P*(*a*(*t*)|*a*(*t-1*)) have been provided in the previous section. The discussion of *g*(*a*(*t*)|*a*(*t-1*)) will be introduced later. This weighting is shown in Figure 6 (c) step 2. Note that in the initial step, *w*(*t*)= *P*(*e*(*t*)|*a*(*t*)).

It has been shown that [34] the above obtained set of random draws and their weights {*S*(*t*),*W*(*t*)} *is properly weighted* with respect to $\pi(a(t))$. It means that the following equation is true:

$$\lim_{n\to\infty}\frac{\sum_j^n h(s_j)w_j}{\sum_j^n w_j} = E_\pi(h(a)) \qquad (14)$$

where *h* is any integrable function, E_π is the expectation, and the notation of time *t* has been dropped for the sake of simplicity of the expression.

The fundamental idea of the *SIS* algorithm is to use both a set of discrete samples obtained by the importance function *g*(*a*(*t*)|*a*(*t-1*)) and the weights obtained by equation (13) to approximate the *a posteriori* density. In another words, the distribution information is embedded both in the samples *S*(*t*) and the weights *W*(*t*). It is suitable for the applications which only require to get the expectation *E*(*h*(*a*(*t*)) like in tracking problems. However, in our application, our final aim is to obtain the relevant data on which the analysis task can be performed. We need the samples *S*(*t*) (*i.e.* location of the

```
The Algorithm Dynamic_SBADA(S(t-1), W(t-1) )
Results={}
begin
    {S(t), W(t)} ← {S(t-1), W(t-1)} by employing the SIS
    algorithm as per equation (13)
    {S(t), W(t)}= resampling(S(t), W(t))
    for i = 1 to N do /* for each sample in {S(t), W(t)}*/
        begin
        F_L ← feature extraction from a_i in the multimedia
        data.
        Compute P(H=1|f_L)
        Calculate P(H=1|f_L, a)= P(H=1|f_L).w_i
        SID = 1? 0: P(H=1|f_L, a)>0.5
        results=Results + {s_i, SID}
        end
    end
output(Results, {S(t), W(t)})
end
```

Figure 10. The Dynamic SBADA algorithm.

```
The Algorithm resampling(S(t),W(t), N')
/* N' denotes number of re-sampled samples*/
begin
    Normalize W(t) so that $\sum_{n=1}^{N} w_n(t) = 1$
    Interpret each weight as a probability, use
    $c_i(t) = c_{i-1}(t) + w_i(t)$ in order to obtain the
    cumulative probability distribution:
     C(t) ={c_1(t),c_2(t),… ,c_N(t)}
    for i = 1 to N' do
        begin
        Find by binary subdivision, the smallest j for
        which c_j(t)>=i/ N'
        Create the new samples s'_i(t):
        s'_i(t)= s_j(t)+ r  ( r is small random perturbation
        value )
        w'_i(t)= w_j(t)
        end
    S(t)' = {a_1', a_2', a_3',…, a_N'} , W'(t) = {w_1', w_2', w_3'
    ,…, w_N'}
output(S'(t),W'(t) )
end
```

Figure 11. The Re-sampling algorithm.

samples) themselves to fully cover the entire information about the distribution of the attention $\pi(a(t))$. To this end, after the *SIS* algorithm, a re-sampling step is required to relocate the samples in the higher attended regions as shown in Figure 6(c) step 3.

Based on the above discussion, after adding the *SIS* algorithm to dynamically obtain the distribution of $P(a(t)|E(t))$, the Dynamic Sampling based Attention Driven Analysis (Dynamic SBADA) algorithm is formulated as shown in Figure 10. The re-sampling algorithm is listed in Figure 11. This algorithm treats the weights as contiguous intervals of (0,1). These intervals are randomly ordered and it is sampled such that the weight of chosen samples in every interval is the same. Note that adding random perturbation value *r* in the step 5 is to prevent the creation of identical samples.

Next, we will discuss how to update the samples from the current experiential environment according to equation (13).

3) Environment sampling

Since we obtain the attention value from the experiential environments, samples used in our approach have two tasks: sense the environment and maintain the attention. Therefore, we define samples $S(t)$ to include both sensor samples $SS(t)$ and attention samples $AS(t)$:

$$S(t)=\{SS(t), AS(t)\} \quad (15)$$

The samples $S(t)$ comprises of *sensor samples* $SS(t)$ and the *attention samples* $AS(t)$. The sensor samples are basically uniform random samples at any time *t* which constantly sense the environment. The attention samples are the dynamically changing samples which essentially represent the data of interest at time *t*.

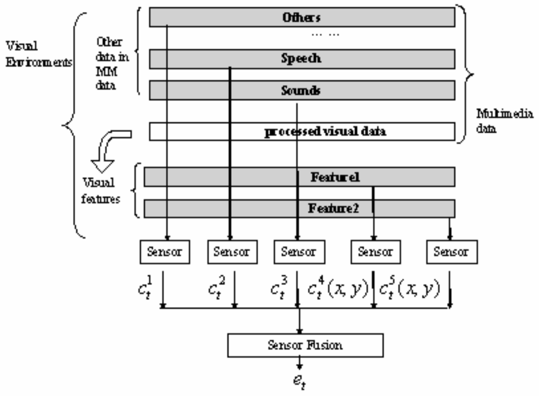


Figure 12. The framework for the sensing of environment.

Since both the types of samples have different uses, we define different importance functions ($g(a(t)|a(t\text{-}1))$) for them. The sensor samples are used to constantly sense the environment. Therefore, we define a uniform importance function $g_S(a(t))$ = *uniform sampling* for sensor samples. It allows the sensor samples to quickly notice any changes in the environments. Thus, sensor samples constantly scan the environment, looking out for sudden changes in the attention. For example, in the video face detection scenario, the sensor samples can alert the fact that a new face has entered the scene which cannot be inferred merely by the dynamical evolution of the attention samples of the previous time instant. So sensor samples perform the task of current context estimation from the extracted clues c_t^n, n=1,...,N. The attention samples are the dynamically changing samples which essentially represent the data of interest at time *t*. The attention samples are therefore derived dynamically and adaptively at each time instance from the sensor samples in our framework through sensor fusion of the current environmental context and the assimilation of the past experience. Once we have the attention samples, the multimedia analysis task at hand can work only with these samples instead of the entire multimedia data. These focused attended samples are the most relevant data for that purpose. Figure. 12 illustrates our framework for doing sensor fusion within the experiential environment. It should be understood that our data assimilation process is sampling based. Not all data need to be processed. Our aim now is to obtain these sensor samples to infer the attention. They can be sensed by multiple cues from the environment which can subsequently be fused to create $e(t)$.

The cues for obtaining experiences in the *visual environments* can be classified as temporal cues and spatial cues. They can be visual features extracted from the visual data or information from its accompanying data (speech, sound, text etc.). Basically, sensors can sense these cues in order to infer the state of the environment. Based on the above, the experiential sampling technique can also be defined as follows:

Experiential Sampling: *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. Higher attended samples will be given more weight and temporally, attention is controlled by the total number of attention samples.*

4) Sensor Sampling

Studies on human visual system show that the role of experience used in top-down visual perception increases in importance and can become indispensable when the viewing conditions deteriorate or when a fast response is desired. In addition, humans get information about the objects of interest from different sources of different modalities [7]. Therefore, when we analyze one particular data type (say spatio-temporal visual data) in multimedia, we cannot constrain our analysis to this data type only. Sensing other accompanying data like audio, speech, music, and text can help us find out where is the important data. Therefore, it is imperative to develop a sampling framework which can sense and fuse all environmental context data for the purpose of multimedia analysis.

In our framework, $SS(t)$ is a set of $N_S(t)$ sensor samples at time *t* which estimates the state of the multimedia environment. As mentioned above, these sensor samples are randomly and uniformly generated in order to sense the changes in the environments. Therefore, we define a uniform importance function $g_S(a(t))$ = *uniform sampling* for them. It makes sensor samples to quickly spot any changes in the environments.

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. $SS(t)$ is then defined as:

$$SS(t) = \left\{ss(t); \Pi^S(t)\right\} \tag{16}$$

where $ss(t)$depends on the type of multimedia data. For spatial data, $ss(t) = \{(x_1, y_1), (x_2, y_2), \Lambda\ , (x_{N_S}, y_{N_S})\}$ 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 $\Pi^S(t) = \{\pi_1^S(t), \pi_2^S(t), \mathrm{K}\ , \pi_{N_S}^S(t)\}$. Now each $\pi_i^S(t)$ is obtained by performing sensor fusion of the q 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 cue $c\_sp_i(t)$ is given by $c\_sp_i(t) = \left\{(x_i^1, y_i^1, w\_sp_i^1), \mathrm{K}\ , (x_i^{N_S}, y_i^{N_S}, w\_sp_i^{N_S})\right\}$ 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

$$\pi_i^S(t) = \sum_{j=1}^{q} \alpha_j \cdot w\_sp_j^i \tag{17}$$

where α_j is the importance of the jth cue. So we basically employ the linear combination as the sensor fusion strategy. But this can be replaced by a more sophisticated sensor fusion strategy, which has been investigated in our previous research in [22, 23], if the application so requires. Also, 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. This can be denoted as:

$$w\_sp_j = VF_j(I_t(x,y), \ldots, I_1(x,y), m_j) \tag{18}$$

where VF_j is the feature extraction function of the jth cue and m_j is its function parameters. $I_t(x,y)$ denotes the image intensity at time t.

For instance, in a video, the motion cue is a spatial cue since it varies according to its spatial position. It can be simply defined as

$$w\_mot(x,y) = \left|I_t(x,y) - I_{t-1}(x,y)\right| \tag{19}$$

Here the feature extraction function is the absolute difference of corresponding pixel intensity values of two neighboring frames. However, there is no adjustable parameter in this function.

5) Attention Sampling

We know that the attention changes dynamically. In a manner different from that of the sensor samples, which use uniform random sampling as the importance function, we use another probability distribution as an importance function $g_A(a(\mathrm{t})|a(t\text{-}1))$ to create the attention samples:

$$g_A(a(t) \mid a(t-1)) = P(a(t) \mid a(t-1)) \tag{20}$$

where $P(a(t)|a(t\text{-}1))$ is the dynamics of attention which can be obtained by equation (11). Consequently, the equation to compute the weights (in equation (13)) becomes:

$$w_i(t) = w_i(t-1) P(e(t) \mid a(t)) \tag{21}$$

The notation for attention sampling is introduced as follows:

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

$$AS(t) = \left\{as(t); \Pi^A(t)\right\} \tag{22}$$

where as(t) again depends on the type of multimedia data. For spatial data, $as(t) = \{(x_1, y_1), (x_2, y_2), \Lambda\ , (x_{N_A(t)}, y_{N_A(t)})\}$, 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 $\Pi^A(t) = \left\{\pi_1^A(t), \pi_2^A(t), \mathrm{K}\ , \pi_{N_A(t)}^A(t)\right\}$. 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.

However, there still have one question: how to determine the number of attention samples N_A(t) which varies with time? N_A(t) intuitively models the *attention saturation* which is defined in the next section.

