Goal-Oriented Optimal Subset Selection of Correlated Multimedia Streams

PRADEEP K. ATREY and MOHAN S. KANKANHALLI
National University of Singapore
and
JOHN B. OOMMEN
Carleton University

A multimedia analysis system utilizes a set of correlated media streams, each of which, we assume, has a confidence level and a cost associated with it, and each of which partially helps in achieving the system goal. However, the fact that at any instant, not all of the media streams contribute towards a system goal brings up the issue of finding the best subset from the available set of media streams. For example, a subset of two video cameras and two microphones could be better than any other subset of sensors at some time instance to achieve a surveillance goal (e.g. event detection). This article presents a novel framework that finds the optimal subset of media streams so as to achieve the system goal under specified constraints. The proposed framework uses a dynamic programming approach to find the optimal subset of media streams based on three different criteria: first, by maximizing the probability of achieving the goal under the specified cost and confidence; second, by maximizing the confidence in the achieved goal under the specified cost and probability with which the goal is achieved; and third, by minimizing the cost to achieve the goal with a specified probability and confidence. Each of these problems is proven to be NP-Complete. From an AI point of view, the solution we propose is heuristic-based, and for each criterion, utilizes a heuristic function which for a given problem, combines *optimal* solutions of small-sized subproblems to yield a potential near-optimal solution to the original problem. The proposed framework allows for a tradeoff among the aforementioned three criteria, and offers the flexibility to compare whether any one set of media streams of low cost would be better than any other set of higher cost, or whether any one set of media streams of high confidence would be better than any other of low confidence. To show the utility of our framework, we provide the experimental results for event detection in a surveillance scenario.

Categories and Subject Descriptors: H.5.1 [**Information Interfaces and Presentation**]: Algorithms

General Terms: Algorithms, Security

Additional Key Words and Phrases: Agreement coefficient, confidence fusion, media fusion, optimal media selection

ACM Reference Format:

Atrey, P. K., Kankanhalli, M. S., and Oommen, J. B. 2007. Goal-Oriented optimal subset selection of correlated multimedia streams. ACM Trans. Multimedia Comput. Commun. Appl. 3, 1, Article 2 (Feb. 2007), 24 pages. DOI = 10.1145/1198302.1198304 http://doi.acm.org/10.1145/1198302.1198304.

Authors' addresses: P. K. Atrey (contact author), M. S. Kankanhalli, School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Republic of Singapore; email: pradeepk@comp.nus.edu.sg; J. B. Oommen, School of Computer Science, Carleton University, Ottawa, ON: K1S 5B6, Canada.

1. INTRODUCTION

Most media analysis tasks can be better performed by using *multiple* correlated media, as compared to using only *single* medium. This is because a single type of media device can only partially help in achieving a system goal (e.g. event detection in multimedia surveillance scenario) due to its ability to sense only a part of the environment, and also due to the inaccuracies in capturing and processing media data. Moreover, multiple media devices can capture different aspects of the environment to provide complementary information which is not available from a single medium. On the other hand, the system designer can have various confidence levels in different media for different system goals. For instance, we can have more confidence in a video stream than an audio stream if the goal is 'to detect faces,' and a better (or more accurate) face detector is available.

The fact that at any instant, all employed media streams do not necessarily contribute towards a goal brings up the issue of finding the most informative subset of media streams with a high confidence level. The most informative subset of streams dynamically changes over time. Once this subset is found, we can continue using it for a certain period while ignoring the remaining streams. This eliminates the cost of using a redundant and less informative stream. The cost of using a media stream usually includes the cost of a media device, its installation and maintenance costs, and the cost of energy to operate and process it.

The selection of the optimal subset of streams is an important research problem in application scenarios including, but not limited to, surveillance and monitoring, media search etc. In surveillance systems, people employ multiple sensors, such as video cameras, microphones, and infra-red cameras, and also use other nonsensory data. Such systems "on-the-fly" should be able to find whether a set of two video cameras and two microphones would be better when compared to any other set of one video camera and three microphones in achieving a specific surveillance goal of detecting a suspicious activity. In media search systems (which have gained fair attention within the research community [Jain 2004]), the system should be able to select the optimal subset from stored media streams on a server for download and play.

In this article, we describe a proposed framework for determining a near-optimal[1] media selection scheme with a detailed heuristic explanation and the corresponding experimental results.[2]

The proposed framework essentially addresses the following research issues:

(1) What is the optimal number of media streams required to achieve the goal under the specified constraints?

(2) Which subset of streams is the optimal?

(3) In case the most suitable subset is unavailable, can we use alternate media streams without much loss of cost-effectiveness and confidence?

(4) How frequently should this optimal subset be computed so that the overall cost of the system is minimized?

Given a set of media streams, which subset is the optimal? This question can be answered in many ways. The optimal subset may be the one which:

(1) maximizes the probability of achieving the system goal under the specified maximum cost and with a specified minimum confidence;

[1] We will show that the problem of obtaining the optimal solution is NP-Complete. Thus, while we seek the optimal solution, our goal will be to attain one that is reasonably close to optimal.

[2] The earlier version of some of the results found here was published in Atrey and Kankanhalli [2005].

(2) maximizes confidence in the media streams used, with a specified minimum probability of achieving the system goal under a specified maximum cost; and

(3) minimizes the cost of using the media streams so as to attain a specified minimum probability of achieving the system goal with a specified minimum confidence.

We thus study the problem of optimal stream selection from the three different aforementioned angles.

We reduce the 0-1 KNAPSACK problem to that of optimal media selection and use a dynamic programming approach to solve it. In our problem, for each stream, its probability of contributing towards the goal and the system designer's confidence level in it are analogous to the *profit*, while its cost is analogous to the *weight*, of a KNAPSACK problem. The fundamental difference is that we fuse the probabilities and confidence levels using a Bayesian approach [Atrey et al. 2005], while the profits are added in the 0-1 KNAPSACK problem.

From a theoretical perspective, the problem is proven to be NP-Complete. Thereafter, the proposed framework uses a dynamic programming approach that finds the optimal subset of streams based on the preceding three criteria. From an AI point of view, the solution we propose is heuristic-based, and for each criterion, utilizes a heuristic function which, for a given problem, combines *optimal* solutions of small-sized subproblems to yield a potential near-optimal solution to the original problem. To achieve the latter, we resort to a recent result proven in Oommen and Rueda [2005], where the authors showed that the quality of a heuristic algorithm[3] is determined by the accuracy of the heuristic function it uses. The details of how this result is used in this context is also discussed.

The rest of this article is organized as follows. We highlight related work in Section 2. In Section 3, we formulate the problem of determine the optimal subset. In Section 4, we present our framework in detail. We provide the experimental results in Section 5. Finally, Section 6 concludes the article with some discussions on future work.

2. RELATED WORK

In the past, the optimal sensor selection problem has been widely studied in various contexts. In the context of discrete event systems and failure diagnosis Debouk et al. [2002] formulated the optimization issue as a Markovian decision problem (MDP) with the objective of identifying instances where it is possible to explicitly determine optimal strategies. The sequence of tests is applied to identify the least costly sensor combination that satisfies a set of system properties (such as diagnosability) with the minimum expected number of tests. The method works under specified assumptions which are overconstrained. For instance, the authors assume an uniform cost for all sensors, which is impractical in a multimedia environment where different *types* of media are employed. Their work also does not integrate the confidence in sensors, as does our proposed framework. Jiang et al. [2003] presented a formal method for optimal sensor selection for discrete event systems with partial observation. The sensor subset (or observation mask) that qualifies for selection must follow desired formal properties, such as (co-)observability, or normality (for control), state observability (for state estimation), diagnosability (for failure diagnosis) under partial observation, etc. However, their method neither considers the cost of obtaining a subset of sensors nor the system designer's confidence in this subset while attempting to determine the optimal observation mask.

A sensor selection method for the execution of continuous probabilistic queries [Lam et al. 2004] has also been proposed. Their method meets the accuracy requirement by selecting the set of highly

[3] This conjecture, which was unproven earlier, has been the basis for designing numerous algorithms such as the A* algorithm and its variants.

correlated sensors. The correlation is computed assuming that all the sensors are of the same type. Therefore, their method is not suitable for a set of heterogeneous sensors. Also, they do not explicitly consider the cost of each sensor.

In the area of wireless sensor networks, [Pahalawatta et al. 2004] proposed to solve the problem of optimal sensor selection by maximizing the information utility gained from a set of sensors subject to a constraint on the average energy consumption in the network. However, their method does not consider the confidence in sensors. Moreover, our framework also takes into account the processing cost of sensor data.

Recently, Isler and Bajcsy [2005] proposed a generic sensor model, where the measurements can be interpreted as polygonal, convex subsets of the plane. They used an approximation algorithm so as to minimize the error in estimating the position of a target. However, their work also does not explicitly have a notion of the cost of using streams nor the confidences in them.

In contrast to all the solutions previously described, our proposed work provides a *tradeoff* between the extent to which the goal is achieved, the confidence in the streams, and the cost of using streams. In addition, our method provides the system designer with the flexibility of choosing next best sensor if the best sensor is not available.

Siegel and Wu [2004] have also pointed out the importance of considering confidence in sensor fusion and have used Dempster-Shafer's 'theory of evidence' to fuse the confidences. In contrast, we model confidence fusion by using a Bayesian formulation because it is both simple and computationally efficient.

3. PROBLEM FORMULATION

We use the following model of computation.

$\mathcal{M}1$. **S** is a multimedia analysis system designed for a goal G, and employs a set $\mathbf{M}^n(t) = \{M_1, M_2, \ldots, M_n\}$ of n media streams at time t.

$\mathcal{M}2$. For $1 \leq i \leq n$, let $0 < p_i < 1$ be the *probability* of achieving the system goal G using the individual i^{th} media stream. p_i is also denoted as $P(G|M_i)$. Also, let P_Φ (also denoted as $P(G|\Phi)$) be the 'fused probability' of achieving the system goal G using a subset $\Phi \in \mathcal{P}(\mathbf{M}^n)$ of media streams. The 'fused probability' is the overall probability of achieving the system goal using a group of media streams [Atrey et al. 2005].

$\mathcal{M}3$. For $1 \leq i \leq n$, c_i, let be the *cost* per unit time of using stream i. Also, let $C_n = \sum_{i=1}^{n} c_i$ be the *total cost*.

$\mathcal{M}4$. For $1 \leq i \leq n$, let $0 < f_i < 1$ be the system designer's *confidence* in the i^{th} stream.

We make the following assumptions:

$\mathcal{A}1$. All media capture the same environment (but optionally, different aspects of the environment) and provide correlated observations.

$\mathcal{A}2$. The system designer's confidence level in each of the media streams is at least 0.5 i.e. $f_i > 0.5$. This assumption is reasonable, since it is not useful to employ a media device which is found to be inaccurate more than half of the time.

$\mathcal{A}3$. The system goal G is to test a specified hypothesis H. Examples of a hypothesis could be : 'There is a person is knocking at the door in the corridor.'

$\mathcal{A}4$. There are multiple system goals and each can be accomplished by using a subset of the total number of streams. Hence, there is a need to select the best subset for a specific system goal.

$\mathcal{A}5$. The fused probability of achieving the goal, as well as the overall confidence, increase monotonically as more streams providing similar evidence are used.

We formulate three different problems referred to as *multimedia selection* (MS) problems: MaxGoal, MaxConf, and MinCost as follows.

