Sensor Network Operations(Chapter in a IEEE press Monograph 2004 Sept.).

A Multisensor Network Based Framework for Video Surveillance: Realtime Super-resolution Imaging

Guna Seetharaman Electrical
and Computer Engineering Air Force
Institute of Technology Wright
Patterson AFB, OH 45433
guna@afit.edu

Ha V. Le
Department of Electrical and Computer Engineering
Vietnam National University, Hanoi
144 Xuan Thuy, Hanoi, Vietnam
hvle@hn.vnn.vn

S. S. Iyengar
Department of Computer Science
Louisiana State University
Baton Rouge LA 70803
iyengar@bit.csc.lsu.edu

N. Balakrishnan
Super Computing Research Center
Indian Institute of Science
Bangalore
India

R. Loganantharaj
Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette LA 70504
logan@cacs.louisiana.edu

Abstract

A network of multi modal sensors with distributed and embedded computations is considered for a video surveillance and monitoring application. Practical factors limiting the video surveillance of large areas are highlighted. A network of line-of-sight sensors and mobile-agents based computations are proposed to increase the effectiveness. CMOS digital cameras in which both sampling and quantization occur on the sensor focal plane are more suitable for this application. These cameras operate at very high video frame rates and are easily synchronized to acquire images synaptically across the entire network. Also, they feature highly localized short term memories and include some SIMD parallel computations as an integral part of the image acquisition. This new framework enables distributed computation for piecewise stereovision across the camera network, enhanced spatio-temporal fusion, and super resolution imaging of steadily moving subjects. A top level description of the monitor, locate and track model of a surveillance and monitoring task is presented. A qualitative assessment of several key elements of the mobile agents based computation for tracking persistent tokens moving across the entire area is outlined. The idea is to have as many agents as the number of persons in the field of view, and perform the computations in a distributed fashion without introducing serious bottlenecks. The overall performance is promising when compared against that of a small network of cameras monitoring large corridors with a human operator in the loop.

Keywords: super-resolution, motion compensation, optical flow

The author was with the Center for Advanced Computer Studies, University of Louisiana, when this work was started.

1 Introduction

Increased access to inexpensive fabrication of CMOS circuits has vitalized research in intelligent sensors with embedded computing and power aware features. Video image sensors [10], [14], and infra red detectors have been built using standard CMOS processes, with embedded digital signal processing in the pixel planes. They indicate an emerging trend in modeling the flow of information in an image processing system. The old 'acquire, plumb, and process – in chain' model of image processing systems would be replaced. The chain would most likely evolve into a topsorted graph comprised of: acquire-and-fuse, macro-assimilate, and meta-process stages, in which several nodes within each stage may be connected via lateral parallelism (fusion at that level). In this new paradigm, the data – at each stage of the chain would be subjected to appropriately designed intelligent processing, giving rise to a pipeline of incrementally inferred-knowledge. Insight into data representation and modeling of data flow in this framework could have a profound impact on making large surveillance systems more tractable through simpler distributed and parallel computing. The new approach would require sensor and data fusion at all levels, in both time and space. Smart CMOS pixel planes with simple and networked computational features could make video surveillance more effective and tractable. This paper is an effort to introduce a frame work, and examine at least one of the newly enabled benefits of such a basic multisensor network system.

The paper is organized as follows. The basic structure of a large area video surveillance and monitoring system in the context of a busy airport is described. Some key parameters that may be used for gaging its performance are identified. Practical issues contributing to design tradeoffs are outlined. A set of currently known video imaging sensors and associated monitoring algorithms are described. A line of sight sensor network is introduced to effectively decouple the subtasks of detecting events of interest, and of tracking known tokens in space and time. A sub pixel accurate motion compensated super resolution imaging algorithm is applied as a suitable candidate to benefit from this decoupling. A description of mobile agents based computations is presented in the context of this multisensor surveillance and monitoring network. Some challenges that may have to be addressed are listed in the conclusion.

2 Basic Model of Distributed Multi-Sensor Surveillance System

We present the basic factors influencing the analysis and performance of the dataflow and computation in a large-scale multisensor network, in the context of a airport surveillance system. The relevant factors are captured in Figure 1, including the dynamic computational states of the embedded software agents. The basic model assumes that some mechanism is available to distinctly locate multiple objects (persons) within its field of view, as they enter. Let m be the number of people that can check in simultaneously and pass through the pre-determined points of entry. Their exact locations are picked up by fast 3D sensors.

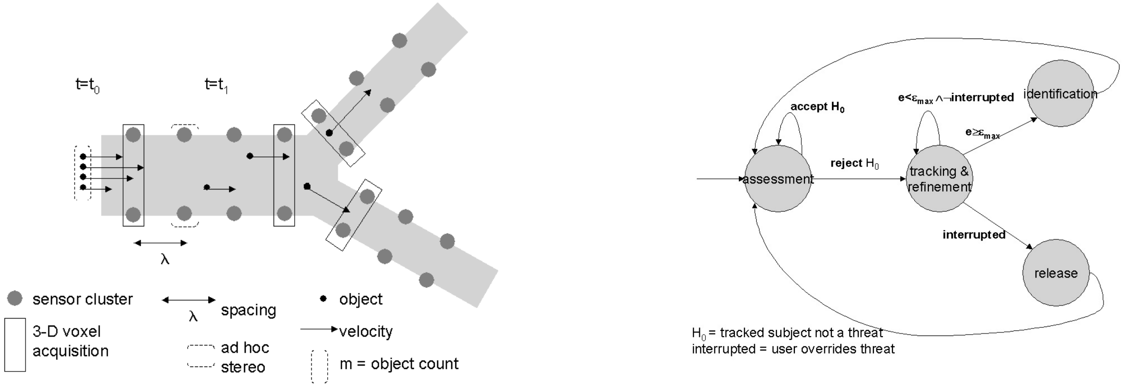


Figure 1: *The central theme is to observe video image of a busy corridor, with multiple lanes and branches. The principal paradigm is to track every moving entity until a threshold has been reached to raise an alarm or abandon tracking.*

Several multi sensors, such as X-ray and infrared cameras, may acquire additional data and tag the data to the voxel. This is in fact the origin of a spatio-temporal thread associated with the events triggered by the moving person. Each person moves at an arbitrary pace. Let v be the average velocity. They are monitored by sensor clusters downstream. The local agents at these sensor clusters will need a speed and power proportional to $m \cdot v$. The local memory required and the monitoring complexity are $m\lambda v$. Network data flow is $m \cdot v$ The spatio-temporal registration between two stations separated by a distance λ will be proportional to: m^2. It is envisioned that large variations in velocity can complicate the matter. In that case, simple dynamic programming approach using last recently known location as the index of search space could be exploited to reduce the complexity of the problem. Along these lines, we estimate the tracking complexity to be proportional to $m^2(1+\alpha|v_{max}-v_{min}|)$, for the purpose of spatio-temporal registration. Further inspection reveals that the peak load on the agents would also be influenced by factor proportional to the variation. The complexity of local agent processing power, and memory requirements will scale up by a factor $(1+\alpha|v_{max}-v_{min}|)$. In all these cases, the value of α would assume different values, one for each context.

Given a voxel and a camera whose field of view covers the same, the location of the image of the voxel in the captured video image is trivially determined if the cameras have been fully calibrated. We assume this to be the case. In a sense, this abstraction treats some clusters of sensors to be more adept at identifying distinct events; whereas, others downstream are more efficient in tracking them. Also, we do not preclude the possibility of any pair of adjacent sensors in forming ad hoc means to resolve 3D locations should there be a need triggered by local temporal events. That is, more rigorous voxel acquisition sensors are placed sporadically, and loosely coupled video cameras are widespread.

