Geodesic Active Contour Based Fusion of Visible and Infrared Video for Persistent Object Tracking

F. Bunyak, K. Palaniappan, S. K. Nath
Department of Computer Science
University of Missouri-Columbia
MO 65211-2060 USA
bunyak,palaniappank,naths@missouri.edu

G. Seetharaman
Dept of Electrical and Computer Engineering
Air Force Institute of Technology
OH 45433-7765 USA
guna.seetharaman@afit.edu

Abstract

Persistent object tracking in complex and adverse environments can be improved by fusing information from multiple sensors and sources. We present a new moving object detection and tracking system that robustly fuses infrared and visible video within a level set framework. We also introduce the concept of the flux tensor as a generalization of the 3D structure tensor for fast and reliable motion detection without eigen-decomposition. The infrared flux tensor provides a coarse segmentation that is less sensitive to illumination variations and shadows. The Beltrami color metric tensor is used to define a color edge stopping function that is fused with the infrared edge stopping function based on the grayscale structure tensor. The min fusion operator combines salient contours in either the visible or infrared video and drives the evolution of the multispectral geodesic active contour to refine the coarse initial flux tensor motion blobs. Multiple objects are tracked using correspondence graphs and a cluster trajectory analysis module that resolves incorrect merge events caused by under- segmentation of neighboring objects or partial and full occlusions. Long-term trajectories for object clusters are estimated using Kalman filtering and watershed segmentation. We have tested the persistent object tracking system for surveillance applications and demonstrate that fusion of visible and infrared video leads to significant improvements for occlusion handling and disambiguating clustered groups of objects.

1 Introduction

Successful application of computational vision algorithms to accomplish a variety of tasks in complex environments requires the fusion of multiple sensor and information sources. Significant developments in micro-optics and micro-electomechanical systems (MEMS), VCSELS, tunable RCLEDS (resonant cavity LEDS), and tunable microbolometers indicate that hyperspectral imaging will rapidly become as ubiquitous as visible and thermal videos are today [13, 22]. On board lidars and radars have been used successfully in unmanned autonomous vehicles, extending their versatility well beyond what was demonstrated in the 1990's based on dynamic scene analysis of visible video only. Most of the autonomous vehicles competing in the recent DARPA Grand Challenge events used one or more lidar sensors to augment the video imagery, demonstrating intelligent navigation using fusion of multiple information sources [17]. Autonomous navigation in city traffic with weather, signals, vehicles, pedestrians, and construction will be more challenging.

Effective performance in persistent tracking of people and objects for navigation, surveillance, or forensic behavior analysis applications require robust capabilities that are scalable to changing environmental conditions and external constraints (ie visibility, camouflage, contraband, security, etc.) [3]. For example, monitoring the barrier around sensitive facilities such as chemical or nuclear plants will require using multiple sensors in addition to a network of (visible) video cameras. Both infrared cameras and laser-scanner based lidar have been used to successfully enhance the overall effectiveness of such systems. In crowds or busy traffic areas even though it may be impractical to monitor and track each person individually, information fusion that characterizes objects of interest can significantly improve throughput. Airport surveillance systems using high resolution infrared/thermal video of people can extract invisible biometric signatures to characterize individuals or tight groups, and use these *short-term* multispectral blob signatures to resolve cluttered regions in difficult video segmentation tasks.

Persistent object detection and tracking are challenging processes due to variations in illumination (particularly in outdoor settings with weather), clutter, noise, and occlusions. In order to mitigate some of these problems and to improve the performance of persistent object tracking, we investi-

IEEE Workshop on Applications of Computer Vision (WACV'07)
0-7695-2794-9/07 $20.00 © 2007 IEEE

gate the fusion of information visible and infrared imagery. Infrared imagery is less sensitive to illumination related problems such as uneven lighting, moving cast shadows or sudden illumination changes (i.e. cloud movements) that cause false detections, missed objects, shape deformations, false merges etc. in visible imagery. But use of infrared imagery alone often results in poor performance since generally these sensors produce imagery with low signal-to-noise ratio, uncalibrated white-black polarity changes, and "halo effect" around hot or cold objects [9]. "Hot spot" techniques that detect moving objects by identifying bright regions in infrared imagery are inadequate in the general case, because the assumption that the objects of interest, people and moving cars are much hotter than the surrounding is not always true.

In this paper, we present a new moving object detection and tracking system for surveillance applications using infrared and visible imagery. The proposed method consists of four main modules, motion detection, object segmentation, tracking, and cluster trajectory analysis, summarized below and elaborated in the following sections.

A coarse *motion detection* is done in the infrared domain using the flux tensor method. A foreground mask $\mathrm{FG_M}$ identifying moving blobs is outputted for each frame. *Object segmentation* refines the obtained mask $\mathrm{FG_M}$, using level set based geodesic active contours with information from visible and infrared imagery. Object clusters are segmented into individual objects, contours are refined, and a new foreground mask $\mathrm{FG_R}$ is produced. *Multi-object tracking module* resolves frame-to-frame correspondences between moving blobs identified in $\mathrm{FG_R}$ and outputs moving object statistics along with trajectories. Lastly, a *cluster trajectory analysis* module combines segments and analyzes trajectories to resolve incorrect trajectory merges caused by undersegmentation of neighboring objects or partial and full occlusions.

2 Motion Detection

Fast motion blob extraction is performed using a novel *flux tensor* method which is proposed as an extension to the 3D grayscale structure tensor. By more effectively using spatio-temporal consistency, both the grayscale structure tensor and the proposed flux tensor produce less noisy and more spatially coherent motion segmentation results in comparison to classical optical flow methods [15]. The flux tensor is more efficient in comparison to the 3D grayscale structure tensor since motion information is more directly incorporated in the flux calculation which is less expensive than computing eigenvalue decompositions as with the 3D grayscale structure tensor.