6) Attention Saturation

The temporal attribute of the spatio-temporal data requires the multimedia system to possess the ability of varying the amount of attention at different times. We introduce the concept of *attention saturation* to measure the attention in a given time slice. For instance, the attention saturation of motion in Figure 13 (a) is higher than that in Fig. 13 (d). The attention saturation in this case can be calculated as the sum of attention in the spatial extent. Its value ranges from 0 (lowest, no attention) to 1 (highest, full attention). We define the attention saturation as $ASat$(t):

$$ASat(t) = f_N\Big(\int_{Spatial} P(a(t) \mid E(t))\Big) \tag{23}$$

where f_N is the mapping function which is used to normalize the value into range [0,1]. f_N is defined as the squashing function [1] shown in the equation (24)(the relationship of input and output is shown in Fig.14).

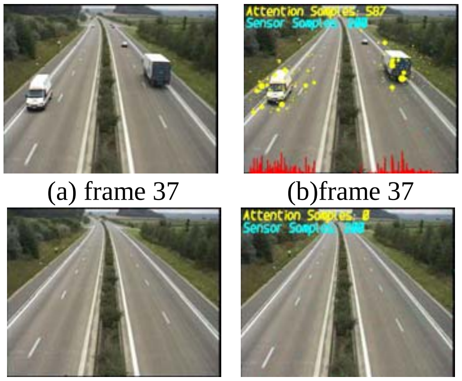
$$f_N(x) = \frac{1 - \exp(-\lambda \cdot x)}{1 + \exp(-\lambda \cdot x)} \tag{24}$$

where λ is a scaling factor. As shown in Fig. 14, the benefit of employing equation (24) is that it can map a very large input domain to the interval [0, 1]. We select λ so that the output scatters in the interval [0, 1] as much as possible.

The current attention is essentially captured by the sensor samples. The sensor samples are updated by each of the cues. Of course, some cues may only have temporal attributes and no spatial coordinate (e.g. audio volume). Such cues can be defined as $c\_tp_j(t) = \{w\_tp_j\ \}$, where w_tp_j is the weight of the jth cue. Therefore, the discrete form of the equation (23) is given below:

$$\begin{aligned} &ASat(t) \\ &= f_N\Big(\frac{1}{n} \sum_{t'=[t-n,t]} \Big(\frac{1}{N_S} \sum_{i=1}^{N_S} \pi_i^S(t') + \sum_{j=1}^{p} \beta_j w\_tp_j(t')\Big)\Big) \end{aligned} \tag{25}$$

where β_j is the importance of the j^{th} temporal cue and p is the number of the temporal cues. Thus, the attention saturation of the current state is captured by the average weight of all the sensor samples and temporal cues. 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. In our audio-visual face detection, we set $\beta_j = 0.8$ for the sound volume cue and n=3 for the web-camera video stream.



(a) frame 37 (b)frame 37

(c)frame 479 (d)frame 479

Figure 13. Temporal motion attention.(a) more motion activity (b) 567 attention samples are employed to represent this motion attention.(c) need less attention at this time(d) No attention samples are needed at this time. The number of attention samples is calculated by using equation (8).

Note that for sensor samples, the number of samples was fixed a priori at N_S in equation (25) and these samples are generated uniformly and randomly at every time instant. But the number of attention samples varies with time. For instance, in the traffic monitoring application shown in Figure 13, Figure 13 (a) has more motion activity and hence needs more attention samples to represent this motion attention. As shown in Figure 13 (b), 587 attention samples (marked as yellow points) are required to represent this motion attention using our method. In contrast, Figure 13 (c) has less motion and needs fewer attention samples. As shown in Figure 13 (d), no attention samples are needed. However, all previous image based attention models [9, 10, 12, 13] lack the ability to model this adaptive behavior.

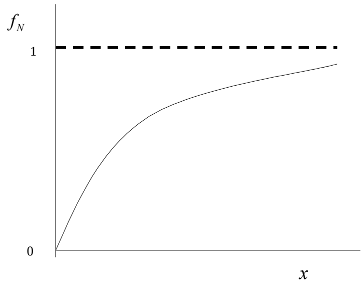


Figure 14. Relationship between the temporal experiences and the probability of the temporal attention (Attention Saturation).

Thus, we can utilize attention saturation to measure the attention at a given time instance (the temporal attention) as shown in the experiment in Figure 29.

We are now ready to determine the number of attention samples at time t using:

$$N_A(t) = N_{Max} ASat(t) \qquad (26)$$

where N_{Max} is the maximum number of samples the system can handle.

7) Past Experiences

We have introduced how the attention guides the analysis task. Contrastingly, in this section, we will discuss how the local analysis task guides the attention in the form of the past experiences. This is also an important concept in Neisser's Perceptual Cycle, *i.e.* how the perceiver use the results of analysis to modify the current schema (current environment model).

Our attention model is employed to obtain attention from the experiential environment. The current environment model in our case is the attention model. As formulated in section III.C.4, the attention model is parameterized by each cue's feature extraction function VF_j, its function parameter m_j and its importance α_j (see equation (17) and (18)). The data to be dealt with is dynamic with temporal variations. Therefore, the attention model itself should change dynamically. It is non-trivial to accurately model the dynamical evolution of the attention model itself due to these variations. Thus we want to simultaneously model the dynamically varying attention as well as the evolving attention model (from which the attention is derived). We add the time variable t to our formulation and define the parameters of the attention model for q feature cues at time t as $APara(t)$={ $\alpha_1, \ldots, \alpha_q . m_1, \ldots, m_q$ }.

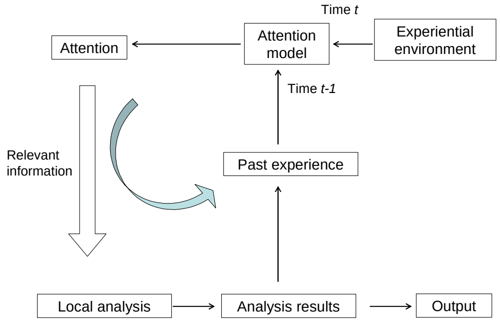


Figure 15. Analysis guides attention model evolution by past experiences.

The local analysis task, though time-consuming, provides us the most reliable measurements about the multimedia data. Like human beings, the results of the analysis can be stored as the accumulated knowledge. This knowledge can be utilized as the past experience when a future data assimilation process starts. In our framework, we want those past experiences to help in *adjusting* (*adapting*) the attention model and let the analysis task guide that attention model evolution. Figure 15 describes this process graphically.

Suppose we are doing multimedia analysis by mapping low level features to a *semantic symbolic identity*, named *Tar* (target) in the spatio-temporal data. The attention represented by the attention samples should be focused on regions which have concentrated relevant information about the identity *Tar*. Due to the reliability of the local analysis task, we can actually employ the local analysis task to judge the accuracy of the current attention samples. At time *t*, after performing the local analysis task on the attention samples *AS*(*t*), we divide the attention samples *AS*(*t*) into two sets: $AS_+(t)$ containing *reliable attention samples,* and $AS_-(t)$ containing *unreliable attention samples* by using the following equations:

$$AS_+(t) = \{AS(t) : S_M(f_L, AS(t)) = Tar\} \qquad (28)$$

$$AS_-(t) = \{AS(t) : S_M(f_L, AS(t)) \neq Tar\}$$

where S_M is the feature to semantics mapping function defined in section III.B.

By employing equation (28), we treat the attention samples which are finally proven to have the relevant information about the target *Tar* as the reliable attention samples and the others are not reliable. We call these classified attention samples $AS(t)=\{AS_+(t), AS_-(t)\}$ the *past experience.* Intuitively, the past experience can be used as labeled training samples to learn or update the attention model parameters *AP*(*t*+1) for next time slice. This procedure is defined as follows:

$$APara(t+1) = L(AS_+(t), AS_-(t)) \qquad (29)$$

where *L* denotes the inductive learning method to be used to obtain the parameters of the attention model.

D. The Experiential Sampling Technique

We are now ready to fully describe the experiential sampling technique based on the background developed so far. Now we assume we are dealing with spatio-temporal data denoted by stream(*t*). Our experiential sampling technique (a two-class classification problem is considered) is summarized as shown in Figure 16.

From the algorithm, we can see that the analysis task itself collects relevant data (step 4-7) in the form of attention samples and performs the analysis task on these attention samples (step 11-14). In addition, the local analysis can classify the attention samples and these gained past experiences can re-train the attention model to make it more adaptive to the environments (step 14-16).

E. Evaluation

We use ideas from foraging theory to evaluate the efficacy of the experiential sampling technique. When people explore data and assimilate information, people try to maximize their rate of gaining valuable information over cost. In the information foraging theory [35], it has been formulated as maximizing the rate of gain of valuable information per unit cost *R*:

$$R = \frac{G}{T_B + T_W} \qquad (30)$$

where *G* is the total net amount of valuable information gained (the attended samples), T_B is the total amount of time spent between information patches (time to sense the environment using sensor samples and compute the context) and T_W is the time within the information patches (time to obtain the attention samples and to perform analysis time on them). Therefore a "good" method should have the ability to maximize *R* at any given time. The intuitive idea is that the amount of computation required for determining attention should be small enough so that the savings obtaining by doing the task only on the attended samples clearly dominates this factor. Thus, we can obtain an overall gain.

```
The Algorithm ExperientialSampling(stream(t),
  AS(t-1),N_A(t-1),t)
begin
   /* Experiential environment sensing */
1. Initialization: $N_s$ , $N_{Max}$ , and  if (t == 0), $N_A(t) = 0$.
2. AS(t)← AS(t-1)  by equation (11)
3. SS(t) ←create_ramdom_sampling().  /*create SSs*/
4. for SS(t)(i = 1 to $N_s$ ), AS(t)( k= 1 to $N_A(t)$) do   /* weight
   updating*/
    begin
         for j = 1 to q  do /* each multimedia cue*/
           begin
           $w\_sp^i_j$ ← equation (18) with $VF_j$ and $m_j$ /*cues  for
           SS(t) sensor samples*/
           $w\_sp^k_j$ ← equation (18) with $VF_j$  and $m_j$ /*cues  for
           AS(t) attention samples*/
           end
      $\pi_i^S(t)$ ← by equation (17) with $w\_sp^i_j, \alpha_j, j = 1,\ldots,q$
      /*weight for SS(t) sensor samples*/
      $\pi_k^A(t)$ ← by equation (17) with $w\_sp^k_j, \alpha_j, j = 1,\ldots,q$
      /*weight for AS(t) attention samples*/
    end
5.  Calculate $ASat(t)$ by equation (25)  /*Overall Attention
    saturation*/
6.  $N_A(t)$ ← equation (26)  /*current number of attention samples
    */
    /* Building of  the attention model and attention driven
    analysis */
7.  AS'(t)  = resampling((ss(t),as(t)), ($\Pi^S(t), \Pi^A(t)$ ),$N_A(t)$)  /*
    Create current attention samples, see Fig. 11*/
8.  AS(t) = AS'(t)
9.  if $N_A(t)$ > 0 then do   /*Attention driven analysis*/
      begin
10.   for i = 1 to $N_A(t)$  do  /*for each attention sample in AS(t) */
          begin
11.       $F_L$ ← feature_extraction(stream(t), $as(t)$ ) at
            location of as(t) in stream(t).
12.       Compute $P(H=1|f_L)$
13.       Calculate $P(H=1|f_L, a)= P(H=1|f_L)\times\pi_i^A(t)$
14.       $SID = 1 ? 0 : P(H=1|f_L, a)>0.5$
15.       Classify $AS(t)$ into { $AS_+(t)$ , $AS_-(t)$ } by
            equation (28) /*the past experience*/
16.       update $\alpha_j, VF_j, j = 1,\ldots,q$  by equation (29)
            /*adaptation shaped by experience*/
          end
    end  /* end of "if $N_A(t)$ > 0"  */
output(results from step 15, AS(t-1), $N_A(t-1)$)
end.
```

Figure 16. The Experiential Sampling Technique.

