An iterative integrated framework for thermal-visible image registration, sensor fusion, and people tracking for video surveillance applications

Atousa Torabi*[,a], Guillaume Massé[a], Guillaume-Alexandre Bilodeau[a]

[a]*LITIV laboratory, Department of Computer and Software Engineering,*
École Polytechnique de Montréal, P.O. Box 6079, Station Centre-ville, Montréal
(Québec), Canada, H3C 3A7

Abstract

In this work, we propose a new integrated framework that addresses the problems of thermal-visible video registration, sensor fusion, and people tracking for far-range videos. The video registration is based on a RANSAC trajectory-to-trajectory matching, which estimates an affine transformation matrix that maximizes the overlapping of thermal and visible foreground pixels. Sensor fusion uses the aligned images to compute sum-rule silhouettes, and then constructs thermal-visible object models. Finally, multiple object tracking uses blobs constructed in sensor fusion to output the trajectories. Results demonstrate the advantage of our proposed framework in obtaining better results for both image registration and tracking than separate image registration and tracking methods.

Key words: Thermal camera, visible camera, thermal-visible image registration, sensor fusion, multiple people tracking

*Corresponding author
Email addresses: `atousa.torabi@polymtl.ca` (Atousa Torabi), `guillaume.masse@polymtl.ca` (Guillaume Massé), `guillaume-alexandre.bilodeau@polymtl.ca` (Guillaume-Alexandre Bilodeau)

1. Introduction

In the recent years, there has been a growing interest in visual surveillance using multimodal sensors, such as thermal and visible cameras in both civilian and military applications. Zhu and Huang give a comprehensive introduction about multimodal surveillance systems in [1]. The advantages of jointly using a thermal camera and a visible camera have been studied and discussed extensively in some few works such as [1, 2]. Two main benefits of the joint use of thermal and visible sensors are first the complementary nature of different modalities that provides the thermal and color information of the scene and second, the redundancy of information captured by the different modalities, which increases the reliability and robustness of a surveillance system. These advantages motivated the computer vision community to study and investigate algorithms for thermal-visible video surveillance systems.

For approximately planar far-range videos at different zoom settings, where extracting low level features inside ROIs are difficult due the small size of objects, using the spatio-temporal information of the scene, such as object trajectories and performing sequence-to-sequence matching rather than low level image-to-image matching is an interesting solution. In Caspi *et al.*, a feature-based video sequence-to-sequence matching technique is proposed based on matching object trajectory points [3]. However, trajectory-based matching involves another problem, which is computing trajectories of moving objects in the scene for a pair of video sequences. Since the features to match are trajectory points, the accuracy of computed trajectories in both thermal and visible video has a crucial effect on the image registration result.

In our previous work [4, 5], we proposed trajectory-based sequence-to-sequence video registration, where the object trajectories were computed separately offline for thermal and color video sequences using multiple object tracking, but with an improved trajectory matching that uses foreground pixel overlapping as well as trajectory point matching as registration criteria. In [4, 5], the image registration is similar to the one we used in this paper; however, since the trajectories were estimated separately from tracking using data of a single modality, some trajectories (registration input data) were inaccurate and disconnected. Furthermore, the foreground pixel overlapping criterion could be misleading for some video frames due to the background subtraction errors. In this paper, we address the problem of image registration and object tracking in a novel integrated framework with the final goal of improving both registration and tracking. We propose an iterative, integrated, thermal-visible video registration, sensor fusion, and multimodal tracking for two synchronized streams of long-range videos recorded by collocated visible and thermal cameras at different zoom settings. For our proposed methods, no camera calibration is needed. The only assumption is the intersection of field of view between thermal and visible cameras. In this paper, we mainly focus on a feedback scheme and collaboration between the three modules of our system (image registration, sensor fusion, and tracking), but we also suggest a fusion score computed in the sensor fusion module of our system as an improved registration criterion.

Contribution. Our proposed integrated framework improves both registration and tracking by providing better quality for their input data. Thermal-visible sensor fusion improves the input data for tracking in thermal and visible videos, which results in more accurate object trajectories. Using accurate trajectories as registration input data results in more accurate image registration. In our experiments, we show that our proposed framework outperforms similar image registration methods previously proposed in the-state-of-the-art [3, 5]. Also, we propose a new transformation matrix selection method based on the fusion scores computed in our sensor fusion step. The algorithms presented in this manuscript are based on [6], but they are further developed with detailed analysis and new evaluations.

In the remainder of this paper, we present some background (section 2), then the architecture of the whole system (section 3), followed by a description of our image registration, sensor fusion, and tracking (sections 4, 5, and 6). Then, we discuss the performance of our proposed method (section 7). Finally, we conclude our paper (section 8).

2. Related works

Despite the advantages of multimodal surveillance systems, jointly using two sensors of different modalities increases the complexity of a surveillance system and raises new problems such as image registration and multimodal data fusion. Several works are related to algorithms for thermal-visible data fusion. Conaire *et al.* compared the various fusion methods by evaluating the tracking performance of systems using different fusion methods for aligned pairs of images [7]. Their image alignment is done by estimating the optimum planar homography using a manual process and then warping the thermal images. Also Sadjadi gave a comparative analysis of various fusion methods by proposing a set of measures to study directly their performance [8]. Furthermore, Conaire *et al.* proposed a framework that performs data fusion and tracking in one integrated system [9]. In their framework, data fusion is based on fusing the output of multiple spatiogram trackers. In another work, Kumar *et al.* proposed a multimodal object detection based on fusion of blobs in thermal and visible foreground images [10]. Their method addresses the problem of uncertainty in object detection for dynamic environment such as outdoor scenes. Their fusion method is based on a feedback scheme that performs a simple blob matching between fuse blobs in the previous frame and blobs detected individually in the current thermal and visible frames, followed by a belief fusion that determines the validity of foreground regions detected for each modality and a Kalman filter fusion method. However, in their method, they did not address the problem of object tracking (tracking is based on a simple blob matching) and image registration.

Moreover, a number of works have been published on computer vision methods appropriate for thermal-visible video surveillance applications including background subtraction, object detection [11, 12], multi-pedestrian tracking, and classification [13, 14, 9, 15]. In the works mentioned above, especially the ones designed for approximately planar far-range scenes [10, 9], the problem of automatic video registration is not studied. However, in thermal-visible video surveillance applications, where the thermal and visible videos are captured by two synchronized cameras with different lenses or zooms and with different FOVs, the primary problem before data fusion or any further analyses is automatic image registration. Due to the numerous differences in imaging characteristics of thermal and visible cameras, finding appropriate correspondence measure for matching multimodal images is challenging. Most methods used for registering images of single imaging modality are not applicable. It is also very difficult to find correspondence for an entire scene.