2.1 Sensors: Design, Deployment, Data Acquisition and Low Level Fusion.

Rapid acquisition of the 3D location (voxel occupancy) of people in their field of view is essential. A very high speed three dimensional sensor published in literature [1] may be used to acquire the 3D image of physical scenes. The sensor is constructed with an array of smart analog-pixel sensors. Each smart pixel is made of a photo cell, an analog comparator and a sample-and-hold circuit. The 3D data acquisition requires a planar laser beam to sweep through the scene. When the laser sweeps through the scene, it would produce an event of significance at various pixels at different times, – easily detected by an increase in intensity. The exact time of the event is captured which amounts to sensing the depth.

It is possible to acquire 3D data by a set of two video cameras, and not require a laser beam, or use laser minimally when necessary. Such a hybrid range and passive video sensor [21]. The sensor consists of two video cameras with significant overlap in their fields of view and similarity in their parallax. It seeks to detect a number of feature points in each view, in an attempt to establish point correspondence [Huang84] and compute 3D data. It involves spatial search for concurring observation(s) across the views. The spatial search has a well defined geometric pattern known as epipolar lines. Then, a SIMD parallel computing array makes it possible to acquire up to 15000 voxels per second [21]. In addition two image sequences are delivered by the cameras.

The computation described above is an example of low level fusion, facilitated by a pair of networked sensors with mutual access to very low-level data of each other. A number of higher-level approaches exist for stereovision; and, they differ in terms of the feature space used to detect the points of interest, and the methods used for matching them across two views, and the controllability of the observed space. Such computations, in our view, fall in the classification of macro and mid level fusion. Often they use scene knowledge and object knowledge and not the temporal signatures. That is, mutual access to the raw data of neighboring sensors and spatio-temporal fusion at low-level, we believe would offer speed, albeit with a need for post processing.

A set of omni directional (isotropic) light emitting diodes driven by a multiphase clock, and a set of omni directional light sensors can be deployed in large numbers along the corridor. These would be packaged in easily installed and networked strips. The spatio-temporal signals acquired through these rudimentary sensor nets would help reduce the camera count and increase their separation. Some of them come packaged with local micro-controllers for communication purposes. Such a sensor network is illustrated in Figure 2.

2.2 Overview of Cooperative Agents that Monitor and Track

An agent plays a pivotal role in integrating raw sensor inputs and information coming from sensor fusion and image analysis so as to achieve effective monitoring and tracking objects by collaborating with other agents. A software agent is a program that perceives an environment through sensors and acts on the environment [19] [6] , so as to achieve the

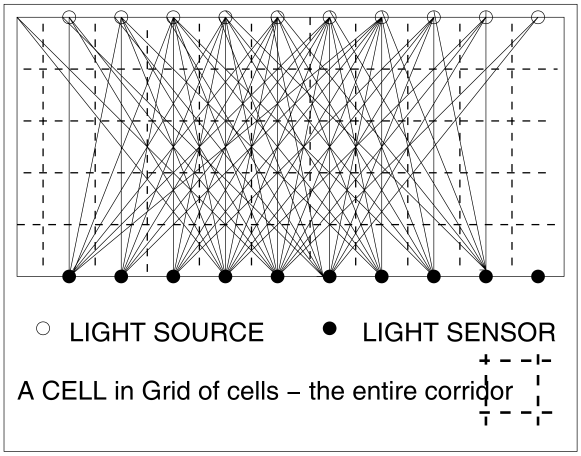


Figure 2: *A Line-of-Sight sensor network comprised of light emitters and light sensors shown above effectively eliminates the bottleneck of locating people (points of interest) in video frame rate.*

intended purposes, which is monitoring and tracking all the objects with its monitoring boundaries. Agents are autonomous within the context of its intended purposes and thus an ideal choice to perform continuous monitoring and tracking purposes. A functional architecture of our agent is given in Figure 3, which is inspired by the architecture of remote agent that was successfully deployed by NASA. The agent has five functional components namely execution monitor, planning and scheduler, knowledge-base, detector of unusual behavior and a communicator.

We envision a set of agents are collaborating together to achieve the overall purpose. The communicator module of an agent is responsible for maintaining all relevant information of all other agents in the system and respond to the request from the execution monitor to send message to another agent. Here we assume that the agents do have a global knowledge of other agents in the system their capabilities and their layout along with their physical monitoring boundaries. If for some reason an agent fails and another agent becomes active in covering the void left by the failing agent, the new agent communicates with all other agents, specially the one in its neighborhood to inform about its existence, its coverage area and its capability. Steps have been taken to safeguard against some malicious agents pretending to be part of the cooperating agents.

To monitor and to track an object, each object entering into the monitoring space is given a unique identification number, which in our case is a time stamp followed by a predefined number of digits, say two digit number, generated randomly. If many objects enter at he same time through different entrance, there may be a possibility (1/1000 in the case of two digit random number) of assigning the same id for two objects. If such violation occurs, it will be corrected appropriately. In addition to assigning a unique id for a 3D voxel, other features of the object is also stored, which will help to identifying the voxel and reassigning

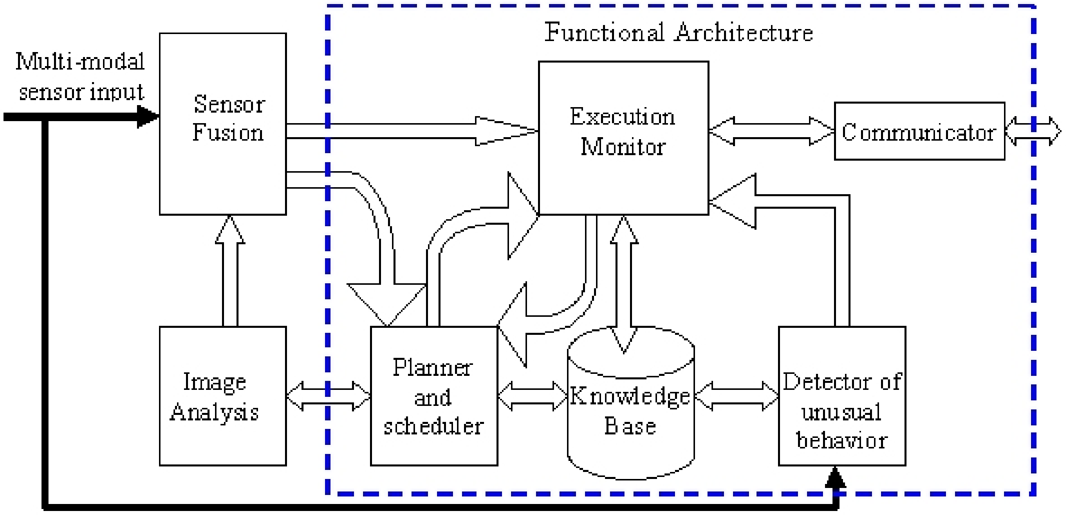


Figure 3: *Functional architecture of the intelligent agent*

the id after the object temporarily leave out of the monitoring area, such as rest room etc.

For tracking and monitoring purposes, spatial temporal history of an object, that is, its location at different time points, is maintained by the module for planning and scheduling. From the information of location over the time period, the current speed and the direction is inferred and it is being used to predict the future location in the next time slot. The approximate locations in the next time slot of objects are fed into the image analyzers. The image analyzer along with the sensors, confirm or obtain the new locations of the requested objects and update the planning and scheduling module. For some reason no object is identified in the region, then the area is gradually increased and the identification process continues. This is the basis of tracking. If many objects appeared in an area, it is resolved by matching with the candidates who may have moved from their previous positions. The projected path of an object is being used to predict whether it will leave the monitoring boundary of the current agent. The id and the coordinates of the objects that are predicted to cross the boundary are handed over to the agents who will subsequently monitor those objects. The knowledge-base has predefined template of features and the expected behaviors of the objects that matched the features.

