

Human segmentation by geometrically fusing visible-light and thermal imageries

Jian Zhao · Sen-ching S. Cheung

Published online: 5 December 2012
© Springer Science+Business Media New York 2012

Abstract From depth sensors to thermal cameras, the increased availability of camera sensors beyond the visible spectrum has created many exciting applications. Most of these applications require combining information from these hyperspectral cameras with a regular RGB camera. Information fusion from multiple heterogeneous cameras can be a very complex problem. They can be fused at different levels from pixel to voxel or even semantic objects, with large variations in accuracy, communication, and computation costs. In this paper, we propose a system for robust segmentation of human figures in video sequences by fusing visible-light and thermal imageries. Our system focuses on the geometric transformation between visual blobs corresponding to human figures observed at both cameras. This approach provides the most reliable fusion at the expense of high computation and communication costs. To reduce the computational complexity of the geometric fusion, an efficient calibration procedure is first applied to rectify the two camera views without the complex procedure of estimating the intrinsic parameters of the cameras. To geometrically register different blobs at the pixel level, a blob-to-blob homography in the rectified domain is then computed in real-time by estimating the disparity for each blob-pair. Precise segmentation is finally achieved using a two-tier tracking algorithm and a unified background model. Our experimental results show that our

Part of this material is based upon work supported by the National Science Foundation under Grant No. 1018241. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

J. Zhao
Windows Phone, Microsoft Corporation, One Microsoft Way,
Redmond, WA 98052, USA
e-mail: Jian.Zhao@microsoft.com

S.-c. S. Cheung (✉)
Center for Visualization and Virtual Environments, University of Kentucky,
329 Rose Street, Lexington, KY 40506, USA
e-mail: cheung@engr.uky.edu

proposed system provides significant improvements over existing schemes under various conditions.

Keywords Sensor fusion · Human segmentation · Multi-camera fusion · Thermal cameras

1 Introduction

A central theme in many modern vision systems is to identify and segment human shapes in video sequences. Despite decades of efforts, it remains a challenging problem due to the significant variations in visual appearances caused by occlusion, illumination change, highlight, shadow, and color confusion. One approach that has garnered a great deal of interest in recent years is to utilize multimodal sensors to improve the segmentation results. As human bodies usually have different temperature than the ambient environment, thermal infrared sensors are popular choices to be used in conjunction with regular visible-light cameras.

While the introduction of thermal camera can potentially benefit the segmentation process, it also introduces a number of technical challenges. First, unless expensive optical realignment apparatus is used, the thermal camera and the visible-light camera are not spatially aligned. Error in registration between the two cameras can significantly degrade the performance of any segmentation algorithm. As the texture information in the two types of cameras are completely different, typical stereo vision approaches cannot be used to solve the problem. Second, the new modality can bring new channels of noises which could further confuse the segmentation classifier. It is thus imperative to develop proper sensor fusion techniques to take advantage of both modalities in order to achieve the optimal results.

The general problem of sensor fusion has been studied for decades [16, 19]. Most of the existing works in sensor fusion, however, remain at a relatively abstract level as the problem heavily depends on the specific applications and the type of sensors. While the conclusions from these studies might be general enough to cover all kinds of sensor networks, they provide little guidance for the design of any particular sensor fusion application.

More specifically, we are tackling a camera fusion problem which consists of two distinct fusion processes: geometric fusion and data fusion. Geometric fusion refers to the set of techniques that align the images from different cameras while data fusion aims at making prediction or decision based on the aligned data. Key algorithmic operations used in these processes are categorically illustrated in Fig. 1. They include:

1. Determining a proper *camera model* is the first step of designing a camera fusion algorithm. It has a significant impact on the subsequent steps, especially the calibration and the registration procedure. While the pinhole camera model is used in most applications, there exists a range of other camera models that can be useful in different situations—some models can offer a more precise description of the optical process while others can provide complexity reduction with fewer parameters for specific applications.
2. *Calibration* is the off-line operation that estimates the position and internal parameters of all the cameras in the network. The complexity of this procedure mainly depends on the chosen camera model.

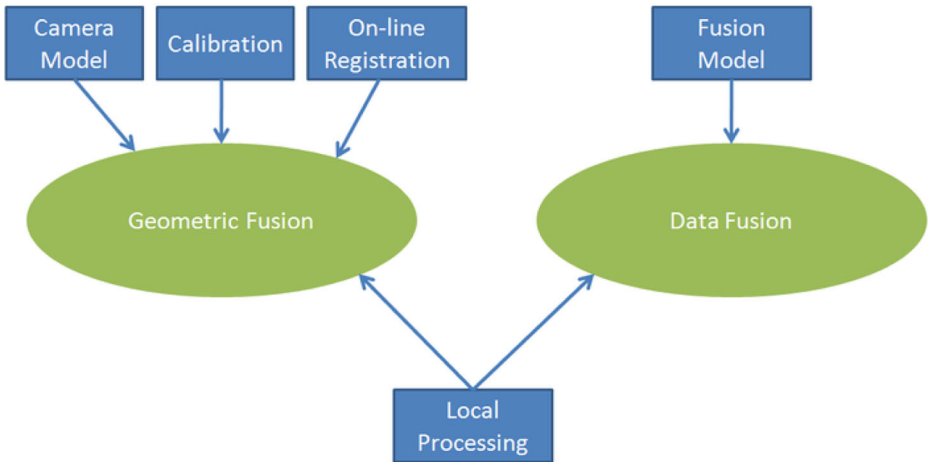


Fig. 1 Camera fusion diagram

3. Camera projection reduces the 3D world into 2D images, making the alignment between different camera views an ill-posed task. Even in the best scenario where all the cameras are stationary and calibrated off-line, the relationship between images obtained from different cameras still depends on the unknown 3D positions of the objects. To further complicate the problem, cameras may move during operations when they can be mounted on movable platforms. To cope the dynamic 3D environment and camera positions, it is desirable to use an *online adaptive registration* procedure that can adjust the registration parameters in real-time and can incorporate any a priori information about the geometric relationship between the objects and the environment.
4. *Local processing* must be conducted at each camera to reduce the communication and computation burden on a camera network. While it is important to carry out a case-by-case analysis when designing the local processing algorithm for a given camera network, there are a range of image processing algorithms including background subtraction, feature extraction and feature tracking that are frequently applied due to their simplicity and effectiveness.
5. Lastly, a proper *data fusion model* is needed to aggregate information received from all cameras in the network. There are a myriad of sensor fusion models applicable for camera networks [43]. Traditional sensor fusion techniques such as fusion-by-selection techniques are useful in identifying a specific camera that best describes the situation. Bayesian techniques provide a unified probabilistic framework to fuse all the temporal and spatial information together and have the capability of injecting domain knowledge as prior probabilities. If the ultimate goal is to produce a classification result, fusion-by-classification techniques are appropriate as they treat all available data as a single feature vector and feed it into a classification engine.

The main contribution of this paper is a novel camera fusion design that combines thermal and visible-light images for robust segmentation of humans in video sequences. Our proposed system closely follows the aforementioned camera

fusion pipeline: a novel approximation camera model named *blob homography* is specifically designed for geometric fusion between these two modalities. By rectifying the homography matrix, our proposed model significantly reduces the computational complexity of parameter estimation compared with the standard approach in homography estimation. Communication between the two cameras are kept at the minimum as local background subtraction is applied to extract foreground blobs. Our proposed algorithm estimates in real-time the blob-to-blob homography matrices to achieve pixel-level registration based the disparity between each blob pair from the two cameras. The multi-modality information is then combined under a two-tier tracking algorithm and a unified background model to mitigate segmentation noise from either modalities. An earlier version of this work has appeared in [44]. In this extended work, we provide a more detailed review of related work, a new formulation of the blob homography to describe the registration process, and additional experimental results to demonstrate the performance of the proposed system.