In the literature, some works have been proposed on multimodal image registration for various computer vision applications. Krotosky and Trivedi give a comparative analysis of multimodal image registration methods [16]. Most of these works address the image registration problem as a low-level image-to-image feature-based matching problem. In this approach, image features are first extracted and then a matching is done between the dense or sparse extracted features of a pair of images. For example, Irani *et al.* proposed an image registration method by which local correlation values of the features extracted from a Gaussian pyramid of visible and thermal images are computed, and a global alignment using an iterative Newtonian method is performed [17]. In Coiras *et al.*, image registration is estimated from an affine transformation that maximizes the global edge-formed triangle matching [18]. In Han *et al.*, a hierarchical genetic algorithm-based method is applied for matching the human silhouette in thermal and visible images using two pairs of corresponding points of a human walking on a straight line at a fixed distance from the camera [19]. In these methods, the quality of image alignment is limited to the quality of low-level image feature extraction. Especially for far-range scene people monitoring, extracting features inside blobs is more difficult because blobs are small. Therefore, low-level feature extraction is quite problematic. The other image-to-image matching approach for thermal-visible image registration is the dense stereo correspondence method which is basically a scanline- search box matching followed by a dense disparity map estimation based on the winner takes all (WTA) approach. For example, in Krotosky and Trivedi work, a mutual information (MI) based image registration method is proposed for calibrated pair of thermal and visible images in a close range scene [16]. The robustness of this method is limited by MI window sizes that are needed to be large enough to sufficiently populate the joint probability histogram of MI computation. For far-range people monitoring applications, this

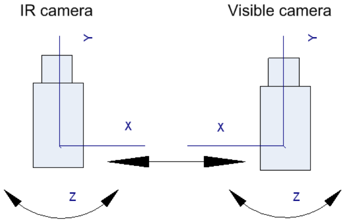


Figure 1: Camera setup

assumption is usually not satisfied due to the small size of blobs and lack of details of patterns inside blobs. Moreover, a simpler camera setup that does not need further pre-processing such as multimodal calibration is desirable.

3. Overview of methods

The input data of the system are synchronized video streams captured by a thermal and a visible camera that are collocated with intersecting fields of view (FOVs) at different zoom settings. We assume that the scene is planar, which means that difference of the distances of moving objects in the scene are much smaller than the distance of the scene from the camera. Fig. 1 shows the camera setup. Cameras can rotate around the *z*-axis and move along the *x*-axis and *y*-axis relative to each other. The only requirement is the intersection of fields of view of the two cameras.

The input data of our system at each frame are pair of thermal and visible foreground images. We apply the background subtraction background method proposed by [20] to separate the foreground pixels from the background. Any reasonable background subtraction method with a fair number of false negative and false positive foreground pixels may be used. Fig. 2 shows the flowchart of our algorithm, which consists of two stages: 1) initialization; and 2) the main loop for image registration, sensor fusion, and tracking. Initialization is performed at the beginning of the videos, where, for some frames, tracking is performed separately for the thermal and the visible video frames until we obtain enough object trajectory points in the scene to estimate a good transformation matrix. The second part of the algorithm consists of a loop on pairs of thermal and visible video frames, where image registration, sensor fusion, and thermal-visible tracking are performed respectively. The image registration estimates an affine transformation matrix, which is used to transform one image into the coordinates of the second one. The sensor fusion matches the color and thermal pixels of blobs using this transformation matrix, and combines thermal and color information. At this step, the matching quality of the computed blobs is also evaluated to decide whether a new transformation matrix should be estimated or if it should be skipped at the next frame. Finally, tracking is performed for thermal and visible videos using fuse blobs obtained from the sensor fusion. These new trajectory points will be used for image registration computation at the next frame.

4. Thermal-visible image registration

At the beginning of the videos, a few trajectory points that are not collinear are required to compute a reasonable initial estimate of the transformation matrix that will be used for sensor fusion. For a fixed number of frames, tracking is performed separately in thermal and visible videos. Then, videos are registered and the overlapping error (Eq. 3) is computed. The registration is repeated until reaching a frame for which the overlapping error is less than a fixed threshold, to ensure the acceptable quality of image alignment required for sensor fusion. The number of initialization frames is subject to change from one video sequence to

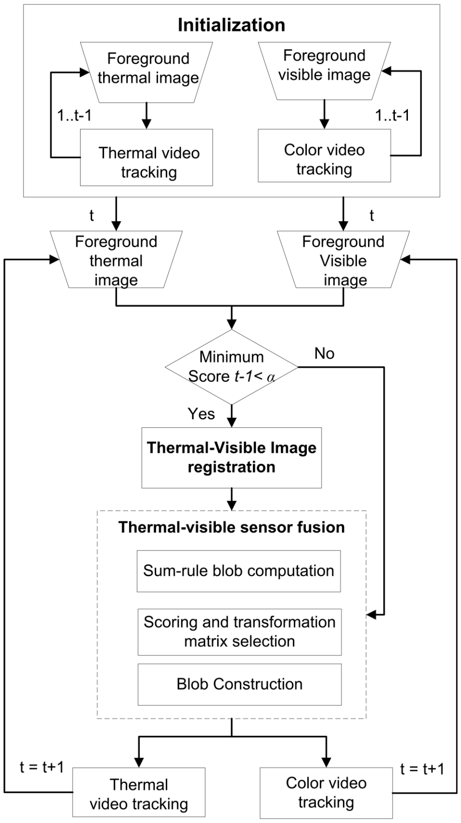


Figure 2: Flowchart of our system

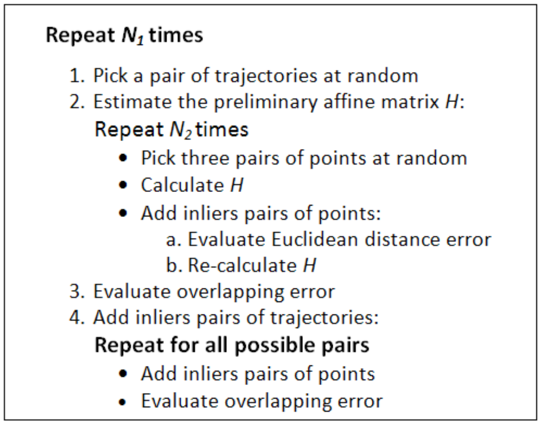
another, based on the frame rate of the video, the trajectory pattern of the moving objects in the scene, and the number of the people walking in the FOV of the cameras at the beginning of the video.

Image registration is performed by aligning the thermal and color images using an affine transformation matrix H [21] computed by matching object trajectory pairs and point pairs from thermal and visible videos. Points are matched using a RANSAC-based algorithm. Our RANSAC-based method is based on matching randomly selected points on the object trajectories of synchronized thermal and visible videos, and finding the best matching points. The affine transformation matrix H is estimated using the normalized Direct Linear Transform (DLT) method [21] to find the least squares solution.

A pair of trajectories is composed of a trajectory from the thermal video and another from the visible video. For example, at frame t, if there are three trajectories for thermal video (T^1_{left}, T^2_{left} and T^3_{left}) and if there are two trajectories for visible video (T^1_{right} and T^2_{right}), then we have six pairs of trajectories that are used as the data pool for the RANSAC algorithm. We used the top-most point position of the human silhouette during tracking to construct a trajectory, since it is less sensitive to shadows on the floor that are falsely detected as part of the human silhouette. Fig. 4 shows matching trajectory points of a pair of trajectories.

Since the videos are synchronized, a pair of corresponding trajectory points in a trajectory pair is a pair of points with the same time stamp. Matching a possible pair of points with the same time stamp, instead of all the points, reduces the combinatorial complexity of the matching problem considerably.