Our attention samples are used to collect the relevant information. If the attention model is accurate, the attention saturation *ASat(t)* intuitively measures the relevant information regarding the goal in a given time slice. We can define $G = ASat(t)$ (*when* $ASat(t) \neq 0$). The cost of obtaining the sensor samples C_S can be treated as T_B while the cost of obtaining attention samples C_A and performing local analysis (the equation (1)) on attention samples C_{F_L} can be treated as $T_{W.}$

Based on the above, the rate of gain of valuable information per unit cost of our approach R_E is equal to:

$$R_E(t) = \frac{ASat(t)}{N_A(C_A + C_{F_L}) + N_S C_S} \tag{31}$$

Since the cost of obtaining both sensor samples C_S and attention samples C_A is much smaller than the cost of performing local analysis C_{F_L}, the second part of the denominator in the equation ($N_S C_S$ as well as C_A) can be removed. Consequently, by replacing N_A from equation (26), equation (30) becomes:

$$R_E(t) \approx \frac{ASat(t)}{N_{Max} ASat(t) C_{F_L}} = \frac{1}{N_{Max} C_{F_L}} \tag{32}$$

From equation (32), we can see our algorithm is adaptive to the experiential environment and keeps maximizing the rate of gaining valuable information over cost. When there is more relevant information (increasing the attention saturation in the numerator), the number of attention samples will be larger and consequently the cost increases (as the increase in the denominator) and *vice versa*. This keeps the valuable information gain per unit cost near the maximal value.

In contrast, the rate of gain of valuable information per unit cost for the local feature centered approach R_{F_L} (if we only use equation (1) for content analysis) is equal to:

$$R_{F_L}(t) = \frac{ASat(t)}{MC_{F_L}} \tag{33}$$

where M is number of times that the local analysis needs to be performed. M is much bigger than the maximum number of attention samples $N_{Max.}$ Especially when there is less relevant information, there still will be a constant local analysis cost. Therefore, it is not efficient compared to our approach.

IV. HANDLING MULTIPLE DATA STREAMS

A. Optimal Selection of Data Streams

We have seen in the earlier section that for a single media stream case (which could have multiple local feature streams), the goal oriented attention driven analysis can be succinctly described by:

$$SID = S_M(f_L, a_G) = \arg\max_H P(H \mid f_L, a_G)$$

$$\therefore SID = \arg\max_H P(H \mid f_L) \cdot P(a_G \mid E)$$

We will now extend this scenario to the real multimedia case when multiple correlated media streams are considered. Our work adopts an approach similar to that of [38] and generalizes their ideas for multimedia systems. As described earlier, there are n media data streams S_1, S_2 …Sn. These data streams consist of K types of data such as image sequence, audio stream, motion detector, annotations, symbolic streams, and any other type that may be relevant. We assume that these streams are synchronized. Further, we assume that metadata MD_1, MD_2, …MD_n for each stream is available from the original sources that helps in interpreting the data stream in the context of the environment. And since in most cases feature detectors will be applied to each data stream in the context of the corresponding metadata for each data stream, we can represent the multimedia data as a (possibly) correlated features stream set $\overline{\mathrm{F}} = \{f_j\}$, where f_j is the j^th^ feature stream where $1 \le j \le N$ such that $N \ge n$ and there is at least one feature stream derived from every multimedia data stream. So, now our equation (7) can be modified to include the multiple correlated multimedia data streams scenario to:

$$SID = f_M(\overline{\mathrm{F}}, a_G) = \arg\max_H P(H \mid \overline{\mathrm{F}}, a_G)$$

$$\therefore SID = \arg\max_H P(H \mid \overline{\mathrm{F}}) \cdot P(a_G \mid E) \tag{34}$$

Clearly, there is some amount of noise in every data stream f_j and also there is a tremendous amount of redundancy among them. The questions raised earlier in Section 3 boil down to the question of selection of appropriate features stream set for the goal to be achieved. More formally, let us assume that a set of $\overline{\mathrm{F}}$ feature streams allows the system to achieve goal G. We also assume that each feature stream contains only partial information to achieve the goal and there is redundancy (overlap) of information among the various feature streams. Let us also assume that there is a cost function associated with the use of each subset of $\overline{\mathrm{F}}$. Our problem now can be defined as:

(a) to identify a lowest cost subset of feature streams $\Phi^* \subseteq \overline{\mathrm{F}}$ such that the goal G can be accomplished.

(b) to develop an optimal procedure for determining this subset Φ^*.

Assume that when the full set of data streams $\overline{\mathrm{F}}$ is available, we have:

$$P_{\overline{\mathrm{F}}}(G \mid H, f_j) > \alpha, 1 \le j \le N \tag{35}$$

where $P_{\overline{\mathrm{F}}}(G \mid H, f_j)$ denotes the probability that the goal of correctly identifying the hypothesis of the symbolic identity when it is actually true, given the N feature streams information and $0 < \alpha < 1$ denotes the confidence level. Our problem can now be restated as:

(a) identify a lowest cost subset Φ^* of feature streams such that

$$P_{\Phi^*}(G \mid H, f_j) > \alpha, 1 \le j \le N. \tag{36}$$

(b) determine the optimal procedure to identify the feature stream subset Φ^* assuming we have a method to determine whether an arbitrary subset $\Phi \subseteq \overline{\mathrm{F}}$ satisfies

$$P_{\Phi}(G \mid H, f_j) > \alpha, 1 \leq j \leq N. \qquad (37)$$

Note that the total cost is normally related to the total computation cost of the feature streams subset or perhaps can be the hardware cost of obtaining the feature streams or could be related to the energy consumption of obtaining the feature streams (particularly in case of low power appliances). Let us quantify the cost of using a subset of feature streams Φ by c_{Φ} and let us assume an *a priori* probability p_{Φ} that this subset can achieve the goal G. The idea of having these probabilities is that it allows for an identification strategy to be developed to obtain the lowest cost feature stream set. So we can not only identify *which* subset that can achieve the goal but also provides a mechanism to determine *how* to identify this optimal subset. This optimization problem is posed as a Markovian decision process. We also try to provide a set of assumptions under which this optimal strategy can be developed. Of course, by changing these assumptions, we can better study the structure of this problem and can lead to better identification algorithms for different problem instances.

B. The General Multiple Stream Problem

We will first present the results in a general setting and then narrow down some specific instances of the problem. In the general case, let us assume that we are given a multimedia system with a set of $\overline{\mathrm{F}}$ feature streams. We make the following assumptions:

(1) The goal G can be achieved when the full set of *N* feature data streams $\overline{\mathrm{F}}$ is available. If we do not have this assumption, there is no optimization problem to solve.

(2) Any combination of *i* feature streams $(i < N)$ has a lower cost than any combination of $i+1$ feature data streams. This allows for the fact that for any specific combination of *i* data streams to be of less cost than that of any other set of *i* data streams. Note that this may not be a realistic assumption. Relaxing this assumption is an open problem.

(3) If the *a priori* probability that the multimedia system can achieve goal G using a combination of *i* feature streams is p_i, then we have:

$$0 = p_0 < p_1 \leq p_2 \leq \mathrm{K} \leq p_{N-1} < p_N = 1 \qquad (38)$$

What this essentially states is all feature data streams have an equal capability of providing information for achieving goal G. We will modify this assumption later on for a specific instance of the general problem.

(4) If a combination of feature data streams Φ_A cannot achieve the goal G, the probability $p_{|B|}$ remains the same for all sets of feature data streams $B \supset A$. Moreover, if a combination of feature data streams Φ_A achieves the goal G, the probability $p_{|C|}$ remains the same for all sets of feature data streams $C \subset A$.

(5) The cost of finding out whether a subset of feature data streams can help achieve the system goal or not is a constant equal to *c*. This assumption basically states that there is a constant cost procedure to determine whether the given subset Φ is sufficient to achieve the system goal G. One can conceivably have a benchmark data set with ground truth to perform this test.

We can now cast the feature stream subset selection problem as a decision problem on a directed graph. The nodes of the directed graph are the elements of the power-set of $\overline{\mathrm{F}}$. Each node of the graph represents a combination of feature data streams. Two vertices *A* and *B* are connected by an edge directed from B to A iff $|B| = |A| + 1$ and $A \subset B$. Node ϕ of the graph is the empty node which corresponds to the use of zero feature data streams. An example of a directed graph for a multimedia system with three feature data streams is shown in Figure 17.

We note that the directed graph provides the combinations of possible feature data streams. The idea of the identification procedure is to quickly identify the node with the least cost which allows for the multimedia system to achieve goal G. Each subset of feature data streams (corresponding to a node) can be tested for fact whether it achieves the goal G or not. Note that the node containing $\overline{\mathrm{F}}$ i.e. containing all the feature data streams does achieve goal G (from assumption 1). The node ϕ cannot achieve the goal G. If a node *A* can achieve the goal G, then node $B \supset A$ can also achieve the goal G. Conversely, if a node *A* cannot achieve the goal G, then node $B \supset A$ also cannot achieve the goal G. These are fairly obvious statements. We now need a set of definitions:

(1) Node B of the directed graph is a child of node C in the graph iff there exists a directed path from C to B.

(2) Node B of the directed graph is a parent of node C iff there exists a directed path from B to C.

(3) Let ξ be a set of nodes with $A \in \xi$. A reachable set from ξ conditioned on the fact that A can achieve goal G, is a set composed of all nodes in ξ whose cost is less than that of A.

(4) Let ξ be a set of nodes with $A \in \xi$. A reachable set from ξ conditioned on the fact that A cannot achieve goal G, is a set composed of all nodes in ξ except node A and its children in ξ.

(5) A reachable set is a set that results from applying an arbitrary sequence of tests (for testing whether a node can achieve goal G) according to the definitions 3 and 4 above.

Now, we are ready to pose the problem as a Markovian decision problem with perfect observations. The information state of the process is the set of nodes of the directed graph which have not yet been checked whether they can achieve goal G and could potentially correspond to a least cost combination of feature data streams. Therefore, an information state is a reachable set. Let V(Q) denote the minimum expected cost (of testing) when the state is Q. Then V(Q) satisfies the optimality equation:

$$V(Q) = \min_{i \in Q}\{c + p_i * V(N_i^G) + (1 - p_i) * V(N_i^{\overline{G}})\} \quad (39)$$

Note that G denotes that the system goal is achievable and $\overline{G}$ denotes that the system goal is not achievable. We will now provide a solution to the above equation under the assumptions stated at the beginning of this section.