2.1 3D Structure Tensors

Orientation estimation using structure tensors have been widely used for low-level motion estimation and segmentation [14, 15]. Under the constant illumination model, the optic-flow (OF) equation of a spatiotemporal image volume $\mathbf{I}(\mathbf{x})$ centered at location $\mathbf{x} = [x, y, t]$ is given by Eq. 1 [12] where, $\mathbf{v}(\mathbf{x}) = [v_x, v_y, v_t]$ is the optic-flow vector at $\mathbf{x}$,

$$\begin{aligned} \frac{d\mathbf{I}(\mathbf{x})}{dt} &= \frac{\partial \mathbf{I}(\mathbf{x})}{\partial x}\, v_x + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial y}\, v_y + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial t}\, v_t \\ &= \nabla \mathbf{I}^T(\mathbf{x})\, \mathbf{v}(\mathbf{x}) = 0 \qquad (1) \end{aligned}$$

and $\mathbf{v}(\mathbf{x})$ is estimated by minimizing Eq. 1 over a local 3D image patch $\mathbf{\Omega}(\mathbf{x}, \mathbf{y})$, centered at $\mathbf{x}$. Note that v_t is not 1 since we will be computing spatio-temporal orientation vectors. Using Lagrange multipliers, a corresponding error functional $e_{ls}(\mathbf{x})$ to minimize Eq. 1 using a least-squares error measure can be written as Eq. 2 where $W(\mathbf{x}, \mathbf{y})$ is a *spatially invariant* weighting function (e.g., Gaussian) that emphasizes the image gradients near the central pixel [14].

$$\begin{aligned} e_{ls}(\mathbf{x}) = \int_{\mathbf{\Omega}(\mathbf{x},\mathbf{y})} \left(\nabla \mathbf{I}^T(\mathbf{y})\, \mathbf{v}(\mathbf{x})\right)^2\, W(\mathbf{x}, \mathbf{y})\, d\mathbf{y} \\ +\lambda \left(1 - \mathbf{v}(\mathbf{x})^T \mathbf{v}(\mathbf{x})\right) \qquad (2) \end{aligned}$$

Assuming a constant $\mathbf{v}(\mathbf{x})$ within the neighborhood $\mathbf{\Omega}(\mathbf{x}, \mathbf{y})$ and differentiating $e_{ls}(\mathbf{x})$ to find the minimum, leads to the standard eigenvalue problem for solving $\hat{\mathbf{v}}(\mathbf{x})$ the best estimate of $\mathbf{v}(\mathbf{x})$, $\mathbf{J}(\mathbf{x}, \mathbf{W})\ \hat{\mathbf{v}}(\mathbf{x}) = \lambda\ \hat{\mathbf{v}}(\mathbf{x})$.

The 3D structure tensor matrix $\mathbf{J}(\mathbf{x}, \mathbf{W})$ for the spatiotemporal volume centered at $\mathbf{x}$ can be written in expanded matrix form, without the spatial filter $W(\mathbf{x}, \mathbf{y})$ and the positional terms shown for clarity, as Eq. 3.

$$\mathbf{J} = \begin{bmatrix} \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial x}\frac{\partial \mathbf{I}}{\partial x} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial x}\frac{\partial \mathbf{I}}{\partial y} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial x}\frac{\partial \mathbf{I}}{\partial t} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial y}\frac{\partial \mathbf{I}}{\partial x} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial y}\frac{\partial \mathbf{I}}{\partial y} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial y}\frac{\partial \mathbf{I}}{\partial t} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial t}\frac{\partial \mathbf{I}}{\partial x} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial t}\frac{\partial \mathbf{I}}{\partial y} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial \mathbf{I}}{\partial t}\frac{\partial \mathbf{I}}{\partial t} d\mathbf{y} \end{bmatrix} \qquad (3)$$

The elements of $\mathbf{J}$ (Eq. 3) incorporate information relating to local, spatial, or temporal gradients. A typical approach is to threshold on $\mathbf{trace}(\mathbf{J}) = \int_{\mathbf{\Omega}} ||\nabla I||^2 d\mathbf{y}$ but this fails to capture the nature of these gradient changes, and results in ambiguities in distinguishing responses arising from stationary versus moving features (e.g., edges and junctions with and without motion). Analyzing the eigenvalues and the associated eigenvectors of $\mathbf{J}$ can usually resolve this ambiguity, which can then be used to classify the video regions experiencing motion [16]. However eigenvalue decompositions at every pixel is computationally expensive especially if real time performance is required.

2.2 Flux Tensors

In order to reliably detect only the moving structures *without* performing expensive eigenvalue decompositions, we propose the concept of the *flux tensor*, that is the temporal variations of the optical flow field within the local 3D spatiotemporal volume.

Computing the second derivative of Eq. 1 with respect to t, we obtain Eq. 4 where, $\mathbf{a}(\mathbf{x}) = [a_x, a_y, a_t]$ is the acceleration of the image brightness located at $\mathbf{x}$.

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{d\mathbf{I}(\mathbf{x})}{dt}\right) \quad = \quad & \frac{\partial^2\mathbf{I}(\mathbf{x})}{\partial x \partial t}\, v_x + \frac{\partial^2\mathbf{I}(\mathbf{x})}{\partial y \partial t}\, v_y + \frac{\partial^2\mathbf{I}(\mathbf{x})}{\partial t^2}\, v_t \\ & + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial x}\, a_x + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial y}\, a_y + \frac{\partial \mathbf{I}(\mathbf{x})}{\partial t}\, a_t \end{aligned} \quad (4)$$

which can be written in vector notation as,

$$\frac{\partial}{\partial t}(\nabla\mathbf{I}^T(x)\mathbf{v}(\mathbf{x})) = \frac{\partial \nabla\mathbf{I}^T(\mathbf{x})}{\partial t}\,\mathbf{v}(\mathbf{x}) + \nabla\mathbf{I}^T(\mathbf{x})\,\mathbf{a}(\mathbf{x}) \qquad (5)$$