Our RANSAC algorithm is a non deterministic iterative algorithm that estimates the transformation matrix based on the matching of object trajectory points



```
Repeat $N_1$ times
  1. Pick a pair of trajectories at random
  2. Estimate the preliminary affine matrix $H$:
     Repeat $N_2$ times
       • Pick three pairs of points at random
       • Calculate $H$
       • Add inliers pairs of points:
           a. Evaluate Euclidean distance error
           b. Re-calculate $H$
  3. Evaluate overlapping error
  4. Add inliers pairs of trajectories:
     Repeat for all possible pairs
       • Add inliers pairs of points
       • Evaluate overlapping error
```

Figure 3: RANSAC-based algorithm for trajectory point matching

from a pair of thermal and visible videos. Fig. 3 shows the steps of our object trajectory point matching. It is composed of two RANSAC loops, one for the pairs of trajectories with N_1 iterations, and one for the pairs of points in a selected pair of trajectories with N_2 iterations. The number of iterations N is computed with

$$N = \frac{log(1-p)}{log(1-(1-\varepsilon)^s)}, \tag{1}$$

where p is the confidence (in our experiments p is 0.99) and s is the minimum number of points required for the homography (e.g. $s = 3$ for affine transformation). ε, the probability of outliers, is computed by

$$\varepsilon = 1 - \frac{N_p}{N_t}, \tag{2}$$

where N_p is the number of inlier pairs of points/trajectories and N_t is the total number of pairs of points/trajectories. In fact, the number of iterations depends on the number of inlier pairs of points/trajectories. The larger the number of inlier pairs, the less iteration is required. In our algorithm (Fig. 3), N_1 and N_2 are determined by Eq. 1 and 2.

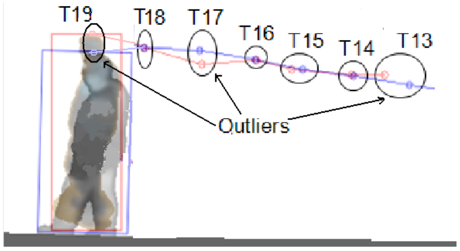


Figure 4: Matching trajectory points from thermal and visible video. $T14$, $T15$, $T16$, $T18$, and $T19$ are inliers.

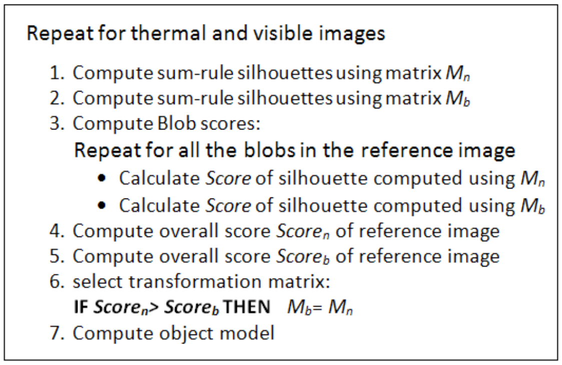
H is calculated using three pairs of points selected at random. After that, all the points of the trajectory of the thermal video frame are transformed using the estimated H. Then, the Euclidean distance between these transformed points, and their corresponding points in the visible video are computed. Pairs of points for which the Euclidean distance is smaller than a threshold T (typically, $T = 5$ pixels) are considered as inlier pairs. The best estimation of H is that computed with the largest number of inlier pairs of points. H is re-estimated using all the inliers pairs of points. Fig. 4 illustrates the matching of selected pairs of trajectory points.

After the first estimation of the transformation matrix H, its quality is evaluated using an overlapping error function OE defined for the foreground pixels of the pairs of thermal and visible video frames.

$$OE = 1 - \frac{N_{c \cap t}}{N_{c \cup t}}, \tag{3}$$

where $N_{c \cap t}$ is the number of overlapping foreground color and thermal image pixels, and $N_{c \cup t}$ is the number of foreground pixels from the union of the color and thermal images. Evaluating the overlapping error allows our method to perform, even when there are few trajectories in the scene.

For each possible pair of trajectories, the thermal image trajectory points are



```
Repeat for thermal and visible images
  1. Compute sum-rule silhouettes using matrix $M_n$
  2. Compute sum-rule silhouettes using matrix $M_b$
  3. Compute Blob scores:
     Repeat for all the blobs in the reference image
       • Calculate Score of silhouette computed using $M_n$
       • Calculate Score of silhouette computed using $M_b$
  4. Compute overall score $Score_n$ of reference image
  5. Compute overall score $Score_b$ of reference image
  6. select transformation matrix:
     IF $Score_n > Score_b$ THEN  $M_b = M_n$
  7. Compute object model
```

Figure 5: Our sensor fusion algorithm

transformed into visible image coordinates, and then the inlier pairs of points are selected using Eq. 3. Using all inlier points, the H matrix is recalculated. Then, the overlapping error is computed for the new estimated matrix H. If the overlapping error for the new estimated matrix is less than the overlapping error of the previous estimation of H, the pair of trajectories is added to the set of inlier pairs of trajectories. This procedure is continued until all the possible pairs of trajectories have been evaluated.

5. Thermal-visible sensor fusion

Thermal-visible sensor fusion combines the information of the registered color and thermal foreground images. Fig. 5 shows our sensor fusion algorithm. M_n represents the transformation matrix estimated by image registration in the current frame, and M_b represents the current best matrix. If the image registration is not performed in the current frame, computations related to M_n shown in 5 are simply skipped.

In this work, a silhouette is defined as a binary object region, and a sum-rule silhouette is defined as a silhouette constructed using a sum of probabilities of foreground pixels in thermal and visible images. To compute a sum-rule silhouette, either foreground pixel coordinates of the thermal image should be transformed into visible image coordinates, or vice versa. Using either method, the computed sum-rule silhouette is the same. The sum-rule method was proposed by [22], and is defined as

$$(X,Y) \in S : \textbf{IF}\ P(S \mid t(X,Y)) + P(S \mid c(X,Y)) > \alpha_{sum}, \tag{4}$$

where $t(X,Y)$ represents the thermal value at image coordinates (X,Y), $c(X,Y)$ represents the color value at image coordinates (X,Y) after transformation, S represents the sum-rule silhouette, and α_{sum} represents a threshold. The probabilities that a pixel belongs to the foreground in each sensor are computed as

$$P(S|t(X,Y)) = 1 - e^{\|t(X,Y)-\mu_t(X,Y)\|^2} \tag{5}$$

where $\mu_t(X,Y)$ is the mean background value of the coordinates (X,Y) for the thermal. $P(S|c(X,Y))$ is computed similarly for transformed visible image. The quality of a sum-rule silhouette is evaluated using a score function. A transformation matrix is selected, based on the scoring results of all the silhouettes inside one image. The score function for the thermal image is defined as follows:

$$SF_t(i) = \frac{sum\left(B^t_{j\in\{1,...n\}} \cap S^t_i\right)}{sum\left(B^t_{j\in\{1,...n\}}\right)}, i \in \{1,...,m\} \tag{6}$$

where m is the number of computed sum-rule silhouettes inside the intersecting FOVs of the two cameras, S^t_i represents the i^{th} sum-rule silhouette computed in the thermal image, $SF_t(i)$ represents its score, and B^t_j are blobs in the original thermal foreground image that intersect with S_i^t. Since background subtraction is not perfect, object regions might be fragmented into smaller ones in the original foreground image. So, the blobs B_j^t that intersect S_i^t should all be fragments belonging to one object. If all blobs B_j^t are inside S_i^t, then S_i^t is perfectly aligned and its score will be 1 (the maximum value). The same applies for visible images for computation of score function in visible $SF_c(i)$. The score of matrix M_n for one image is,

$$Score_n = \left\{ \frac{\sum_{i=1}^{m} (SF_c(i) + SF_t(i))}{2 \times m} \right\}_{M_n} \tag{7}$$

where m is the number of sum-rule silhouettes, $Score_n$ is the score of matrix M_n.The $Score_b$ (the score of matrix M_b for one image) is computed similarly using matrix M_b. Finally, if the score $Score_n$ of the new estimated matrix is higher than the score $Score_b$ of the best matrix, M_n replaces M_b.