Theorem 1: If $p_l + p_{l+1} \geq 1$ for $l = 1, \mathrm{K}\ , N - 2$, then an optimal test strategy for identifying the feature stream subset Φ^* is to test the combinations of feature data streams in an increasing order of feature data stream cost.

Proof: The proof for this theorem is structurally similar to the proof of Theorem 3.1 of [38].

What this theorem states is that if one tests the combination of feature data streams in this manner, an optimal feature stream subset Φ^* is *guaranteed* to be identified with the least cost. What is a more interesting result is the following corollary which precisely computes the value of V(Q) for the optimal subset:

Theorem 2: Let Q be a reachable set. Then the minimum expected cost associated with Q is

$$V(Q) = c * [\sum_{n=0}^{n_l - 1}(1 - p_l)^n + (1 - p_l)^{n_l} * \sum_{n=0}^{n_{l+1}-1}(1 - p_{l+1})^n \quad (40)$$
$$+ \mathrm{K} + (1 - p_l)^{n_l} * (1 - p_{l+1})^{l+1} \mathrm{K} * (1 - p_{h-1})^{n_h - 1} * \sum_{n=0}^{n_h - 1}(1 - p_h)^n$$

Proof: This proof is similar to the proof of corollary 3.1 of [38].

We will now examine a special instance of the above generalized setting.

C. Analysis of the Constant Fusion Probability Instance

Let us now examine the generalized setting under a more constrained assumption 3 of section IV.B. If we assume the following modified assumption:

(3') The *a priori* probability that the multimedia system can achieve goal G when a combination of *i* feature data streams is utilized for $1 < i < N - 1$, is equal to a constant probability *p*.

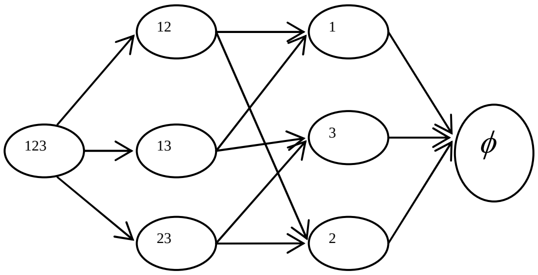


Figure 17: The directed graph for three feature data streams case.

The earlier assumption is constrained to consider the fact that any subset of feature data stream set has equal probability of achieving the system goal. This may not be a very realistic assumption but it is a practical assumption to make when no prior empirical evidence is available in which case it is fair to assume $p = p_i = \frac{1}{2}$ for all subsets *i*. This essentially means that any subset is equally like to achieve the system goal and we would like to identify the subset with the minimum cost.

Theorem 3: Let Q be an information state with *i* and *j* being two elements of Q. If $p \geq \frac{1}{2}$, then:

(1) If $|Q_i^G| + |Q_i^{\overline{G}}| \leq |Q_j^G| + |Q_j^{\overline{G}}|$ and $|Q_i^G| \leq |Q_j^G| \leq |Q_i^{\overline{G}}|$, $|Q_i^G| \leq |Q_j^{\overline{G}}| \leq |Q_i^{\overline{G}}|$ then $V_i(Q) \leq V_j(Q)$ where $V_i(Q)$ denotes the expected cost of testing all $k \in Q$ when the information state is Q and follow the optimal test strategy afterwards.

(2) An optimal test strategy is to test combinations of feature stream sets in an increasing order of their cost. The minimum expected cost associated with Q is

$$V(Q) = c * \sum_{n=0}^{|Q|-1}(1 - p)^n \ . \quad (41)$$

Proof: This proof is structurally similar to that of Theorem 3.2 of [38].

What is interesting is the above theorem provides an upper-bound of V(Q) as $\frac{c}{p}$.

D. Attention Saturation for Multiple Data Streams

If we have multiple data streams, we need to be able to decide how many sensor and attention samples to allocate to each data stream. The case of sensor samples is quite straightforward. For one data stream, we had N_S sensor sample. If we have $\left|\Phi^*\right|$ data streams, then we can allocated a fixed $N_S(i)$ number of sensor samples for sensing the environment for each data stream where $1 < i \leq \left|\Phi^*\right|$. Again, the notion of attention saturation can also be used with generalization. For one data stream case, we had N_A attention samples. We now define attention saturation for a single feature stream $\mathrm{F_j}$ as follows:

$$ASat^{F_i}(t) = f_N\left(\int_{f_i} P\left(a(t) \mid E(t)\right)\right) \quad (42)$$

Then for all the data streams, we have the total amount of attention saturation as:

$$ASat(t) = \sum_{f_i} ASat^{f_i}(t)$$

Now we can easily compute the number of attention samples for each individual data stream using:

$$N_A^{f_i}(t) = \frac{N_{\max} \bullet ASat^{f_i}(t)}{ASat(t)} \tag{43}$$

Thus, we can compute the number of attention samples required for each data stream in way which proportional to the amount of attention required for that stream normalized over the total attention saturation.

V. APPLICATIONS

Since it is a general framework, our proposed experiential sampling technique can be used for a variety of multimedia analysis tasks, especially in real-time applications like monitoring systems [49]. We have applied our framework in the multiple-camera video surveillance domain [50], video adaptation [48] and camera control [37]. In this article, as test examples, we apply our framework in the three test examples ranging from activity monitoring, face detection to monologue detection.

```
The Algorithm
MotionMonitoring_by_ES(image(t),AS(t-1), N_A(t-1), t)
begin
1. Initialization: $N_s \leftarrow 200$, $N_{Max} \leftarrow 1000$, if ($t == 0$), $N_A(t) = 0$.
2. $SS(t) \leftarrow$ ramdom_sampling(image(t).width,image(t).height).
3. $AS(t) \leftarrow AS(t-1)$ by equation (11)
4. $SS(t)(i = 0$ to $N_s)$, $AS(t)(i = 0$ to $N_A(t)) \leftarrow$
   updateweight(image(t)) /* weight updating using
   equation (19)*/
5. $ASat(t) \leftarrow$ by equation (25)
6. $N_A(t) \leftarrow$ equation (26) /*number of attention samples
     needed in the current environment*/
7. $AS'(t)$ = resampling((ss(t),as(t)), $(\Pi^S(t), \Pi^A(t))$,$N_A(t)$) /*
     create attention samples*/
8. $AS(t) = AS'(t)$
output(ASat(t), AS(t))
end.
```

Figure 18. Algorithm of Motion Activity Monitoring by Experiential Sampling.

A. *Activity Monitoring*

In the traffic monitoring and surveillance applications, the most important task is to monitor the motion activity. The experiential sampling technique can use motion as the cue to maintain the motion attention in both spatial and temporal directions. Without fully processing the spatio-temporal data, locations of the attention samples actually reflect the spatial location of the motion activity while the attention saturation indirectly indicates the total amount of the motion activity.

Sensors samples in equation (16) again can be defined as $ss(t) = \left\{ (x_1, y_1), (x_2, y_2), \ldots, (x_{N_S}, y_{N_S}) \right\}$. Their associated weight is defined as $\Pi^S(t) = \left\{ \pi_1^S(t), \pi_2^S(t), \ldots, \pi_{N_S}^S(t) \right\}$. The can then be obtained by calculating the spatial cue of motion. We define the spatial cue of motion as $c\_sp_{MT}(t) = \{ (x_{MT}^1, y_{MT}^1, w\_sp_{MT}^1), \ldots, (x_{MT}^{N_S}, y_{MT}^{N_S}, w\_sp_{MT}^{N_S}) \}$. The weight of each sensor sample and attention can be updated by using equation (19). Based on the experiential sampling technique (Figure 16), the algorithm of motion activity monitoring is summarized in Figure 18. Since in these experiments, we want to show that our sampling method captures the motion attention, the final attention driven analysis steps (shown in steps 9-15 in Figure 16) are discarded to depict only pure attention.

Figure 19 illustrates the procedure for a pedestrian monitoring scenario. For each time instance, the algorithm outputs the temporal attention (*ASat*(*t*)) and spatial attention (*AS(t)*). The procedure can be described as follows:

Step 1. *The SSs are randomly created to sense the entire scene in (b), (f) and (j). If the ASs exist in the previous time slice, they are dynamically updated (Step 4) to the current time slice like the dark dots shown in (f) and (j). Note that time 0 indicates the situation when there are no previous ASs, i.e. start of the system or a sequence of interest.*

Step 2. *The weights of the SSs and the ASs (in (c), (g) and (k)) are adjusted by the measurements from the motion features (by using Eq. (3)). The weights are indicated by the size of the points in (c), (g) and (k).*

Step 3. *The temporal attention (ASat(t)) is calculated from the weight-adjusted SSs. Consequently, AS(t) is created by resampling the SSs (shown from (c) to (d)) or both the SSs and ASs (shown from (g) to (h) and from (k) to (l), respectively). The number of the created AS(t)s is controlled by the obtained ASat(t). Obviously, the AS(t)s created represent the distribution of the spatial attention as shown in (d), (h) and (l).*

Step 4. *Each AS follows its own dynamics (e.g., constant velocity) and is dynamically updated to the next time slice (dark dots shown from (h) to (j)).*

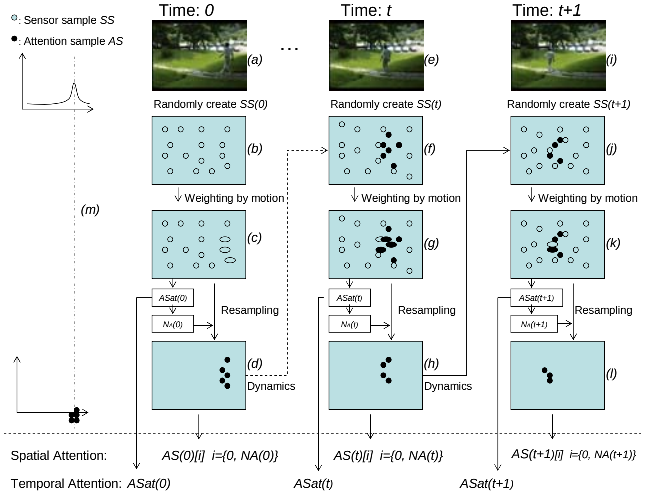


Figure 19. Illustration of our framework for motion activity monitoring. (*a,e,i*) Sample frames. (*b,f,j*) Randomly created *SSs* and dynamical-updated *ASs* (only in *f&j*) from previous time slice. (*c,g.k*) weighted *SSs* and *ASs* (only in *g&k*). (*d,h,j*) N_A number of the *ASs.* (*j*) *ASs* in (*d*) represent the attention distribution.