Using the same approach for deriving the classic 3D structure, minimizing Eq. 4 assuming a constant velocity model and subject to the normalization constraint $||\mathbf{v}(\mathbf{x})|| = 1$ leads to Eq. 6,

$$\begin{aligned} e^F_{ls}(\mathbf{x}) = \int_{\mathbf{\Omega}(\mathbf{x},\mathbf{y})} & \left(\frac{\partial(\nabla\mathbf{I}^T(\mathbf{y})}{\partial t}\,\mathbf{v}(\mathbf{x})\right)^2 W(\mathbf{x},\mathbf{y})\,d\mathbf{y} \\ & + \lambda\left(1 - \mathbf{v}(\mathbf{x})^T\mathbf{v}(\mathbf{x})\right) \end{aligned} \qquad (6)$$

Assuming a constant velocity model in the neighborhood $\mathbf{\Omega}(\mathbf{x},\mathbf{y})$, results in the acceleration experienced by the brightness pattern in the neighborhood $\mathbf{\Omega}(\mathbf{x},\mathbf{y})$ to be zero at every pixel. As with its 3D structure tensor counterpart $\mathbf{J}$ in Eq. 3, the 3D flux tensor $\mathbf{J_F}$ using 6 can be written as $\mathbf{J_F}(\mathbf{x},\mathbf{W}) = \int_{\mathbf{\Omega}} W(\mathbf{x},\mathbf{y})\frac{\partial}{\partial t}\nabla\mathbf{I}(\mathbf{x}) \cdot \frac{\partial}{\partial t}\nabla\mathbf{I^T}(\mathbf{x})d\mathbf{y}$ and in expanded matrix form as Eq. 7.

$$\mathbf{J_F} = \begin{bmatrix} \int_{\mathbf{\Omega}} \left\{\frac{\partial^2\mathbf{I}}{\partial x\partial t}\right\}^2 d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2\mathbf{I}}{\partial x\partial t}\frac{\partial^2\mathbf{I}}{\partial y\partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2\mathbf{I}}{\partial x\partial t}\frac{\partial^2\mathbf{I}}{\partial t^2} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial^2\mathbf{I}}{\partial y\partial t}\frac{\partial^2\mathbf{I}}{\partial x\partial t}\, d\mathbf{y} & \int_{\mathbf{\Omega}} \left\{\frac{\partial^2\mathbf{I}}{\partial y\partial t}\right\}^2 d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2\mathbf{I}}{\partial y\partial t}\frac{\partial^2\mathbf{I}}{\partial t^2} d\mathbf{y} \\ \int_{\mathbf{\Omega}} \frac{\partial^2\mathbf{I}}{\partial t^2}\frac{\partial^2\mathbf{I}}{\partial x\partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \frac{\partial^2\mathbf{I}}{\partial t^2}\frac{\partial^2\mathbf{I}}{\partial y\partial t} d\mathbf{y} & \int_{\mathbf{\Omega}} \left\{\frac{\partial^2\mathbf{I}}{\partial t^2}\right\}^2 d\mathbf{y} \end{bmatrix} \qquad (7)$$

As seen from Eq. 7, the elements of the flux tensor incorporate information about temporal gradient changes which leads to efficient discrimination between stationary and moving image features. Thus the trace of the flux tensor matrix which can be compactly written and computed as, $\mathbf{trace}(\mathbf{J_F}) = \int_{\mathbf{\Omega}} ||\frac{\partial}{\partial t}\nabla\mathbf{I}||^2 d\mathbf{y}$ can be directly used to classify moving and non-moving regions without the need for expensive eigenvalue decompositions. If motion vectors are needed then we can minimize Eq. 6 to get $\hat{\mathbf{v}}(\mathbf{x})$ using $\mathbf{J_F}(\mathbf{x},\mathbf{W})\ \hat{\mathbf{v}}(\mathbf{x}) = \lambda\ \hat{\mathbf{v}}(\mathbf{x})$. In this approach the eigenvectors need to be calculated at just moving feature points.

3 Motion Constrained Object Segmentation

As described in Section 2.2, each pixel in an infrared image frame $\mathbf{I}_{IR}(\boldsymbol{x},t)$ is classified as moving or stationary by thresholding trace of the corresponding flux tensor matrix $(\mathbf{trace}(\mathbf{J_F}))$ and a motion blob mask $\mathrm{FG_M(t)}$ is obtained. This module refines $\mathrm{FG_M(t)}$ by addressing two problems of motion detection: holes and inaccurate object boundaries. Motion detection produces holes inside slow moving homogeneous objects, because of the aperture problem. Motion blobs are larger than the corresponding moving objects, because these regions actually correspond to the union of the moving object locations in the temporal window, rather than the region occupied in the current frame. Beside inaccurate object boundaries this may lead to merging of neighboring object masks and consequently to false trajectory merges and splits at the tracking stage.

In order to refine the coarse $\mathrm{FG_M}$ obtained through flux tensors, we rely on the fusion of multi-spectral image information and motion information, in a level set based geodesic active contours framework. This process is summarized in Algorithm 1 and elaborated in the following sub-sections.