The agility of the application demands immediate recognition of the unusual behavior of an object. To realize such recognition, we implement reactive behavior of an agent using the sensor input. The abnormal detecting module is trained to recognize a classes of abnormal behaviors from the raw sensor input and thereby avoiding processing time. Based on the thread levels of abnormal behavior, the execution module of the agent will take appropriate action. For example, if the sensor detects smoke, the agent will immediately enabled the fire alarm and inform all the security personnel who are trained to deal with the physical situation. In the context of mobile sensor networks, such as UAVs equipped with video cameras and wireless commucnication, the agents should also take into account the cost of communication with the other nodes. The optimal computation of local temporal computations, such as image sequence analysis should be designed to exploit local data first, and use stereovision sparingly.

3 Super Resolution Imaging

The objective of super-resolution imaging is to synthesize a higher resolution image of objects from a sequence of images whose spatial resolution is limited by the operational nature of the imaging process. The synthesis is made possible by several factors that effectively result in sub-pixel level displacements and disparities between the images.

Research on super-resolution imaging has been extensive in recent years. Tsai and Huang were the first trying to solve the problem. In [24], they proposed a frequency domain solution which uses the shifting property of the Fourier transform to recover the displacements between images. This as well as other frequency domain methods like [13] have the advantages of being simple and having low computational cost. However, the only type of motion between images which can be recovered from the Fourier shift is the global translation, therefore, the ability of these frequency domain methods is quite limited.

Motion-compensated interpolation techniques [23, 9] also compute displacements between images before integrating them to reconstruct a high resolution image. The difference between these methods and the frequency domain methods mentioned above is that they work in the spatial domain. Parametric models are usually used to model the motions. The problem is, most parametric models are established to represent rigid motions such as camera movements, while in the real world motions captured in image sequences are often non-rigid, too complex to be described by a parametric model. Model-based super-resolution imaging techniques such as back-projection [12] also face the same problem.

More powerful and robust methods such as the *projection onto convex sets* (POCS)-based methods [17], which are based on set theories, and stochastic methods like *maximum a posteriori* (MAP)-based [8] and *Markov random field* (MRF)-based [20] algorithms are highly complex in term of computations, hence unfit for applications which require real-time processing.

We focus our experimental study [15] on digital video images of objects moving steadily in the field of view of a camera fitted with a wide-angle lens. These assumptions hold good for a class of video based security and surveillance systems. Typically, these systems routinely perform MPEG analysis to produce a compressed video for storage and offline processing. In this context, the MPEG subsystem can be exploited to facilitate super-resolution imaging through a piecewise affine registration process which can easily be implemented with the MPEG-4 procedures. The method is able to increase the effectiveness of camera security and surveillance systems.

The flow of computation in the proposed method is depicted in Fig. 4. Each moving object will be separated from the background using standard image segmentation techniques.

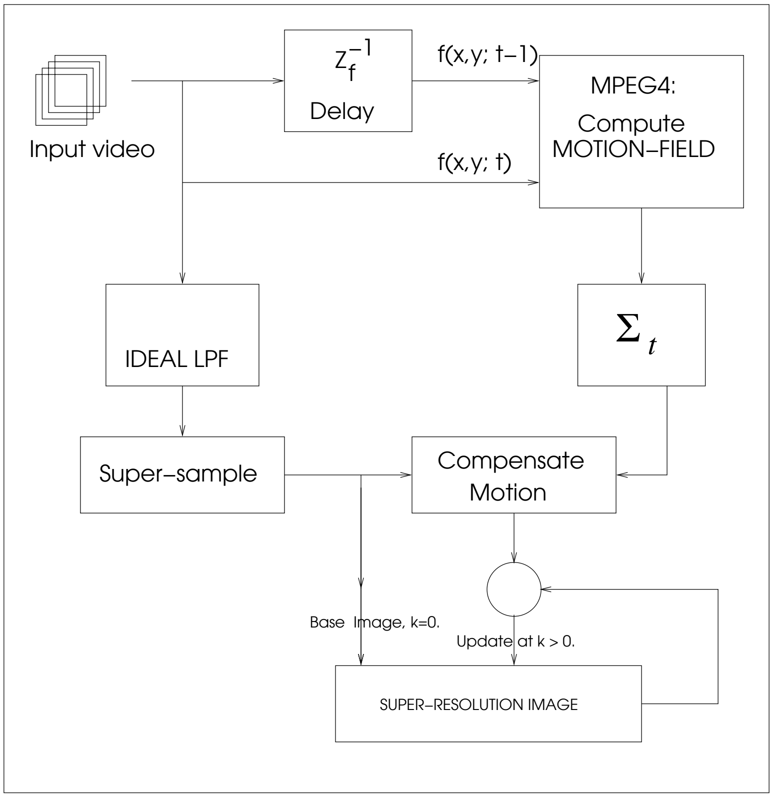


Figure 4: *The schematic block diagram of the proposed super-resolution imaging method.*

Also, a set of feature points, called the points-of-interest, will be extracted. These points include places were the local contrast patterns are well defined, and/or exhibit a high degree of curvature, and such geometric features. We track their motions in the 2-D context of a video image sequence. This requires image registration, or some variant of point correspondence matching. The net displacement of the image of an object between any two consecutive video frames will be computed with subpixel accuracy. Then, a rigid coordinate system is associated with the first image, and any subsequent image is modeled as though its coordinate system has undergone a piecewise affine transformation. We recover the piecewise affine transform parameters between any video frame with respect to the first video frame to a subpixel accuracy. Independently, all images will be enlarged to a higher resolution using a bilinear interpolation [16] by a scale factor. The enlarged image of each subsequent frame is subject to an inverse affine transformation, to help register it with the previous enlarged image. Given K video frames, then, in principle, it will be feasible to synthesize $K - 1$ new versions of the scaled and interpolated and inverse-motion-compensated image at the first frame instant. Thus, we have K high resolution images to assimilate from.

4 Optical Flow Computation

We follow a framework proposed by Cho et al. [7] for optical flow computation based on a piecewise affine model. A surface moving in the 3-D space can be modeled as a set of small planar surface patches. Then, the observed motion of each 3-D planar patch in the 2-D image plane can be described by an affine transform. Basically, this is a mesh-based technique for motion estimation, using 2-D content-based meshes. The advantage of content-based meshes over regular meshes is their ability to reflect the content of the scene by closely matching boundaries of the patches with boundaries of the scene features [2], yet finding feature points and correspondences between features in different frames is a difficult task. A multiscale coarse-to-fine approach is utilized in order to increase the robustness of the method as well as the accuracy of the affine approximations. An adaptive filter is used to smooth the flow field such that the flow appears continuous across the boundary between adjacent patches, while the discontinuities at the motion boundaries can still be preserved. Many of these techniques are already available in MPEG-4.

Our optical flow computation method includes the following phases:

1. *Feature extraction and matching*: in this phase the feature points are extracted and feature matching is performed to find the correspondences between feature points in two consecutive image frames.

2. *Piecewise flow approximation*: a mesh of triangular patches is created, whose vertices are the matched feature points. For each triangular patch in the first frame there is a corresponding one in the second frame. The affine motion parameters between these two patches can be determined by solving a set of linear equations formed over the known correspondences of their vertices. Each set of these affine parameters define a smooth flow within a local patch.

Finding the correspondences between feature points in consecutive frames is the key step of our method. We devised a matching technique in which the cross-correlation, curvature, and displacement are used as matching criteria. The first step is to find an initial estimate for the motion at every feature point in the first frame. Some matching techniques described in [22] would consider all of $M \times N$ pairs where M and N are the number of feature points in the first ans second frames, respectively. Some others assume the displacements are small to limit the search for a match to a small neighborhood of each point. By giving an initial estimate for the motion at each point, we are also able to reduce the number of pairs to be examined without having to constrain the motion to small displacements. We have devised a multiscale scheme, in which the initial estimation of the flow field at one scale is given by the piecewise affine transforms computed at the previous level. At the starting scale, a rough estimation can be made by treating the points as if they are under a rigid 2-D motion. It means the motion is a combination of a rotation and a translation. Compute the centers of gravity, C_1 and C_2, the angles of the principal axes, α_1 and α_2, of the two sets of feature points in two frames. The motion at every feature points in the first frame can be roughly estimated by a rotation around C_1 with the angle $\phi = \alpha_2 - \alpha_1$, followed by a translation represented by the vector $\mathbf{t} = \mathbf{x}_{C_2} - \mathbf{x}_{C_1}$, where $\mathbf{x}_{C_1}$ and $\mathbf{x}_{C_2}$ are the vectors representing the coordinations of C_1 and C_2 in their image frame.