Find the subset $\Phi \in \mathcal{P}(\mathbf{M}^n)$ that:

- **Problem** MaxGoal : maximizes P_Φ subject to $C_\Phi \leq C_{spec}$ and $F_\Phi \geq F_{spec}$.
- **Problem** MaxConf : maximizes F_Φ subject to $C_\Phi \leq C_{spec}$ and $P_\Phi \geq P_{spec}$.
- **Problem** MinCost : minimizes C_Φ subject to $F_\Phi \geq F_{spec}$ and $P_\Phi \geq P_{spec}$.

The preceding notations are:

P_Φ is the fused probability of achieving the goal when the subset Φ of media streams is used by system **S**;
C_Φ is the cost of using the subset Φ of streams;
F_Φ is the overall confidence when the subset Φ of streams is used;
P_{spec} is the specified minimum fused probability of achieving the goal;
C_{spec} is the specified maximum overall cost (note that $C_\Phi \leq C_n$); and
F_{spec} is the specified minimum overall confidence.

3.1 Complexity of Computing Optimal Solutions to MS Problems

We endeavour to formulate a heuristic-based solution to the problem of obtaining the optimal subset of multimedia streams. We discuss why such a solution is necessary in subsequent paragraphs.

Each of these three MS problems is structurally similar to the 0-1 KNAPSACK problem. We now prove that the MS Problems are NP-Complete problems.

THEOREM 3.1. *MS Problems are NP-Complete problems whenever the number of media streams* $n \geq 2$.

PROOF. The three MS problems are optimization problems. They can be restated as decision problems in the following manner

MaxGoal $= \{$Does a subset Φ with $P_\Phi \geq P_{spec}$ exist : $F_\Phi \geq F_{spec}$ and $C_\Phi \leq C_{spec}\}$
MaxConf $= \{$Does a subset Φ with $F_\Phi \geq F_{spec}$ exist : $P_\Phi \geq P_{spec}$ and $C_\Phi \leq C_{spec}\}$
MinCost $= \{$Does a subset Φ with $C_\Phi \leq C_{spec}$ exist : $P_\Phi \geq P_{spec}$ and $F_\Phi \geq F_{spec}\}$

The proof for this theorem is similar for all three problems MaxGoal, MaxConf, and MinCost. We consider the case of problem MaxGoal. To prove problem MaxGoal to be an NP-Complete problem, we provide Lemmas 3.2, 3.3, and 3.4, which together prove Theorem 3.1. □

LEMMA 3.2. *The* 0-1 KNAPSACK *problem is reducible to problem* MaxGoal *in polynomial time*, that is, 0-1 KNAPSACK $\geq_{Polynomial}$ MaxGoal.

PROOF. We pick a known NP-Complete 0-1 KNAPSACK problem and define an instance of it as a 5-tuple

$$\langle \mathbf{U}^n, \mathbf{X}, \mathbf{W}, X_{spec}, W_{spec} \rangle$$

with a set $\mathbf{U}^n = \{u_i\}_{i=1}^n$ of n items, their profits $\mathbf{X} = \{x_i\}_{i=1}^n$, weights $\mathbf{W} = \{w_i\}_{i=1}^n$, specified minimum profit X_{spec}, knapsack capacity W_{spec}; and an objective of determining whether a subset $\Lambda \subseteq \mathbf{U}^n$ of items having overall profit $X_\Lambda \geq X_{spec}$ exists under the constraint $W_\Lambda \leq W_{spec}$, where W_Λ is the total weight of items of subset Λ.

The corresponding instance of MaxGoal is defined by a six-tuple

$$\langle \mathbf{M}^n, \mathbf{P}, \mathbf{F}, \mathbf{C}, P_{spec}, C_{spec}, F_{spec} \rangle$$

with a set $\mathbf{M}^n = \{M_i\}_{i=1}^n$ of n streams, the probabilities $\mathbf{P} = \{p_i\}_{i=1}^n$ of individually helping in achieving the goal, their confidences $\mathbf{F} = \{f_i\}_{i=1}^n$, costs $\mathbf{C} = \{c_i\}_{i=1}^n$, minimum specified fused probability P_{spec}, maximum specified cost C_{spec}, and minimum specified confidence F_{spec}; and with an objective of determining whether a subset $\Phi \subseteq \mathbf{M}^n$ of streams, based on which we obtain the fused probability $P_\Phi \geq P_{spec}$ of achieving the goal, exists under the constraints $C_\Phi \leq C_{spec}$ and $F_\Phi \geq F_{spec}$, where C_Φ and F_Φ are the total cost of using and overall confidence in, respectively, subset Φ.

A transformation function $T_r : K \rightarrow T_r(K)$ which maps an instance K of the 0-1 KNAPSACK problem into the given instance $T_r(K)$ of the MaxGoal problem is defined as $T_r(\mathbf{U}^n, \mathbf{X}, \mathbf{W}, X_{spec}, W_{spec})\{\mathbf{M}^n = \mathbf{U}^n, \mathbf{P} = \mathbf{X}, \mathbf{F} = NULL, \mathbf{C} = \mathbf{W}, P_{spec} = X_{spec}, C_{spec} = W_{spec}, F_{spec} = 0, \Phi = \Lambda, P_\Phi = X_\Lambda, C_\Phi = W_\Lambda\}$. Note that relaxing the constraint of confidence (i.e., making $F_{spec} = 0$) reduces the given instance of the MaxGoal problem into an instance of the 0-1 KNAPSACK problem.

We now argue that K has a solution if and only if $T_r(K)$ has a solution. If a subset Λ of items, with the overall profit X_Λ (by adding the profits obtained from individual items) within the weight $W_\Lambda \leq W_{spec}$, exists in an instance K of the 0-1 KNAPSACK problem; in the corresponding instance $T_r(K)$ of the MaxGoal problem, there exists a subset Φ of media streams based on which an overall probability $P_\Phi \geq P_{spec}$ of achieving the goal is estimated (by fusing with a Bayesian approach the probabilities of achieving the goal based individual streams) within the total cost $C_\Phi \leq C_{spec}$ and with the overall confidence $F_\Phi \geq F_{spec}$. Note that although X_Λ in the 0-1 KNAPSACK problem and P_Φ in the MaxGoal problem are computed using different methods, they are equivalent, as both are computable in polynomial time and both increase monotonically (as stated in the assumption $\mathcal{A}5$). We prove this using Lemma 3.3.

It is obvious that the transformation T_r of instances of the two problems can be done in polynomial time because there is a one-to-one correspondence, and K will have a solution iff $T_r(K)$ has a solution. This proves that the 0-1 KNAPSACK problem is reducible to the MaxGoal problem in polynomial time. □

LEMMA 3.3. *The functions to compute the overall profit X_Λ in 0-1 KNAPSACK problem and the overall probability P_Φ in the* MaxGoal *problem are equivalent.*

PROOF. As already known, in 0-1 KNAPSACK problem, the function to compute the overall profit is additive, whereas in the MaxGoal problem, the overall probability of the occurrence/non-occurrence of event is computed using a Bayesian formulation (Eq. (1) in Section 4.2), which is given as:

$$P_i = \frac{P_{i-1}.p_i.e^{\overline{\gamma}_i}}{P_{i-1}.p_i.e^{\overline{\gamma}_i} + (1 - P_{i-1})(1 - p_i).e^{-\overline{\gamma}_i}}.$$

By making the term $\overline{\gamma}_i = 0$, the preceding equation becomes

$$= \frac{\rho.\sigma}{\rho.\sigma + (1 - \rho)(1 - \sigma)},$$

where $\rho = P_{i-1}$, $\sigma = p_i$, and $0 < P_{i-1}, p_i < 1$. This equation, which contains multiplication and division steps, can easily be transformed into an additive function by replacing the multiplication and division steps with successive additions and subtractions, respectively, as

$$= \frac{\rho.\sigma}{2.\rho.\sigma + 1 - \rho - \sigma} = \frac{\overbrace{\rho.\rho\ldots}^{\sigma-times}}{2.\overbrace{\rho.\rho\ldots}^{\sigma-times} + 1 - \rho - \sigma} = \frac{\sum_1^\sigma \rho}{2.\sum_1^\sigma \rho + 1 - \rho - \sigma} = \rho'/\sigma',$$

where $\rho' = \sum_1^\sigma \rho$ and $\sigma' = 2.\sum_1^\sigma \rho + 1 - \rho - y$. Note that ρ' and σ' can be computed in time of polynomial order $O(d)$, where d is the degree of precision in considering the probability value σ.

Further transformation can be done as follows:

$$= \sum_{1}^{\rho'} 1 + \left(- \sum_{1}^{\sigma'} 1 \right)$$

which are simply additive steps.

The previous transformation will also hold for the case when $\overline{\gamma}_i \neq 0$. The only difference would be that the time complexity of computing the overall probability using the aforementioned equation will be of polynomial order $O(n \times d)$, since the computation of $\overline{\gamma}_i$ (refer to Section 4.2) would also require $O(n)$ time.

The preceding arguments prove Lemma 3.3. □

LEMMA 3.4. *Problem* MaxGoal *is in NP.*

PROOF. To prove that the problem MaxGoal is NP, we show that the solution to the decision version of MaxGoal problem can be verified in polynomial time.

To verify if there exists a subset Φ of media streams based on which we obtain a fused probability $P_\Phi \geq P_{spec}$ of achieving the goal within the total cost $C_\Phi \leq C_{spec}$ and with the overall confidence $F_\Phi \geq F_{spec}$; one can simply make the choices of streams in $O(n)$ time, and can fuse the probabilities (of achieving the goal based on individual streams) and their confidence levels. Their costs can simply be added. We can then compare the overall confidence and the total cost of using streams with the specified constraints. If $C_\Phi \leq C_{spec}$ and $F_\Phi \geq F_{spec}$ are true, then the solution is correct, else it is not. This proves that problem MaxGoal does belong to the NP class.

Lemmas 3.2, 3.3, and 3.4 together prove that the problem MaxConf is NP-Complete.

In the case of problem MaxConf, the proof follows the same lines of reasoning for problem MaxGoal, except that in this case, we would present the same arguments for F_Φ, instead of P_Φ. Similarly, in the case of problem MinCost, the proof is similar with the arguments for C_Φ, instead of P_Φ. The details are omitted due to space constraints. □

In light of Theorem 3.1, we proceed with the major thrust of this article, namely, to develop techniques for obtaining approximate solutions to the problems.

3.2 Developing Approximate Solutions to Problems MaxGoal, MaxConf, and MinCost

From a computational and practical perspective, Theorem 3.1 justifies the research for developing heuristic-based solutions because the optimal solution can only be obtained by exhaustive search of the entire solution space. The computation of the exact solution by a brute-force strategy would require a combinatorially explosive number of operations, which is infeasible for typical values of n occurring in any large-scale application. Finally, as mentioned before, there doesn't seem to be any systematic way by which any partial solution can be discarded, except by some type of branch-and-bound philosophy in which a particular subset is discarded (after being initially investigated) when its current *partial* solution is already more expensive that the *total* solution of another subset.

3.2.1 *Rationale for the Heuristic.* In an effort to develop good heuristic solutions to the various problems MaxGoal, MaxConf, and MinCost, we resort to the recent results of Oommen and Rueda [2005].