The rest of the paper is organized as follows. Related work in geometric and data fusion are first reviewed in Section 2. In Section 3, we propose a novel camera model called blob-homography model and show its merits over other camera models. The joint calibration procedure for the thermal and RGB cameras is demonstrated in Section 4. The real-time algorithms for blob-based registration and parameter estimation are presented in Section 5. In Section 6, we describe our data fusion component which includes robust tracking scheme and fused background modeling. Experimental results presented in Section 7 demonstrate the effectiveness of our proposed system over the standard approach of image warping. We conclude the paper with a discussion on future work in Section 8.

2 Related work

The sensor fusion model described by the US Joint Directions of Laboratories or the JDL model is arguably the earliest framework for sensor fusion. First proposed in [40] and later revised in [30, 33], the JDL model consists of five different levels [16]:

1. Source preprocessing (*Level 0*): This is a pre-processing level performed locally at each sensor. The main objective of any level-0 process is to reduce the communication and computation burden needed at the later fusion stages. Common level-0 processes include signal denoising and compression.
2. Object/Entity Assessment (*Level 1*): This level represents the core signal processing of sensor data including data alignment, data association, object tracking, and identification.
3. Situation Assessment (*Level 2*): The objective of this level is to find a contextual description of the relationship between the observed objects and the anticipated events.
4. Threat Assessment (*Level 3*): By combining any prior knowledge and predictions about the situation, processes in this level infer about vulnerabilities and opportunities in the sensor network. Typical processes include the estimation of the security threat level and the locations of target objects.
5. Process Refinement (*Level 4*): This level focuses on the fusion of information from different processes and the control of different sensors. Sensor management

is the central functionality in this level and is the most important task in an active sensor network.

Typical sensor fusion techniques in camera network corresponding to each level in the JDL model are illustrated in Fig. 2.

While the JDL model provides a comprehensive framework for all sensor fusion applications, the camera network fusion problem focuses more narrowly on two aspects within level one of the JDL model: geometric fusion aimed at registering disparate camera views and data fusion aimed at making prediction or decision based on aligned data. Table 1 lists the recent work in these areas that are relevant to the fusion problem between visible-light and thermal cameras. We categorize each work based on its approach used in geometric fusion as well as the image features and statistical inference tools used in data fusion. Geometric fusion techniques range from hardware-based optical fusion and 3D reconstruction which are of high cost but produce accurate pixel-alignment, and medium computational-cost scheme like blob homography that achieves blob-alignment, to fast image warping scheme with only frame-level alignment. For the data fusion phase, the key objective is to produce accurate object segmentation. In general, pure image-based technique produce sub-par segmentation results. Improved segmentation can be produced with background (BG) subtraction and the most accurate results come from a combination of BG subtraction and temporal tracking with probabilistic inference tools to handle uncertainty. We have included our assessment in computational cost, alignment accuracy and segmentation accuracy for each work in Table 1. We believe our system, first described in [44] and refined in this manuscript, has better performance than other state-of-the-art schemes as it can achieve pixel-level alignment with an efficient blob homography system in rectified domain and produce accurate segmentation results with a unified scheme that combines background subtraction and tracking. In the following subsections, we survey all of these techniques and contrast them with our design in both geometric and data fusion.

2.1 Geometric fusion

Most existing systems tackle the geometric fusion problem by either optical fusion [11, 39, 42], full 3D reconstruction [23, 28], or image warping [1, 9, 13, 14, 20, 27, 36]. The optical fusion approaches use specially-designed optical devices to merge

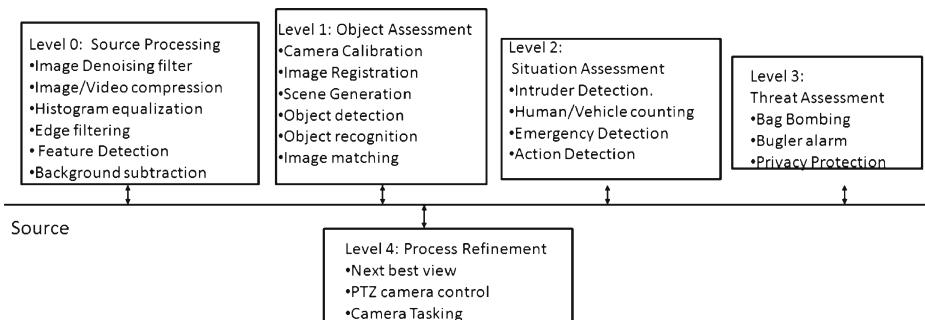


Fig. 2 Sensor fusion techniques in camera network

Table 1 Related work in thermal and visible-light camera fusion in human detection

Fusion scheme	Geometric fusion	Cost	Alignment	Image features	Inference tools	Accuracy
[42]	Optical fusion	High	Pixel	Region grow	None	Low
[39]	Optical fusion	High	Pixel	BG subtraction	None	Medium
[31]	Optical fusion	High	Pixel	BG subtraction	None	Medium
[12]	Alignment assumed	–	Pixel	BG subtraction	Kalman filter	High
[7]	Alignment assumed	–	Pixel	Intensity	Bayesian net	High
[5]	Alignment assumed	–	Pixel	Active contour	Flux tensor	High
[15]	Alignment assumed	–	Pixel	Motion	Particle filter	High
[29]	Alignment assumed	–	Pixel	BG subtraction	Bayesian	High
[10]	Alignment assumed	–	Pixel	YUV spatiogram	Mean-shift net	High
[14]	Image warping	Low	Frame	Active contour	None	Medium
[27]	Image warping	Low	Frame	BG subtraction	Fuzzy logic	High
[20]	Image warping	Low	Frame	BG subtraction	None	Medium
[18]	Image warping	Low	Frame	Thresholding	None	Low
[9]	Image warping	Low	Frame	Region grow	None	Low
[36]	Image warping	Low	Frame	BG subtraction	None	Medium
[1]	Image warping	Low	Frame	BG subtraction	Simple logic	Medium
[28]	3D reconstruction	High	Pixel	BG subtraction	None	Medium
[23]	3D reconstruction	High	Pixel	BG subtraction	None	Medium
[26]	Blob homography	Medium	Blob	BG subtraction	None	Medium
[35]	Blob homography	Medium	Blob	BG subtraction	Blob tracking	High
[44]	Blob homography	Low	Blob	BG subtraction	2-tier tracking	High

the optical axes of the two cameras so that the two cameras can see exactly the same view. Despite their computational efficiency and registration accuracy, they are inflexible in terms of matching cameras with different resolutions and field of views. Some designs suffer from high manufacturing costs [39] while others have significant degradation in image quality due to signal absorption in the optical components [11].

Instead of the hardware approach, the most accurate software approach for geometric fusion is to reconstruct the 3D world using multiple disparate views obtained from different cameras [23, 28]. While these system provide accurate alignment between different modalities, they impose heavy computational complexity on the geometric fusion process and are not suitable for real-time operations—for example, in [23], a full calibration of both infrared and visible-light cameras needs to be conducted to estimate the intrinsic and extrinsic projection matrices. Then stereo cameras are used to replace the single visible-light camera to perform online registration in order to get the 3D position of each scene point. The geometric fusion is achieved by projecting the 3D scene points onto the image plane of thermal camera.

Image warping method is the most widely adopted method for geometric fusion due to its simplicity. It calculates a homography matrix between the two camera views from a set of corresponding points, which can either be hand-picked or obtained by an automatic calibration process. The homography is then used to warp segmentation results from one modality to the other. If the same homography matrix is applied to all objects in the scene, significant distortion may occur if there is a large variation in depth among different objects in the scene. Systems such as [14, 20] adopt additional search procedures to correct the registration error. The search procedure, however, can adversely affect the computational performance of the registration process.