Algorithm 1 Object Segmentation Algorithm

```
Input : Visible image sequence $\mathbf{I}_{RGB}(\boldsymbol{x},t)$, infrared image sequence
        $\mathbf{I}_{IR}(\boldsymbol{x},t)$, foreground mask sequence $\mathrm{FG_M}(\boldsymbol{x},\mathrm{t})$ with $N_M(t)$ regions
Output : Refined foreground (binary) mask sequence $\mathrm{FG_R}(\boldsymbol{x},\mathrm{t})$ with
        $N_R(t)$ regions

for each time $t$ do
    Compute edge indicator functions $g_{IR}(\boldsymbol{x},t)$ and $g_{RGB}(\boldsymbol{x},t)$
        from infrared $\mathbf{I}_{IR}(\boldsymbol{x},t)$ and visible $\mathbf{I}_{RGB}(\boldsymbol{x},t)$ images.
    Fuse $g_{IR}(\boldsymbol{x},t)$ and $g_{RGB}(\boldsymbol{x},t)$ into a single edge indicator function $g_F(\boldsymbol{x},t)$.
    Initialize refined mask, $\mathrm{FG_R(t)} \leftarrow 0$
    Identify disjoint regions $\mathrm{R_i(t)}$ in $\mathrm{FG_M(t)}$ using connected component analysis.
    for each region $\mathrm{R_i}(t)$ $\{i = 1, 2, ...N_M(t)\}$ in $\mathrm{FG_M(t)}$ do
        Fill holes in $\mathrm{R_i}(t)$ using morphological operations.
        Initialize geodesic active contour level sets $\mathcal{C}_i(t)$ using contour of $\mathrm{R_i}(t)$.
        Evolve $\mathcal{C}_i(t)$ using $g_F(t)$ as edge stopping function.
        Check stopping/convergence condition to subpartition $\mathrm{R_i}(t) =$
            $\{\mathrm{R_{i,0}}(t), \mathrm{R_{i,1}}(t), ..., \mathrm{R_{i,N_{R_i}(t)}}(t)\}$ into $N_{R_i}(t) \geq 1$ foreground regions and one background region $\mathrm{R_{i,0}}(t)$ .
        Refine mask $\mathrm{FG_R}$ using foreground partitions as
            $\mathrm{FG_R} = \mathrm{FG_R} \cup \mathrm{R_{i,j}};\ j = 1 : N_{R_i}(t)$
    end for// $N_M(t)$ regions
end for// $T frames$
```

3.1 Fusion of Visible and Infrared Information using Edge Feature Indicator Functions

Contour feature or edge indicator functions are used to guide and stop the evolution of the geodesic active contour when it arrives at the object boundaries. The edge indicator function is a decreasing function of the image gradient that rapidly goes to zero along edges and is higher elsewhere.

The magnitude of the gradient of the infrared image is used to construct an edge indicator function g_{IR} as shown below where $G_\sigma(x,y) * \mathbf{I}_{IR}(x,y)$ is the infrared image smoothed with a Gaussian filter,

$$g_{IR}(\mathbf{I}_{IR}) = exp(-|\nabla G_\sigma(x,y) * \mathbf{I}_{IR}(x,y)|) \tag{8}$$

Although the spatial gradient for single channel images lead to well defined edge operators, the gradient of multi-channel images (i.e. color edge strength) is not straight forward to generalize since gradients in different channels can have inconsistent orientations. We explored the use of several color feature operators and selected the Beltrami color tensor [10] as the best choice based on robustness and speed.

The Beltrami color metric tensor operator for a 2D color image defines a metric on a two-dimensional manifold $\{x, y, R(x,y), G(x,y), B(x,y)\}$ in the five-dimensional spatial-spectral space $\{x, y, R, G, B\}$. The color metric tensor is defined below where $\mathcal{I}_2$ is the 2×2 identity matrix and $\mathbf{J_C}$ is the 2D color structure tensor [10],

$$\mathcal{E} = \mathcal{I}_2 + \mathbf{J_C} \tag{9}$$

$$\mathbf{J_C} = \begin{bmatrix} \displaystyle\sum_{i=R,G,B} \left(\frac{\partial \mathbf{I}_i}{\partial x}\right)^2 & \displaystyle\sum_{i=R,G,B} \frac{\partial \mathbf{I}_i}{\partial x}\frac{\partial \mathbf{I}_i}{\partial y} \\ \displaystyle\sum_{i=R,G,B} \frac{\partial \mathbf{I}_i}{\partial x}\frac{\partial \mathbf{I}_i}{\partial y} & \displaystyle\sum_{i=R,G,B} \left(\frac{\partial \mathbf{I}_i}{\partial y}\right)^2 \end{bmatrix} \tag{10}$$

The determinant of $\mathcal{E}$ is considered as a generalization of the intensity image gradient magnitude to multispectral image gradients. The Beltrami color edge stopping function can then be defined as,

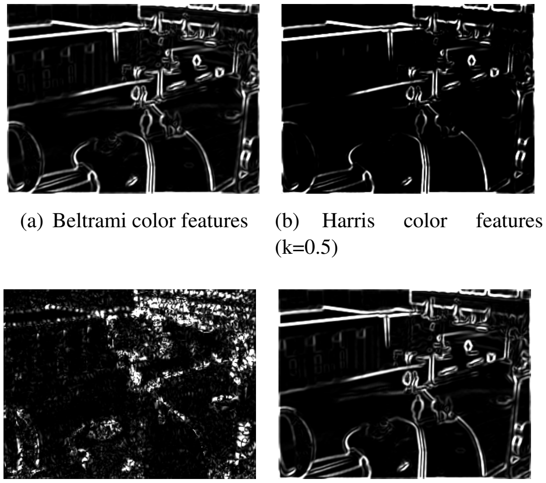
$$g_{RGB}(\mathbf{I}_{RGB}) = exp(-abs(\mathbf{det}(\mathcal{E}))) \tag{11}$$

$$\begin{aligned} \mathbf{det}(\mathcal{E}) &= 1 + \mathbf{trace}(\mathbf{J_C}) + \mathbf{det}(\mathbf{J_C}) \\ &= 1 + (\lambda_1 + \lambda_2) + \lambda_1\lambda_2 \end{aligned} \tag{12}$$

where λ_1 and λ_2 are the eigenvalues of $\mathbf{J_C}$.

Although any robust accurate color edge response function can be used, we found the Beltrami color tensor to be the most suitable for persistent object tracking. Several common color (edge/corner) feature indicator functions were evaluated for comparison purposes, including Harris [11] and Shi-Tomasi operators [18]. The Harris operator (Eq. 13) uses the parameter k to tune edge versus corner responses (i.e. $k \rightarrow 0$ responds primarily to corners).

$$\begin{aligned} H(\mathbf{I_{RGB}}) &= \mathbf{det}(\mathbf{J_C}) - k\ \mathbf{trace^2}(\mathbf{J_C}) \\ &= \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \end{aligned} \tag{13}$$



(a) Beltrami color features

(b) Harris color features (k=0.5)

(c) Shi-Tomasi color features $min(\lambda_1, \lambda_2)$

(d) Modified Shi-Tomasi color features $max(\lambda_1, \lambda_2)$

Figure 1: Color features for frame #1256 obtained using Beltrami, Harris, and Shi-Tomasi operators.