To explain these results of Oommen and Rueda [2005], consider a heuristic algorithm A. Suppose that A could invoke one of two possible heuristic functions. The question of determining which heuristic function is superior has typically demanded a yes/no answer, often substantiated by empirical evidence. In Oommen and Rueda [2005], the authors proposed a formal, rigorous theoretical model that provided a *stochastic* answer to this problem. They proved that given a heuristic algorithm A that could utilize either of two heuristic functions $\mathcal{H}1$ or $\mathcal{H}2$ used to find the solution to a particular problem, if the accuracy of evaluating the cost of the *optimal* solution by using $\mathcal{H}1$ is greater than that of evaluating the cost using $\mathcal{H}2$, then $\mathcal{H}1$ has a higher probability than $\mathcal{H}2$ of leading to the optimal solution. Informally speaking, this means that whenever we seek to find a heuristic solution for a problem, it is always advantageous to utilize heuristic functions that use "clues", which in turn lead to good solutions with *high probability*. In this vein, the heuristic criterion we propose for each of these problems involves approximately optimal solutions of lower-dimensional subspaces, which are then fused using well-established laws of fusion. We emphasize, though, that since these individual solutions are not necessarily optimal, their fused solution need not necessarily be the optimal. However, for the case when the dimension n is small, the search could result in a solution which is very close to optimal. The question, then, is one of getting a superior solution in a high-dimensional space, given a set of reasonably accurate solutions obtained for lower-dimension subspaces, that is, for smaller values of $m < n$. We then advocate the concept that if these solutions are fused, the accuracy of the heuristic solution in the higher-dimensional space increases, implying (as a consequence of the results due to Oommen and Rueda [2005]) that the fused result could lead to the optimal solution with a higher probability. Merging the fused result from the lower-dimensional result is done, in our case, by dynamic programming.

The only question remaining is that of knowing which specific heuristics are to be used in each of the problems MaxGoal, MaxConf, and MinCost.

$\mathcal{H}1$. In the case of MaxGoal, the heuristic is the fused probability of n streams, which we quantify as the result obtained from the fusion of $n-1$ streams and the n^{th} stream (and the corresponding method of computation utilizing dynamic programming), as explained in Section 4.2.

$\mathcal{H}2$. In the case of MaxConf, the heuristic is the fused confidence of n streams, again quantified as the result obtained from the fusion of confidences of $n-1$ streams and that of the n^{th} stream. Again, the corresponding dynamic programming determines how the latter is computed.

$\mathcal{H}3$. In the case of MinCost, the heuristic for n streams is determined as follows. If we select the n^{th}, the best cost would be c_n plus the cost of the approximated optimal solution for using the remaining $n-1$ streams so that the overall probability of achieving the goal is at least P_{spec}. However, if we don't select this, then the best cost would possibly be that of using the remaining $n-1$ streams.

In each case, we utilize the quality of the solution for the low-dimensional subproblems as the quantifying heuristic function.

The solution we propose is a heuristic-based method that operates in two stages. The first stage looks for these particular lower-dimensional "optimal" solutions for small values of n. It then determines whether these lead to subsets that (possibly) *have* to be included in the overall solution. As per the results of Oommen and Rueda [2005], including more accurately estimated lower-dimensional solutions in the higher-dimensional subset will (stochastically) tend to lead to a superior solution. Indeed, it experimentally turns out that the contribution of these values with respect to the overall objective function (for MaxGoal, MaxConf, and MinCost) is of fundamental and primal importance.

4. PROPOSED FRAMEWORK

4.1 Overview

Given a set of n media streams and the system goal (i.e., to test a hypothesis H) in hand, the solution which approximates the optimal subset of media streams to test a hypothesis H is obtained as follows:

(1) For $1 \leq i \leq n$, we first estimate the probability $p_i = P(H|M_i)$ that hypothesis H is true. For example, for the hypothesis "a person is running in the corridor," a standard Bayes classifier can be first trained and then used to obtain these probabilities along a timeline.

(2) We then experimentally learn the confidence level f_i of each stream i, $1 \leq i \leq n$ by letting the system use only the stream M_i. The confidence level is assigned to a stream based on how it has helped in accurately testing a hypothesis.

(3) Using a voting strategy, we divide the n streams into two subsets S_1 and S_2 based on the fact whether, at the current instant, they agree or disagree in support of the true hypothesis. Precisely, those streams that support the hypothesis with more than 0.50 probability are put in set S_1 and the rest in set S_2.

(4) For the two subsets S_1 and S_2, we compute fusion probabilities $P_{S_1} = P(H|S_1)$ and $P_{S_2} = P(\bar{H}|S_2)$ of achieving the goal using a Bayesian approach [Atrey et al. 2005], and also find the overall confidence F_{S_1} and F_{S_2} for the subsets S_1 and S_2, respectively (as described in Section 4.2).

(5) We assign the weights to two subsets based on their respective overall confidence values and conclude that the hypothesis H is true if $P_{S_1}.F_{S_1} \geq P_{S_2}.F_{S_2}$. The system then finds the optimal subsets Φ_1 and Φ_2 from the sets S_1 and S_2, respectively, and continues to use them until the probability of the hypothesis being true, based on both optimal subsets, becomes more than a user-specified threshold (i.e., P_{spec}).

4.2 Preliminaries

4.2.1 *Fusion of Correlated Probabilities.* We combine the probabilities of achieving the system goal based on two sources $\mathbf{M}^{i-1}$ and M_i using a Bayesian approach, where $\mathbf{M}^{i-1}$ is group of $i-1$ streams (i.e., $\mathbf{M}^{i-1} = \{M_1, M_2, \ldots, M_{i-1}\}$) and M_i is an individual i^{th} stream to be fused with $\mathbf{M}^{i-1}$. The fusion model (described in our previous work [Atrey et al. 2005]) is given as follows:

$$P_i = \frac{P_{i-1}.p_i.e^{\overline{\gamma}_i}}{P_{i-1}.p_i.e^{\overline{\gamma}_i} + (1 - P_{i-1})(1 - p_i).e^{-\overline{\gamma}_i}}, \tag{1}$$

where $P_i = P(H|\mathbf{M}^i)$ and $P_{i-1} = P(H|\mathbf{M}^{i-1})$ are the probabilities of the hypothesis being declared true by the system **S** based on fusion of a group of i and $i-1$ streams, respectively. The quantity p_i is the probability of the hypothesis being true based on the ith stream individually. Note that one possible way to compute $p_i = P(H|M_i)$ for $1 \leq i \leq n$ is by using a Bayes classifier. However, we could also use an alternative method. The $\overline{\gamma}_i$ is an agreement coefficient between two sources $\mathbf{M}^{i-1}$ and M_i. We describe this in more detail in Section 4.2.2.

4.2.2 *Modeling the Agreement Coefficient.* The correlation among streams refers to the measure of their agreement or disagreement with each other [Atrey et al. 2005]. We call this measure of agreement the *Agreement coefficient* among the streams. The agreement coefficient $\gamma_{ij}(t)$ between the streams M_i and M_j at time instant t is computed by iteratively averaging past agreement coefficients with the current observation. The $\gamma_{ij}(t)$ is precisely computed as

$$\gamma_{ij}(t) = \frac{1}{2}[(1 - 2 \times abs(p_i(t) - p_j(t)|)) + \gamma_{ij}(t-1)], \tag{2}$$

where $p_i(t) = P(H|M_i)$ and $p_j(t) = P(H|M_j)$ are the individual probabilities of the hypothesis H being true based on media streams M_i and M_j, respectively, at time $t > 1$. These probabilities represent decisions about the hypothesis. Exactly the same decisions would imply full agreement ($\gamma_{ij} = 1$), whereas totally dissimilar ones would mean that the two streams fully contradict each other ($\gamma_{ij} = -1$).

The agreement coefficient between two sources $\mathbf{M}^{i-1}$ and M_i is fused as

$$\overline{\gamma}_i = \frac{1}{i-1}\sum_{k=1}^{i-1} \gamma_{ki}, \tag{3}$$

where γ_{ki} for $1 \leq k \leq i-1$, $1 \leq i \leq n$ are the agreement coefficients between the k^{th} and i^{th} streams. The fused agreement coefficient $\overline{\gamma}_i$ is used for combining M_i with $\mathbf{M}^{i-1}$, as described in Section 4.2.1.

4.2.3 *Confidence Fusion.* In the context of media streams, we relate the confidence in a stream to its accuracy. The higher the accuracy of a stream, the higher the confidence we would have in it. We compute the accuracy of a stream by determining the number of times this stream correctly confirms the hypothesis out of the total number of observations. Note that in our case, the accuracy of a media stream includes measurement accuracy of the media device, as well as the accuracy of the algorithm used for processing the media data.

Confidence fusion refers to the process of finding the overall confidence in a group of media streams, where individual media streams have their own confidence level. Considering the confidence values as probabilities, we propose a Bayesian method to fuse the confidence levels in individual streams, *and this constitutes one of the "heuristic functions" used by our strategy* on which the results of Oommen and Rueda [2005] rest. For n number of media streams, the overall confidence is iteratively computed. Let F_{i-1} be the overall confidence in a group of $i-1$ streams. By fusing the confidence f_i of i^{th} stream with F_{i-1}, the overall confidence F_i in a group of i streams is computed as

$$F_i = \frac{F_{i-1} \times f_i}{F_{i-1} \times f_i + (1 - F_{i-1}) \times (1 - f_i)}. \tag{4}$$

In the preceding formulation, although the media streams are correlated in content, we assume that they are mutually independent in terms of their confidence levels.

4.3 Solution for MaxGoal

We first find all the subsets Φ_i, $1 \leq i \leq n'$ of streams whose cost $C_{\Phi_i} \leq C_{spec}$, for $1 \leq i \leq n'$. Then, we pick a subset Φ from the subsets Φ_i, $1 \leq i \leq n'$ for which the confidence F_Φ is maximum.

The dynamic programming approach for approximating the optimal subset Φ works as follows. We begin by considering the selection of the n^{th} stream. If we select the n^{th} stream, then the fused probability would be the result obtained from fusion of the n^{th} stream with the remaining $n-1$ streams (with a specified cost $C_{spec} - c_n$, where $c_n < C_{spec}$). However, if we do not select it, the fused probability would possibly be the result obtained from fusion of the remaining $n-1$ streams (with a specified cost C_{spec}). The fused probability (of achieving the goal) will be the maximum of these two possible best options, which is also an integral part of the heuristic function that the solution for MaxGoal utilizes.

We thus describe the structure of our solution, which converges to the optimal one by the following recurrence relation:

$$Prob(i, m) = \begin{cases} Prob(i-1, m) & , c_i > m \\ max[Prob(i-1, m), \mathbf{PFusion}(Prob(i-1, m-c_i), p_i, \Gamma) & , c_i \leq m \end{cases},$$

where $Prob(i, m)$, $1 \leq i \leq n$, $1 \leq m \leq C_{spec}$, approximates the optimal fused probability (of achieving the goal) based on streams 1 to i with the cost m. The initial conditions for the recursive relation are:

$$Prob(1, m) = \begin{cases} 0 & , c_1 > m \\ p_1 & , c_1 \leq m \end{cases}.$$

The **PFusion** function combines the probabilities of the system **S** achieving the goal based on two sources $\mathbf{M}^{i-1}$ and M_i using the fusion model given in Eq. (1). Here, Γ is the set of agreement coefficients among media streams.

We approximate the optimal fused probability by recursively computing $Prob(n, m)$. As soon as the *Prob* table is constructed, the proposed solution which approximates the optimal subset Φ is computed by backtracking through the table.