Instead of using a single homography for the entire image, the authors of [32] propose to align foreground blobs from different cameras by identifying shape feature points to estimate the homography. Similar approaches are also used in [26]. Our proposed system is also based on estimating homography between individual blob correspondences. Compared with these prior approaches, there are three key contributions of our proposed system: first, we are the first to reformulate such a blob-to-blob registration as a camera model which enables better understanding of the proposed approach within the well-studied framework of camera models. Second, our approach significantly reduces the complexity of the online parameter estimation process. Specifically, we reduce the traditional eight parameters in homography to just a single parameter. Finally, by including the parameter estimation into a combined background-subtraction and tracking system, we utilize temporal information to provide a more robust estimation of the blob homographies.

2.2 Data fusion

While our primary goal is to geometrically fuse multiple camera views, our system includes components such as background subtraction and tracking that make decision from disparate sensor data. Traditional data fusion techniques are pervasively used to improve the segmentation from information obtained by multiple sensors. Kumar et al. [27] adopt fuzzy logic to evaluate the confidence from each sensor. Han and Bhanu [20] compares different fusion rules under a Bayesian framework. Statistical inference techniques, such as Kalman filter, particle filter, and mean-shift are also commonly used to fuse the multi-modality observations to make an informative decision [6, 12, 15, 29, 45].

Alternatively, the data fusion can be performed at the image level. In [42], region grow segmentation algorithm is performed using the output of thermal camera as seeds. Similar techniques are used in [7]. Morphological operations are adopted in [13, 14, 35]. In our proposed system, we combine the thermal and color image into a fused non-parametric background model similar to the scheme in [24]. While we do not claim novel contribution to the theoretical fusion problem, we argue that a simple two-tier background modeling with adaptive parameters in each tier is robust enough to provide efficient and accurate segmentation results.

3 Camera model

Depending on the application requirements, different camera models can be used to describe the process of camera projection. The most popular model is the pinhole

camera model. It captures the key characteristics of the projection process and has a convenient mathematical representation in homogeneous coordinates. However, it requires 11 parameters for each camera and the robust estimation of these parameters can be computationally intensive. The parallel projective model is a simplification of the pinhole camera model where the lines of projection are assumed to be parallel. Not only does the parallel projective model have less degrees of freedom, it admits a simpler mapping between disparate camera views. The weak perspective model, a special case of the parallel projective model, further assumes the average distance of the objects to the camera to be much larger than the distance variation among the objects. The weak perspective model can further reduce the number of parameters. Despite its simplicity, the assumption of constant depth in weak perspective model is often violated in real-world applications such as video surveillance. Our proposed camera model combines multiple weak perspective models to simultaneously improve the registration and maintain low computational complexity. In this section, we review the mathematical underpinning of existing camera models that are relevant to our applications, and provide the precise formulation of the new model.

3.1 Pinhole camera model

Under the pinhole camera model [21], a 3×4 projection matrix P is used to map a 3D point \mathbf{X} , represented in homogeneous coordinates $[x, y, z, 1]^T$, into 2D camera coordinates $\mathbf{x} = [x', y', 1]^T$

$$\mathbf{x} = P\mathbf{X}$$

The camera projection matrix P can be decomposed into two matrices:

$$P = K[R|\mathbf{t}]$$

$[R|\mathbf{t}]$ is the *extrinsic* matrix which is composed of a rotation matrix R and a translation vector t of the camera coordinate system with respect to the world coordinate system. K is called the *intrinsic* matrix which encapsulates the camera's internal parameters:

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

where f_x, f_y are the focal length of the camera converted into image pixel units; c_x, c_y are the coordinates of the center of the image plane; the skew factor s is non-zero only when the x and y image axes are not perpendicular.

The pinhole camera model has 11 degrees of freedom. Since they are coupled together during the projection process, a carefully designed calibration is required to estimate them.

3.2 Lens distortion

Camera lenses sometimes introduce non-linearity into the projection process. If the camera suffers from significant lens distortion, an equalization process is needed before applying the pinhole camera model. Let $\mathbf{x}_d = [x_d, y_d]^T$ be the image coordinate under distortion and $\mathbf{x}_u = [x_u, y_u]^T$ be the coordinate with the distortion corrected. $\mathbf{x}_c = [c_x, c_y]^T$ is the center of the image and $r = \sqrt{(x_d - c_x)^2 + (y_d - c_y)^2}$

is the distance of the pixel in question to the image center. There are generally two types of distortion [4]:

1. *Radial distortion* models the radially symmetric part of lens distortion. In practice, it accommodates most of the distortion. It is usually modeled as a polynomial with only even power terms:

$$\mathbf{x}_u = \mathbf{x}_c + \left(1 + \sum_{i=1}^n \kappa_i r^{2i} \right) (\mathbf{x}_d - \mathbf{x}_c)$$

where κ_i 's are the lens-specific parameters for the radial distortion.

2. *Tangential distortion* is caused by improperly-aligned lens and is mathematically modeled as follows:

$$\mathbf{x}_u = \mathbf{x}_d + \begin{bmatrix} D_1(r^2 + 2(x_d - c_x)^2) + 2D_2(x_d - c_x)(y_d - c_y)(1 + \sum_{i=1}^n D_{i+2}r^{2i}) \\ D_2(r^2 + 2(y_d - c_y)^2) + 2D_1(x_d - c_x)(y_d - c_y)(1 + \sum_{i=1}^n D_{i+2}r^{2i}) \end{bmatrix}$$

where D_i 's are parameters for the tangential distortion and n controls the highest degree term used in the model.

3.3 Parallel projection model

To reduce the number of parameters in the pinhole camera model, one can move the camera center to infinity so that the projection lines become parallel. While it is obviously an unrealistic assumption, this parallel projection model can generate different approximations that can significantly simplify the parameter estimation process.

Under the parallel projection assumption, the projection matrix P becomes P_∞ defined below [21]:

$$P_\infty = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where $r_{i,j}$ are the elements of the rotation matrix R . Under this model, the intrinsic matrix remains the same as in the pinhole camera model but the extrinsic matrix is simplified as the camera center is at infinity. The model in (1) is called the *affine camera model*. The degree of freedom in the extrinsic matrix reduces to three, resulting in a total of eight parameters in the affine camera model.

Further simplifications of the affine camera model are possible. By setting the skew factor to zero, the *weak perspective* model has only seven degrees of freedom. It is a good approximation when the average distant of the object to the camera is much larger than the distance variation among different objects. In fact, it is equivalent to first projecting the object onto the object plane by a set of parallel rays orthogonal to the plane, then projecting the image from the object plane to the image plane by scaling the whole image by a fixed factor proportional to the inverse of the average depth of the object. This process is shown in Fig. 3.

A further simplification of the weak perspective model is the *scaled orthographic projection* model obtained by setting $f_x = f_y$ which results in only 6 degrees of freedom. The *orthographic projection* model can be derived from the scaled orthographic projection model by setting $f_x = f_y = 1$.

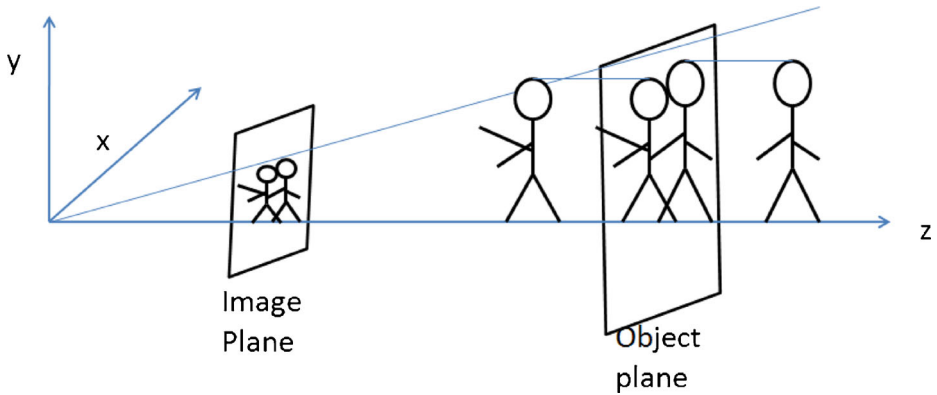


Fig. 3 Weak perspective model. The object is first projected onto the object plane by lines parallel to the image axis. It is then scaled by a factor related to the average depth of the object to the image plane

3.4 Plane homography model and blob homography

Using the pinhole camera model, any 3D point observed in one camera can be seen anywhere along the epipolar line on the image plane of a second camera [21, ch. 9]. As such, there is no bijective mapping between two camera views. However, if we restrict the 3D points to be co-planar, there does exist a bijective mapping between image points from any two camera views. Specifically, if $\mathbf{x}_1 = [x_1 \ y_1 \ 1]^T$ and $\mathbf{x}_2 = [x_2 \ y_2 \ 1]^T$ are points from the two camera views of any 3D point on a given plane, there exists an invertible linear mapping H that maps one to the other [21, ch. 2]:

$$\mathbf{x}_1 = H\mathbf{x}_2 \quad (2)$$

The matrix H , known as the homography matrix, is a 3×3 matrix with eight degrees of freedom. Unfortunately, the homography matrix depends on the depth and pose of the object plane [21, ch. 13.1]. As such, the homography matrix has to be estimated on-line.