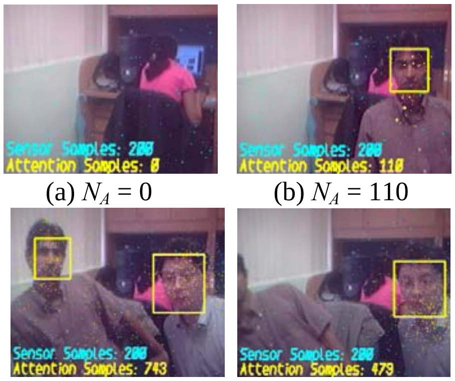
B. Face Detection

In the face detection problem [2, 3, 4, 5], current robust detection methods all rely on exhaustive scanning of the entire image with different scale factors i.e. every position of the image is probed at different scales by employing either a Gaussian model [2], or Neural Networks [3] or boosted classifiers [4, 5]. 193737 probe computations are needed for a single 320x240 sized image (using 20x20 size scan window and a scale factor of 1.2). However, in most of the cases, human faces only occupy a small part of a given frame. Obviously, most of these probes are conducted where faces do not possibly exist. The computations in such low probability areas are wasteful and can even lead to false detections. It would be ideal if the expensive face detection computations were carried out only where faces are very likely to occur. Our experiential sampling framework precisely facilitates this.

In this test example, we try to perform face detection task by using data coming from two streams (one visual data, one audio data). We utilize experiences (domain knowledge and accompanying audio (speech) and visual cues (skin color and motion)) to infer the attention samples. These attention samples are adaptively maintained by the sampling based attention framework proposed in the previous sections. We use the *adaboost* face detector [4] for performing the multimedia analysis task. Therefore, the mapping function in equation (7) becomes:

$$SID = S_{Adaboost}(f_L, AS(t)) \qquad (44)$$

where $S_{Adaboost}$ is the *signal to symbol* mapping function from the *adaboost* face detector. f_L is the input feature to the detector, which is obtained by feature extraction from the location of the attention samples *AS(t)*. Face detection is only performed on the attention samples to achieve robust real time processing. This processing (by using spatial cues from motion and skin color) is shown in Figure 20. Note that depending on the amount of attention, the number of attention samples is different. For instance, N_A in Figure 20 (c) is 743, which is bigger than in Figure 20 (b) and (d) since Figure 20 (c) has two attention areas whereas Figure 20 (b) and (d) only have one attention area. Figure 20 (c) also shows our sampling technique can maintain more than one attention region.



(a) N_A = 0 (b) N_A = 110

(c) N_A = 743 (d) N_A = 479

Figure 20. Face detection sequence.

Most importantly, past face detection results serve as the past experience to adaptively correct the attention samples and the skin color model in the attention inference stage. This provides the face detector the ability to cope with a variety of changing visual environments

1) Cues from audio-visual data

In the face detection application, the attended regions and frames are those where the probability of finding a face is high. The face attention information can be inferred from the contextual information in the experiential environment. In this application, we would like to use the cues of visual features (motion and skin color) and accompanying audio data to sense the experiential environment. The methods of obtaining the cues are now described:

Skin color cue: Since skin color is clustered well in the color space [30], we use the 1-D histogram of hues (color) channel from HSV color system to represent the skin color [30].

We define

$$c\_sp_{Skin}(t) = \{(x^1_{Skin}, y^1_{Skin}, w\_sp^1_{Skin}), \ldots, (x^{N_S}_{Skin}, y^{N_S}_{Skin}, w\_sp^{N_S}_{Skin})\}$$

as the skin color cues. As shown in Figure 21, the stored skin color histogram H_t in time t is employed as a lookup table to calculate the weight $w\_sp^i_{Skin}$. This lookup procedure (looking for bin's value) is defined as

$$w\_sp^i_{Skin}(x, y) = lookup(hue(x, y), H_t) \qquad (45)$$

Figure 22(a) shows that attention samples congregate at the skin color region and the face is finally detected in that region by the final analysis (face detector). Figure 22 (b) shows the weight map as measured by equation (35).

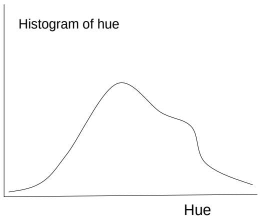
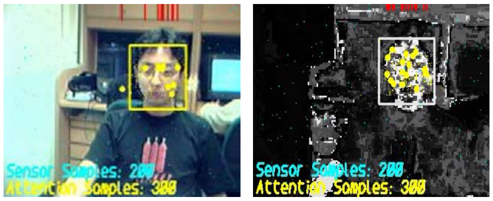
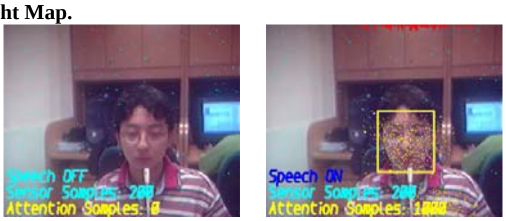


Figure 21. 1 D histogram of the hue channel.



(a) (b)

Figure 22 Face detection by Skin color histogram H_t. (a) Sample Image (b) Weight Map.



(a)Frame 2 (b)Frame 10

Figure 23. Sound cue (a) speech off N_A=0.(b) speech on. N_A becomes 1000. face detected.

```
The Algorithm FaceDetection_by_ES(visual(t),audio(t), AS(t-1), $N_A$(t-1),t)
Begin
 /* Experiential environment sensing */
1. Initialization: $N_s$ ← 200 , $N_{Max}$ ← 1000, if (t == 0), $N_A(t)$ = 0.
2. AS(t)← AS(t-1) by equation (11) /* importance sampling for the attention samples*/
3. SS(t) ← ramdom_sampling(visual(t).width, visual(t).height)./* importance sampling for the sensor samples*/
4. $ASat_C(t)$ ← $audio(t)$ by equation (48) /*Attention saturation for the audio data stream*/
5. for SS(t)(i = 0 to $N_s$ ), AS(t)( k= 0 to $N_A(t)$) do
    begin
      $\pi_i^S(t)$ ← by equation (47) /*weight for SS(t) sensor samples*/
      $\pi_i^A(t)$ ← by equation (47) /*weight for AS(t) attention samples*/
    end
6. Calculate $ASat(t)$ using equation (25)
7. $N_A(t)$ ← equation (26) /*number of attention samples for each data stream*/
/* Building of the attention model and attention driven analysis */
8. AS'(t) = resampling((ss(t),as(t)), ($\Pi^S(t), \Pi^A(t)$ ),$N_A(t)$) /* create attention samples*/
9. AS(t) = AS'(t)
10. if $N_A(t) > 0$ then do
        begin
11.     for i = 1 to $N_A(t)$ do /*Attention driven analysis*/
            begin
12.         $f_L$ ← feature_extraction(as(t), visual(t)).
13.         Compute $P(H=face|f_L)$
14.         Calculate $P(H=face|f_L, a)= P(H=face|f_L). \pi_i^A(t)$
15.         SID = face?nonface: $P(H=face|f_L, a)>0.5$
16.         find $AS_+(t)$ by equation (49) /*the experience*/
17.         update $H(t)$ by equation (50)/*adaptation*/
            end
        end /* end of "if $N_A(t) > 0$" */
output(results form step 15, AS(t), $N_A(t)$)
end
```

Figure 24. Face detection by Experiential Sampling.

Motion cue: The motion cue is defined as

$$c\_sp_{MT}(t) = \{(x_{MT}^1, y_{MT}^1, w\_sp_{MT}^1), \mathrm{K}, (x_{MT}^{N_S}, y_{MT}^{N_S}, w\_sp_{MT}^{N_S})\} \quad (46)$$

The weight $w\_sp_{MT}^i$ is obtained by equation (19).

By using the motion and skin color cue, the equation (17) becomes

$$\pi_i^S(t) = \alpha_{MT} \cdot w\_sp_{MT}^i + \alpha_{Skin} \cdot w\_sp_{Skin}^i \quad (47)$$

Speech/Sound cue: The detection of speech/sound implies the possible existence of the face in the visual data. This is the case of utilizing information from multiple data-streams. We can then determine the attention saturation for this accompanying audio channel:

$$ASat_C(t) = \beta_S vol(t) \quad (48)$$

where *vol(t)* is the current volume of the audio channel and β_S measures the importance of the sound volume cue.

By utilizing this stream, we can dynamically update the number of attention samples on the visual when audio is detected. This consequently leads the face detector to work on more attention samples during the time slice when speech/audio occurs. For instance, Figure 23 shows that speech cue can help to create the attention samples when there is not any motion attention initially.

2) Effect of past experience on skin color model

As discussed earlier, we use the 1-D histogram of hues (color) channel to build the skin color model. In order to ensure that the past experience helps future data assimilation, we make the histogram model adaptive i.e. it is responsive to the current environment and the analysis results. Thus, the model parameters can dynamically adapt to the varying scene illumination. We denote it as H_t at time *t*. It is learned from the final analysis of the previous time slice. In our method, the face regions obtained by the face detection in time *t-1* serve as the feedback experience for the computation of the skin color histogram H_t for time *t*. This is formulated as follows by replacing equation (28) and (29).

$$AS_+(t) = \{AS(t) : S_{Adaboost}(f_L, AS(t)) = Tar\} \quad (49)$$

$$H(t+1) = Hist(hue(AS_+(t))) \quad (50)$$

where *Hist()* is the histogram of hue channel from the *reliable attention samples* $AS_+(t)$ while $AS_+(t)$ is obtained by equation (49). Based on this past experience, the skin color model is dynamically updated.

3) Audio-visual face detection by our experiential sampling technique

By utilizing the cues from visual data and companying audio data, our face detection method is summarized in Figure 24.

In addition, $ASat_C(t)$, which is the attention in the accompanying audio stream, can be used as the trigger to arouse the sensors in the visual stream:

if $ASat_C(t) > T$ **then start** *FaceDetection_by_ES(…)*

if $ASat_C(t) < T$ *and* $N_A(t) == 0$ *for a period of time* **then stop** *FaceDetection_by_ES(…)*

5.2 Monologue Detection

In this test example, we illustrate how to use our experiential sampling to deal with multiple data streams and multiple sub-tasks. We use two cameras and one audio sensor. One camera focuses on the whole scene (consisting of two people) while the other is supposed to be dynamically focused on the person speaking. We try to locate the speakers using one visual stream and one audio stream. From the detected faces and the lip-region analysis, we infer who the speaker is. Consequently the second camera is then zoomed onto the speaker. To this end, a face detector and a lip motion detector have to be used.

We use the method developed in [37] to adaptively adjust the camera parameters for zooming.

We show in Figure 25 (which summarizes the algorithm) how to locate relevant data for multiple sub-tasks (face detector and lip motion detector).

```
The Algorithm
MonologueDetection_by_ES(visualstream1(t),visualstream2, audio(t), AS(t-1), $N_A$(t-1),t)
begin
 /*  face detection*/
1. $ASat_C(t) \leftarrow audio(t)$ by equation (48)
2. if  $ASat_C(t) < T$ then exit
/* audio stream as the trigger for face detector*/
3. FaceRegions =
FaceDetection_by_ES(visualstream1(t),audio(t), ),
AS(t-1), $N_A$(t-1),t)
4. if  FaceRegions == 0 then exit
/* face detector as the trigger for lip motion detector*/
5. LipRegions ← find_lip_region(FaceRegions)   /* locate the lip regions*/
6. A SpeakerRegion ← lip_motion_detector(LipRegions)
/* find which face is speaking and obtain the zoom factor to control the second camera*/
7. visualstream2(t) ← zoom(SpeakerRegion, visualstream2(t)) /* control the second camera to focus on the speaker*/
 8.
lip_motion_detector(visualstream2(t))
/* making sure about the monologue by
checking for lip movement */
end
```

Figure 25.Algorithm of Monologue Detection.