Algorithm MaxGoal. The algorithm **MaxGoal** outlines the idea described earlier

```
MaxGoal($n$, $p$, $\Gamma$, $c$, $f$, $C_{spec}$, $F_{spec}$)
Inputs
$n$ : Number of input media streams.
$p[1 \ldots n]$ : Probability of each stream achieving the goal.
$f[1 \ldots n]$ : Confidence in each media stream.
$c[1 \ldots n]$ : Cost of using each media stream.
$\Gamma$: Set of agreement coefficients among media streams.
$C_{spec}$ : Specified maximum cost.
$F_{spec}$ : Specified minimum confidence.
Steps
   1. Initialize $Prob$, $Conf$ and $Select$ array to zero.
   2. for $i$ = 1 to $n$, $m$ = 0 to $C_{spec}$
   3.      if ($c[i] \leq m$)
   4.           Compute fused probability $P_i$ using equation (1)
   5.           Compute overall confidence $F_i$ using equation (4)
   6.           if ($P_i > Prob[i-1, m]$)      $Prob[i, m] = P_i$, $Conf = F_i$, $Select[i, m]$ = 1
   7.           else      $Prob[i, m] = Prob[i-1, m]$, $Conf[i, m] = Conf\,[i-1, m]$, $Select[i, m] = 0$
   8.      else      $Prob[i, m] = Prob[i-1, m]$, $Conf\,[i, m] = Conf\,[i-1, m]$, $Select[i, m] = 0$
   9. $k = m - 1$, $P_\Phi = Prob[n, k]$, $C_\Phi = 0$
  10. for $i = n$ to 1 in steps -1
  11.      if ($Select[k] = 1$)
  12.           Output the stream $i$ into $\Phi$
  13.           $C_\Phi = C_\Phi + c[i]$, $k = k - c[i]$
  14. $F_\Phi$ = maximum confidence at $C_\Phi$
Outputs
$P_\Phi$: An approximation to the optimal probability to achieve the goal.
$\Phi$: The set of media streams used to obtain $P_\Phi$.
$C_\Phi$: The cost of using $\Phi$ to obtain $P_\Phi$.
$F_\Phi$: The confidence in subset $\Phi$.
```

4.4 Solution for MaxConf

Similar to problem MaxGoal in Section 4.3, for problem MaxConf, we first find all the subsets Φ_i, $1 \leq i \leq n'$ of streams whose cost $C_{\Phi_i} \leq C_{spec}$, for $1 \leq i \leq n'$. Then, we pick a subset Φ from those subsets Φ_i, $1 \leq i \leq n'$ for which the overall probability P_Φ of achieving the goal is maximum.

The dynamic programming solution for MaxConf works as follows. We approximate the optimal solution by the following recurrence relation, given as

$$Conf(i, m) = \begin{cases} Conf(i-1, m) & , c_i > m \\ max[Conf(i-1, m), \mathbf{CFusion}(Conf(i-1, m-c_i), f_i) & , c_i \leq m \end{cases},$$

where $Conf(i, m)$, $1 \leq i \leq n$, $1 \leq m \leq C_{spec}$, approximates the optimal overall confidence in the streams 1 to i with the cost m, and is the "local" heuristic function that MaxConf resorts to. The initial conditions for the recursive relation are

$$Conf(1, m) = \begin{cases} 0 & , c_1 > m \\ f_1 & , c_1 \leq m. \end{cases}$$

CFusion combines the confidence levels in two sources $\mathbf{M}^{i-1}$ and M_i using the fusion model given in Eq. (4). We approximate optimal overall confidence by recursively computing $Conf(n, m)$. Once the $Conf$ table is constructed, the reported solution (which is the approximation to the optimal subset Φ) is found by backtracking through it. The algorithm **MaxConf** can be outlined similarly to **MaxGoal**. Details are omitted in the interest of brevity.

4.5 Solution for MinCost

The problem MinCost differs from MaxGoal and MaxConf in that the optimization functions in MaxGoal and MaxConf are to *maximize* probability and confidence, respectively, while in MinCost, we *minimize* the cost.

We first find all the subsets Φ_i, $1 \leq i \leq n'$ of streams whose fused probabilities $P_{\Phi_i} \geq P_{spec}$, for $1 \leq i \leq n'$. Then, we pick a subset Φ from the subsets Φ_i, $1 \leq i \leq n'$ for which the confidence F_Φ is maximum.

To solve MinCost using a dynamic programming approach, we begin by considering the nth stream. If we select it, the best cost would be c_n plus the cost of the approximated optimal solution of using the remaining $n-1$ streams so that the overall probability of achieving the goal is at least P_{spec}. However, if we don't select it, then the best cost would possibly be that of using the remaining $n-1$ streams. The optimal cost of achieving the goal will be the minimum of these two "best" options, and this will be the heuristic function that MinCost depends on so as to invoke the results of Oommen and Rueda [2005].

Let *Cost*(i, m) denote the cost of using media stream $1 \ldots i$ for achieving the goal with probability m. Assuming this probability takes one of the L discrete values, we characterize the recursive relation for *Cost*(i, m) as follows:

$$Cost(i,m) = \left\{ \begin{array}{ll} min(Cost(i-1,m), c_i) & , m \leq min(p_i, P_{spec}) \\ \textbf{while}(l[ss] \neq 0)\{ & \\ \quad min(Cost(i,m), fcost) & , p_i < m \leq R \textbf{ and } Cost(i,m) \neq \infty \\ \quad min(Cost(i-1,m), fcost) & , p_i < m \leq R \textbf{ and } Cost(i,m) = \infty \\ \} & \\ Cost(i-1,m) & , m > R' \end{array} \right. ,$$

where $1 \leq i \leq n$, $1 \leq m \leq L$. The initial conditions are

$$Cost(1,m) = \left\{ \begin{array}{ll} c_1 & , m \leq min(p_1, P_{spec}) \\ \infty & , m > p_1 \end{array} \right. .$$

In the recursive formulation previously described, *fcost*, R, and R' are computed as

$$fcost = \left\{ \begin{array}{ll} Cost(i-1, l[ss]) & , ss > 0 \textbf{ and } l[ss] \neq p_i \\ c_i & , ss > 0 \textbf{ and } l[ss] = p_i \\ 0 & , ss = 0 \end{array} \right.$$

$$R = \left\{ \begin{array}{ll} \textbf{PFusion}(l[ss], p_i) & , ss > 0 \textbf{ and } l[ss] \neq p_i \\ p_i & , ss > 0 \textbf{ and } l[ss] = p_i \\ 0 & , ss = 0 \end{array} \right. .$$

$$R' = \left\{ \begin{array}{ll} max(R', R) & , ss > 0 \\ 0 & , ss = 0 \end{array} \right.$$

The $l[ss]$ is an array that contains the probabilities based on the individual streams, as well as the fusion probabilities. After constructing the *Cost* table, the *Select* array is traced back to find the solution which approximates the optimal subset Φ.

4.6 Complexity Analysis

Any brute-force approach to solve each of the three problems MaxGoal, MaxConf, and MinCost requires $O(2^n)$ time, since all 2^n combinations of streams need to be checked so as to find the optimal subset. We

Algorithm Min Cost. The algorithm **MinCost** is given as follows.

```
MinCost($n, p, c, f, \Gamma, L, P_{spec}, F_{spec},$ )
Input
    $n, p, c, f, \Gamma$ and $F_{spec}$: Similar to MaxGoal
    $L$: Number of discrete levels of probability values
    $P_{spec} \leq L$: Specified minimum fused probability of achieving the goal
Steps
    1. Initialize $Cost$ to $\infty$, $L$ to 100, and $Prob$, $Conf$ and $Select$ array to zero.
    2. for $i$ = 0 to $n$
    3.     for $m$ = 0 to $Min(p_i, L)$
    4.         $Cost[i, m]$ = $Min(Cost[i-1, m], c_i)$
    5.         if $(Cost[i, m] = Cost[i-1, m])$
    6.             $Conf[i, m] = Conf[i-1, m]$, $Prob[i, m] = Prob[i-1, m]$, $Select[i, m]$ = 0
    7.         else    $Conf[i, m] = f_i$, $Prob[i, m] = p_i$, $Select[i, m]$ = 1
    8.     Initialize variables- $R = R' = 0$, $ss = 0$, $fcost = 0$, $fconf = 0$, $fprob = 0$
    9.     $ss$ = Number of unique values in $Cost$ array, copy them into $l$ array
   10.     while $(l[ss] \neq 0)$
   11.         if $(l[ss] \neq p_i)$
   12.             $fprob$ =PFusion($l[ss], p_i, \Gamma$), $fconf$ =CFusion($l[ss], f_i$), $fcost = Cost[i-1, l[ss]] + c_i$
   13.         else $fprob = p_i$, $fconf = f_i$, $fcost = c_i$
   14.         $R = fprob$
   15.         for $m = m'$ to $R$
   16.             if $(Cost[i, m] \neq \infty)$    $Cost[i, m] = min(Cost[i, m], fcost)$
   17.                 if $(Cost[i, m] = fcost)$    $Conf[i, m] = f_i$, $Prob[i, m] = p_i$
   18.             else    $Cost[i, m] = min(Cost[i-1, m], fcost)$
   19.                 if $(Cost[i, m] = fcost)$    $Conf[i, m] = f_i$, $Prob[i, m] = p_i$
   20.                 else $Conf[i, m] = Conf[i-1, m]$, $Prob[i, m] = Prob[i-1, m]$
   21.             if $(Cost[i, m] \neq Cost[i-1, m]$ and $Cost[i, m] \neq \infty)$    $Select[i, m] = 1$
   22.             else $Select[i, m] = 0$
   23.         $m' = R + 1$, $R' = max(R', R)$, $ss = ss + 1$
   24.     for $m = R' + 1$ to $L$
   25.         $Cost[i, m] = Cost[i-1, m]$, $Conf[i, m] = Conf[i-1, m]$, $Prob[i, m] = Prob[i-1, m]$
   26.         $Select[i, m] = 0$
   27. $OptProb = P_{spec}$
   28. if $(OptProb < L)$
   29.     while $(Cost[i, OptProb + 1] = Cost[i, OptProb])$    $OptProb = OptProb - 1$
   30. else
   31.     while $(Cost[i, OptProb] = Cost[i, OptProb - 1])$    $OptProb = OptProb - 1$
   32.     $OptProb = OptProb - 1$
   33. $P_\Phi = OptProb$, $C_\Phi = 0$, $i = i - 1$, $m = OptProb$, $C_\Phi = k = Cost[i, OptProb - 1]$
   34. while $(k > 0)$
   35.     while $(Cost[i, m] \neq k)$    $m = m - 1$
   36.     if $(Select[i, m] = 1)$    Output $i$ into $\Phi$, $k = k - c_i$
   37.     $i = i - 1$
   38. $F_\Phi$ = maximum confidence at $P_\Phi$
Outputs
    $\Phi, C_\Phi, P_\Phi, F_\Phi$: Same as MaxGoal
```

have also proven these three MS problems to be NP-Complete in Section 3.1. However, the proposed dynamic programming-based approach solves them in polynomial time under the assumptions that:

—the total cost of media streams is not exponential in terms of total number of media streams, that is, $C_n \neq O(2^n)$ (for problems MaxGoal and MaxConf); and

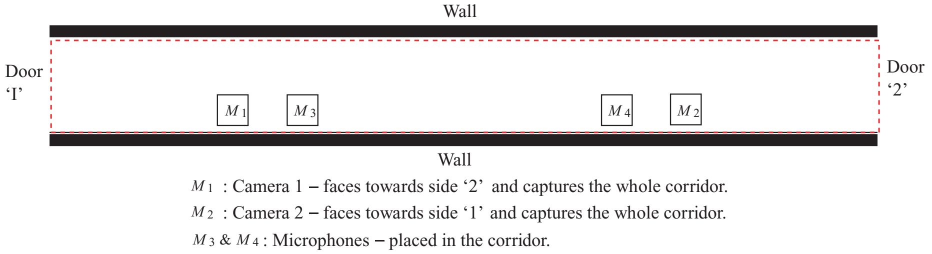

Wall
Door
'I'
M 1
M 3
M 4
M 2
Door
'2'
Wall
M 1 : Camera 1 – faces towards side '2' and captures the whole corridor.
M 2 : Camera 2 – faces towards side '1' and captures the whole corridor.
M 3 & M 4 : Microphones – placed in the corridor.