The Shi-Tomasi operator [18] is defined as $min(\lambda_1, \lambda_2)$ above a certain threshold. The Shi-Tomasi operator responds strongly to corners and filters out most edges (since one of the eigenvalues is nearly zero along edges). This is not suitable for a geodesic active contour edge stopping function, so we tested a modified operator $max(\lambda_1, \lambda_2)$ which responds nicely to both edges and corners.

It is interesting to note that all of the above color feature detectors can be related to the eigenvalues of the color structure tensor matrix $\mathbf{J_C}$. Since these values are correlated with the local image properties of edgeness and cornerness i.e. $\lambda_1 >> 0$, $\lambda_2 \approx 0$ or $\lambda_1 \approx \lambda_2 >> 0$ respectively. The best operators for the geodesic active contour edge stopping functions will respond to all salient contours in the image. In our experiments, as shown in Figure 1, the Beltrami color (edge) features was the most suitable function and is fast to compute. The Harris operator misses some salient contours around the pedestrians, the Shi-Tomasi operator responds primarily to corners and is not suitable as an edge stopping function, the modified Shi-Tomasi operator produces a contour map that is nearly the same as the Beltrami color metric tensor map but is slightly more expensive to compute due to the square root calculation for the eigenvalues.

The fused edge indicator function $g_F(x,y)$ should respond to the strongest edge at location (x,y) in either channel. So the fusion operator is defined as the minimum of the two *normalized* $(0,1)$ edge indicator functions $g_{IR}(x,y)$, and $g_{RGB}(x,y)$,

$$g_F(\mathbf{IR}, \mathbf{RGB}, x, y) = min\{g_{IR}(x,y), g_{RGB}(x,y)\} \tag{14}$$

This ensures that the curve evolution stops where there is an edge in the visible imagery or in the infrared imagery. Infrared imagery could have been considered as a fourth channel and the metric tensor in Eq. 10 could have been defined in the six-dimensional spatial-spectral space $\{x, y, R, G, B, IR\}$. But the infrared imagery should have more weight in our decision than any single channel of the visible imagery, since moving infrared edges are highly salient for tracking. In order not to miss any infrared edges, independent of the gradients in the visible channels, the *min* statistic Eq. 14 is used. *min* fusion operator handles cases where the visible RGB appearance of the moving object is similar to the background but there is a distinct infrared signature, and when the background and foreground have similar infrared signatures but distinct appearances.

3.2 Level Set Based Active Contours

Active contours evolve a curve $\mathcal{C}$, subject to constraints from a given image. In level set based active contour methods the curve $\mathcal{C}$ is represented implicitly via a Lipschitz function ϕ by $\mathcal{C} = \{(x, y)|\phi(x, y) = 0\}$, and the evolution of the curve is given by the zero-level curve of the function $\phi(t, x, y)$. Evolving $\mathcal{C}$ in a normal direction with speed F amounts to solving the differential equation [7],

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| F; \quad \phi(0, x, y) = \phi_0(x, y) \tag{15}$$

Unlike parametric approaches such as classical snake, level set based approaches ensure topological flexibility since different topologies of zero level-sets are captured implicitly in the topology of the energy function ϕ. Topological flexibility is crucial for our application since coarse motion segmentation may result in merging of neighboring objects, that need to be separated during the segmentation stage. To refine the coarse motion segmentation results obtained using flux tensors, we use geodesic active contours [6] that are tuned to edge/contour information effectively. The level set function ϕ is initialized with the signed distance function of the motion blob contours ($\mathrm{FG_M}$) and evolved using the geodesic active contour speed function,

$$\frac{\partial \phi}{\partial t} = g_F(\mathbf{I})(c + \mathcal{K}(\phi))|\nabla \phi| + \nabla \phi \cdot \nabla g_F(\mathbf{I}) \tag{16}$$

where $g_F(\mathbf{I})$ is the fused edge stopping function (Eq. 14), c is a constant, and $\mathcal{K}$ is the curvature term,

$$\mathcal{K} = div\left(\frac{\nabla \phi}{|\nabla \phi|}\right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{\frac{3}{2}}} \tag{17}$$

The force $(c + \mathcal{K})$ acts as the internal force in the classical energy based snake model. The constant velocity c pushes the curve inwards or outwards depending on its sign (inwards in our case). The regularization term $\mathcal{K}$ ensures boundary smoothness. The external image dependent force $g_F(\mathbf{I})$ (Section 3.1) is the fused edge indicator function and is used to stop the curve evolution at visible or infrared object boundaries. The term $\nabla g_F \cdot \nabla \phi$ introduced in [6] is used to increase the basin of attraction for evolving the curve to the boundaries of the objects.