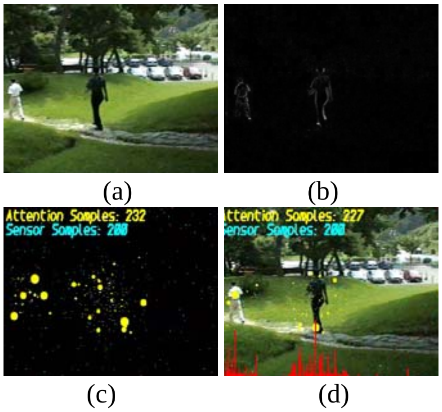
VI. EXPERIMENTS

In this section, we present results from the three test examples. The result videos are available for viewing at http://www.comp.nus.edu.sg/~mohan/ebs/.

A. *Activity Monitoring*

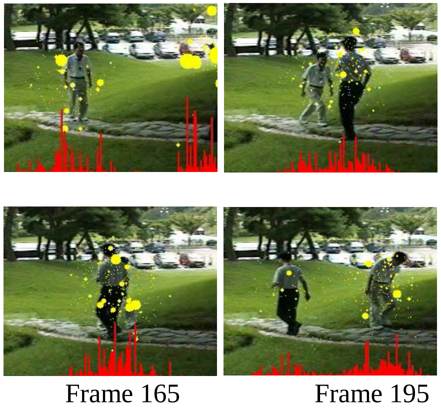
We test our method for the video of several pedestrians (Figure 26 and 27) and traffic monitoring sequences (Figure 28). There are 200 sensor samples randomly scattered spatially to sense the motion experience. Based on the sensor output, attention samples are created. Their numbers and spatial distribution are all determined by the motion experience. Figure 26 shows that, unlike the saliency map based attention model (indicated by Figure 26(b)), only 227 attention samples and 200 sensor samples are sufficient to maintain the motion attention.

The weight of each attention sample is drawn using red bars along with the x direction to visualize the spatial attention in *x* direction. From Figure 26, 27 and 28, we can see that our experiential sampling technique can model multi-modal motion attention quite well without maintaining the saliency map (which requires higher computation). The evolution of temporal attention (attention saturation) is shown in Figure 29.



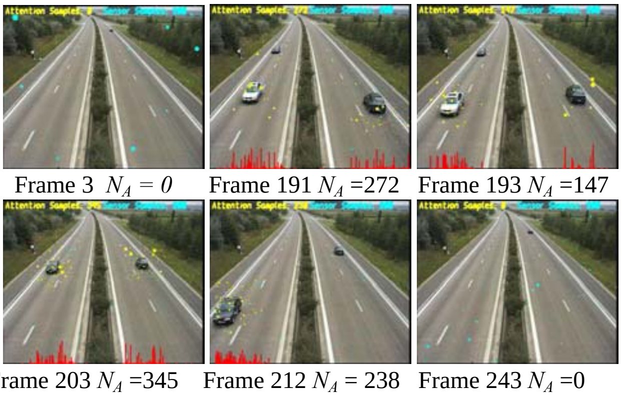
(a) (b) (c) (d)

Figure 26. Mpeg 7 Test sequence 1. (a) original frame.(b) saliency map for motion. (c) 232 attention samples (yellow points) for motion. (d) motion attention by attention samples with original frame. Red bar shows the spatial visual attention in *x* direction; yellow points show the 227 attention samples. Point size indicates the confidence of this sample. Blue points show the 200 sensor samples.



Frame 165 Frame 195

Figure 27. Mpeg 7 Test sequence 2. Red bar shows the spatial visual attention in *x* direction; Yellow points show the attention samples. Point size indicates the confidence of this sample. This figure illustrates the ability of maintaining multi-modal attention. Both visual attention emerge and split during and after the crossing of the subjects.



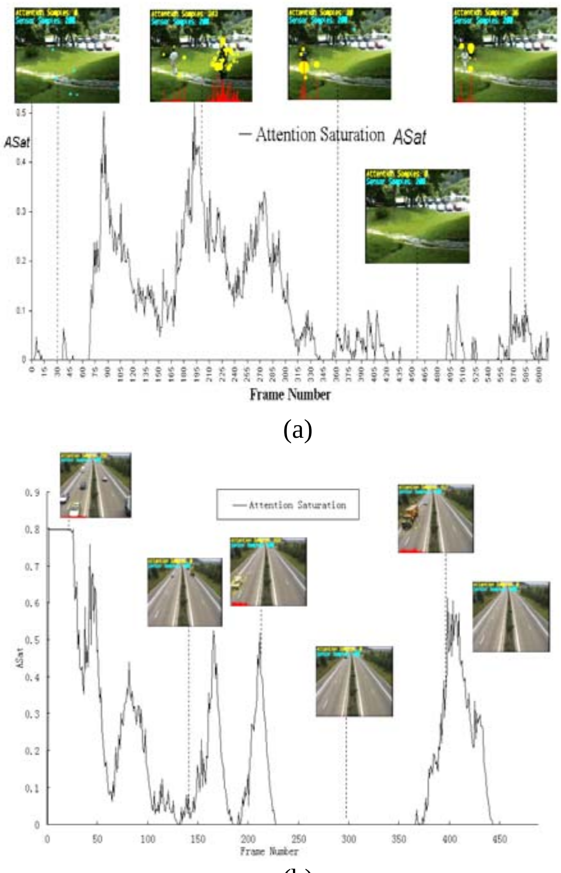
Frame 3 $N_A = 0$ Frame 191 $N_A = 272$ Frame 193 $N_A = 147$

Frame 203 $N_A = 345$ Frame 212 $N_A = 238$ Frame 243 $N_A = 0$

Figure 28. Traffic monitoring sequence. This figure illustrates the both spatial and temporal visual attention inferred from motion experience. Blue points are sensor samples while yellow points are attention samples. Red bar shows the spatial attention in *x* direction. It evolves according to the spatial experience. N_S number of sensor samples is set to 200. N_A number of attention samples changes each time based on the temporal experience.

Figure 29(a) shows that the temporal motion attention, calculated from the equation (24), evolves according to the motion activity in a pedestrian sequence. In Figure 29(b), the

N_A roughly reflects the traffic status at each time step. Therefore, our method here can be used for monitoring the traffic also. It also shows that the temporal attention is aroused only when the cars pass by. At other times, when N_A is zero, there are no attention samples. During this time, the only processing and analysis done is the sensor sampling. 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 process the entire data. It fulfills our aims of providing analysis have the ability to select the data to be processed.



(a)

(b)

Figure 29. Activity monitoring. (a) The pedestrian activity monitoring by attention saturation. (b) Traffic monitoring by attention saturation.

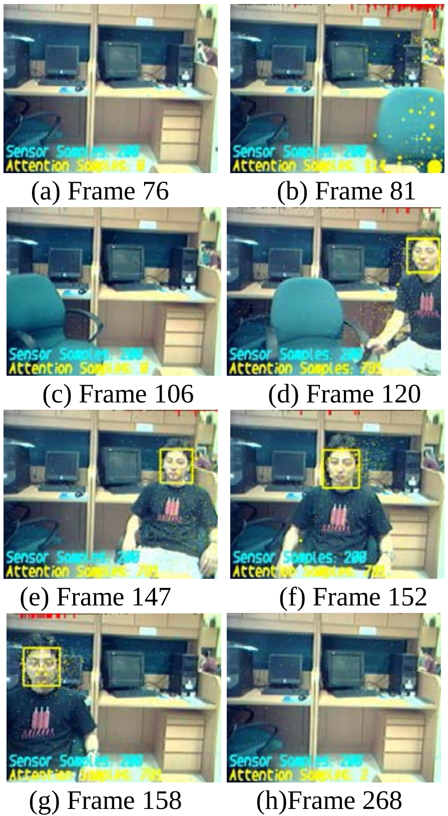
B. Face Detection Results

We use our experiential sampling technique to solve the face detection problem. Sensor samples are employed to obtain the current environment from the skin color, motion and audio cues. The face attention is maintained by the attention samples.

1) Face Detection by spatial cues

Because of our sampling method, the *adaboost* face detector is not applied on all the pixel and regions. The face detector is only executed on the attention samples which indicate the most probable face data regions.

Figure 30 shows that face detection results by using the motion cue. As shown in Figure 30, N_S number of sensor samples is set to 200.The number and spatial distribution of attention samples can dynamically change according to the face attention. In Figure 30 (a), there is no motion in the frame, so N_A, the number of attention samples is zero. No face detection is performed. In Figure 30 (b), when a chair enters, it alerts the motion sensor and attention is aroused. N_A increases to 414. Face detection is performed on the 414 attention samples. But the face detector verifies that there is no face there. In Figure 30 (c) as the chair stops, there is no motion and so the attention samples vanish. In Figure 30(d)-(h) attention samples come on with the face until the face vanishes.



(a) Frame 76 (b) Frame 81

(c) Frame 106 (d) Frame 120

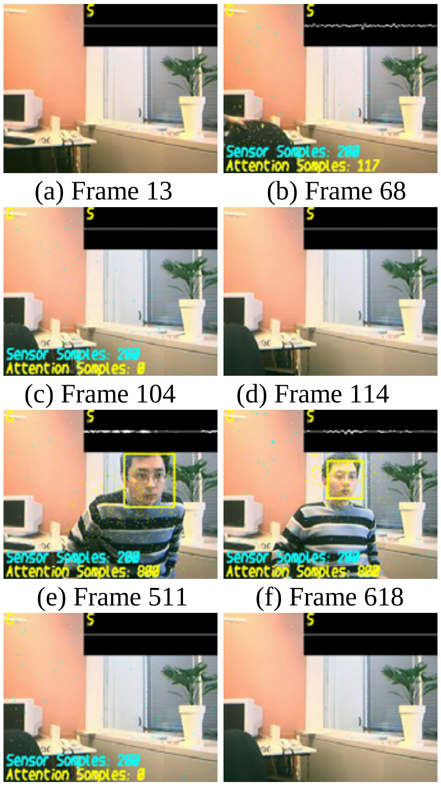
(e) Frame 147 (f) Frame 152

(g) Frame 158 (h)Frame 268

Figure 30. Face detection by using spatial cues. (a) static frame *N_A=0* (b)A chair moves *N_A=414* (c)the chair stopped. *N_A=0* (d) a person comes. *N_A=791* (e) a person. *N_A=791* (f) one person. *N_A=791* (g) one person. *N_A=791* (h) static frame. *N_A=2.*

2) Audio-Visual Face Detection

Figure 31 shows the face detection by using the audio-visual data from the two different streams. Figure 31(a) shows the initial status: there is no face detection working in the visual stream. The only processing is in the audio stream for the purpose of detecting the sound volume. In Figure 31(b), when a chair enters, it alerts the volume sensor in the audio stream and triggers the face detection module in the visual stream. Thus, sensors in the visual streams start to work: 200 sensor samples are uniformed randomly sampled and sense the visual scene. Based on this, 117 motion attention samples are aroused to follow the moving object (chair). Face detection is performed on those attention samples. But the face detector verifies that there is no face there. In Figure 31(c), the chair stops. It causes the volume in audio stream becomes zero and the attention samples vanish. If this state remains for a short period of time, the face detection module in the visual stream is shut down again as shown in Figure 31(d). In Figure 31(e)-(f), the volume sensor arouses the face detection module again when a person enters. Attention samples are aroused by both the spatial cues in the visual stream (motion/ skin color) and the temporal cue in audio stream (volume). The attention samples come on with the face until the face vanishes and audio stream become silent again (In Figure 31(g)). If the system is in this state for a while, face detection is shut down again due to the no activity in both the audio and visual streams. Only the sensor sampling of the audio-visual environment continues to take place.