In 3D computer vision, it is a common practice to rectify camera views before estimating the scene structure. The rectification process finds linear mappings in homogeneous coordinates that move the epipoles of the camera pair into infinity. As a result, the pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. Denoting the rectification matrices for the two cameras as H_1 and H_2 , the points after rectification in two images as $\mathbf{x}'_1 = [x'_1 \ y'_1 \ 1]^T$ and $\mathbf{x}'_2 = [x'_2 \ y'_2 \ 1]^T$, and the homography between the two rectified image planes as H' , we have the following set of relationships:

$$\begin{aligned} \mathbf{x}'_1 &= H_1\mathbf{x}_1 \\ \mathbf{x}'_2 &= H_2\mathbf{x}_2 \\ \mathbf{x}'_2 &= H'\mathbf{x}'_1 \end{aligned} \quad (3)$$

Using (2) and (3), the original homography matrix H and the rectified homography matrix H' are related by the following equation:

$$H = H_1^{-1}H'H_2 \quad (4)$$

The rectification matrices depends only on the epipoles of the two camera views which can be obtained from the fundamental matrix. The fundamental matrix does not depend on the scene structure and can be robustly estimated offline using a calibration object. The significance of the rectification process is that the rectified homography matrix H' is very simple. In fact, we can show that H' has only three parameters. The precise form of H' is described in Theorem 31 and the proof can be found in the [Appendix](#). An important consequence is that after rectification, a pair of corresponding image points \mathbf{x}'_1 and \mathbf{x}'_2 in the two camera views have identical y -coordinate. The difference in the x -coordinate between corresponding points is called disparity.

Theorem 31 (Homography in rectified images) *The homography H' in rectified image domain is in the form*

$$H' = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If an object in the scene is far away from the camera, all parts of the same object can be assumed to have the same depth to the camera. This is a valid assumption in common surveillance scenarios because the foreground human objects are at least several meters away from the cameras, while the depth variation of the different parts of a human body is within several centimeters. Based on this assumption, we propose the *blob homography model* in which a separate homography is used to map between the blobs at different views that correspond to the same foreground object. The projection process under the blob homography is shown in Fig. 4. The blob homography model represented in the rectified domain can lead to significant simplification in H' —since the entire object is assumed to have the same depth, the disparity of every pair of corresponding points from the two rectified camera views

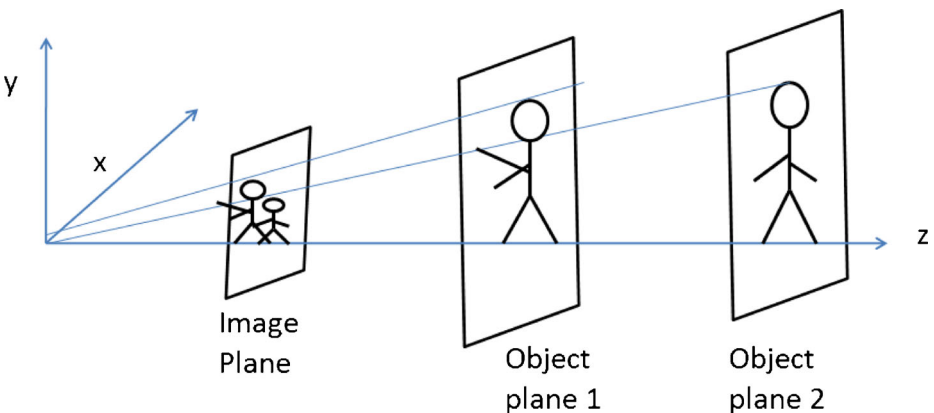


Fig. 4 Blob homography model. The foreground image is firstly segmented into blobs. The weak perspective model characterized by different depth is then applied for each blob

of the same object must be identical to each other. For every pair of corresponding points $\mathbf{x}'_1 = [x'_1 \ y' \ 1]^T$ and $\mathbf{x}'_2 = [x'_2 \ y' \ 1]^T$, we have

$$\begin{aligned} x'_2 - x'_1 &= d \\ (a_{11} - 1)x'_1 + a_{12}y' + a_{13} &= d \end{aligned} \quad (5)$$

The constraint (5) is derived based on Theorem 31 and must hold for all points on the same blob, thereby implying that $a_{11} = 1$, $a_{12} = 0$ and $a_{13} = d$. The blob homography between correspondent blobs thus depends only on one coefficient, a_{13} , which is the disparity in the rectified image. The estimation of this single parameter can be robustly performed with very few corresponding points and the estimation process is described in Section 5. This is particularly important for registration of thermal images which do not have prominent texture information to produce a large number of point correspondences. In the subsequent sections, we will describe how correspondences are established in our proposed system.

4 Joint camera calibration

Camera calibration has been studied for decades and a handful of robust multi-camera calibration toolboxes have emerged. For example, the tools provided in [2] are useful for estimating intrinsic parameters, including the distortion parameters for individual cameras. The tools described in [8, 34] are frequently adopted for estimating the extrinsic parameters for multiple cameras in a unified world coordinate system. However, these popular toolboxes are usually designed for homogeneous camera networks with regular visible light cameras. When cameras with different modalities are concerned, they must be adapted to handle the significant disparity between views. This is particular challenging for color and thermal images due to the significant differences in object appearances.

The blob homography model described in Section 3.4 has the advantage that it does not explicitly require the intrinsic and extrinsic parameters of the cameras. All it needs are the rectification matrices which can be estimated based on a small set of corresponding point pairs from both camera views. In order to provide a rapid calibration process, we have constructed a simple calibration object shown in Fig. 5. It is a circular metal plate painted with a distinctive color. The distinct color can

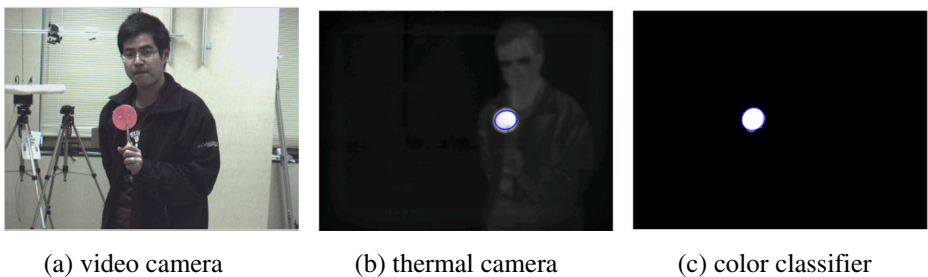


Fig. 5 The colored metal plate in **a** is used as calibration object. **b** shows the ellipse-fitting result in the thermal camera and **c** shows the classification and ellipse-fitting result in the visible-light camera

be easily identified in the color frame using a color classifier. To be robust against change in illumination, a mixture-of-Gaussian classifier is applied to the hue channel of the image to robustly identify all pixels that match the target hue. To make the same object equally distinct in the thermal image, the metal plate is slightly heated so that it is at a higher temperature than the ambient environment. As a result, it becomes very prominent in the thermal image as shown in Fig. 5 and a simple threshold can easily identify the entire shape. We then use a least-square ellipse fitting algorithm to determine the centers of the detected regions from both modalities. The pair of centers is used as a single correspondence and a set of corresponding points are obtained by moving the calibration objects to different positions.