Let i^t and j^{t+1} be two feature points in two frames t and $t+1$, respectively. Let i'^{t+1} be the estimated match of i^t in frame $t+1$, $d(i', j)$ be the Euclidean distance between i'^{t+1} and j^{t+1}, $c(i, j)$ be the cross-correlation between i^t and j^{t+1}, $0 \le c(i, j) \le 1$, and $\Delta\kappa(i, j)$ be the difference between the curvature measures at i^t and j^{t+1}. A *matching score* between i^t and j^{t+1} is defined as follows

$$\begin{array}{rcl} d(i', j) > d_{\max} : & & \\ s(i, j) & = & 0 \\ d(i', j) \le d_{\max} : & & \\ s(i, j) & = & w_c c(i, j) + s_k(i, j) + s_d(i, j), \end{array} \tag{1}$$

where

$$\begin{array}{rcl} s_k(i, j) & = & w_k(1 + \Delta\kappa(i, j))^{-1} \\ s_d(i, j) & = & w_d(1 + d(i', j))^{-1} \end{array} \tag{2}$$

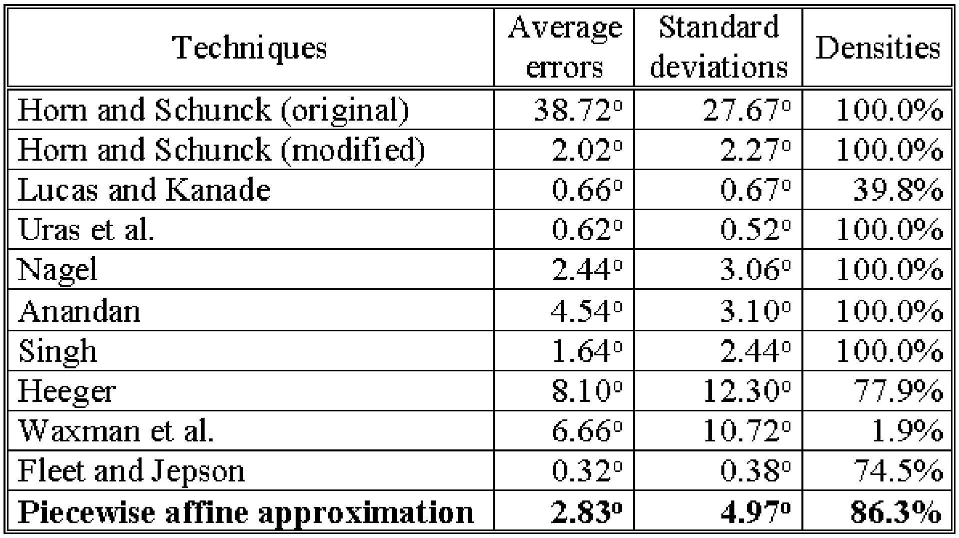
The quantity $d_{\max}$ specifies the maximal search distance from the estimated match point. w_c, w_k, and w_d are the weight values, determining the importance of each of the matching criteria. The degree of importance of each of these criteria changes at different scales. At a finer scale, the edges produced by Canny edge detector become less smooth, meaning the curvature measures are less reliable. Thus, w_k should be reduced. On the other hand, w_d should be increased, reflecting the assumption that the estimated match becomes closer to the true match. For each point i^t, its optimal match is a point j^{t+1} such that $s(i, j)$ is maximal and exceeds a threshold value t_s. Finally, inter-pixel interpolation and correlation matching are used in order to achieve subpixel accuracy in estimating the displacement of the corresponding points.

Using the constrained Delaunay triangulation [11] for each set of feature points, a mesh of triangular patches is generated to cover the moving part in each image frame. A set of line segments, each of which connects two adjacent feature points on a same edge, is used to constrain the triangulation, so that the generated mesh closely matches the true content of the image. Each pair of matching triangular patches, results in six linear equations made of piecewise local affine motion parameters, which can be solved to produce a dense velocity field inside the triangle.

4.1 Evaluation of Optical Flow Computation Technique

We conducted experiments with our optical flow estimation technique using some common image sequences created exclusively for testing optical flow techniques and compared the results with those in [3] and [5]. The image sequences used for the purpose of error evaluation include the Translating Tree sequence (Fig. 6), the Diverging Tree sequence (Fig. 7), and the Yosemite sequence (Fig. 8). These are simulated sequences for which the ground truth is provided.

As in [3] and [5], an angular measure is used for error measurement. Let $\mathbf{v} = [u \;\; v]^T$ be the correct 2-D motion vector and $\mathbf{v}_e$ be the estimated motion vector at a point in the



Techniques	Average errors	Standard deviations	Densities
Horn and Schunck (original)	38.72°	27.67°	100.0%
Horn and Schunck (modified)	2.02°	2.27°	100.0%
Lucas and Kanade	0.66°	0.67°	39.8%
Uras et al.	0.62°	0.52°	100.0%
Nagel	2.44°	3.06°	100.0%
Anandan	4.54°	3.10°	100.0%
Singh	1.64°	2.44°	100.0%
Heeger	8.10°	12.30°	77.9%
Waxman et al.	6.66°	10.72°	1.9%
Fleet and Jepson	0.32°	0.38°	74.5%
Piecewise affine approximation	**2.83°**	**4.97°**	**86.3%**

Table 1: *Performance of various optical flow techniques on the Translating Tree sequence.*

image plane. Let $\tilde{\mathbf{v}}$ be a 3-D unit vector created from a 2-D vector $\mathbf{v}$:

$$\tilde{\mathbf{v}} = \frac{[\mathbf{v} \;\; 1]^T}{|[\mathbf{v} \;\; 1]|} \tag{3}$$

The angular error ψ_e of the estimated motion vector $\mathbf{v}_e$ with respect to the correct motion vector $\mathbf{v}$ is defined as follows:

$$\psi_e = \arccos(\tilde{\mathbf{v}}.\tilde{\mathbf{v}}_e) \tag{4}$$

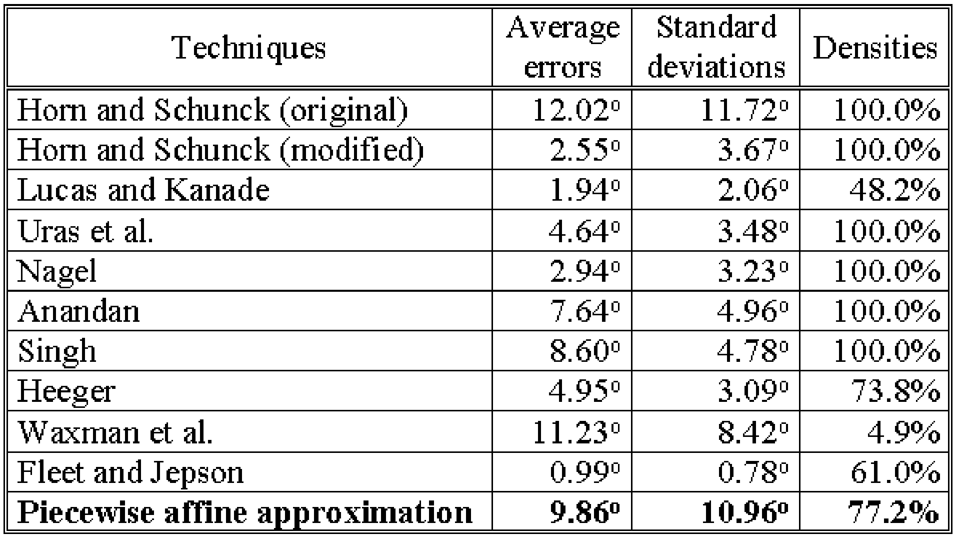
Using this angular error measure, bias caused by the amplification inherent in a relative measure of vector differences can be avoided.