Fig. 1. The layout of the environment under surveillance and monitoring.

—the total discrete levels L of probability values are not exponential in terms of total number of media streams, that is, $L \neq O(2^n)$ (for problem MinCost).

The time complexity of both **MaxGoal** and **MaxConf** algorithms is $O(n^2 \times C_{spec})$, where $C_{spec} \leq C_n$. This is, on average, lower than that of the brute-force approach. Note that $O(n^2 \times C_{spec})$ also includes the time complexity of **PFusion**, which is $O(n)$. The space complexity of the **MaxGoal** algorithm is $O(n \times C_{spec})$.

The algorithm **MinCost** has a time complexity of $O(n^2 \times L)$ to approximate the optimal subset, which is again better than the brute-force approach. Note that the higher the discrete levels L of probability value, the higher the time complexity. In the algorithm **MinCost**, we have used $L = 100$. The space complexity is $O(n \times P_{spec})$, where $P_{spec} \leq L$.

5. RESULTS

To demonstrate the utility of our proposed framework, we present experimental results in a surveillance and monitoring scenario. The surveillance environment is the corridor of our school building and the system goal is to detect events such as humans running, walking, standing, talking, shouting, and door-knocking in the corridor. The environment layout is shown in Figure 1. We use two video sensors (cameras M_1 and M_2) to record the video from two opposite sides of the corridor, and two audio sensors (microphones M_3 and M_4) to capture ambient sound.

To describe the events, we have introduced the notions of *compound event* and *atomic event* [Atrey et al. 2005]. This description of events is inspired by Nevatia et al. [2003], though the authors have used it in a different context and have not considered the optimal selection of streams. An atomic-event is an event in which exactly one object having one or more attributes is involved in exactly one activity at a location over a period of time, whereas a compound event is the union of two or more atomic events. For example, a compound event "a person is walking and shouting in the corridor" is composed of two atomic events "a person is walking in the corridor" and "a person is shouting in the corridor."

5.1 Preliminary Steps

5.1.1 *Video Processing.* The video is processed to detect human motion (running, walking, and standing). Video processing involves two major steps: background modeling and blob detection. The background is modeled using an adaptive Gaussian method [Stauffer and Grimson 1999]. Blob detection is performed by first segmenting the foreground from the background using simple 'matching' on the three RGB color channels, and then using morphological operations (i.e., erode and dilation) to obtain connected components (i.e., blobs). The matching is defined as a pixel value being within 2.5 standard deviations of the distribution. We assume that the blob of an area greater than a threshold corresponds to a human. We extract for each detected blob its bounding rectangle and area, as shown in Figure 2. Based on the presence of potential blobs in a sequence of video frames, we estimate human motion using these two features. We map the blob's bottom point (i.e., approximating corresponding

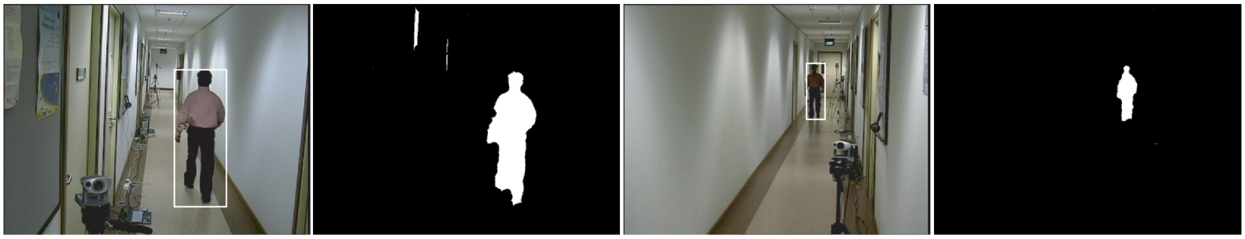


Fig. 2. Blob detection in camera 1 and camera 2.

Table I. The Feature Used for Video and Audio Streams

(a) Video

Classification Task	Stream 1	Stream 2
Foreground/Background	RGB channels	RGB channels
Running/Walking/Standing	Blob's displacement	Rate of change in Blob's area

(b) Audio

Classification task	Stream 1	Stream 2
Foreground/Background	Zero Crossing Rate	Root Mean Square
Excited/Normal	Zero Crossing Rate	Root Mean Square
Vocal/Nonvocal	Zero Crossing Rate	Linear Predictor Coefficients

to the human feet) in the image to a point in the 3D world (i.e., on the corridor's floor). To achieve this mapping, we calibrate the cameras and obtain a transformation matrix that maps image points to points on the corridor's floor. This provides the exact ground location of the human in the corridor at a particular time instant. Another way of estimating the human motion which we used is to observe the rate of change of a blob's area. We exploit the fact that the blob area increases at a certain rate as the person moves towards the camera, and vice versa. A summary of the video features used for various classification tasks is provided in Table I(a).

The system identifies the start and end of an atomic event in video as follows. If a person moves towards the camera, the start of an atomic event is marked when the blob's area becomes greater than a threshold and the atomic event ends when the blob intersects the image plane. However, if the person walks away from the camera, the start and end of the atomic event are inverted. Once an atomic event is detected, we divide the time duration for which the event occurred into time-windows of t_w. Here, t_w is the *minimum time period* in which an atomic event can be detected. We do this timeline division to determine key points for the purpose of assimilation. In our experiment, we set $t_w = 1$ second. Using the actual location of the person on the corridor's floor at the end of each time-window t_w, we compute the *average distance* travelled by a person on the ground. Based on this average distance, a Bayes classifier is first trained and then used to classify an atomic event as one of the classes standing, walking, and running. Similarly, these events are detected based on the average rate of change in the blob's area over a period t_w.

5.1.2 *Audio Processing.* The system detects events such footsteps, talking, shouting, and door-knocking based on audio streams. Based on the number of footsteps in a fixed time-window, an event is classified as either 'running' or 'walking.' The audio (of 44.1 MHz frequency) is divided into "audio frames" of 50ms each. The frame size is chosen by experimentally observing that 50ms is the minimum period during which an event such a footstep can be represented. Similar to the video, we model the audio background using an adaptive Gaussian method [Stauffer and Grimson 1999] and segment the foreground for each audio frame using a matching within 2.5 standard deviations of the distribution. Once the foreground audio events are detected, the system classifies them into "excited" and "normal"

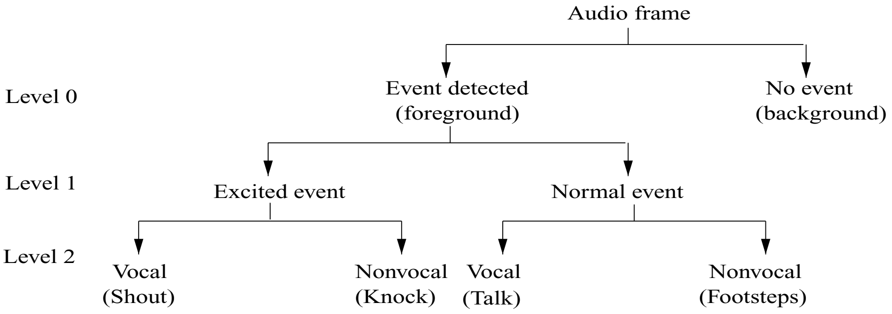


Fig. 3. Audio event classification.

categories. These excited and normal events are further classified into "vocal" and "nonvocal" events, as shown in Figure 3. Table I(b) summarizes audio features used for foreground/background segmentation and for classification.

As shown in Table I(b), in stream 1 option, we used a zero crossing rate (ZCR) feature for all three classification levels, while in the stream 2 option, we used a root mean square (RMS) for foreground/background segmentation and distinguishing between excited and normal events. Linear predictor coefficients (LPC) are used for categorizing between vocal and nonvocal events. The zero crossing rate measures the number of times in the given time interval (50ms, in our case) that the signal amplitude passes through a value of zero, moving from negative to positive and vice versa. The root mean square is a two-norm of the vector containing the samples in one audio frame (of 50ms). Note that these two features are sensitive to excited events and have been found to have a higher value for excited events as compared to that for normal ones. Linear predictor coefficients have been widely used in the speech processing community. LPCs are filter coefficients described in all pole models which approximate the characteristics of a speech production system. Therefore, LPCs are sensitive to vocal sounds. This motivated us to use LPCs for the detection of vocal and nonvocal events. We used an LPC algorithm from the MATLAB toolbox. A Bayesian classifier is first trained and then employed to classify the atomic audio events at three different levels (level 0 to level 2, as shown in Figure 3) [Atrey et al. 2006].

To assimilate the information obtained from all eight streams (two streams each of two audio and two video sensors), the probabilistic decisions about the audio events are obtained after every t_w time intervals (two seconds, in our experiments). Note that in two seconds, we have 20 audio frames of 50ms each. The audio event classification for the audio data of t_w time period is performed as follows. First, the system learns via training the number of audio frames corresponding to an event in the audio data of t_w time period. Then, a Bayesian classifier is employed to estimate the probability of occurrence of an audio event at a regular time interval t_w.

To demonstrate how our framework works, we consider a compound event $\mathbf{E}_c$, that is, "a person walked in the corridor from side A, stood near the door and knocked it, and then walked to side B of the corridor." In order to detect the compound event $\mathbf{E}_c$, we decompose it into its constituent atomic events $\mathbf{e}_1$ = "a person walked/stood in the corridor" and $\mathbf{e}_2$ = "a person knocked the door in the corridor." The probabilistic decisions for these two atomic events obtained using four video and four audio streams are aligned along a timeline, as shown in Figure 4. In the figure, the x-axis denotes key points along the timeline and, y-axis shows the probability of occurrence of an atomic event based on a particular stream. The legends used are: '$\circ$'—standing, '$\Box$'—walking, '∇'—knocking, and '$\star$'—no event. For example, the legend '$\circ$' shown at key point '8' for the stream $\mathbf{V}_{11}$ indicates the probability of occurrence of an event "person is standing" based on the stream 1 (refer to Table I(a)) of video camera 1. We will shortly describe in Section 5.2 how the optimal subset is selected from the set of these eight streams.