After the correspondence have been established, Hartley's method for rectifying noncalibrated cameras is applied [22]. A RANSAC algorithm is first performed to remove outliers before the estimation of the fundamental matrix. The epipoles are then obtained by decomposing the fundamental matrix. Finally, by minimizing the projection error, an iterative method is used to obtain the two homographies that maps the epipoles to infinity. Note that after rectification, the rectified images will have the same resolutions regardless of their original resolutions.

5 Online registration

As stated in Section 3.4, the blob homography after rectification is completely described by the constant disparity along the horizontal direction between a pair of corresponding blobs. In order to produce a robust estimate of this disparity, we scan through all possible pairs of corresponding points along the contours of the blobs from the color and thermal frames. We then employ a number of geometric constraints to filter out the outliers and compute our final estimate based on the *statistical mode* of the measured disparities among the remaining pairs of correspondences. The details of the algorithm are given in Algorithm 1.

Algorithm 1: Online registration algorithm

Input: Rectification matrices H_1 , H_2 , disparity range $[d_{min}, d_{max}]$ and correspondent blob pairs

Output: Blob wise homography H

foreach pair of corresponding blobs B_1 and B_2 **do**

 Extract the contours of B_1 , B_2 ;

 Rectify the contours using H_1 and H_2 ;

foreach Horizontal Scan line **do**

if Both contours have same number of points **then**

 match the points between two blobs into pairs according to the scan line

 order;

 Filter out the pairs with disparity out of $[d_{min}, d_{max}]$;

 Collect the disparity histogram;

end

if there are enough counts in the histogram **then**

 Get the mode of the disparity histogram \hat{d} ;

else

 return failure;

end

end

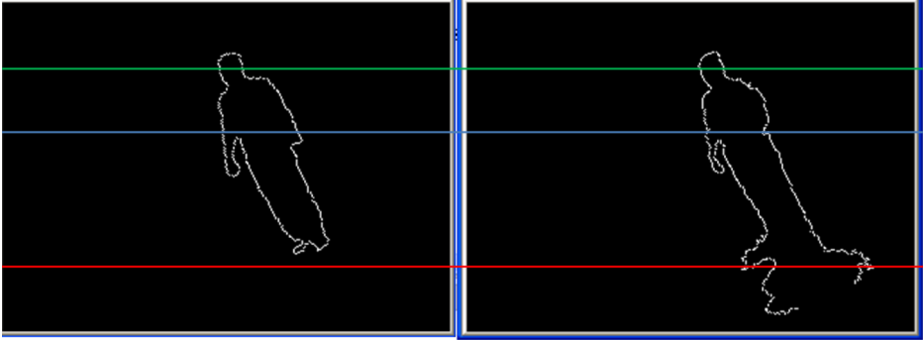


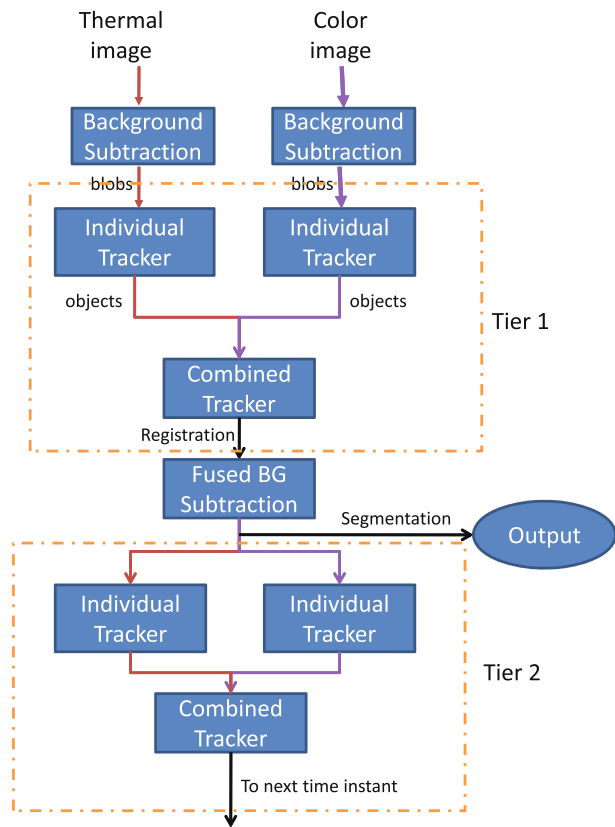
Fig. 6 Removing erroneous correspondences at different scan-lines: *green line (upper)* gives good match while *red line (bottom)* contains uneven number of points due to the shadow, which will be ruled out by our algorithm. Along the *blue line (middle)*, the rightmost points of each image is a false match and is likely to be filtered out by disparity range

Two different mechanisms are used in Algorithm 1 to effectively filter out the outliers. We illustrate these mechanisms based on a real example shown in Fig. 6. The rectified color and thermal contours of an individual are shown along with three sample scan-lines: green, blue, and red. Among the three scan-lines, only the green one produces two reliable correspondences from the left and right contour points that will be used for disparity estimation. The faulty correspondences from the blue and red lines are identified using the following procedure. First, when a scan line have different numbers of points in the two views, it is treated as unreliable and discarded without calculating any disparity. This process helps to rule out difficult situations due to occlusion, shadows, and defective foreground/background segmentation. The red line in Fig. 6 is an example of such a mismatch pruned away by this first step. Second, we remove problematic scan-lines based on disparity range $[d_{\min}, d_{\max}]$. We obtain this disparity range during the calibration process by placing the calibration objects at extreme ends within the field of views of both cameras. After performing the rectification, the disparity range is measured by projecting the calibration points into the rectified domain and finding the minimum and maximum disparities. Disparities measured during the online registration that are out of the disparity range are simply discarded. Along the blue scan-line in Fig. 6, the disparity of the pair of right contour points from the color and thermal images is bigger than d_{\max} and is thus eliminated from the pool of correspondences.

6 Robust fusion via tracking and background modeling

In Section 5, we present the core algorithm in estimating disparity between corresponding blobs that is robust against minor occlusions and segmentation defects. In this section, we address the key issues of extracting blobs from the videos, establishing correspondences between blobs from the two camera views, and handling major occlusions and other segmentation problems. Our approach is based on a two-tier blob tracking scheme coupled with a joint color-thermal background subtraction module. The different components of our approach are shown in Fig. 7.

Fig. 7 Block diagram of our robust fusion via two-tier tracking and background modeling



Before describing the technical details of our approach, we first provide an overview of the entire process and motivate our approach with an example. As shown in Fig. 7, background subtraction is first performed locally at each camera to extract the color and thermal foreground blobs. The detected blobs are fed into separate trackers to detect occlusions and filter out possible false positives. A combined tracker is then used to match objects between the two camera views, calculate and track the disparity of each object. Using the disparity estimated from the combined tracker, a blob homography matrix for each blob can be calculated using (4) which will be used to align each pair of matched objects to perform a second joint background subtraction. Finally, the improved segmentations from the joint background subtraction are used to update the state of each tracker for processing subsequent frames.