To verify if the accuracies are indeed sub-pixel, we use the distance error $d_e = |\mathbf{v} - \mathbf{v}_e|$. For the Translating Tree sequence, the mean distance error is 11.40% of a pixel and the standard deviation of errors is 15.69% of a pixel. The corresponding figures for the Diverging Tree sequence are 17.08% and 23.96%, and for the Yosemite sequence are 31.31% and 46.24%. It is obvious that the flow errors at most points of the images are sub-pixel.

5 Super-Resolution Image Reconstruction

Given a low-resolution image frame $\mathbf{b}_k(m, n)$, we can reconstruct an image frame $\mathbf{f}_k(x, y)$ with a higher resolution as follows [16]:

$$\mathbf{f}_k(x, y) = \sum_{m,n} \mathbf{b}_k(m, n) \frac{\sin \pi(x\lambda^{-1} - m)}{\pi(x\lambda^{-1} - m)} \frac{\sin \pi(y\lambda^{-1} - n)}{\pi(y\lambda^{-1} - n)} \tag{5}$$



Techniques	Average errors	Standard deviations	Densities
Horn and Schunck (original)	12.02°	11.72°	100.0%
Horn and Schunck (modified)	2.55°	3.67°	100.0%
Lucas and Kanade	1.94°	2.06°	48.2%
Uras et al.	4.64°	3.48°	100.0%
Nagel	2.94°	3.23°	100.0%
Anandan	7.64°	4.96°	100.0%
Singh	8.60°	4.78°	100.0%
Heeger	4.95°	3.09°	73.8%
Waxman et al.	11.23°	8.42°	4.9%
Fleet and Jepson	0.99°	0.78°	61.0%
Piecewise affine approximation	**9.86°**	**10.96°**	**77.2%**

Table 2: *Performance of various optical flow techniques on the Diverging Tree sequence.*

where $\frac{\sin\theta}{\theta}$ is the ideal interpolation filter, and λ is the desired resolution step-up factor. For example, if $\mathbf{b}_k(m,n)$ is a 50×50 image and $\lambda = 4$, then, $\mathbf{f}_k(x,y)$ will be of the size $200\times\ 200$.

Each point in the high-resolution grid corresponding to the first frame can be tracked along the video sequence from the motion fields computed between consecutive frames, and the super-resolution image is updated sequentially:

$$x^{(1)} = x, y^{(1)} = y, \mathbf{f}_1^{(1)}(x,y) = \mathbf{f}_1(x,y) \tag{6}$$

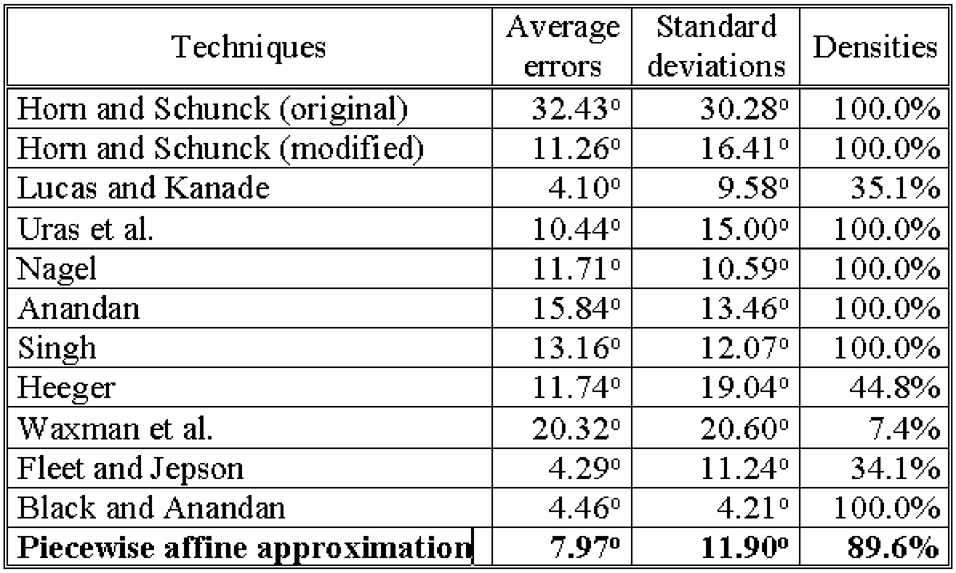
$$x^{(k)} = x^{(k-1)} + u_k(x^{(k-1)}, y^{(k-1)}), y^{(k)} = y^{(k-1)} + v_k(x^{(k-1)}, y^{(k-1)}) \tag{7}$$

$$\mathbf{f}_k^{(k)}(x,y) = \frac{k-1}{k}\mathbf{f}_{k-1}^{(k-1)}(x,y) + \frac{1}{k}\mathbf{f}_k(x^{(k)}, y^{(k)}) \tag{8}$$

for $k = 2,3,4\cdots$. The values u_k and v_k represent the dense velocity field between $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$. This sequential reconstruction technique is suitable for online processing, in which the super-resolution images can be updated every time a new frame comes.

6 Experimental Results

In the first experiment we used a sequence of 16 frames capturing a slow- moving book (Fig. 9). Each frame was down-sampled by a scale of four. High resolution images were reconstructed from the down-sampled ones, using $2, 3, ..16$ frames, respectively. The graph in Fig. 5 shows errors between reconstructed images and their corresponding original frame keep decreasing when the number of low-resolution frames used for reconstruction is increased, until the accumulated optical flow errors become significant. Even though this is



Techniques	Average errors	Standard deviations	Densities
Horn and Schunck (original)	32.43°	30.28°	100.0%
Horn and Schunck (modified)	11.26°	16.41°	100.0%
Lucas and Kanade	4.10°	9.58°	35.1%
Uras et al.	10.44°	15.00°	100.0%
Nagel	11.71°	10.59°	100.0%
Anandan	15.84°	13.46°	100.0%
Singh	13.16°	12.07°	100.0%
Heeger	11.74°	19.04°	44.8%
Waxman et al.	20.32°	20.60°	7.4%
Fleet and Jepson	4.29°	11.24°	34.1%
Black and Anandan	4.46°	4.21°	100.0%
Piecewise affine approximation	**7.97°**	**11.90°**	**89.6%**

Table 3: *Performance of various optical flow techniques on the Yosemite sequence.*

a simple case because the object surface is planar and the motion is rigid, it nevertheless presented the characteristics of this technique.

The second experiment was performed on images taken from a real surveillance camera. In this experiment we tried to reconstruct high-resolution images of faces of people captured by the camera (Fig. 10). Results show obvious improvements of reconstructed super-resolution images over original images.

For the time being, we are unable to conduct a performance analysis of our super-resolution method in comparison with others', because: 1) There has been no study on quantitative evaluation of the performance of super-resolution techniques so far; and 2) There are currently no common metrics to measure the performance of super-resolution techniques (in fact, most of the published works on this subject did not perform any quantitative performance analysis at all). The number of super-resolution techniques are so large that a study on comparison of their performances could provide enough contents for another paper.

7 Conclusion

We have presented a method for reconstructing super-resolution images from sequences of low-resolution video frames, using motion compensation as the basis for multi-frame data fusion. Motions between video frames are computed with a multiscale piecewise affine model which allows accurate estimation of the motion field even if the motion is non-rigid. The reconstruction is sequential – only the current frame, the frame immediately before it and the last reconstructed image are needed to reconstruct a new super-resolution image.