Algorithm 2 Tracking Algorithm

Input : Image sequence $\mathbf{I}(\boldsymbol{x}, t)$, and refined foreground mask sequence $\mathrm{FG_R}(\boldsymbol{x}, t)$

Output : Trajectories and Temporal Object Statistics

```
for each frame $\mathbf{I}(\boldsymbol{x}, t)$ at time $t$ do
    Use the refined foreground mask, $\mathrm{FG_R(t)}$ from the motion constrained object segmentation module.
    Partition $\mathrm{FG_R(t)}$ into disjoint regions using connected component analysis $\mathrm{FG_R(t)} = \{\mathrm{R_1(t), R_2(t), \ldots, R_{N_R}(t)}\}$ that ideally correspond to $N_R$ individual moving objects.
    for each region $\mathrm{R_i(t)}$ $\{i = 1, 2, ...N_R(t)\}$ in $\mathrm{FG_R}(t)$ do
        Extract blob centroid, area, bounding box, support map etc.
        Arrange region information in an object correspondence graph $\mathcal{O_{GR}}$. Nodes in the graph represent objects $\mathrm{R_i(t)}$, while edges represent object correspondences.
        Search for potential object matches in consecutive frames using multi-stage overlap distance $\mathcal{D_{MOD}}$.
        Update $\mathcal{O_{GR}}$ by linking nodes that correspond to objects in frame $\mathbf{I}(\boldsymbol{x}, t)$ with nodes of potential corresponding objects in frame $\mathbf{I}(\boldsymbol{x}, t-1)$. Associate the confidence value of each match, $\mathcal{C_M}(i, j)$ with each link.
    end for
end for
Trace links in the object correspondence graph $\mathcal{O_{GR}}$ to generate moving object trajectories.
```

Since the geodesic active contour segmentation relies on edges between background and foreground rather than the color or intensity differences, the method is more stable and robust across very different appearances, non-homogeneous backgrounds and foregrounds. Starting the active contour evolution from the motion segmentation results prevents early stopping of the contour on local non-foreground edges.

4 Multi-object Tracking Using Object Correspondence Graphs

Persistent object tracking is a fundamental step in the analysis of long term behavior of moving objects. The tracking component of our system outlined in Algorithm 2 is an extension of our previous work in [4, 5]. Object-to-object matching (correspondence) is performed using a multi-stage overlap distance $\mathcal{D_{MOD}}$, which consists of three distinct distance functions for three different ranges of object motion as described in [4].

Correspondence information is arranged in an acyclic directed graph $\mathcal{O_{GR}}$. Trajectory-Segments are formed by tracing the links of $\mathcal{O_{GR}}$ and grouping ”inner” nodes that have a single parent and a single child. For each Trajectory-

Algorithm 3 Cluster Trajectory Analysis

```
Input : Merged trajectory segment TM, parent and child trajectory segments $\mathrm{TP(1{:}\,n_p)}$, $\mathrm{TC(1{:}\,n_p)}$.
Output : Individual trajectory segments $\mathrm{TS(1{:}\,n_p)}$, updated parent and child trajectory segments $\mathrm{TP(1{:}\,n_p)}$, $\mathrm{TC(1{:}\,n_p)}$.
Initialize state matrix $\mathbf{X}(t_0)$ consisting of position and velocity informations for each individual trajectory segments $\mathrm{TS(1{:}\,n_p)}$ using the parent segments' states, $\mathbf{X}(t_0) \leftarrow \mathrm{TP(1{:}\,n_p).\,states}$
for each node $\mathrm{TM.\,node(t)}$ of the merged segment do
    Predict using Kalman filter [1] $\hat{\mathbf{X}}(t)$, the estimated state matrix of the trajectory segments $\mathrm{TS(1{:}\,n_p)}$ corresponding to individual objects in the object cluster, from the previous states $\mathbf{X}(t-1)$.
    Project masks of the sub-nodes $\mathrm{TS(1{:}\,n_p).\,node(t\text{-}1)}$ from the previous frame, to the predicted positions on the current merged node $\mathrm{TM.\,node(t)}$, and use these projected masks as markers for the watershed segmentation.
    Using watershed segmentation algorithm and the markers obtained through motion compensated projections of the sub-nodes from the previous frame, segment the merged node $\mathrm{TM.\,node(t)}$ corresponding to a cluster of objects into a set of sub-nodes corresponding to individual objects $\mathrm{TS(i).\,node(t)}, i = 1 : n_p$.
    Use refined positions of the sub-nodes obtained after watershed segmentation to update the corresponding states $\mathbf{X}(t)$.
end for
Update object correspondence graph $\mathcal{O}_{\mathcal{GR}}$ by including sub-node informations such as new support maps, centroids, areas etc.
Update individual trajectory segments's parent and children links ($\mathrm{TS(1{:}\,n_p).\,parents}$,$\mathrm{TS(1{:}\,n_p).\,children}$), parent segments' children links ($\mathrm{TP(1{:}\,n_p).\,children}$), and children segments' parent links ($\mathrm{TC(1{:}\,n_p).\,parents}$) by matching TSs to TPs and TCs.
Propagate parent segments' labels to the associated sub-segments, which are subsequently propagated to their children ($\mathrm{TP(1{:}\,n_p).\,label} \rightarrow \mathrm{TS(1{:}\,n_p).\,label} \rightarrow \mathrm{TC(1{:}\,n_p).\,label}$ ).
```

Segment, parent and children segments are identified and a label is assigned. Segment labels encapsulate connectivity information and are assigned using a method similar to connected component labeling.

4.1 Cluster Trajectory Analysis

Factors such as under-segmentation, group interactions, occlusions result in temporary merging of individual object trajectories. The goal of this module is to resolve these merge-split events where n_p parent trajectory segments TPs, temporarily merge into a single trajectory segment TM, then split into n_c child trajectory segments TCs, and to recover individual object trajectories TS s. Currently we only consider symmetric cases where $n_p = n_c$.

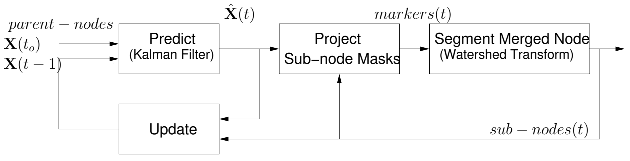


Figure 2: Cluster segmentation using Kalman filter and watershed segmentation.

Most occlusion resolution methods rely heavily on appearance. But for far-view video, elaborate appearance-based models cannot be used since objects are small and not enough support is available for such models. We use prediction and cluster segmentation to recover individual trajectories. Rather than predicting individual trajectories for the merged objects from the parent trajectories alone, at each time step, object clusters are segmented, new measurements are obtained, and object states are updated. This reduces error accumulation particularly for long lasting merges that become more frequent in persistent object tracking.