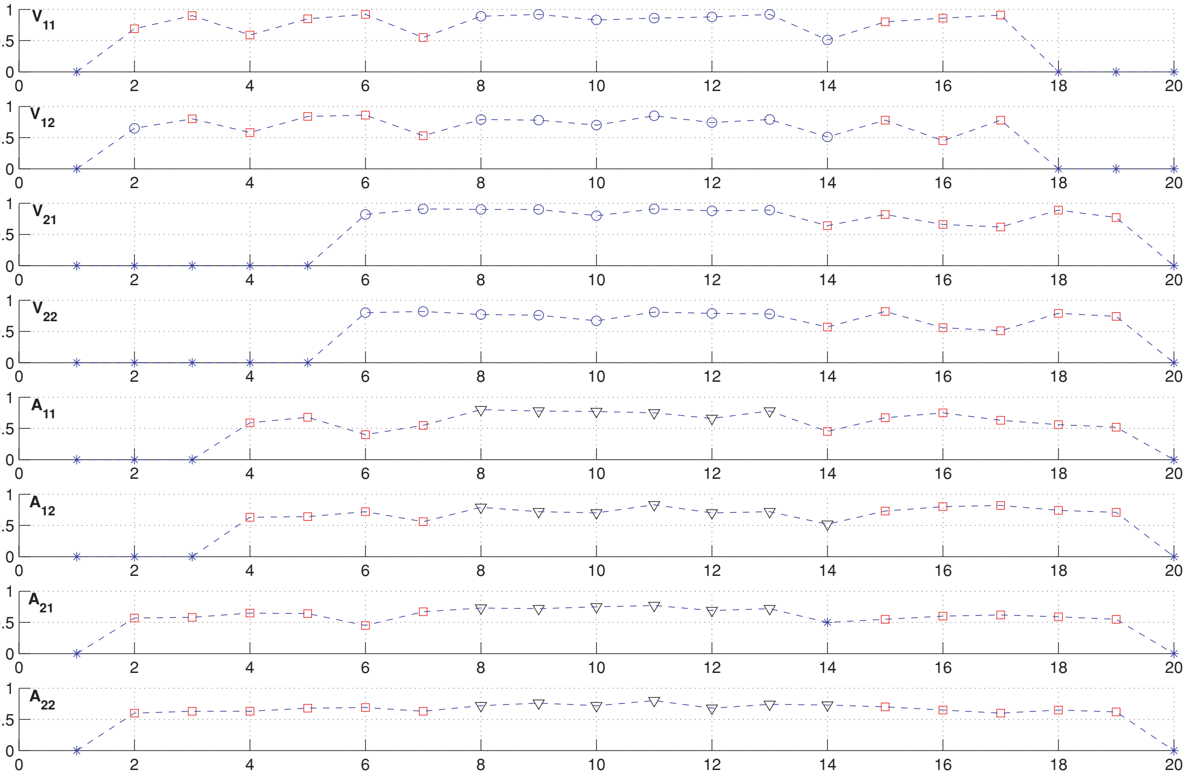


Fig. 4. Timeline-based probabilistic decisions for events using all eight streams.

5.1.3 *Cost Estimation.* As described in the Introduction, the cost of streams is usually comprised of two types: the one-time cost and the running cost. Note that the one-time cost (such as installation cost, the cost of training classifiers, etc.) is optimized by the system designer during system design. Our focus in this article is on the on-the-fly optimization of running cost by the system. The running cost consists of the costs of processing and operating, as well as the wear and tear of the media stream. Note that the operating and wear and tear cost can be computed based on the statistics of power consumption and the diminishing cost of video sensors. For our experiments, we consider only the processing cost of streams and describe how this can be estimated for various video and audio streams.

Processing a stream usually consists of two steps: feature extraction and event classification. We compute the processing cost by estimating the time taken in feature extraction and in event classification steps for all the streams. Table II(a) shows the same for a video stream. For an audio stream, Table II(b) shows the cost of extracting different features (ZCR, RMS, and LPC) and that cost of event classification at three different levels. Based on the data shown in Tables II(a) and II(b), we provide the total estimated cost for all eight streams in Table II(c). Note that when two video streams obtained from the same camera (e.g., V_{11},V_{12} from camera 1 or V_{21},V_{22} from camera 2) are selected together in the optimal subset, the cost of only one stream is counted, since the major cost of blob detection remains common to both.

5.1.4 *Computing Confidences in Streams.* We computed the confidences in all four video streams used by running the experiments for 30 events of walking, standing, and door knocking. By comparing

Table II. Processing Cost of Video and Audio Streams

(a) Video stream

Blob detection (BD)	0.66 frames (each of size 756 × 568) per second
Event classification (EC)	0.010 seconds

(Assuming that there are 8 frames per second in video, it takes $8/0.66 \approx 12.12$ seconds for processing of 1 second of video)

(b) Audio stream

Feature extraction	ZCR	RMS	LPC
Cost	1.5642 seconds	0.8628 seconds	1.5072 seconds
Event classification	Foreground/Background (F/B)	Excited/Normal (E/N)	Vocal/Nonvocal (V/NV)
Cost	0.0082 seconds	0.0076 seconds	0.0100 seconds

(These processing costs are for 1 second of audio)

(c) The total estimated cost for all the streams

Stream	Cost breakup	Estimated total cost (in Unit money)
$\mathbf{V}_{11}, \mathbf{V}_{12}, \mathbf{V}_{21}, \mathbf{V}_{22}$	(12.12 (BD) + 0.010 (EC)) × 10	≈ 12.0
$\mathbf{A}_{11}$,$\mathbf{A}_{21}$	(1.5642 (ZCR) + 0.0082 (F/B) + 0.0076 (E/N) + 0.0100 (V/NV))× 10	≈ 1.5
$\mathbf{A}_{12}$, $\mathbf{A}_{22}$	(0.8628 (RMS) + 1.5072 (LPC) + 0.0082 (F/B) + 0.0076 (E/N) + 0.0100 (V/NV)) × 10	≈ 2.5

(These costs are for processing of streams of 1 second. In calculating the final cost, we assume that the processing of every second of data costs 1 unit money)

Table III. Confidences in All the Streams

Stream	$\mathbf{V}_{11}$	$\mathbf{V}_{12}$	$\mathbf{V}_{21}$	$\mathbf{V}_{22}$	$\mathbf{A}_{11}$	$\mathbf{A}_{12}$	$\mathbf{A}_{21}$	$\mathbf{A}_{22}$
Confidence	0.62	0.55	0.60	0.54	0.55	0.58	0.55	0.58

results with the ground-truth, we noticed that the event detection was found 60% correct using the feature stream 1 (i.e., blob's displacement) of both cameras; while it was found 55% and 54% with feature stream 2 (i.e., blob's area) for camera 1 and camera 2, respectively. The audio analysis was done separately [Atrey et al. 2006] and it was found that the overall accuracy of event detection using audio sensors was 55% based on ZCR and 58% based on (RMS+LPC). Based on this experimental evidence, we assigned the confidence levels to different streams, as shown in Table III.

5.2 Optimal Subset Selection of Streams

Using the preliminary data obtained in Section 5.1, we now show how our framework selects the optimal subset of streams for detecting the event $\mathbf{E}_c$. Note that, due to the placement and coverage space of sensors, not all of the sensors may detect the event at the same time instance. Therefore, environment information is needed to determine the right set of streams out of which the optimal subset would be selected. As shown in Figure 4, the event $\mathbf{E}_c$ is detected based on the set (V_{11}, V_{12}, A_{21}, A_{22}) of streams at key point '2.' We first show in Section 5.2.1 how the optimal subset is computed at a key point. Next, in Section 5.2.2, we demonstrate how frequently the optimal subset is recomputed along the timeline and also how much of the cost is saved by using only the optimal subset.

5.2.1 *Finding an Optimal Subset at a Key Point.* The system computes the optimal subset at key point '2' as follows. First, since the probabilistic decisions based on three (V_{11}, A_{21}, A_{22}) of four streams favor the "walking" event, they are kept in group S_1 and the rest (V_{12}) are kept in group S_2 (refer to Section 4.1, step 3). Next, we assimilate the probabilistic decisions obtained from streams within each

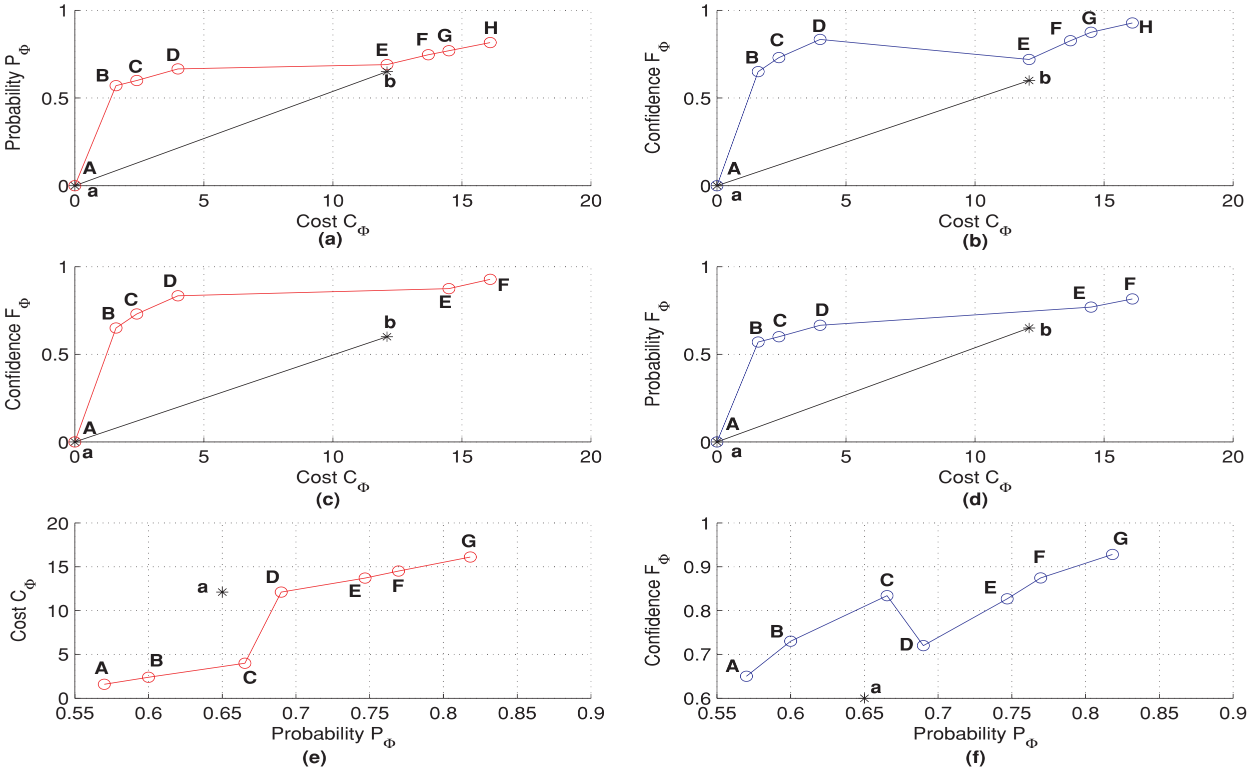


Fig. 5. (a) and (b) **MaxGoal**: **A** = (Nil), **B** = (A_{21}), **C** = (A_{22}), **D** = (A_{21}, A_{22}), **E** = (V_{11}), **F** = (V_{11}, A_{21}), **G** = (V_{11}, A_{22}), **H** = (V_{11}, A_{21}, A_{22}) represent the subsets in favor of event "walking"; (c) and (d) **MaxConf**: **A** to **D** (just the same as **MaxGoal**) **E** = (V_{11}, A_{22}), **F** = (V_{11}, A_{21}, A_{22}) represent the subsets in favor of event "walking"; (e) and (f) **MinCost**: **A** to **C** (just the same as **MaxGoal**) **D** = (V_{11}), **E** = (V_{11}, A_{21}), **F** = (V_{11}, A_{22}), **G** = (V_{11}, A_{21}, A_{22}) represent the subsets in favor of event "walking"; and the symbols **a** = (Nil), **b** = (A_{12}) represent the subsets in favor of event "standing" for all three MS problems.

of the two sets and obtain the fused probabilities $P(\mathbf{E}_c|S_1)$ and $P(\bar{\mathbf{E}}_c|S_2)$ using Eq. (1) by assuming a uniform agreement coefficient $\gamma = 0$ among the streams. Note that we have described in our previous works Atrey and Kankanhalli [2004, 2005] how the agreement or disagreement among the streams affects fused probabilities. We also find the overall confidence F_{S_1} and F_{S_2} of the two sets S_1 and S_2, respectively, using Eq. (4). We obtain $P(\mathbf{E}_c|S_1) = 0.82$, $P(\bar{\mathbf{E}}_c|S_2) = 0.65$, $F_{S_1} = 0.93$, and $F_{S_2} = 0.60$. Since $P(\mathbf{E}_c|S_1).F_{S_1} = 0.7544) > (P(\bar{\mathbf{E}}_c|S_2).F_{S_2} = 0.3900)$, we conclude that there is more evidence in support of the "walking" event as compared to those evidence in favor of the "standing" event.