Blobs are also constructed. In our work, a blob is defined as all the pixels (either connected or disconnected) with their visual features that belong to one object in an image. Blobs are the input data of tracking step. The sensor fusion improves the quality of input data by computing a sum-rule silhouette that handles the shortcomings of the background subtraction using a single sensor, such as blob fragmentation. Furthermore, sensor fusion provides the color and thermal information of the blob pixels that are used as features for tracking. For blob construction, if the score of a sum-rule silhouette (Eq. 6) is maximum which is 1, the sum-rule silhouette will be considered as a detected blob in the reference image. Otherwise, the original blob's fragments computed by background subtraction that intersect with the computed sum-rule silhouette will be clustered as one blob. In this way, the fragmentation problem is solved.

6. Multiple people tracking method

The object model used in our tracking method is the color-thermal histogram of the input blobs. This histogram has 54 bins for the HSV colors and 16 bins for the thermal intensities. For tracking, any method that computes and updates the trajectory of the objects frame by frame is applicable. Here, we use an online Multiple Hypothesis Tracking (MHT) method, which we proposed in previous work [23]. Our tracking method identifies objects at each frame and estimates the best trajectories computed up to the current frame. In our previous work [23], the tracking was performed only for videos captured by a single visible camera. Therefore, we presented a method for handling blob fragmentation that used the spatial and temporal characteristics of blobs for a few frames, in order to reattach the blob fragments belonging to one object. In this work, instead of this fragmentation handling method, we applied data fusion, which combines the information from the thermal and color videos and improves the quality of the input data for tracking, and, consequently, improves the tracking results considerably. Tracking is performed separately for thermal and visible videos using constructed blobs with thermal-visible histogram as tracking feature.

Our tracking algorithm has three main steps that are described in the following sections. We use two graphs for tracking: an event graph to record all blob's events and store their appearance information while they are being tracked, and a hypothesis graph to generate hypotheses for handling data association of split objects.

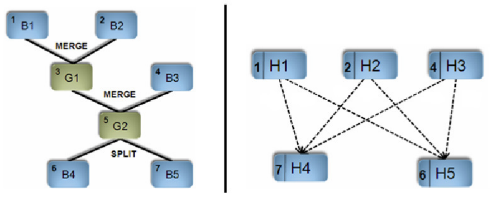


Figure 6: Event graph (left) and hypothesis graph (right). In the hypothesis graph, the number on the left of each hypothesis node corresponds to a track node in the event graph, with the corresponding number in the upper left corner.

6.1. Definition of event graph and hypothesis graph

Fig. 6 shows an event graph with its corresponding hypothesis graph. The event graph represents all blobs with their merging and splitting events during tracking. Each vertex of this graph (track node) stores a blob's appearance, including top-most point coordinates, its adaptive thermal-color histogram, blob events such as correspondence, merging, and splitting, and the frame number of the last update in the node. Edges represent merging and splitting events among the blobs. The hypothesis graph is a directed, weighted graph. The vertices of this graph (hypothesis nodes) simply correspond to the track nodes of the event graph that belong to entering blobs (blobs that appear in the scene) and split blobs (blobs that break away from a group, or a single blob). A group blob does not have hypothesis nodes. This is because these nodes are used to solve the data association problem before and after object interactions. The weight of each edge n_in_j that represents a hypothesis is defined as,

$$\omega\left(n_in_j\right)=\left|AH\left(n_i\right)-AH\left(n_j\right)\right|, \tag{8}$$

where $\omega\left(n_in_j\right)$ is the Euclidean distance between two adaptive color-thermal histograms of the two blobs belonging to the hypothesis nodes n_i and n_j. In practice, the edge information is stored in the nodes. Thus, for each hypothesis node n_i, three sets of nodes, called S (Source), E (End), and BH (Best Hypotheses), are defined as,

$$S\left(n_i\right) = \left\{\forall n_j | \exists n_j n_i\right\}, \tag{9}$$

$$E\left(n_i\right) = \left\{\forall n_k | \exists n_i n_k\right\} \text{and} \tag{10}$$

$$BH\left(n_i\right) = \left\{\forall n_j \in S\left(n_i\right) | E_1\left(n_j\right) = n_i\right\}. \tag{11}$$

The sets defined by Eq. 9 and Eq. 10 are ordered increasingly based on the weights of their common edges with n_i. In Eq. 11, BH can be empty or contain one or more elements. E_1 is the first element of E.The sets S, E, and BH are used for object labelling and for finding trajectories. It is important to note that the event graph and the hypothesis graph may be composed of more than one component (subgraph), since the connections between nodes represent the interactions that have occurred between the blobs during tracking (two blobs that do not interact are not connected).

6.2. Step1: matching blobs

In the first step of our algorithm, a distance matrix is computed to find the blobs $B_i(t-1)$ and $B_j(t)$ that possibly correspond, along with their appearance dissimilarities in two consecutive frames. The appearance dissimilarity $D_{t-1}^{t}(i,j)$ is defined as

$$D_{t-1}^{t}(i,j) = \begin{cases} d(h_{B_i(t-1)}, h_{B_j(t)}) & \text{if overlapped} \\ -1 & \text{otherwise} \end{cases}, \tag{12}$$

where $d(h_{B_i(t-1)}, h_{B_j(t)})$ is the thermal-color histogram intersection between the ith blob in frame $t-1$ and the jth blob in frame t if the bounding boxes of the two blobs overlap. Otherwise, these two blobs cannot match each other and their corresponding element in the matrix is -1. This assumption is based on the fact that a blob should move on a short distance in two consecutive frames because of the frame rate of the camera. Therefore, its bounding boxes in the previous and the current frames should overlap. The size of the distance matrix is $N \times M$, where N is the number of blobs in the frame $t-1$ and M is the number of blobs in the frame t. The thermal-color histogram intersection is defined as

$$d(h_{B_i(t-1)}, h_{B_j(t)}) = \frac{\sum_{k=1}^{K} min(h_{B_i(t-1)}(k), h_{B_j(t)}(k))}{\sum_{k=1}^{K} h_{B_i(t-1)}(k)}, \tag{13}$$

where $h_{B_i(t-1)}$ and $h_{B_j(t)}$ are the thermal-color histogram of the ith blob in frame $t-1$ and the jth blob in frame t, and K is the number of the thermal-color histogram bins.

A blob in frame $t-1$ matches a blob in frame t if the dissimilarity is not -1. Events such as entering, leaving, merging, and splitting are detected by finding the matching blobs in two consecutive frames using the distance matrix.

6.3. Step 2: updating the graphs

The event graph and the hypothesis graph are updated based on the events detected in the matching process:

- If a blob in the current frame t is an appearing object, a track node in the event graph and a hypothesis node in the hypothesis graph are added.