Figure 8 illustrates this process. In the first time instant shown in Fig. 8a, there is only one object in each view. However, in the next time instance in Fig. 8b, due to the shadow of the object on a nearby wall, the visible-light camera produces two blobs after background subtraction and the tracker mistakenly takes the shadow blob as a new object. In the combined tracker, no correspondences can be established in the calculation of disparity in Algorithm 1 as all the point pairs are filtered out. Using the temporal prediction from the tracker in the visible-light camera, we can still produce a reasonable estimate of the disparity. By feeding the estimated disparity to the joint

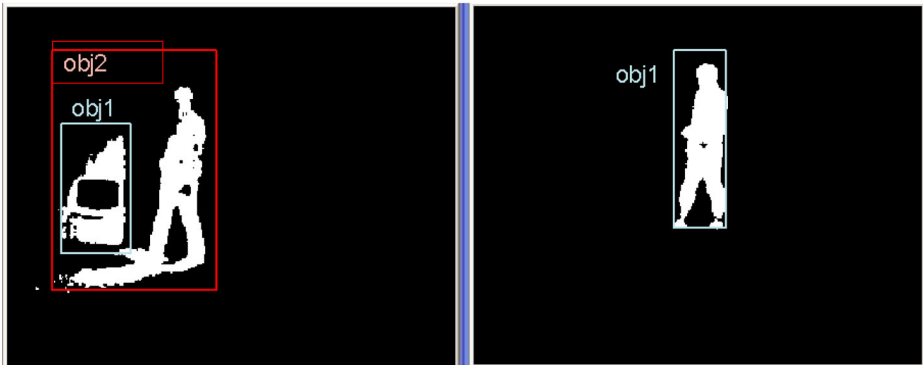
(a) result in time t (b) local tracker at $t + 1$ (c) fused result in $t + 1$

Fig. 8 Snapshot of segmentation result in successive frames. The *color bounding box* shows the state in local trackers, we see how the second tier of the tracking correct the wrong estimation from local trackers in the first tier

background subtraction, we are able to get a much better segmentation shown in Fig. 8c. The new information is passed back to the local and combined trackers to update their states. The details of the tracking and background subtraction are provided in the next subsections.

6.1 Robust tracking

Each tier of the two-tier tracking process consists of simple trackers at two different levels—the local level and combined level. The local tracker tracks the objects' bounding box and velocity. The velocity is updated at a fixed adaption rate α using the formula below:

$$v_t = \alpha v_{t-1} + (1 - \alpha) \hat{v}_t \quad (6)$$

where v_{t-1} is the velocity state from the previous time and \hat{v}_t is the current observed velocity. A blob association process is used to associate each observed blob to the closest track within its tracking gate. A candidate track is established for each non-associated blob and it will become a formal track after receiving observations continuously for a few frames. A track will be deleted from the tracker if no observations are associated with the track for an extended period of time. In the case of brief occlusion, the two objects will momentarily merge together into a single blob and reappear as two blobs once the occlusion has passed. Motion segmentation during occlusion is a well studied topic in computer vision and any competitive technique can be applied here. For our system, we rely on the scheme from our earlier work in [37] which uses the velocity of the bounding box and texture similarity for object segmentation.

The combined tracker attempts to smooth out the disparity of the object based on temporal history. The state of the combined tracker z_t is defined as follows:

$$z_t = \begin{cases} z_{t-1} & \text{if } |z_{t-1} - \frac{1}{d_{\max} - \hat{d}}| > \epsilon_1, \\ \alpha z_{t-1} + (1 - \alpha) \frac{1}{d_{\max} - \hat{d}} & \text{otherwise.} \end{cases} \quad (7)$$

The observation $\frac{1}{d_{\max} - \hat{d}}$ is computed based on the disparity output \hat{d} from Algorithm 1 and the maximum possible disparity d_{\max} . This quantity has been shown to be linearly proportional to the depth of the object from the rectified image plane [17, ch. 11.1.1]. This transformation is important as the transformed quantity provides a more uniform variation over the entire range of disparity for noise removal. As the disparity depends on the local blob segmentation which can be quite noisy, we implement a gating process so that the state will not be updated unless the new observation is within ϵ from the previous state. ϵ_1 is a design parameter that controls how conservative the combined tracker is. Implementing the gating process directly on the disparity will unfairly penalize objects that are close to the cameras. In the final step, we compute a robust estimate of disparity $\bar{d} = d_{\max} - \frac{1}{z_t}$ which is used to align corresponding blobs from the two camera views.

The two tiers of tracking are basically identical. The results of the first-tier tracking are used to provide an estimation of the registration parameter \bar{d} between the two camera views. After obtaining \bar{d} , the states of all trackers will be restored to the previous time instance while the two local background models will be aligned to

produce a more reliable foreground segmentation. The improved foreground blobs are then fed to the trackers in the second tier of tracking and the states of the tracker are updated with the new observations. In the next section, we provide the details of both the local and joint background/foreground segmentation.

6.2 Background modeling

There are three different background modeling processes in our system—two of which are performed locally at each camera view and one is performed after the registration process is completed. We first discuss the local background modeling at the thermal camera. Due to the significant temperature difference between the environment and human bodies, the detection of human figures in the thermal image is relatively straightforward. A Gaussian distribution is used to model the variation of ambient temperature measured at each pixel. We collect a number of frames of the background offline to estimate the mean $\mu_{\mathbf{x}}$ and the standard deviation $\sigma_{\mathbf{x}}$ of the ambient temperature at each pixel location \mathbf{x} . By applying this model to an incoming image, a probability map can be generated for foreground detection. The foreground label at each pixel in the thermal image is determined by the following formula:

$$\text{foreground label } L(\mathbf{x}) = \begin{cases} 1 & (T(\mathbf{x}) - \mu_{\mathbf{x}})^2 > \epsilon_2 \cdot \sigma_{\mathbf{x}}^2 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

where $T(\mathbf{x})$ is the temperature intensity at \mathbf{x} . ϵ_2 is an empirically-determined threshold chosen to produce a negligible miss detection rate—this usually results in a fairly accurate human figure segmentation but may occasionally includes background objects such as furniture that are momentarily at a higher temperature after coming in contact with human.

For the local processing of color image, we adopt a non-parametric adaptive background modeling algorithm from [24]. Under this model, the background model at each pixel is described by a list of codewords. Each codeword includes a color vector $C_b(\mathbf{x}) = (R_b(\mathbf{x}), G_b(\mathbf{x}), B_b(\mathbf{x}))$ and a brightness range $(\check{I}(\mathbf{x}), \hat{I}(\mathbf{x}))$. The brightness range is key in removing shadows and highlights, which are major sources of noise for background subtraction in color frames. Shadows and highlights usually have a high brightness variation but a small color variation when compared with the background model. As such, the brightness range is narrowly set so as to fully eliminate these artifacts, possibly at the expense of a less accurate human segmentation. Using this codebook, an incoming color pixel $C(\mathbf{x}) = (R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))$ is classified as background if it matches any codeword based on the following criteria:

$$\text{colorDist}(C(\mathbf{x}), C_b(\mathbf{x})) \leq \epsilon_3 \text{ and } \check{I}(\mathbf{x}) \leq I(\mathbf{x}) \leq \hat{I}(\mathbf{x}) \quad (9)$$

where the color distance is defined as

$$\text{colorDist}(C(\mathbf{x}), C_b(\mathbf{x})) = \sqrt{\|C(\mathbf{x})\|^2 - \frac{\langle C_b(\mathbf{x}), C(\mathbf{x}) \rangle}{\|C_b(\mathbf{x})\|^2}},$$

$I(\mathbf{x}) = 0.3R(\mathbf{x}) + 0.59G(\mathbf{x}) + 0.11B(\mathbf{x})$ is the luminance value, $\langle \cdot \rangle$ is the inner product, and ϵ_3 is an empirically-determined parameter.

After the local foreground extraction processes at both cameras, the results are passed to the first tier tracking system to compute the registration parameters as described in Section 6.1. After aligning each pair of corresponding blobs, we again

run the two background subtraction processes with adjusted parameters under the aligned coordinates system. A pixel is declared as foreground if both processes return foreground. Full details of the joint background subtraction are given in Algorithm 2.