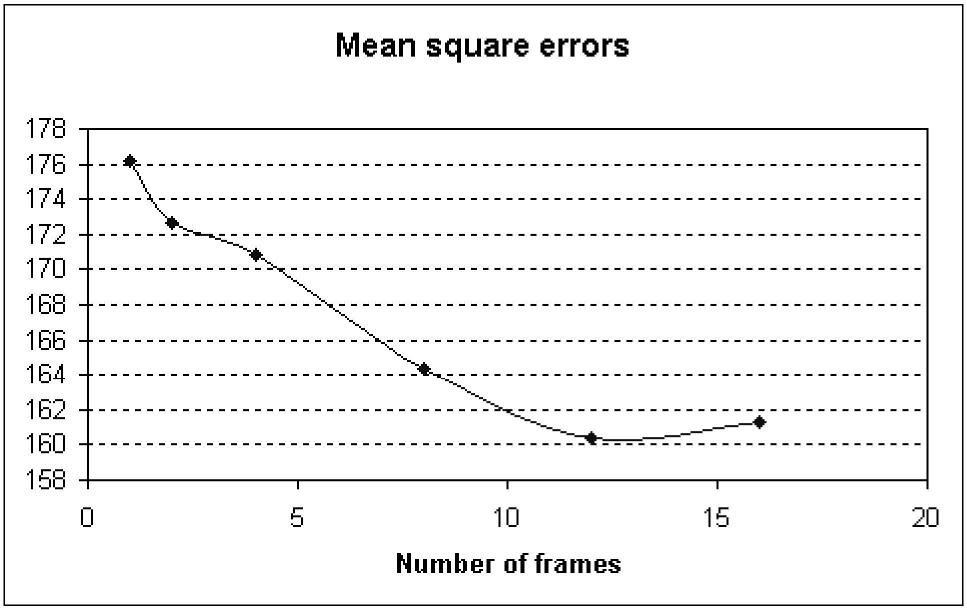


Figure 5: *Graph of mean square errors between reconstructed images and the original frame.*

This makes it suitable for applications that require real-time operations like in surveillance systems. The proposed super resolution is one example of a number of new computations made feasible in a distributed multi sensor network comprised of many video cameras, and line-of-sight sensors. Coarser measurement of people in 3D field of view is made available by the line-of-sight sensors. This in fact helps decouple the tasks of detecting points of interest in images, and recognizing / monitoring the dynamically moving objects giving rise to that points of interest. The decoupling facilitates high frame rate of computation. The task of interpreting the super resolution images, and its dynamics, is currently in progress.

8 Disclaimer

This work was initially conceived when the primary author was with The Center for Advanced Computer Studies, University of Louisiana. The views expressed in this article are those of the authors and do not necessarily reflect the official policy or position of the United States Air Force, the U.S. Department of Defense or the U.S. Government. (AFIT-35-101 pp. 15.5). This article has been cleared for public dissemination with a requirement that this disclaimer be included.

References

[1] L. R. Carley A. Gruss and T. Kanade. Integrated Sensor and Range-Finding Analog Signal Processor. *IEEE Journal of Solid-State Circuits*, 26(3):184–191, March 1991.

[2] Y. Altunbasak and M. Tekalp. Closed-Form Connectivity-Preserving Solutions for Motion Compensation Using 2-D Meshes. *IEEE Transactions on Image Processing*, 6(9):1255–1269, September 1997.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[4] D. Bernard, G. Dorias, E. Gamble, B. Kanefsky, and *et.al.* Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment. In *Proceedings of the AIAA 1999 Annual Conference. Albuquerque, NM*, 1999.

[5] M. J. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Field. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[6] J. M. Bradshaw. *Software Agents*. MIT Press, Cambridge, MA, 1997.

[7] E. C. Cho, S. S. Iyengar, G. Seetharaman, R. J. Holyer, and M. Lybanon. Velocity Vectors for Features of Sequential Oceanographic Images. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3):985–998, May 1998.

[8] M. Elad and A. Feuer. Restoration of a Single Superesolution Image from Several Blurred, Noisy and Undersampled Measured Images. *IEEE Trans. on Image Processing*, 6(12):1646–1658, December 1997.

[9] M. Elad and Y. Hel-Or. A Fast Super-Resolution Reconstruction Algorithm for Pure Translational Motion and Common Space-Invariant Blur. *IEEE Trans. on Image Processing*, 10(8):1187–1193, August 2001.

[10] E. R. Fossum. CMOS Image Sensors: Electronic Camera-on-A-Chip. *IEEE Trans. on Electronic Devices*, 44(10), 1997.

[11] S. Guha. An Optimal Mesh Computer Algorithm for Constrained Delaunay Triangulation. In *Proceedings of the International Parallel Processing Symposium*, pages 102–109, Cancun, Mexico, April 1994.

[12] M. Irani and S. Peleg. Motion Analysis for Image Enhancement: Resolution, Occlusion and Transparency. *Journal of Visual Communications and Image Representation*, 4:324–335, December 1993.

[13] S. P. Kim and W.-Y. Su. Recursive High-Resolution Reconstruction of Blurred Multiframe Images. *IEEE Trans. on Image Processing*, 2(10):534–539, October 1993.

[14] S. Kleinfelder and et.al. A 10000 Frames/sec CMOS Digital Pixel Sensor. *IEEE Journal of Solid-State Circuits*, 36(12), 2001.

[15] Ha Le and G. Seetharaman. A Method of Super-resolution Imaging based on Dense Subpixel Accurate Motion Fields. In *Proceedings of the International Workshop on Digital Computational Video DCV2002*, Nov 2002.

[16] E. Meijering. A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing. *Proc. of The IEEE*, 90(3):319–344, March 2002.

[17] A. J. Patti, M. I. Sezan, and A. M. Tekalp. Superresolution Video Reconstruction with Arbitrary Sampling Lattices and Nonzero Aperture Time. *IEEE Trans. on Image Processing*, 6(8):1064–1076, August 1997.

[18] K. Rajan, M. Shirley, W. Taylor, and B. Kanefsky. Ground Tools for the 21st Centrury. In *Proceedings of the IEEE Aerospace Conference, Big Sky, MT*, 2000.

[19] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, Englewood NJ, 2003.

[20] R. R. Schultz and R. L. Stevenson. Extraction of High-Resolution Frames from Video Sequences. *IEEE Trans. on Image Processing*, 5(6):996–1011, June 1996.

[21] Guna Seetharaman, Magdy Bayoumi, Kimon Valavanis, and Michael Mulder. A VLSI Architecture for Stereo Image Sensors. In *Proceedings of the Workshop on Computer Architecture for Machine Perception. Paris.*, December 1991.

[22] R. N. Strickland and Z. Mao. Computing correspondences in a sequence of non rigid images. *Pattern Recognition*, 25(9):901–912, 1992.

[23] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan. High Resolution Image Reconstruction from Low Resolution Image Sequences, and Space Varying Image Restoration. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 169–172, San Francisco, CA, March 1992.

[24] R. Y. Tsai and T. S. Huang. Multiframe Image Restoration and Registration. In R. Y. Tsai and T. S. Huang, editors, *Advances in Computer Vision and Image Processing*, volume 1, pages 317–339. JAI Press Inc., 1984.