- If correspondence is detected between two blobs in frames $t-1$ and t, the track node in the event graph belonging to the object is updated by adding its top-most point in the current frame t, adding the current frame number, and updating its adaptive thermal-color histogram using

$$AH_{B(t)} = \sum_{k=1}^{K} \alpha AH_{B(t-1)}(k) + (1-\alpha)h_{B(t)}(k). \tag{14}$$

In Eq. 14, $AH_{B(t-1)}$ is the adaptive thermal-color histogram of blob B at frame $t-1$, K is the number of thermal-color histogram bins, $h_{B(t)}$ is the thermal-color histogram of blob B at frame t, and α (varying between 0 and 1) is an adaptation parameter. The adaptive thermal-color histogram is used for generating a hypothesis (likelihood between two nodes), because it gives the global thermal-color information of the blob over several frames and helps reduce the effect of dramatic changes in the thermal-color distribution caused by short-time variations in lighting and temperature, as well as by shadows. Updating a track node for a correspondence event is equivalent to a sequential data association for blobs that are not in a situation of identification uncertainty. This is based on the fact that, if two blobs, one in each of two consecutive frames are found to be similar with a mutual matching, it is very likely that they are associated with the same object.

- If some blobs in frame $t-1$ are merged into a single blob in the current frame t, the tracking of the merging blobs is stopped and a new track node for the group blob is initiated in the event graph.

- If a blob in frame $t-1$ has disappeared from the FOV of the camera, its track node in the event graph is deactivated.

- If splitting is detected, for each split blob a track node in the event graph

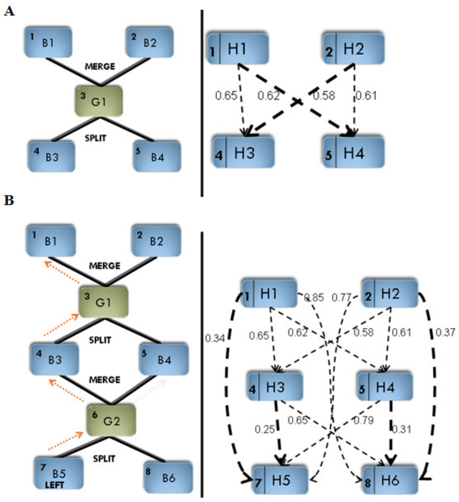


Figure 7: A) An event (left) and a hypothesis graph (right) after a merge/split. B) The same graph updated after a second merging and splitting. The number at the left of each hypothesis node corresponds to a track node in event graph with the same number in the upper left corner of the track node. The dashed arrows in the event graph show the history of one object.

and a hypothesis node in the hypothesis graph are added and hypotheses are generated for the newly added nodes.

To generate the hypotheses for split blobs, hypothesis nodes are added. Then, the S, E, and BH sets of all the nodes that are in the same subgraph as the newly added nodes are updated. Generating a hypothesis only for the nodes in the corresponding subgraph and not for the other nodes in the hypothesis graph is part of our strategy to reduce the number of hypotheses.

To perform the update, newly initiated nodes are added to the E sets of the nodes from the previous frames in the subgraph, and the previous nodes in the subgraph are added to the S sets of the newly initiated nodes. Also, the BH sets of the newly added hypothesis nodes are created according to their S sets. In other words, all the nodes in the subgraph are connected, along with directed edges from the past hypothesis nodes to the new hypothesis nodes. The weight of each directed edge is the likelihood that the source node and the end node have the same appearance, and is calculated using Eq. 8.

If the first elements of the E sets are changed after updating (S sets and E sets are always ordered increasingly), the BH sets in the same subgraph are updated consecutively. This is based on the fact that the intersection of two BH sets for two different nodes should be empty.

6.4. Step 3: object labelling and trajectory computation

The goal of object labelling is to assign a label to each tracked blob in the current frame. For a correspondence event, the blob's label in frame t is the same as it is in frame $t-1$. For merging, the merged blob's label in frame t is the label of all the merging blobs in frame $t-1$. For a blob entering frame t, the label is a new one.

For splitting, the label of a split blob in frame t is determined by processing the hypothesis graph. To do this, we traverse the hypothesis graph in bottom-up fashion, from the current frame, starting from the split blob's hypothesis node n_i. To do this, the TN (Traversing Node) set is initialized by,

$$TN_0(n_i) = \phi, \tag{15}$$

and is updated by

$$TN_t(n_i) = (TN_{t-1}(n_i) \cup BH(n_{current})) - n_{next}. \tag{16}$$

In Eq. 16, $n_{current}$ is the current node during graph traversal (at first $n_{current}$

is n_i and $TN_{t-1}(n_i)$ is ϕ), $TN_t(n_i)$ is a set of possible next destination nodes in the current frame t, and n_{next} is the next node to traverse in the graph chosen with two criteria: 1) n_{next} exists in either $BH(n_{current})$ or $TN_{t-1}(n_i)$; and 2) n_{next} has the closest temporal relationship with $n_{current}$. It is important to note that, if there is more than one node in $BH(n_{current})$ or $TN_{t-1}(n_i)$ that obeys the n_{next} criteria, we traverse these nodes separately. Traversing the graph upward and updating the TN set are continued until we reach a node for which the TN set becomes empty (nowhere to go next). A split blob is given the label of the blob that we reach after traversal of the hypothesis graph. A hypothesis node belonging to a split blob that has an empty BH set before starting graph traversal is a new appearing object that is given a new label.

At each frame, object trajectories are computed by traversing the hypothesis graph in the same way as for labelling, to get its path into the hypothesis graph. However, in the hypothesis graph, some parts of the trajectory (when the object was tracked in a group) are missing, because group blobs have no nodes in the hypothesis graph. The missing parts of the path are recovered by completing it with the help of the event graph. Fig. 8 illustrates an example of trajectory construction for two objects that occlude each other twice.

7. Results and discussion

We have assessed the performance of our method using nine video sequences that we captured (LITIV dataset) and three video sequences of the OTCBVS dataset [11]. The LITIV dataset consists of videos of different tracking scenarios captured by a thermal and visible camera at 30 frames per second with different zoom settings and at different positions. The size of the images is 320×240. Fig.

12 gives qualitative results of our unified image registration, sensor fusion, and tracking. As columns (f) and (g) in the second row of Fig. 12 show, our system tracks objects solely at the intersection of the FOVs of the thermal and visible cameras, since sensor fusion requires the data from both sensors. In section 7.1, we quantitatively assess the performance of our image registration and show that our method outperforms a state-of-the-art image registration methods [3, 5]. In section 7.2, we describe the quantitative results of our thermal-visible multiple people tracking and show the advantage of our integrated framework which performs multimodal tracking compared to separate tracking for thermal and visible videos.

7.1. Image registration evaluation

We have compared our image registration method with the image registration methods proposed by [3] and [5], using the same background subtraction parameters for all methods. In [3] and [5], the input data are trajectories generated from separate tracking for a thermal video and a visible video without sensor fusion. In contrast, in our method, the trajectories are generated by the tracking method described in section 6 performing iteratively with our image registration in a integrated framework. In [3], the registration criterion is the Euclidean point error of the object trajectory points in a pair of thermal and visible videos. In our proposed method and [5], foreground pixel overlapping is used as a matching criterion (more details in section 4). However in [5], image registration is based on a simple iterative scheme where the matrix selection is based on a simple foreground overlapping error rather than the blob fusion score used in this work.