The optimal subset is then found from set S_1 using the dynamic programming-based framework described in Section 4.3 (**MaxGoal**: for maximizing probability), Section 4.4 (**MaxConf**: for maximizing confidence), and Section 4.5 (**MinCost**: for minimizing cost). The optimal subset process at key point '2' is depicted in Figure 5. Figure 5(a) plots how probability is maximized under the given cost constraints, and Figure 5(b) depicts how confidence varies with respect to cost, as a result of maximizing the probability, using the subsets denoted by symbols **A**, **B**, etc. Similar explanations hold true for Figures 5(c)–5(f).

The overall observations from Figures 5(a)–5(f) are:

(1) The proposed framework allows for tradeoff among the extent to which the goal is achieved, the confidence with which it is achieved, and the cost of achieving it. This offers the flexibility to compare

whether any one set of streams of low cost would be better than any other set of streams of higher cost, or any one set of media streams of high confidence would be better than any other set of streams of low confidence. For instance, Figure 5(a) clearly shows that the subset indicated by symbol **D** would be a better choice than the subset indicated by symbol **E**, since there is very small difference in the goal achieved (and overall confidence) using the two subsets (**D** helps in detecting the event with 0.03 less probability than **E** and with overall confidence of more than that in **E**), while there is a significant difference (of ≈ 8) in cost.

(2) The framework also allows for a tradeoff, depending on whether we should opt for maximizing probability, maximizing confidence, or minimizing cost. The plots in Figure 5 suggest how the second factor (say, the probability of occurrence of an event) varies with the third (say, cost) if we opt for maximizing the first factor (say, confidence). The same also holds true for other combinations.

(3) The graphs (in Figure 5) show a pictorial representation of which subset of streams is most suitable in terms of optimal probability, optimal confidence, or optimal cost. They also help in deciding which is the next most suitable subset, in case the best subset is not available. For instance, in Figure 5(e), consider the subset denoted by **G** to be in use. If at some instant the stream A_{21} is unavailable, we can find from the plot that the next best subset is that denoted by **F**.

5.2.2 *Finding the Optimal Subset Along a Timeline.* Once the optimal subset is computed at key point '2,' the system continues using this subset along the timeline, while ignoring the other streams until the probability of occurrence of the event using this subset does not fall below a threshold (0.80, in our experiment). If the probability value falls below the threshold, the optimal subset is recomputed using *all* of the available streams. The processing cost of streams which are ignored is saved.

Timeline-based statistics of the subset used for detecting the event $\mathbf{E}_c$, the loss in probability P_Φ of occurrence of event and in confidence F_Φ in the subset used, and the savings in cost C_Φ (of processing the subset) using all three methods **MaxGoal**, **MaxConf**, and **MinCost** are provided in Tables IV, V, and VI, respectively. Note that the cost of processing the full set (i.e., all eight streams) is 32, the probability of occurrence of an event based on the full set is 0.99, and the overall confidence in the full set is 0.90.

The key observations from Tables IV to VI are as follows:

(1) The proposed framework for optimal subset selection along a timeline provides significant savings in processing cost at a marginal loss in the overall probability of achieved goals and in overall confidence in the subset used. As can be seen from Tables IV–VI, the savings in cost C of 10.2 unit ($\approx$32% for **MaxGoal**), 7.4 unit ($\approx$23% for **MaxConf**), and 16.8 unit ($\approx$50% for **MinCost**) per key point (which occur every two seconds) is achieved at the expense of less than 10% loss in probability P_Φ and confidence F_Φ.

(2) The method **MinCost**, although providing better savings in cost, fails to detect a few atomic events at some key points. For instance, the method (in an effort to minimize the cost) selects only those audio streams in the optimal subset which could detect the "knock" atomic event; but in the absence of video streams, fails to detect whether the person is standing, walking, or running.

(3) Since the processing cost of the optimal subset is significantly reduced compared to the cost of the full set of streams, it helps in achieving real-time performance in event detection.

5.2.3 *The Proposed Method versus the Brute-Force Approach.* We have compared our dynamic programming-based method for stream subset selection with the brute-force approach by recording the computation time for varying numbers of streams, as shown in Figure 6. In **MaxGoal** and

Table IV. Timeline-Based Optimal Subset Selection Using **MaxGoal**

Key point	Description	Loss in P_Φ	Loss in F_Φ	Saving in C_Φ
1	No event	-	-	-
2	**All streams used and the optimal subset is Φ computed** Walk: Φ = (V_{11}, A_{21}, A_{22}), $P_\Phi = 0.95$, $F_\Phi = 0.72$, $C_\Phi = 16$	0	0	0
3	Φ used: (V_{11}, A_{21}, A_{22}), Walk: $P_\Phi = 0.95$, $F_\Phi = 0.72$, $C_\Phi = 16$	0.04	0.18	16
4	Φ used: (V_{11}, A_{21}, A_{22}), Walk: $P_\Phi = 0.77$, $F_\Phi = 0.72$, $C_\Phi = 16$ Since $P_\Phi < P_{spec} \Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), $P_\Phi = 0.89$, $F_\Phi = 0.81$, $C_\Phi = 20$	0	0	0
5	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Walk: $P_\Phi = 0.99$, $F_\Phi = 0.81$, $C_\Phi = 20$	0	0.09	12
6	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Walk (V_{11}, A_{12}, A_{22}): $P_\Phi = 0.99$, $F_\Phi = 0.74$, $C_\Phi = 17$ Stand (A_{11}, A_{21}): $P_\Phi = 0.73$, $F_\Phi = 0.60$, $C_\Phi = 3$	0	0.16	12
7	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}) Walk: $P_\Phi = 0.87$, $F_\Phi = 0.81$, $C_\Phi = 20$	0.12	0.09	12
8	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Stand (V_{11}): $P_\Phi = 0.89$, $F_\Phi = 0.60$, $C_\Phi = 12$ Knock (A_{11}, A_{12}, A_{21}, A_{22}): $P_\Phi = 0.99$, $F_\Phi = 0.74$, $C_\Phi = 8$	0.10	0.30	12
9	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Stand (V_{11}): $P_\Phi = 0.92$, $F_\Phi = 0.60$, $C_\Phi = 12$ Knock (A_{11}, A_{12}, A_{21}, A_{22}): $P_\Phi = 0.99$, $F_\Phi = 0.74$, $C_\Phi = 8$	0.07	0.30	12
10	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Stand (V_{11}): $P_\Phi = 0.83$, $F_\Phi = 0.60$, $C_\Phi = 12$ Knock (A_{11}, A_{12}, A_{21}, A_{22}): $P_\Phi = 0.98$, $F_\Phi = 0.74$, $C_\Phi = 8$	0.16	0.30	12
11	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Stand (V_{11}): $P_\Phi = 0.86$, $F_\Phi = 0.60$, $C_\Phi = 12$ Knock (A_{11}, A_{12}, A_{21}, A_{22}): $P_\Phi = 0.99$, $F_\Phi = 0.74$, $C_\Phi = 8$	0.13	0.30	12
12	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Stand (V_{11}): $P_\Phi = 0.88$, $F_\Phi = 0.60$, $C_\Phi = 12$ Knock (A_{11}, A_{12}, A_{21}, A_{22}): $P_\Phi = 0.96$, $F_\Phi = 0.74$, $C_\Phi = 8$	0.11	0.30	12
13	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Stand (V_{11}): $P_\Phi = 0.92$, $F_\Phi = 0.60$, $C_\Phi = 12$ Knock (A_{11}, A_{12}, A_{21}, A_{22}): $P_\Phi = 0.99$, $F_\Phi = 0.74$, $C_\Phi = 8$	0.07	0.30	12
14	Φ used: (V_{11}, A_{11}, A_{12}, A_{21}, A_{22}), Walk (V_{11}): $P_\Phi = 0.51$, Since $P_\Phi < P_{spec} \Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (V_{21}, V_{22}, A_{11}), $P_\Phi = 0.74$, $F_\Phi = 0.68$, $C_\Phi = 13.5$ Stand (V_{11}, V_{12}): $P_\Phi = 0.52$, $F_\Phi = 0.65$, $C_\Phi = 12$ Knock (A_{12}, A_{22}): $P_\Phi = 0.75$, $F_\Phi = 0.66$, $C_\Phi = 5$	0	0	0
15	Since $P_\Phi < P_{spec}$ at point 14 $\Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (V_{21}, V_{22}, A_{11}, A_{12}), $P_\Phi = 0.99$, $F_\Phi = 0.75$, $C_\Phi = 16$	0	0	0
16	Φ used: (V_{21}, V_{22}, A_{11}, A_{12}), Walk: $P_\Phi = 0.99$, $F_\Phi = 0.75$, $C_\Phi = 16$	0	0.15	16
17	Same as key point 16	0	0.15	16
18	Φ used: (V_{21}, V_{22}, A_{11}, A_{12}), Walk: $P_\Phi = 0.93$, $F_\Phi = 0.67$, $C_\Phi = 16$, No event (V_{21})	0.06	0.23	16
19	Φ used: (V_{22}, A_{11}, A_{12}), Walk: $P_\Phi = 0.88$, $F_\Phi = 0.67$, $C_\Phi = 16$	0.11	0.23	16
20	Φ used: (V_{22}, A_{11}, A_{12}), No event, $C_\Phi = 16$	0	0	16
	Average losses and savings per key point	**0.049**	**0.154**	**10.2**

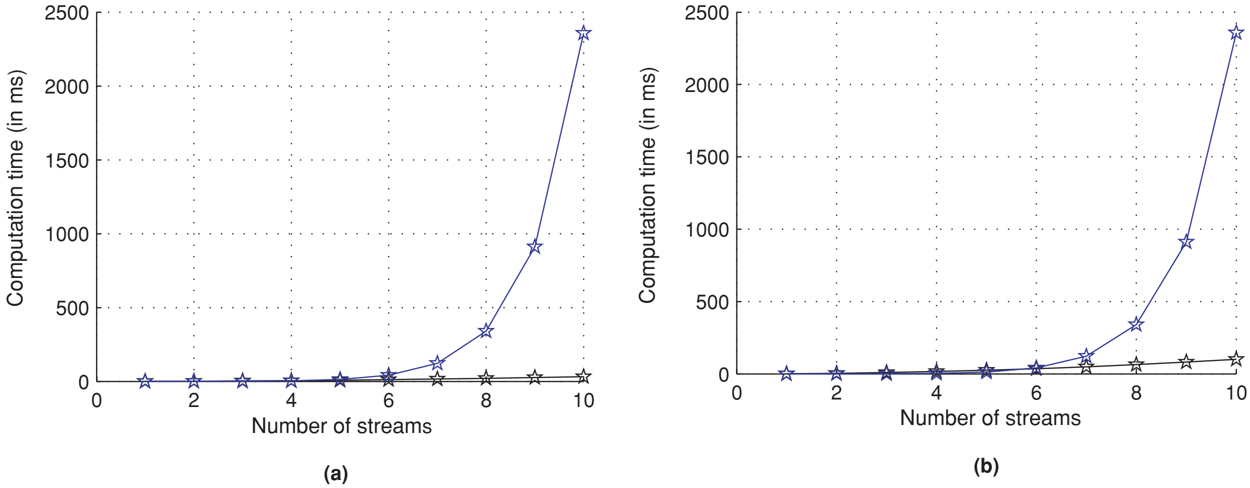
MaxConf, the total cost is taken as 32; and in **MinCost**, the total number of discrete levels L of probability values is taken as 100. The plots in Figure 6 show that the computation time taken by the dynamic programming-based method is significantly less compared to the brute-force approach as the number of streams increases.