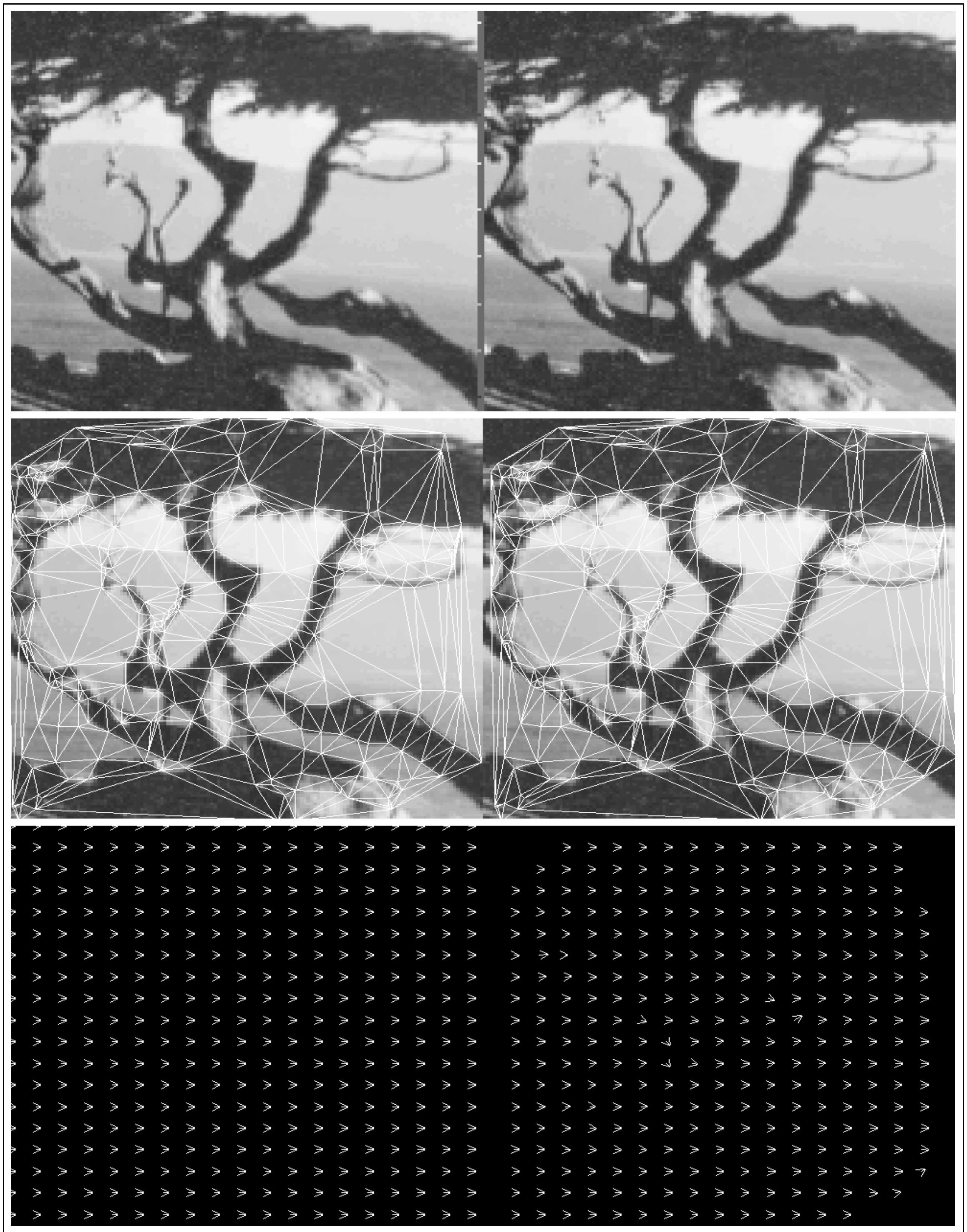


Figure 6: *Top: two frames of the Translating Tree sequence. Middle: generated triangular meshes. Bottom: the correct flow (left) and the estimated flow (right).*

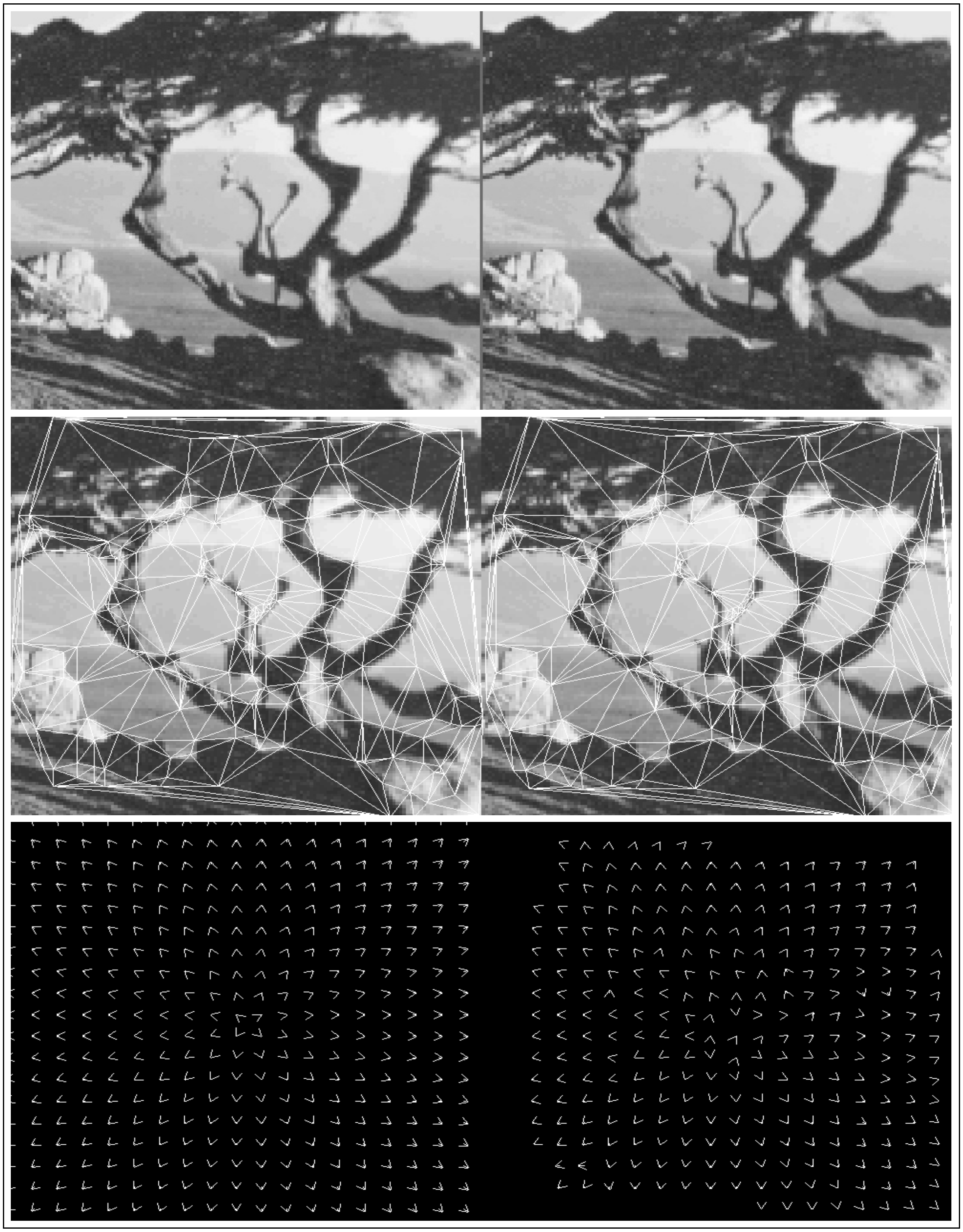


Figure 7: *Top: two frames of the Diverging Tree sequence. Middle: generated triangular meshes. Bottom: the correct flow (left) and the estimated flow (right).*