To quantitatively compare the performance of image registration methods for each pair of videos, we constructed ground-truth (GT) foreground binary images

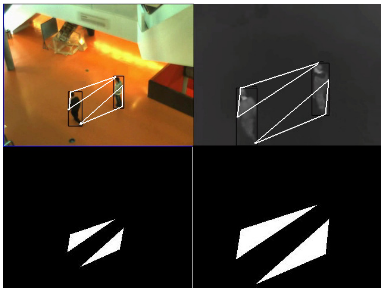


Figure 8: Top: manually selected polygons in IR and in visible images (Frame 90, Seq.1)); bottom: GT binary images

using a manual image registration. For the manual image registration of each pair of videos, one pair of thermal and visible video frames was manually aligned, and, based on this alignment, the affine transformation matrix was computed and used as the GT transformation matrix. Then, two GT binary foreground images are constructed by manually selecting points forming polygons on the thermal image, and by transforming the polygon's pixel coordinates of the thermal image using the GT transformation matrix to obtain a GT foreground for the visible image. Fig. 8 shows the manually selected polygons and the GT thermal and visible binary foreground images. We used the GT foreground images for testing the overlapping error to ensure that the background subtraction error does not contribute to it. We used two metrics to validate our method: 1) the foreground pixel overlapping error (using an equation similar to Eq. 3) of the aligned GT foreground images using the matrices computed by our method and other two methods; and 2) the average point error, which is the average pixel coordinate error in the x and y directions of the aligned polygons' corners after transformation of the GT foreground images.

For foreground pixel overlapping error comparison of our method and Caspi *et al.* [3], we have chosen video sequence 8 of the LITIV dataset. This pair of videos

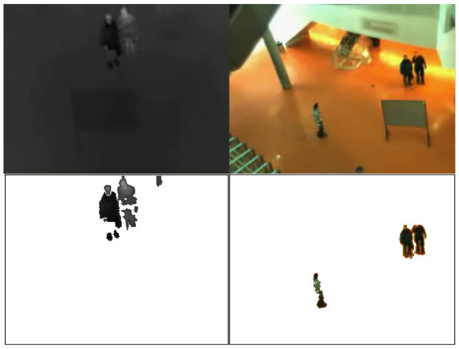


Figure 9: Top: a thermal and a visible video frames (Frame 300, Seq.8), Bottom: corresponding thermal and visible foreground images

is challenging because there are several long term blob fragmentations due to background subtraction misdetection and partial occlusion caused by a stationary object that is part of the background in the scene. In addition, this pair of videos is captured with a thermal and a visible camera at different zoom settings with an approximately small intersection of the FOVs, which makes image registration a challenging problem. Fig. 9 shows the blob fragmentations and the considerable object scale difference in a pair of thermal and visible image frames of video 8 (frame 300).

Fig. 10 shows the foreground pixel overlapping error (Eq. 3) for video pair 8 using our method, the method of [3], and manual image registration. Manual image registration also has a small overlapping error that is caused by rounding polygon coordinate values after transforming the points (our registration precision is in the pixel level). Around frames 350-400, due to several blob fragmentations occurring in the thermal video because of background subtraction misdetection, the overlapping error increases in the method of [3]. Also, in several frames, this method cannot estimate an acceptable transformation matrix, since the trajectories

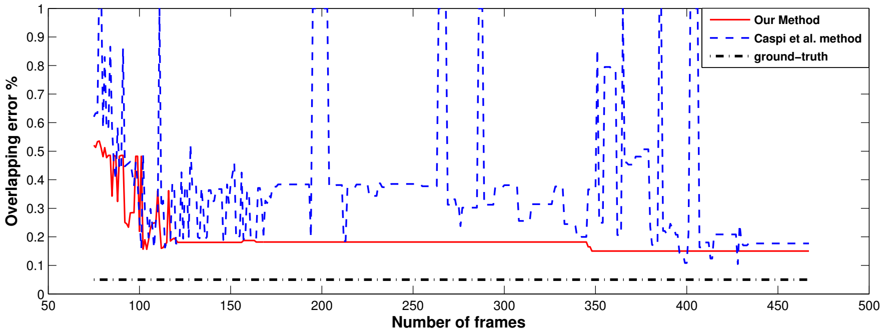


Figure 10: Overlapping error of our image registration method, of [3] image registration method, and of the manual image registration for video 8 frames 62-467.

in the thermal and visible videos are not similar in those frames. Therefore, the RANSAC algorithm did not succeed in estimating a transformation matrix based on matching the trajectories. In general, this plot shows: 1) our method estimates a good transformation matrix (error less than 30 percent) starting from around frames 110-120; 2) the transformation matrix estimated by our method is more stable over time compared to the method of [3], and 3) the overlapping error of our method is smaller than for the method of [3] over most video frames.

Our image registration, which performs iteratively with sensor fusion and tracking in a integrated system, has better image registration results than the method of [3], because: 1) the transformation matrices computed using more accurate trajectory points generated by tracking with sensor fusion are more precise than those computed using trajectories generated by separate tracking, because blob fragmentation is better handled; this is especially true for videos where there are several long term blob fragmentations, such as video sequence 8 (Fig. 9); 2) using the foreground pixel overlapping criterion results in good estimates of the transformation matrix, even when there is a relatively small FOV intersection; this

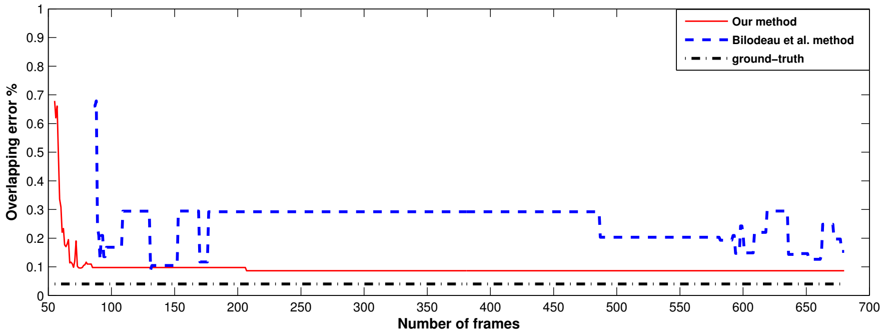


Figure 11: Overlapping error of our image registration method, of [5] image registration method, and of the manual image registration for video 1 frames 55-680.

makes trajectory matching a harder problem, since the trajectory patterns in the two videos are not similar, and 3) by using feedback, the matrix selection based on the fusion score (section 5) replaces the previous transformation matrix by a new one only if it has better fusion score.

Fig. 11 shows the foreground pixel overlapping error (Eq. 3) for video pair 1 using our method, the method of [5], and manual image registration. The reason why we have chosen video pair 1 is because it has a larger intersection of the FOVs (more similar trajectories), which enable us to show the performance of simple matrix selection and compare it with matrix selection based on fusion score that we used in this work. Both plots in fig. 10 and 11 show the transformation matrix selection in our method is more stable since there is less variation in the overlapping foreground errors compared to both state-of-the-art methods

[3, 5]. Fig. 11 shows that even the simple matrix selection used in [5] results in more stable registration results with less foreground overlapping error variations. However, because of the lack of accuracy of computed trajectories and the use of more sophisticated matrix selection such as the one used in our integrated framework, the overlapping errors vary more and even in some frames increase because of erroneous matrix selection compared to the errors of our proposed method.

Table 1 shows the average point errors of our image registration method and the [3] method for 12 video sequences. This table shows that, for video pairs 1, 3, 4, and 8, which are captured at considerably different zoom settings and a relatively small FOV intersection (less similar trajectory patterns) in both X and Y, the Euclidean distance errors of our system are less than with the [3] method. This shows that our method is more robust than the [3] method in challenging videos, where there are fewer similar trajectory patterns in the thermal and visible videos. This is basically because of two features of our method: 1) using the foreground pixel overlap criterion in the RANSAC-based algorithm; and 2) sensor fusion, which handles the fragmentation and gives more similar trajectories in both the thermal and visible videos. For the videos that are captured with the same zoom and with about the same FOV intersection (videos 2, 5, and 7) and in which there is a reasonable amount of short term blob fragmentation that does not significantly change the trajectories, our method and the [3] method give similar results. However, for video 6, where the FOVs of the two cameras are about the same, because of long term blob fragmentation that changes the trajectory patterns considerably, our method produces better results.