Table V. Timeline-Based Optimal Subset Selection Using **MaxConf**

Key point	**Description**	**Loss in P_Φ**	**Loss in F_Φ**	**Saving in C_Φ**
1–14	Same as Table IV			
15	Since $P_\Phi < P_{spec}$ at point 14 $\Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (V_{11}, V_{21}, A_{11}, A_{12}), $P_\Phi = 0.99$, $F_\Phi = 0.79$, $C_\Phi = 28$	0	0	0
16	Walk: Φ = (V_{11}, V_{21}, A_{11}, A_{12}), $P_\Phi = 0.99$, $F_\Phi = 0.79$, $C_\Phi = 28$	0	0.21	4
17	Same as key point 16	0	0.21	4
18	Φ used: (V_{11}, V_{21}, A_{11}, A_{12}), Walk (A_{11}, A_{12}): $P_\Phi = 0.78$, $F_\Phi = 0.63$, $C_\Phi = 4$, No event (V_{11}, V_{21}): $C_\Phi = 24$	0.21	0.27	4
19	Since $P_\Phi < P_{spec}$ at point 18 $\Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (V_{21}, A_{11}, A_{12}, A_{21}, A_{22}), $P_\Phi = 0.95$, $F_\Phi = 0.81$, $C_\Phi = 20$	0	0	0
20	Φ used: (V_{21}, A_{11}, A_{12}, A_{21}, A_{22}), No event, $C_\Phi = 20$	0	0	12
	Average losses and savings per key point	**0.051**	**0.151**	**7.4**

Table VI. Timeline-Based Optimal Subset Selection Using **MinCost**

Key point	**Description**	**Loss in P_Φ**	**Loss in F_Φ**	**Saving in C_Φ**
1	No event	-	-	-
2	**All streams used and the optimal subset is Φ computed** Walk: Φ = (V_{11}, A_{21}, A_{22}), $P_\Phi = 0.95$, $F_\Phi = 0.72$, $C_\Phi = 16$	0	0	0
3	Φ used: (V_{11}, A_{21}, A_{22}), Walk: $P_\Phi = 0.95$, $F_\Phi = 0.72$, $C_\Phi = 16$	0.04	0.18	16
4	Φ used: (V_{11}, A_{21}, A_{22}), Walk: $P_\Phi = 0.77$, $F_\Phi = 0.72$, $C_\Phi = 16$ Since $P_\Phi < P_{spec}$ $\Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (A_{11}, A_{12}, A_{22}), $P_\Phi = 0.81$, $F_\Phi = 0.70$, $C_\Phi = 6.5$	0	0	0
5	Φ used: (A_{11}, A_{12}, A_{22}), Walk: $P_\Phi = 0.81$, $F_\Phi = 0.70$, $C_\Phi = 6.5$	0.18	0.20	25.5
6	Φ used: (A_{11}, A_{12}, A_{22}), Walk (A_{12}, A_{22}): $P_\Phi = 0.85$, $F_\Phi = 0.66$, $C_\Phi = 5$, Stand (A_{11}): $P_\Phi = 0.69$, $F_\Phi = 0.55$, $C_\Phi = 1.5$	0.14	0.24	25.5
7	Φ used: (A_{11}, A_{12}, A_{22}), Walk: $P_\Phi = 0.73$, $F_\Phi = 0.70$, $C_\Phi = 6.5$ Since $P_\Phi < P_{spec}$ $\Rightarrow$ **Optimal subset Φ recomputed**, Walk: Φ = (A_{11}, A_{21}, A_{22}), $P_\Phi = 0.81$, $F_\Phi = 0.67$, $C_\Phi = 5.5$	0	0	0
8	Φ used: (A_{11}, A_{21}, A_{22}), Knock: $P_\Phi = 0.97$, $F_\Phi = 0.67$, $C_\Phi = 5.5$	0.02	0.23	26.5
9	Same as key point 8	0.02	0.23	26.5
10	Same as key point 8 except $P_\Phi = 0.96$	0.03	0.23	26.5
11	Same as key point 8 except $P_\Phi = 0.98$	0.01	0.23	26.5
12	Same as key point 8 except $P_\Phi = 0.90$	0.09	0.23	26.5
13	Same as key point 8 except $P_\Phi = 0.96$	0.03	0.23	26.5
14	Φ used: (A_{11}, A_{21}, A_{22}), Knock (A_{22}): $P_\Phi = 0.73$, Since $P_\Phi < P_{spec}$ $\Rightarrow$ **Optimal subset Φ recomputed**, Knock (A_{11}, A_{12}): $P_\Phi = 0.85$, $F_\Phi = 0.63$, $C_\Phi = 4$	0	0	0
15	Knock (A_{11}, A_{12}): $P_\Phi = 0.85$, $F_\Phi = 0.63$, $C_\Phi = 4$	0.14	0.27	28
16	Same as key point 15 except $P_\Phi = 0.92$	0.07	0.27	28
17	Same as key point 15 except $P_\Phi = 0.89$	0.10	0.27	28
18	Same as key point 15 except $P_\Phi = 0.78$ Since $P_\Phi < P_{spec}$ $\Rightarrow$ **Optimal subset Φ recomputed**, Walk (A_{12}, A_{21}): $P_\Phi = 0.80$, $F_\Phi = 0.63$, $C_\Phi = 4$	0	0	0
19	Φ used: (A_{12}, A_{21}), Walk: $P_\Phi = 0.75$ Since $P_\Phi < P_{spec}$ $\Rightarrow$ **Optimal subset Φ recomputed**, Walk (A_{12}, A_{21}, A_{22}): $P_\Phi = 0.83$, $F_\Phi = 0.70$, $C_\Phi = 6.5$	0	0	0
20	Φ used: (A_{12}, A_{21}, A_{22}), No event, $C_\Phi = 6.5$	0	0	25.5
	Average losses and savings per key point	**0.042**	**0.141**	**16.8**



(a)

(b)

Fig. 6. Comparison of (a) **MaxGoal** and **MaxConf** (with $C_n = 32$); (b) **MinCost** (with $L = 100$) with the brute-force approach.

6. CONCLUSIONS

In this article, we propose a framework that uses a dynamic programming approach to find the optimal subset of media streams for three different objectives: maximizing the probability of achieving the goal under specified cost and confidence constraints; maximizing confidence under specified cost and probability constraints; and minimizing the cost of using the subset to achieve the goal with a specified probability and with a specified confidence. Each of these problems is proven to be NP-Complete, after which we have proposed a dynamic programming approach that finds the optimal subset of media streams based on the aforementioned three criteria. The proposed framework allows for tradeoffs among the three previously mentioned criteria, and offers the flexibility to compare different subsets in terms achieved goal, confidence and the incurred cost. The dynamic programming solution offers the user the flexibility to choose alternative subsets when the best subset is unavailable. The experimental results show the utility of the framework for detecting events in a surveillance scenario. The results show that the subset of a significantly lower cost can help in detecting events at the expense of only minor loss in the probability confidence with which the goal is achieved.

In future work, it would be interesting to explore how the framework can be used in other scenarios, such as selecting streams in media search systems. We will also focus on the formalization of how frequently the approximately optimal subset should be recomputed. Although we have focused on multimedia inputs, we also foresee a similar problem with respect to multimedia output, where we would try to determine the minimal subset of multimedia streams to communicate an intent.

ACKNOWLEDGMENTS

We thank our colleagues Frank Stephan and Wei-Tsang Ooi for their help in developing the proof of the theorem given in this article.

REFERENCES

Atrey, P. K. and Kankanhalli, M. S. 2005. Goal based optimal selection of media streams. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (Amsterdam, The Netherlands). 305–308.

Atrey, P. K., Kankanhalli, M. S., and Jain, R. 2005. Timeline-Based information assimilation in multimedia surveillance and monitoring systems. In *Proceedings of the 3rd ACM International Workshop on Video Surveillance and Sensor Networks* (Singapore). 103–112.

Atrey, P. K., Maddage, N. C., and Kankanhalli, M. S. 2006. Audio based event detection for multimedia surveillance. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. V813–816.

Debouk, R., Lafortune, S., and Teneketzis, D. 2002. On an optimal problem in sensor selection. *J. Discrete Event Dynamic Syst.: Theory Appl. 12*, 417–445.

Isler, V. and Bajcsy, R. 2005. The sensor selection problem for bounded uncertainty sensing models. In *Proceedings of the International Symposium on Information Processing in Sensor Networks* (Los Angeles, CA). 151–158.

Jain, R. 2004. Refining the search engine. *Ubiquity 5,* 29 (Sept.).

Jiang, S., Kumar, R., and Garcia, H. E. 2003. Optimal sensor selection for discrete event systems with partial observation. *IEEE Trans. Autom. Control 48*, 369–381.

Lam, K.-Y., Cheng, R., Liang, B., and Chau, J. 2004. Sensor node selection for execution of continuous probabilistic queries in wireless sensor networks. In *Proceedings of the ACM International Workshop on Video Surveillance and Sensor Networks* (New York). 63–71.

Nevatia, R., Zhao, T., and Hongeng, S. 2003. Hierarchical language-based representation of events in video streams. In *Proceedings of the IEEE Workshop on Event Mining* (Madison, WI).

Oommen, B. J. and Rueda, L. 2005. A formal analysis of why heuristic functions work. *Artif. Intell. J. 164*, 1–22.

Pahalawatta, P., Pappas, T. N., and Katsaggelos, A. K. 2004. Optimal sensor selection for video-based target tracking in a wireless sensor network. In *IEEE International Conference on Image Processing*. Singapore, V:3073–3076.

Siegel, M. and Wu, H. 2004. Confidence fusion. In *Proceedings of the IEEE International Workshop on Robot Sensing*. 96–99.

Stauffer, C. and Grimson, W. E. L. 1999. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2 (Ft. Collins, CO). 252–258.

Received June 2005; revised March 2006; accepted May 2006