Algorithm 2: Joint thermal-*RGB* background subtraction algorithm

Input: Registered thermal image $T(\mathbf{x}')$ where \mathbf{x}' is aligned to \mathbf{x} in the color image; color image $C(\mathbf{x})$; background color codebook $C_b(\mathbf{x})$; thermal background parameter $(\mu_{\mathbf{x}'}, \sigma_{\mathbf{x}'})$; threshold parameters $\epsilon'_2, \epsilon_3, \hat{I}'$ and \hat{I}

Output: foreground label $L(\mathbf{x})$

```

foreach pair of corresponding  $\mathbf{x}$  and  $\mathbf{x}'$  do
  |  $L(\mathbf{x}) = 1$ ;
  | if  $(T(\mathbf{x}') - \mu_{\mathbf{x}'})^2 \leq \epsilon'_2 \sigma_{\mathbf{x}'}$  in thermal image then
  | |  $L(\mathbf{x}) = 0$ ;
  | else
  | | foreach codeword in the background codebook at  $\mathbf{x}$  do
  | | | if  $\text{colorDist}(C_{\mathbf{x}}, C_b) \leq \epsilon_3$  and  $\hat{I}' \leq I(\mathbf{x}) \leq \hat{I}$  then
  | | | |  $L(\mathbf{x}) = 0$ ;
  | | | | break;
  | | end
  | end
end

```

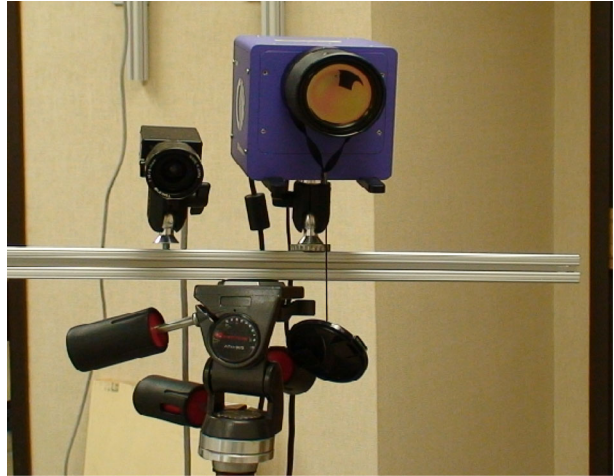
Notice that in Algorithm 2, a new set of threshold parameters are used for background subtraction. The reasons behind such changes are as follows:

1. The threshold ϵ_2 used on the thermal image is raised to a higher value ϵ'_2 so that all warmed background objects are eliminated leaving only human figures at higher temperature.
2. The luminance range $(\hat{I}(\mathbf{x}), \hat{I}'(\mathbf{x}))$ used on the color image is uniformly expanded to (\hat{I}', \hat{I}) so as to produce a more accurate human segmentation. The more lenient brightness range will not introduce any false positives because the subsequent fusion with the thermal foreground will guarantee that any shadow or highlights introduced will be eliminated.

7 Experimental results

Our system consists of a Unibrain™ Fire-i 400 video camera and a ElectroPhysics™ PV320 thermal camera. The two cameras are fixed in a horizontal bar and put next to each other, as shown in Fig. 9. The system runs on a Shuttle computer with 2 GB memory and Athlon Dual core 3800+ CPU at 2.0 GHz. Both cameras capture images at resolution 320×240 . Our single-thread non-optimized implementation of the system can process video frames at 12.8 frame per second.

We have conducted a number of experiments to highlight the weakness of existing approaches and demonstrate the performance of our proposed system. As our primary contributions reside within geometric fusion, the techniques that we should quantitatively compare with fall in three different categories as described in Section 2. They are optical fusion, full 3D reconstruction, and image warping. Optical fusion techniques require specialized optical hardware to align the two modalities

Fig. 9 System setup

and have significant limitations in the choice of cameras. Full 3D reconstruction involves reconstructing the 3D scenes using dense point correspondences which cannot be realized in real-time. Our proposed system can run in real-time without any specialized hardware and the only compatible approach to perform a reasonable comparison is image warping. As such, we focus on comparing our system with image warping.

In the first experiment, we demonstrate the weakness of image warping using simulation. The simulation is based on a typical stereo camera projection models with 8 mm focal length, 5.6 μm square pixel size, and standard VGA (640 \times 480) resolution with a baseline separation of 100 mm. We consider five different scenarios, each characterized by difference ranges of object distances: close-up, conference, indoor surveillance, outdoor mid-range, and outdoor long-range. For each scenario, we first randomly draw 20 3D points within the distance range as calibration points. We then use calibration tools in Mathematica [41], which includes standard RANSAC and global error minimization tools, to calculate the homography between the two camera views. Afterwards, we randomly sample 500 3D points within the range, project them onto one camera view, and map them into another camera view using the homography. The registration error is calculated by comparing the reprojected points with the direct projection of the 3D points onto the second camera using its camera projection matrix. The registration errors under different scenarios are presented in Table 2.

Table 2 Reprojection error using image warping method in different scenarios

	Mean error (in pixels)	Error standard deviation(in pixels)
Close up (0.3–1 m)	224.5	204.3
Conference (1–3 m)	16.70	13.93
Indoor surveillance (3–10 m)	4.674	3.978
Outdoor mid-range (10–20 m)	1.339	0.8405
Outdoor long-range (20–30 m)	0.3301	0.2015

Based on the simulation results, we conclude that image warping methods are not suitable for any indoor applications where the objects are less than 10 m away from the cameras. In the second experiment, we validate the simulation results using a simple indoor sequence. Figure 10 shows two scenarios with different object depths. For each scenario, we show the thermal image and the registration of the thermal blobs (red objects) onto the color image using either image warping or our proposed method in Section 5. In the first row of Fig. 10, we can see that both methods have comparable performances when the calibration points for image warping are chosen roughly at the same depth as the object. However, when the object moves away from the camera in the second row of Fig. 10, the single homography in image warping can no longer accurately register the thermal blob with the color image. On the other hand, our registration algorithm can successfully align the two views regardless of the variation in object depth.

In the third experiment, we show the effectiveness of our combined tracker over the segmentation results using either thermal or visible-light camera alone. For the color background model, we use the implementation from opencv library [3]. We choose the default parameters during the local processing, and then use a relaxed brightness range of $\check{I}' = 10$ and $\hat{I}' = 20$ when combined with registered results from the thermal camera. For the thermal background model, we use $\epsilon_2 = 2$ in the first phase and raise it to $\epsilon_2' = 3$ to obtain the best visual segmentation. The threshold parameter ϵ_1 used in the gating process of the combined tracker is set at 10. Figure 11 is a snapshot of the tracking result. The thermal background subtraction in Fig. 11c shows reasonable segmentation but still has small parts missing due to occlusion and low temperature appurtenance. The color background subtraction in Fig. 11d suffers from illumination changes and shadows. All of these problems are solved by the combined tracker as shown in Fig. 11e. The entire video sequence used in the second

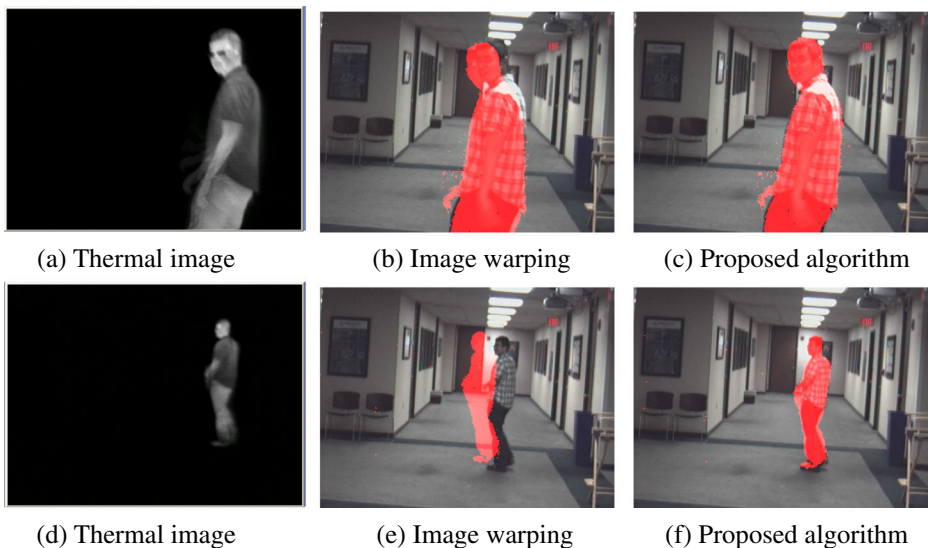


Fig. 10 Performance comparison of registering thermal images to color images using image warping and the proposed algorithm