(a) Frame 13 (b) Frame 68

(c) Frame 104 (d) Frame 114

(e) Frame 511 (f) Frame 618

(g) Frame 698 (h) Frame 700

Figure 31. Audio-visual face detection by Experiential Sampling.

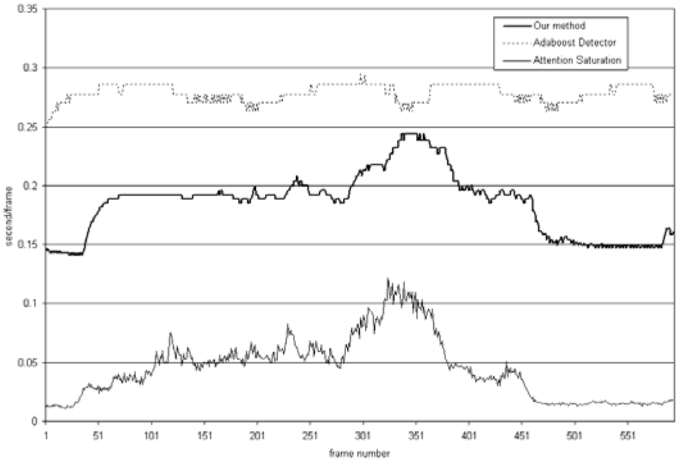


Figure 32. Comparison of the computation speed.

3) Computation Speed

We use a USB web camera to perform real time face detection on a Pentium III 1GHz laptop. The graph of the computation load, indicated by *sec/frame*, in this real time scenario is shown in Figure 32. Note that our absolute speed ((with frame capturing, rendering, recording results (saving to disks), *etc.*)) is constrained by the capture speed of the USB camera. However, we intend to show the adaptability of our computational load rather than the absolute speed. In Figure 32, curve 1 shows the computation load of the *adaboost* face detection while curve 2 indicates the computation load of our experiential sampling with *adaboost* face detector. This figure shows that by using our experiential sampling technique, computation complexity can be significantly reduced. In addition, in order to show the adaptability, we also depict the value of attention saturation in the graph. It shows that the computation complexity varies according to the difficulty of the current task, which is measured by the attention saturation. This is the expected behavior as deduced in equation (32).

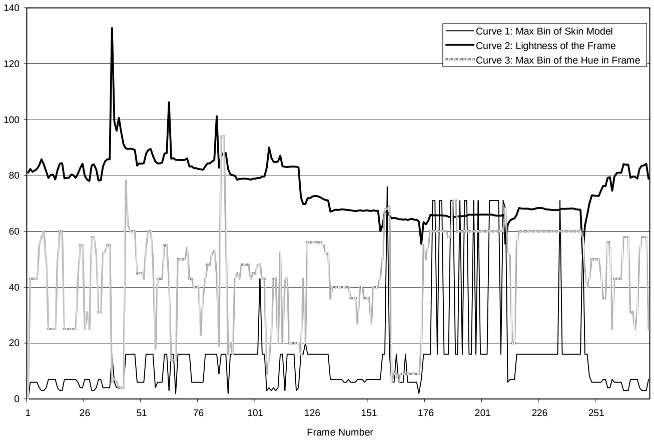


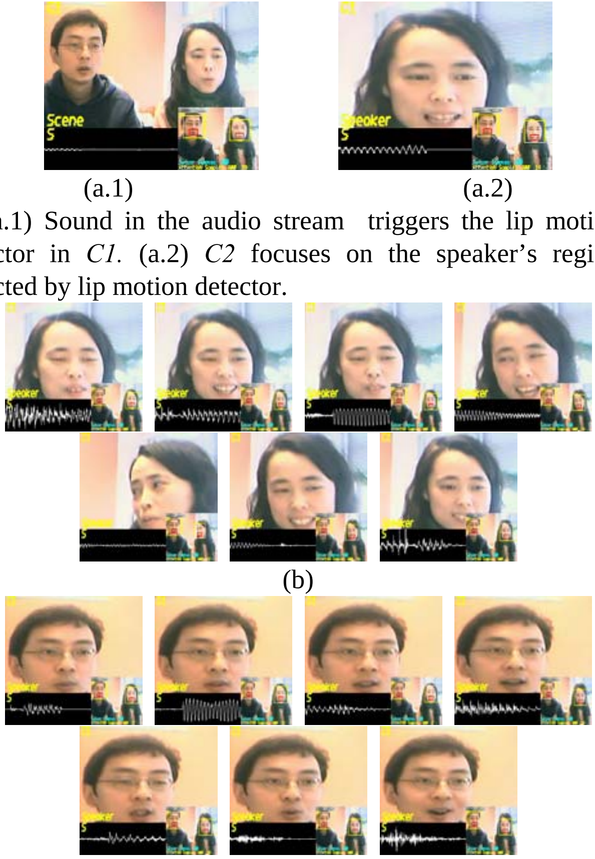
Figure 33. Our Adaptive Model that adapts to the environment.

4) Past Experiences

Based on the discussion in Section III.C.7 and V.B.2, we have implemented the use of the past experience for building the dynamic skin color model. The experimental results are shown in Figure 33. We change the luminance of our visual scene. This consequently causes the global visual environment to vary, which is indicated by the curve 1 (luminance) and curve 2 (Max bin in hue) as shown in Figure 33. By constantly updating the skin color model from the previous analysis, our skin color model can dynamically adapt to the changed visual environment.

C. Monologue Detection

For the monologue detection, we intend to show our approach for integrated analysis on multiple streams and sub-tasks rather than giving quantitative test results. The results of the monologue detection are shown in Figure 34. Figure 34(a) shows the procedure. When there is a sound in the audio stream as shown in Figure 32(a.1), the lip motion detector starts up and speaker is found in camera 1. Then, camera 2 starts to focus on the speaker's region which is detected by the lip motion detector in Camera 1. Detected faces are marked as yellow regions while lip regions are marked as red regions. Face detector and lip motion detector perform measurements on the camera 1, which is indicated in the bottom-right of the frame. Camera 2 zooms in to the speaker's region, which allows further visual analysis to be performed on the output of camera 2 in order to obtain more accurate results. Figure 34(b) shows the detection results for a sequence in which two different speakers speak at different times. Therefore, the second camera focuses on a different person depending on who is speaking.



(a.1) (a.2)

(a.1) Sound in the audio stream triggers the lip motion detector in *C1*. (a.2) *C2* focuses on the speaker's region detected by lip motion detector.

(b)

(c)

Figure 34 Sample frames for Monologue Detection Results

VII. CONCLUSIONS

In this paper, we describe a novel sampling based framework for multimedia analysis called experiential sampling. Based on this framework, we can utilize the context of the experiential environment for efficient and adaptive computations. Inferring from this environment, the multimedia system can select its data of interest while immediately discarding the irrelevant data. As examples, we utilize this framework for the activity monitoring, face detection and monologue detection problems. The results establish the efficacy of the sampling based technique. In the future, other applications like adaptive streaming and surveillance with more sources of different modalities will be further investigated.

What we have essentially done is to formulate the problem of identifying the optimal feature stream subset Φ^* of a multimedia system to accomplish its task. We have formalized the problem to cast it as a Markovian decision problem and have provided an optimal procedure to identify this subset as well as to estimate the cost of identifying this optimal subset. However, much more remains to be done:

- Given this optimal subset Φ^*, how do we best fuse the information from the various feature streams for a particular problem? One possibility is a linear fusion framework. Another possibility is a dynamical system based approach. Model predictive controllers [43] seem to be an attractive option. Or some energy minimization [39] or MDL based approach [40] might turn out to be useful. These are fruitful avenues for future investigations.
- How do we combine continuous feature streams with symbolic feature streams? For example, text stream is often available with video streams. How can the text stream be effectively exploited for video analysis in this case?
- Having identified Φ^*, how do we distribute the attention samples among the various streams belonging to Φ^*? We have suggested one method based on attention saturation. Can it be done in a more efficient manner?
- How off are we from the optimal condition if a particular feature stream from Φ^* drops off? The idea is to gracefully degrade any system and to have a quantitative notion about it. This can have practical implications for handling sensor failures and run-time maintenance of multimedia systems.
- How exactly do we trade one feature stream of Φ^* versus a subset others? The directed graph model will help along with the cost of each feature stream. This can help select different subset of sensors depending on other criteria.
- Our main contribution is the introduction of generalized goal-oriented attention for multiple sensor data streams which are not necessarily biological sensors. Moreover, this attention function has been identified as dynamically varying phenomenon which is continuously updated based on past experience and current context. We have used the sampling framework to mathematically model this phenomenon. Can some other more economical mathematical model be developed for capturing this phenomenon?
- Though we have been inspired by the human phenomenon of attention, we have adopted an engineering approach to solve the problem. However, it may be worthwhile to computationally mimic the biological phenomenon. Building biologically plausible models of attention would be an interesting challenge. Some of the findings by cognitive scientists [6, 8] would be extremely useful for this purpose.

REFERENCES

[1] R. O. Duda, P. E Hart, and D. G. Stork, *Pattern Classification. Second Edition*. Wiley-Interscience, New York. 2001.

[2] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *Tech. Rep. 1532*, M.I.T.: Artificial Intelligence Laboratory and Center for Biological and Computational Learning, 1994.

[3] Rowley, H., Baluja, S., and Kanade, T. 1998. Neural Network-based Face Detection. In *IEEE Trans. Pattern Analysis and Machine Intelligence*,

Vol. 20, No. 1, pp. 23-28.

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

[5] S. Z. Li, L. Zhu, Z-Q. Zhang, A. Blake, H-J. Zhang, and H. Shum, “Statistical Learning of Multi-View Face Detection,” in *Proc. 7th European Conference on Computer Vision,* Copenhagen, Denmark. 2002.

[6] B. J. Scholl, “Objects and Attention: The State of the Art”, *Cognition*, Vol. 80, pp. 1-46, 2001.

[7] R. Jain, “Experiential Computing,” *Communications of the ACM*. Vol. 46, No. 7, pp. 48-55, 2003.

[8] S. B. Most, D.J. Simons, B. J. Scholl, R. Jimenez, E. Clifford, and C. F. Chabris, “How not to be Seen: The Contribution of Similarity and Selective Ignoring to Sustained Inattentional Blindness,” *Psychological Science*, Vol. 12, No. 1, pp. 9-17, 2001.