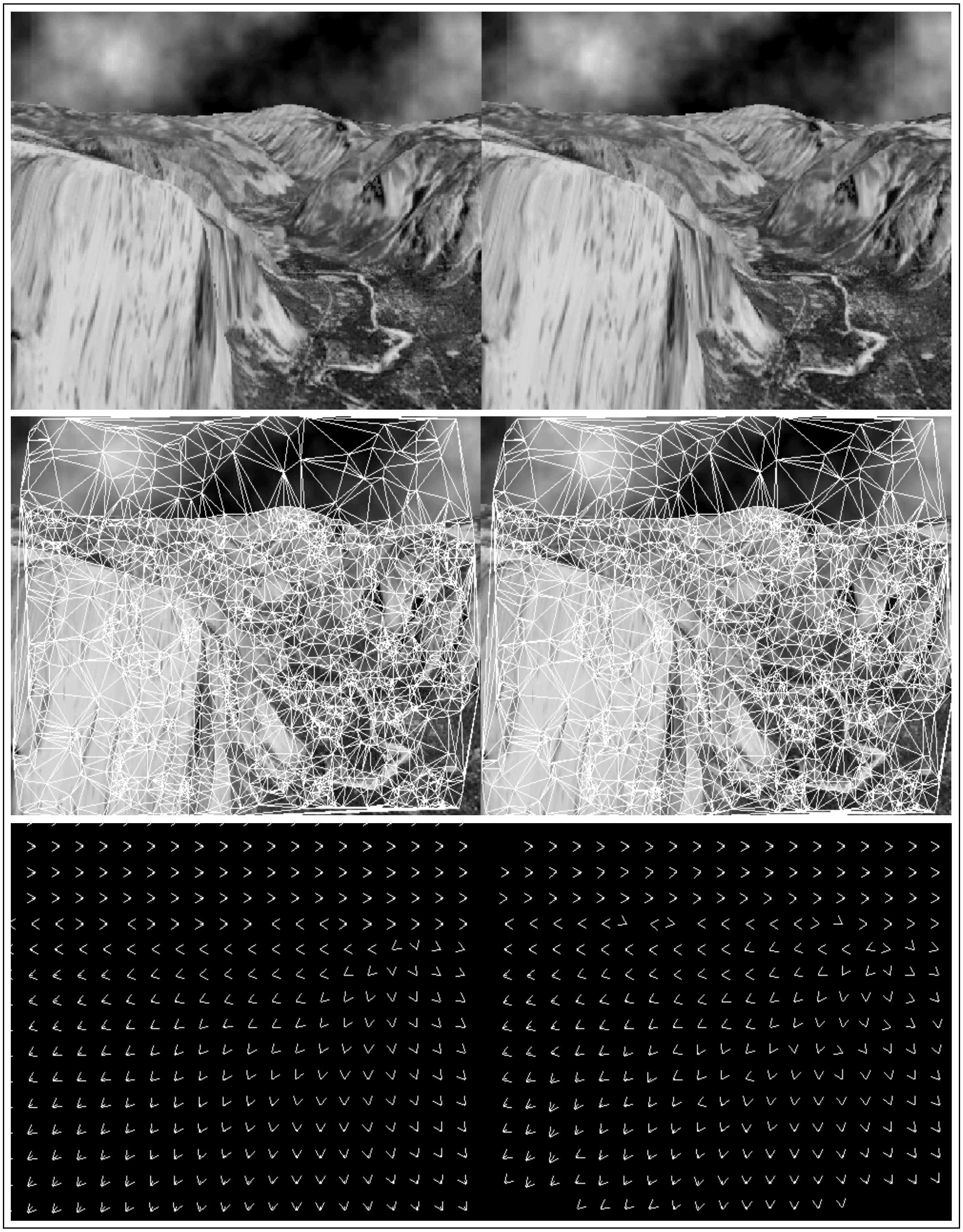


Figure 8: *Top: two frames of the Yosemite sequence. Middle: generated triangular meshes. Bottom: the correct flow (left) and the estimated flow (right).*

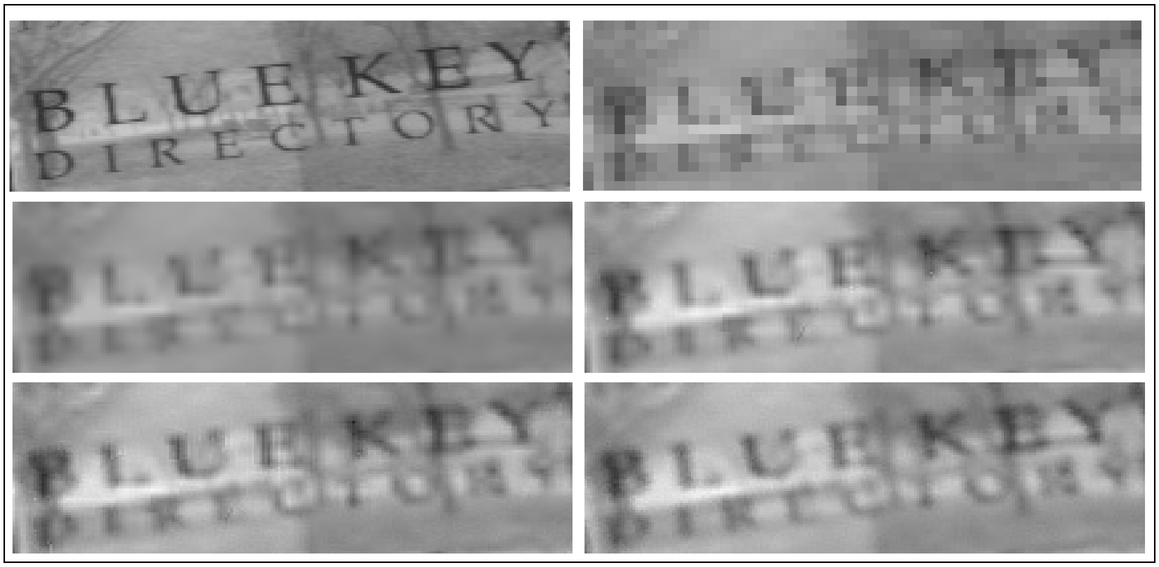


Figure 9: *Top: parts of an original frame (left) and a down-sampled frame (right). Middle: parts of an image interpolated from a single frame (left) and an image reconstructed from 2 frames (right). Bottom: parts of images reconstructed from 4 frames (left) and 16 frames (right).*

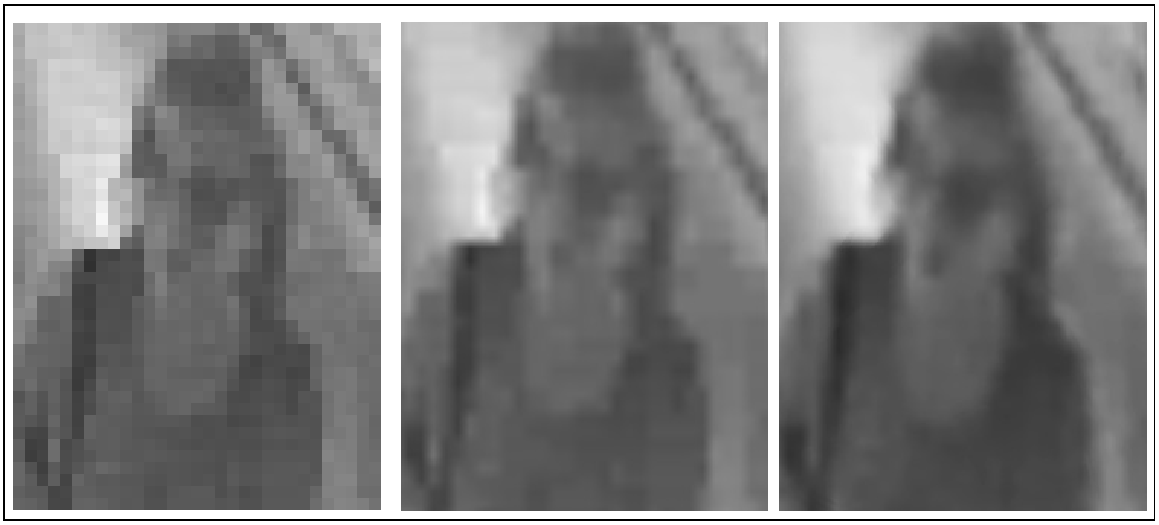


Figure 10: *Left: part of an original frame containing a human face. Center: part of an image interpolated from a single frame. Right: part of an image reconstructed from 4 frames.*