Segmentation of the object clusters into individual objects is done using a marker-controlled watershed segmentation algorithm applied to the object cluster masks [2, 20]. The use of markers prevents over-segmentation and enables incorporation of segmentation results from the previous frame. The cluster segmentation process is shown in Figure 2 and the overall cluster trajectory analysis process is summarized in Algorithm 3, where $\mathbf{X}$ indicates *state* matrix that consists of a temporal sequence of position and velocity information for each individual object segment, and $\hat{\mathbf{X}}$ indicates estimated *state* matrix.

5 Results and Analysis

The proposed system is tested on thermal/color video sequence pairs from OTCBVS dataset collection [8]. Data consists of 8-bit grayscale bitmap thermal images, and 24-bit color bitmap images of 320 x 240 pixels. Images were sampled approximately at 30Hz. and registered using homography with manually-selected points. Thermal sequences were captured using a Raytheon PalmIR 250D sensor, color sequences were captured using a Sony TRV87 Handycam.

Figure 3 shows different moving object detection results. MoG refers to the background estimation and subtraction method by mixture of Gaussians [19] [21]. Flux refers to the flux tensor method presented in Section 2. The parameters for the mixture of gaussians (MoG) method are selected as follows: number of distributions $K = 4$, distribution match threshold $T_{match} = 2.0$, background threshold $T = 70\%$, learning rate $\alpha = 0.02$. The parameters for the flux tensor method use a neighborhood size $W = 9$, and trace threshold $T = 4$.

Visible imagery (Figure 3c) is very sensitive to moving shadows and illumination changes. Shadows (Figure 3c row 1) can alter object shapes and can result in false detections. Illumination changes due to cloud movements covers a large portion of the ground (Figure 3c row 2) which results in many false moving object detections, making detection and tracking of pedestrians nearly impossible. As can be seen from Figures 3d,e infrared imagery is less sensitive to illumination related problems. But infrared imagery is more

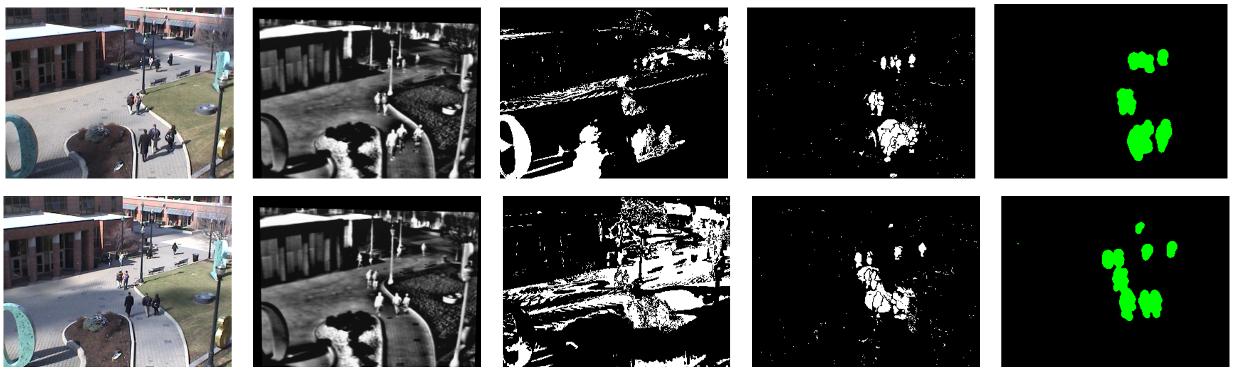

(a) Original image RGB (b) Original image IR (c) MoG on RGB (d) MoG on IR (e) Flux on IR

Figure 3: Moving object detection results for OTCBVS benchmark sequence 3:1. Top row: frame #1048. Bottom row: frame #1256.

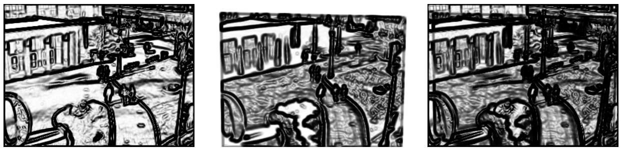

(a) g_{RGB} (b) g_{IR} (c) g_{Fusion}

Figure 4: Edge indicator functions for OTCBVS benchmark sequence 3:1 frame #1256.

noisy compared to visible imagery and suffers from "halo" effects (Figure 3d). The flux tensor method (Figure 3e) produces less noisy and more compact foreground masks compared to pixel based background subtraction methods such as MoG (Figure 3d), since it integrates temporal information from isotropic spatial neighborhoods.

Figure 4 shows visible,IR and fused edge indicator functions used in the segmentation process. Figure 5 illustrates effects of contour refinement and fusion of visible and infrared information. Level set based geodesic active contours refine object boundaries and segment object clusters into individual objects or smaller clusters which is critical for persistent object tracking. When used alone, both visible and infrared video result in total or partial loss of moving objects (i.e. top left person in Figure 5b due to low color contrast compared to background, parts of top right person and legs in Figure 5c due to lack of infrared edges). A low level fusion of the edge indicator functions shown in Figure 5d results in a more complete mask, compared to just combining visible and infrared foreground masks (i.e. legs of top right and bottom persons).

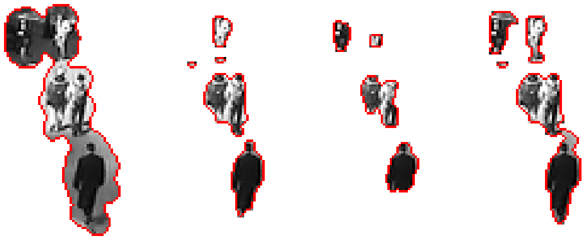
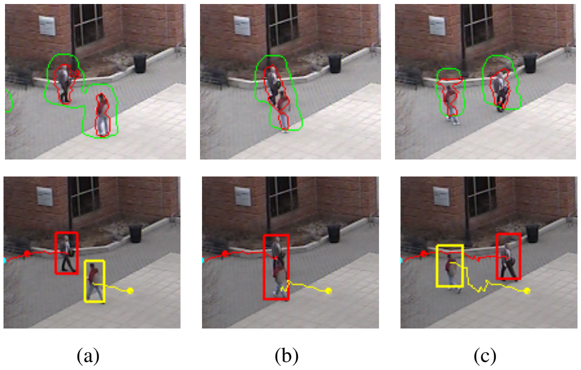

(a) IR Flux (b) Visible (c) Infrared (d) Fusion

Figure 5: (a) Motion blob #2 in frame #1256 using IR flux tensors. Refinement of blob #2 using (b) only visible imagery, (c) only infrared imagery, (d) using fusion of both visible and IR imagery.