[9] D. Chung, R. Hirata, T. N. Mundhenk, J. Ng, R. J. Peters, E. Pichon, A. Tsui, T. Ventrice, D. Walther, P. Williams, and L. Itti, “A New Robotics Platform for Neuromorphic Vision: Beobots,” in *Proc. 2nd Workshop on Biologically Motivated Computer Vision (BMCV'02),* Tuebingen, Germany, 2002.

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

[11] Y. Ma, L. Lu, H-J. Zhang, and M. Li, “A User Attention Model for Video Summarization,” in *Proc. Tenth ACM International Conference on Multimedia*, pp. 533-542, 2002.

[12] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch, “ Attentional Selection for Object Recognition - a Gentle Way,” in *Proc. of 2nd Workshop on Biologically Motivated Computer Vision (BMCV'02),* pp. 472-479, Germany, 2002.

[13] V. Navalpakkam, and L. Itti, “A Goal Oriented Attention Guidance Model,” in *Proc. 2nd Workshop on Biologically Motivated Computer Vision (BMCV'02),* pp. 453-461, Germany, 2002.

[14] E. K. Miller, “The Prefrontal Cortex and Cognitive Control,” *Nature Rev. Neuroscience*. Vol. 1, No. 1, 59-65, 2000.

[15] A. K. Dey, and G. D. Abowd, “Towards a Better Understanding of Context and Context-awareness,” in *H-W Gellerson, Editor,* Handheld and Ubiqitous Computing*, Number 1707 in Lecture Notes in Computer Science*, pp. 304-307. Springer, 1999.

[16] H. Lieberman, and T. Selker, “Out of Context: Computer Systems That Adapt to, and Learn From, Context,” *IBM Systems Journal*, Vol. 39, Nos. 3&4, pp. 617-632, 2000.

[17] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The Active Badge Location System,” *ACM Transactions on Information Systems*, Vol 10, No. 1, pp. 91-102, 1992.

[18] M. Isard and A. Blake, “Condensation-conditional Density Propagation for Visual Tracking,” *International Journal on Computer Vision*, Vol. 29, No. 1, pp. 5-28, 1998.

[19] C. Colombo, M. Rucci, and P. Dario, “Integrating Selective Attention and Space-variant Sensing in Machine Vision. In *Jorge L.C. Sanz, editor, Image Technology: Advances in Image Processing, Multimedia and Machine Vision*, pp. 109-128. Springer, 1996.

[20] E-C. Chang, S. Mallat, and C. Yap, “Wavelet Foveation,” *Journal of Applied and Computational Harmonic Analysis*, Vol. 9, No. 3, pp. 312-335, 2000.

[21] E. L. Schwartz, D. N. Greve, and G. Bonmassar, “Space-variant Active Vision: Definition, Overview and Examples,” *Neural Networks*, Vol. 8 No. 7-8, pp. 1297-1308. 1995.

[22] J. Wang, R. Achanta, and M. S. Kankanhalli, “A Hierarchical Framework for Face Tracking Using State Vector Fusion for Compressed Video,” in *the 28th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, 2003.

[23] J. Wang, “Detecting and Tracking Human Faces in Compressed Domain for Content Based Video Indexing,” M.S. thesis, Sch. of Computing, National Univ. of Singapore, Singapore, 2002.

[24] Z. Ghahramani, and G. Hinton, “Parameter Estimation for Linear Dynamical Systems,” *Technical Report CRG-TR-96-2*, Dept. Comp.Sci., Univ. Toronto, 1996. Available: http://www.cs.toronto.edu/ ~ hinton/absps/tr96-2.html

[25] A. Doucet, S. Godsill, and C. Andrieu, “On Sequential Monte Carlo Sampling methods for Bayesian Filtering,” *Statistics and Computing*, Vol. 10, No. 3, pp. 197-208, 2000.

[26] D. Rubin, “Using the SIR Algorithm to Simulate Posterior Distributions (with discussion),” in *Bayesian Statistics* 3, eds. J.M. Bernard, M.H. DeGroot, D.V. Lindley, and A.F.M. Smith, New York: Oxford University Press, pp. 395-402, 1998.

[27] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation,” in *IEEE Proceedings on Radar and Signal Processing*, Vol. 140, No. 2, pp. 107-113, 1993.

[28] L. Itti, and C. Koch, “Feature Combination Strategies for Saliency-based Visual Attention Systems,” *J. Electronic Imaging*, Vol. 10, No. 1, pp. 161-169, 2001.

[29] J. Carpenter, P. Clifford, and P. Fearnhead, “Building Robust Simulation-based Filters for Evolving Data Sets,” *Technical report*, University of Oxford, Dept. of Statistics, 1999. Avalable: http:// www.stats.ox.ac.uk/pub/clifford/Particle_Filters/jj.Abstract.html

[30] G. R. Bradski, “Computer Vision Face Tracking For Use in a Perceptual User Interface,” *Intel Technology Journal*, 2nd quarter, 1998.

[31] A. Torralba, “Contextual priming for object detection,” *International Journal of Computer Vision*, Vo. 53, No. 2, pp. 169-191, 2003.

[32] A. Jordan, W. Richards, and D. Knill, “Modal structures and reliable inference,” *Perception as Bayesian Inference*, eds. D. Knill and W. Richards, Cambridge Univ. Press, pp. 63-92, 1996.

[33] Moghaddam, B., and Pentland, A. 1997. Probabilistic Visual Learning for Object Representation. IEEE Trans. Pattern Analysis and Machine Vision, Vol. 19, No. 7, pp. 696-710.

[34] Liu, J., Chen, R. 1998. Sequential Monte Carlo for Dynamic Systems. *Journal of the American Statistical Association*, Vol. 93, pp. 1031-1041.

[35] P. Pirolli, and S. K. Card, “Information foraging,” *Psychological Review*, Vol. 106, No. 4, pp. 643-675, 1996.

[36] W. James, *The Principles of Psychology*. Harvard Univ. Press, Cambridge, 1981.

[37] J. Wang, W.Q. Yan, M. S. Kankanhalli, R. Jain, and M. J. T. Reinders, “Adaptive Monitoring for Video Surveillance,” in *The Fourth IEEE Pacific-Rim Conference on Multimedia (PCM 2003),* Singapore, 2003.

[38] R. Debouk, S. LaFortune, and D. Teneketzis, “On an Optimization Problem in Sensor Selection,” *Journal of Discrete Event Dynamical Systems: Theory and Applications*, Vol. 12, No. 4, pp. 417-445, 2002.

[39] G. Chapline, “Minimum Energy Information Fusion in Sensor Networks,” in *Proceedings of The 2nd International Conference on Information Fusion*, 1999.

[40] M. H. Hansen, and B. Yu, “Model Selection and the Principle of Minimum Description Length,” *Journal of the American Statistical Association*, Vol. 96, No. 454, pp. 746-774, 2001.

[41] D. A. Forsyth, J. Haddon, and S. Ioffe, “The Joy of Sampling,” *International Journal of Computer Vision*, Vol. 41, No. 1/2, pp. 109-134, 2001.

[42] A. Cabrales, R. Nagel, and R. Armenter, “Equilibrium Selection through Incomplete Information in Coordination Games: An Experimental Study, ” 2002. [Online] Available: http://pubweb.northwestern.edu/~ arm066/experimental. pdf

[43] M. Nikolaou, “Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs,” *Advances in Chemical Engineering Series*, Academic Press, 2001.

[44] U. Neisser, *Cognition and Reality*, W.H. Freeman, San Francisc, 1976.

[45] C. Neti, B. Maison, A. Senior, G. Iyengar, P. Decuetos, S. Basu and A. Verma, “Joint Processing of Audio and Visual Information for Multimedia Indexing and Human-Computer Interaction,” in *Proc. RIAO (Computer Assisted Information retrieval)*, France, 2002.

[46] G. Iyengar, H. Nock, and C. Neti, “Audio-Visual Synchrony for Detection of Monologues in Video Archives,” in *Proc. ICASSP,* 2003.

[47] D. Li, N. Dimitrova, M. Li, and I. Sethi, “Multimedia Content Processing through Cross-Modal Association,” in *Proc. ACM International Conference on Multimedia (ACM MM 2003)*, Berkeley, 2003.

[48] J. Wang. M. J. T. Reinders, R. L. Lagendijk, J. Lindenberg, and M. S. Kankanhalli, “Video Content Representation on Tiny devices” in *Proc. IEEE International Conference on Multimedia and Expro*. Taipei, July 2004.

[49] J. Wang and M. S. Kankanhalli, “Experience-based Sampling for Multimedia Analysis,” in *Proc. of ACM Multimedia 2003*. (Short paper), Berkeley, pp. 319-322, November, 2003.

[50] J. Wang, M. S. Kankanhalli, W.Q. Yan, and R. JAIN, “Experiential Sampling for Video Surveillance,” in *Proc. 1st ACM Int. Workshop on Video Surveillance*, Berkeley, 2003.

[51] R. Jain, “Semantics in Multimedia Systems,” Keynote talk at *International Conference on Multi-Media Modelling*, Taipei, January 8-10, 2003.

Mohan S. Kankanhalli is a Professor at the Department of Computer Science of the School of Computing at the National University of Singapore. He obtained his BTech (Electrical Engineering) from the Indian Institute of Technology, Kharagpur, and his MS and PhD (Computer and Systems Engineering) from the Rensselaer Polytechnic Institute. He has worked at the Institute of Systems Science (ISS - now Institute for Infocomm Research) in Singapore and at the Department of Electrical Engineering of the Indian Institute of Science, Bangalore. His current research interests are in Multimedia Information Systems (content processing, multimedia retrieval) and Information Security (media watermarking and authentication). He is on the editorial board of several journals including the ACM Multimedia Systems journal and the IEEE Transactions on Information Forensics and Security.

Jun Wang received the BE degree in electrical engineering from the Southeast University in Nanjing, China, and the MS degree in computer science from the National University of Singapore, Singapore. He is now a PhD student with the Information and Communication Theory Group, the Faculty of Electrical Eng., Mathematics and Computer Science (EWI), the Delft University of Technology, the Netherlands. His current research topic is multimedia information personalization and recommender systems.

Ramesh Jain is the Bren Professor of Information and Computer Sciences, the Department of Computer Science, the University of California, Irvine. Ramesh has been an active researcher in multimedia information systems, image databases, machine vision, and intelligent systems. While he was at the University of Michigan, Ann Arbor and the University of California, San Diego, he founded and directed artificial intelligence and visual computing labs. He was also the founding Editor-in-Chief of IEEE MultiMedia magazine and Machine Vision and Applications journal and serves on the editorial boards of several magazines in multimedia, business and image and vision processing. He has co-authored more than 250 research papers in well-respected journals and conference proceedings. Among his co-authored and co-edited books include Machine Vision, a textbook used at several universities. Ramesh has been elected Fellow of ACM, IEEE, IAPR, AAAI, and SPIE. He enjoys working with companies, is involved in research, and enjoys writing. His current research is in experiential systems and their applications.