In our tests, videos from the OTCBVS dataset (videos 10, 11, and 12) are considered as unregistered sequences of images. In video 11, the average point errors are greater because there is only one person in this video and he is walking in a straight line. Thus, all the trajectory points are collinear, and so one of the assumptions required for estimating a precise affine matrix is not met.

7.2. Tracking evaluation

In this section, we quantitatively compare our tracking results using sensor fusion with separate tracking for the visible and thermal videos, but with the same data association method. In separate visible tracking, the color histogram is used as the tracking feature and in separate thermal tracking, the pixel intensity histogram is used as the tracking feature. Table 2 shows the tracking results of our method and separate thermal and visible video tracking.

False positive person identification, $+P$, mostly occurred during blob fragmentation, where a part of the human's body is detected as a new person. This can happen in the short term (1-2 frames) or the long term (several frames). As shown in Table 2, our sensor fusion succeeded in reducing the $+P$ error by handling blob fragmentation for both thermal and visible images in almost all the videos. The other error is the false negative person identification, $-P$. This error mostly occurs because of errors in people identification during a merge-split, or partial occlusion of a person by an object in the scene, where the person is falsely detected as a new object. Our system was able to reduce errors in people identification during a merge-split in our tested videos. The reason is that, in our method, a thermal-visible histogram is used as the tracking feature, which is more robust than separate color or thermal intensity histograms.

In Table 2, we also quantitatively compared the trajectories generated with our method and those generated by the separate video trackers using GT trajectories generated manually. The average Euclidean distance trajectory point error,

AE_{ir-vi}, of our tracking method is significantly smaller than the separate visible/infrared trackers. This shows the effectiveness of sensor fusion for computing more accurate trajectories. In fact, our video registration and tracking results show that our sensor fusion plays a critical role in improving the quality of the whole system.

8. Conclusions

In this paper, we have proposed an iterative integrated framework for thermal-visible video registration, sensor fusion, and multiple people tracking method with feedback designed for a pair of far-range, synchronized thermal and visible videos. Our video registration method is based on a RANSAC trajectory-to-trajectory matching that estimates an affine transformation matrix. Our sensor fusion method handles the object fragmentation caused by imperfect single sensor background subtraction using the aligned thermal and visible video frame pairs. Finally, our multiple people tracking methods inputs blobs constructed in sensor fusion and outputs the trajectories of moving people in the scene.

In our result, we have shown that sensor fusion improves tracking, and ultimately the accuracy of the object trajectories and registration. Our experiments show that our method outperforms similar methods previously developed, such as the [3, 5] method. Our proposed feedback scheme is flexible enough to use any other tracking method that generates trajectories online, and any other sensor fusion and object modeling that is needed for a specific video surveillance application.

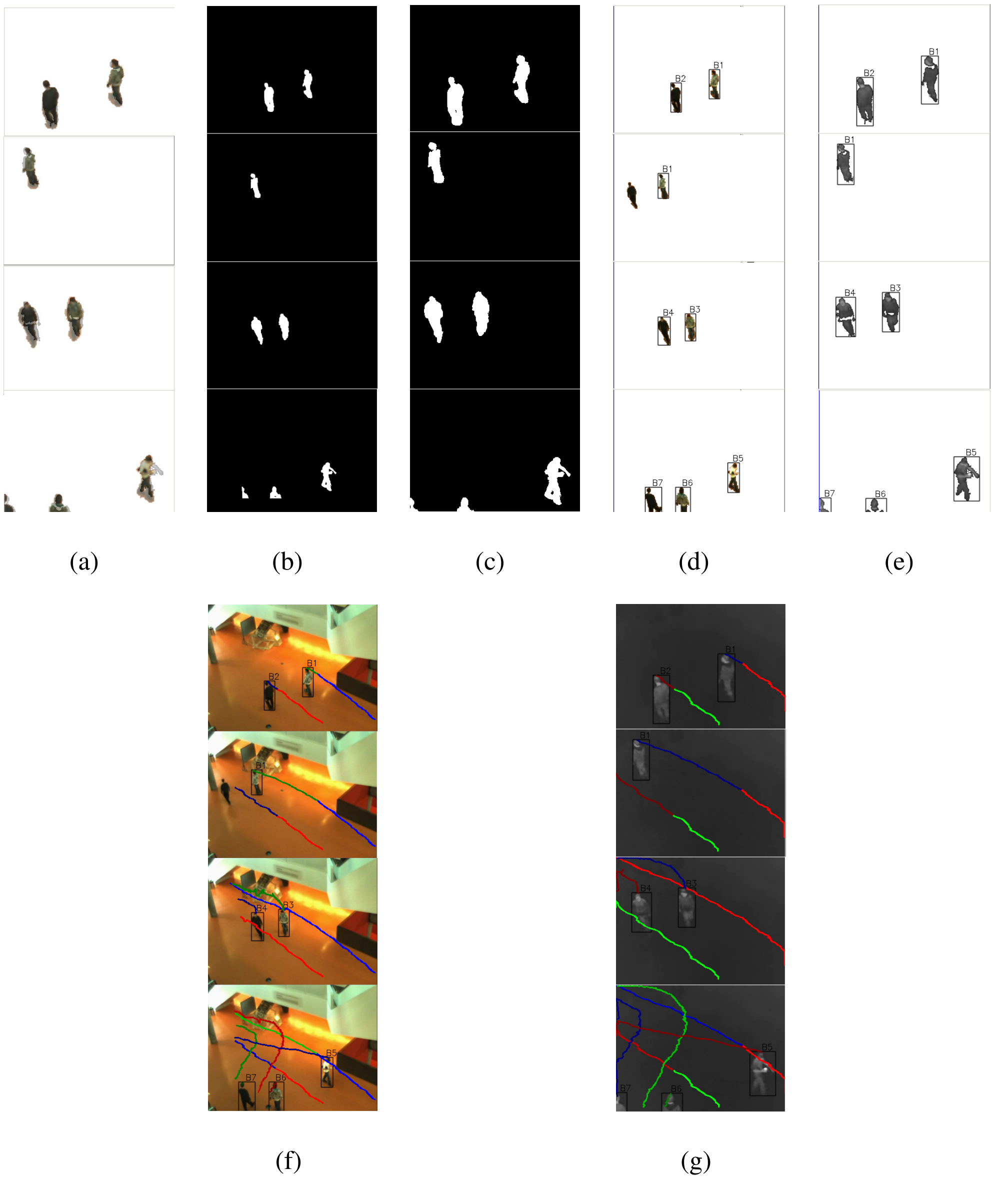


Figure 12: Our results of video 1 at frames 99, 182, 300, and 652. (a) registration of the visible on the thermal image, (b) sum-rule silhouette aligned on the visible image, (c) sum-rule silhouette aligned on the thermal image, (d) and (f) tracking result for the visible image, and (e) and (g) tracking result for the thermal image

References

[1] Z. Zhu, T. Huang, Multimodal surveillance: An introduction, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–6.

[2] D. Socolinsky, Design and deployment of visible-thermal biometric surveillance systems, in: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 2007, pp. 1 –2.