Figure 6 illustrates the effects of contour refinement and merge resolution on object trajectories. Level set based geodesic active contours can separate clusters caused by under-segmentation (Figure 6a) but cannot segment individual objects during occlusions (Figure 6b). In those cases merge resolution recovers individual trajectories using prediction and previous object states (Figure 6 second row). In occlusion events no single parameter (i.e. color, size, shape etc.) can consistently resolve ambiguities in partitioning as evident in Figure 6b first row.

6 Conclusion and Future Work

In this paper we presented a moving object detection and tracking system based on the fusion of infrared and visible imagery for persistent object tracking. Outdoor surveillance applications require robust systems due to wide area coverage, shadows, cloud movements and background activity. The proposed system fuses the information from both visible and infrared imagery within a geodesic active contour framework to achieve this robustness.



(a) (b) (c)

Figure 6: Merge resolution. Left to right: frames #41, #56, and #91 in OTCBVS benchmark sequence 3:4. Top row: motion constrained object extraction results. Flux tensor results are marked in green, refined contours are marked in red. Bottom row: object trajectories after merge resolution.

Our novel flux tensor method successfully segments the coarse motion in the infrared image and produces less noisy and more spatially coherent results compared to classical pixel based background subtraction methods. This motion information serves as an initial mask that is further refined by using level set based geodesic active contours and the fused edges from both infrared and visible imagery. As shown in the experimental results, this improves the segmentation of clustered moving objects.

After the masks are obtained, the tracking module resolves frame-to-frame correspondences between moving blobs and produces moving object statistics along with trajectories. Lastly, the cluster trajectory analysis module analyzes trajectories to resolve incorrect trajectory merges caused by under-segmentation of neighboring objects or partial and full occlusions. The current results show promising performance by fusing multi-spectral information to improve object segmentation. We are currently working on the cluster trajectory analysis module and we will test our system with more sequences to report a complete performance evaluation.

References

[1] Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. John Wiley & Sons, Inc., 2001.

[2] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. In E. Dougherty, editor, *Mathematical Morphology and its Applications to Image Processing*, pages 433–481. Marcel Dekker, New York, 1993.

[3] W. Brown, R. Kaehr, and D. Chelette. Finding and tracking targets: Long term challenges. *Air Force Research Technology Horizons*, 5(1):9–11, 2004.

[4] F. Bunyak, K. Palaniappan, S. K. Nath, T. Baskin, and G. Dong. Quantitive cell motility for *in vitro* wound healing using level set-based active contour tracking. In *Proc. 3^{rd} IEEE Int. Symp. Biomed. Imaging (ISBI)*, pages 1040–1043. Arlington, VA, April 2006.

[5] F. Bunyak and S. R. Subramanya. Maintaining trajectories of salient objects for robust visual tracking. In *LNCS-3212: Proc. ICIAR'05*, pages 820–827, Toronto, Sep. 2005.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. Journal of Computer Vision*, 22(1):61–79, 1997.

[7] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Proc.*, 10(2):266–277, Feb. 2001.

[8] J. Davis and V. Sharma. Fusion-based background-subtraction using contour saliency. In *IEEE Int. Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, San Diego, CA, June 2005.

[9] J. Davis and V. Sharma. Background-subtraction in thermal imagery using contour saliency. *Int. Journal of Computer Vision*, 71(2):161–181, 2007.

[10] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Trans. Image Proc.*, 10(10):1467–1475, Oct 2001.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf.*, volume 15, pages 147–151, Manchester, 1988.

[12] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, Aug. 1981.

[13] Z. Lemnios and J. Zolper. Informatics: An opportunity for microelectronics innovation in the next decade. *IEEE Circuit & Devices*, 22(1):16–22, Jan. 2006.

[14] H. Nagel and A. Gehrke. Spatiotemporally adaptive estimation and segmentation of OF-Fields. In *LNCS-1407: ECCV98*, volume 2, pages 86–102, Germany, June 1998.

[15] S. Nath and K. Palaniappan. Adaptive robust structure tensors for orientation estimation and image segmentation. In *LNCS-3804: Proc. ISVC'05*, pages 445–453, Lake Tahoe, Nevada, Dec. 2005.

[16] K. Palaniappan, H. Jiang, and T. I. Baskin. Non-rigid motion estimation using the robust tensor method. In *IEEE Comp. Vision and Pattern Recog. Workshop on Articulated and Non-rigid Motion*, Washington, DC, June 2004.

[17] G. Seetharaman, A. Lakhotia, and E. Blasch. Unmanned vehicles come of age: The darpa grand challenge. *Special Issue of IEEE Computer*, pages 32–35, Dec. 2006.

[18] J. Shi and C. Tomasi. Good features to track. In *IEEE Conf. on Comp. Vis. and Patt. Recog.(CVPR)*, Seattle, June 1994.

[19] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. and Machine Intell.*, 22(8):747–757, 2000.

[20] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(6):583–598, 1991.

[21] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao. Gaussian mixture density modeling, decomposition and applications. *IEEE Trans. Image Proc.*, 5(9):1293–1302, Sep. 1996.

[22] J. Zolper. Integrated microsystems: A revolution on five frontiers. In *Proc. of the 24th Darpa-Tech Conf.*, Anahiem, CA., Aug. 2005.