[3] Y. Caspi, D. Simakov, M. Irani, Feature-based sequence-to-sequence matching, Int. J. Comput. Vision 68 (2006) 53–64.

[4] F. Morin, A. Torabi, G.-A. Bilodeau, Automatic registration of color and infrared videos using trajectories obtained from a multiple object tracking algorithm, in: Computer and Robot Vision, Canadian Conference, 2008, pp. 311–318.

[5] G. A. Bilodeau, A. Torabi, F. Morin, Visible and infrared image registration using trajectories and composite foreground images, Image Vision Comput. 29 (2011) 41–50.

[6] A. Torabi, G. Masse, G.-A. Bilodeau, Feedback scheme for thermal-visible video registration, sensor fusion, and people tracking, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, 2010, pp. 15 –22.

[7] C. Conaire, N. O'Connor, E. Cooke, A. Smeaton, Comparison of fusion methods for thermo-visual surveillance tracking, in: 9th International Conference on Information Fusion, 2006, pp. 1–7.

[8] F. Sadjadi, Comparative image fusion analysais, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03, 2005, p. 8.

[9] C. O. Conaire, N. E. O'Connor, A. Smeaton, Thermo-visual feature fusion for object tracking using multiple spatiogram trackers, Mach. Vision Appl. 19 (5-6) (2008) 483–494.

[10] P. Kumar, A. Mittal, P. Kumar, Addressing uncertainty in multi-modal fusion for improved object detection in dynamic environment, Information Fusion 11 (4) (2010) 311 – 324.

[11] J. W. Davis, V. Sharma, Fusion-based background-subtraction using contour saliency, in: CVPR '05: IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005, pp. 11–19.

[12] J. W. Davis, V. Sharma, Background-subtraction using contour-based fusion of thermal and visible imagery, Computer Vision and Image Understanding 106 (2-3) (2007) 162 – 182, special issue on Advances in Vision Algorithms and Systems beyond the Visible Spectrum.

[13] A. Leykin, R. Hammoud, Robust multi-pedestrian tracking in thermal-visible surveillance videos, in: CVPRW '06: Conference on Computer Vision and Pattern Recognition Workshop, 2006, pp. 136–144.

[14] A. Leykin, Y. Ran, Thermal-visible video fusion for moving target tracking and pedestrian classification, in: In Object Tracking and Classification in and Beyond the Visible Spectrum Workshop at the International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[15] R. Hammoud, Augmented Vision Perception in Infrared: Algorithms and Applied Systems, Springer Publishing Company, Incorporated, 2009.

[16] S. J. Krotosky, M. M. Trivedi, Mutual information based registration of multimodal stereo videos for person tracking, Comput. Vis. Image Underst. 106 (2-3) (2007) 270–287.

[17] M. Irani, P. Anandan, Robust multi-sensor image alignment, in: ICCV '98: Proceedings of the Sixth International Conference on Computer Vision, 1998, pp. 959–966.

[18] E. Coiras, J. Santamaria, C. Miravet, Segment-based registration technique for visual-infrared images, Optical Engineering 39 (2000) 282–289.

[19] J. Han, B. Bhanu, Detecting moving humans using color and infrared video, in: International Conference on Multisensor Fusion, 2003, pp. 228–233.

[20] B. Shoushtarian, H. E. Bez, A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking, Pattern Recogn. Lett. 26 (1) (2005) 5–26.

[21] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, 2nd Edition, Cambridge University Press, Cambridge, UK, 2003.

[22] J. Han, B. Bhanu, Fusion of color and infrared video for moving human detection, Pattern Recogn. 40 (2007) 1771–1784.

[23] A. Torabi, G.-A. Bilodeau, A multiple hypothesis tracking method with fragmentation handling, in: Computer and Robot Vision, 2009. CRV '09, 2009, pp. 8–15.

Seq.	*Method*	*NF*	*SF*	*NP*	AE_X	AE_Y
1	our method	680	54	7	0.68	2.17
	Caspi *et al.*				4.75	14.79
2	our method	698	143	3	4.14	3.37
	Caspi *et al.*				6.30	3.96
3	our method	1238	200	5	2.84	2.74
	Caspi *et al.*				5.63	4.87
4	our method	329	60	2	3.89	2.84
	Caspi *et al.*				9.85	11.97
5	our method	563	100	3	2.85	3.08
	Caspi *et al.*				4.71	16.12
6	our method	1055	100	4	4.18	5.22
	Caspi *et al.*				9.86	14.07
7	our method	895	107	4	4.38	3.61
	Caspi *et al.*				4.34	2.67
8	our method	467	100	5	3.05	2.22
	Caspi *et al.*				8.89	11.21
9	our method	400	50	3	5.61	4.89
	Caspi *et al.*				7.29	7.79
10	our method	2031	180	2	1.29	1.57
	Caspi *et al.*				1.05	2.87
11	our method	650	123	1	5.92	9.03
	Caspi *et al.*				9.36	8.33
12	our method	1302	100	3	0.83	0.37
	Caspi *et al.*				6.93	2.83

Table 1: Seqs. 1-9, videos from the LITIV dataset, and Seqs. 10-12, videos from the OTCBVS dataset [11]. Our image registration results and Caspi *et al.* [3] registration results. NF: number of video frames, SF: starting frame, which is the first frame after initialization in our method (section 4), AE_X: Average Euclidean error in X of the polygons' corners for frames after initialization, AE_Y: Average Euclidean error in Y of the polygons' corners for frames after initialization.

Seq.	*Method*	NF	NP	$-P_{ir-vi}$	$+P_{ir-vi}$	AE_{ir-vi}
1	Our method	680	7	0-0	0-0	3.57-2.12
	Separate			0-2	1-3	3.98-2.42
2	Our method	698	3	0-0	0-1	2.32-3.57
	Separate			4-4	2-1	2.74-2.47
3	Our method	1238	5	0-0	0-0	2.72-2.83
	Separate			0-4	5-0	3.27-2.74
4	Our method	329	2	0-0	0-0	5.02-3.12
	Separate			2-2	1-3	19.22-15.71
5	Our method	563	3	0-0	2-3	2.86-2.22
	Separate			2-2	3-3	2.83-3.17
6	Our method	1055	4	0-0	2-4	3.60-2.18
	Separate			0-0	4-6	10.48-7.54
7	Our method	895	4	2-2	0-3	2.27-2.46
	Separate			4-4	3-4	2.35-2.43
8	Our method	467	5	0-1	3-3	7.93-5.31
	Separate			2-1	11-8	14.56-5.26
9	Our method	400	3	0-0	2-2	3.06-4.70
	Separate			2-2	2-4	3.27-4.85
10	Our method	2031	2	0-0	1-0	2.51-1.38
	Separate			0-0	6-3	4.87-2.60
11	Our method	650	1	0-0	0-0	1.67-3.03
	Separate			0-0	4-0	1.22-1.92
12	Our method	1302	3	0-0	0-0	1.73-1.77
	Separate			0-0	3-0	0.81-0.75

Table 2: Seq.1-9, videos from the LITIV dataset and Seq. 10-12 videos from the OTCBVS dataset [11]. Our thermal-visible tracking results and separate thermal-visible tracking results without sensor fusion. NF: number of frames, NP: number of tracked people, $+P_{ir-vi}$: false positive identified number of people in thermal and visible, $-P_{ir-vi}$: false negative identified number of people in thermal and visible, and AE_{ir-vi}: Average Euclidean distance trajectory point error compared with manually generated GT trajectories.