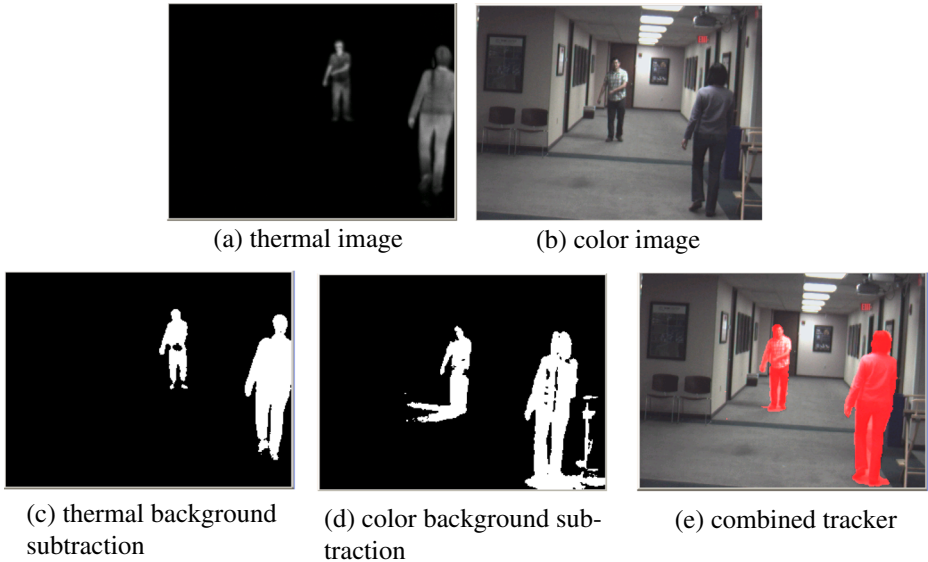


Fig. 11 The result produced by our proposed system in **e** shows the best results over using thermal camera alone in **c** or video camera alone in **d**

and the third experiments along with the segmentation results from different schemes can be found in <http://vis.uky.edu/cheung/MTA2012/hallway/hallway.html>.

In the fourth experiment, we perform quantitative comparisons of different foreground segmentation schemes using ground-truth data. Figure 12 is a snapshot of the footage comparing our proposed system to background subtraction using only color or thermal camera. The full video can be found in <http://vis.uky.edu/~cheung/MTA2012/whiteboard/whiteboard.html>. Twelve frames are randomly chosen and hand-segmented with an interactive graph-cut software [25]. We then compare the accuracy of our proposed algorithm against color background segmentation. In Fig. 13, the leftmost image is the ground truth segmentation. The middle and the rightmost images are color segmentation and fused segmentation results overlaid with the ground truth. The pink-color regions correspond to correct classification, red-colored regions to the false negatives and blue-color regions to false positives. The average accuracy for the twelve frames are tabulated in Table 3. We can see that the fused segmentation algorithm admits a very low false negative rate (1.6 %) while keeping the false positive rate at the same level as the color segmentation.

In the last experiment, we demonstrate the performance of our proposed system using a complicated sequence with three human subjects and multiple instances of occlusions. The one-minute long sequence has three individuals, originally sitting in three distinct locations. The individual in the middle first stood up and started writing on a small whiteboard. Afterwards, this individual walked behind the whiteboard and another person and exit from the left edge of the frame. The person near the right edge then stood up and followed the first individual to leave the scene. A snapshot of the sequence is shown in Fig. 14: the top left image is from the original video; the top-right and the bottom-left foreground masks are obtained by applying background subtraction on the thermal camera and the visible-light camera respectively. We can

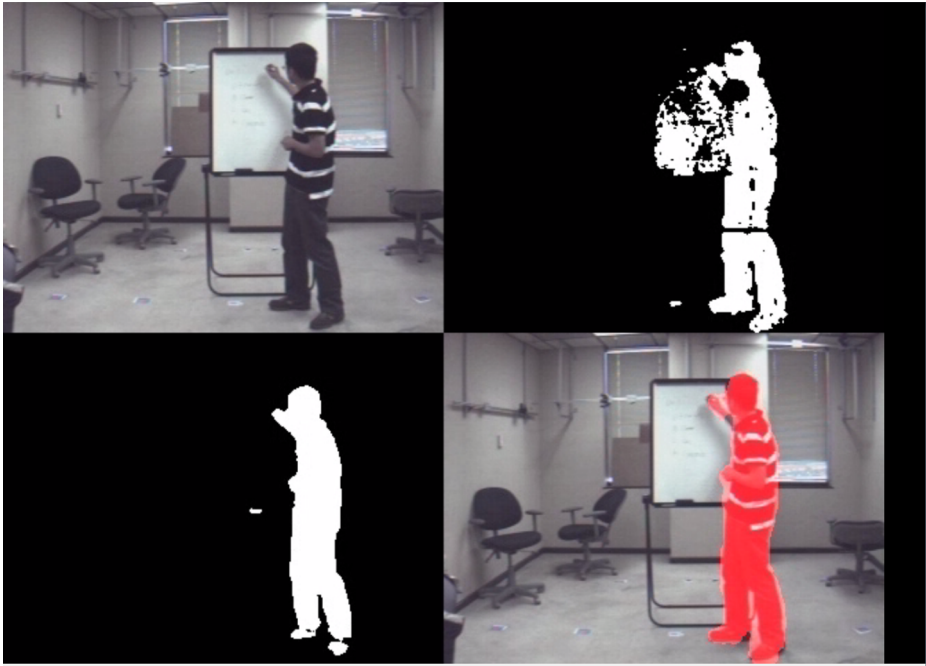


Fig. 12 Ground-truth sequence: the *top left image* is from the visible-light camera; the *top right* and the *bottom left* are the color and thermal background subtraction results respectively. The *bottom right* is the result obtained with our fused system

see that while the thermal mask produces a more accurate silhouette of the human figure, it erroneously includes part of the chair as the foreground—the chair is still warm from being sat on by the same individual. The bottom right image shows an interesting application of the fused segmentation mask obtained by our proposed algorithm—the mask is used to obfuscate the appearance of two individuals in the scene in order to protect their identity. Two different types of obfuscation are used. For the middle individual, the foreground blob is filled with a solid red color and completely covers the color texture information. Such type of obfuscation is useful in hiding the identity while providing information about the action. The shape of



Fig. 13 Quantitative measurement of the segmentation result. *Left*: ground truth; *middle*: color segmentation; *right*: fused segmentation

Table 3 Segmentation error in single camera system and fused camera system

	False positive (type I error)	False negative (type II error)
Color segmentation	0.2071	0.0965
Fused segmentation	0.1857	0.0160

this foreground blob is more accurate than the corresponding blobs from either thermal and visible camera. For the left individual, the foreground blob is filled with the background information making the individual completely transparent. This form of complete object removal has been demonstrated to provide the ultimate protection of visual privacy in surveillance and teleconference [38]. In Fig. 15, we show another frame to highlight the occlusion handling of our proposed algorithm. The segmentation between the two persons on the right is quite accurate except for the lower left edge of the person in the back is slightly enlarged to include part of the person in front. The original sequence and the obfuscated sequence using the fused segmentation mask can be found in <http://vis.uky.edu/~cheung/MTA2012/privacy/privacy.html>.



Fig. 14 Using thermal and visible light camera fusion algorithm to improve human segmentation with applications in privacy protection. The *top left image* is from the original video, the *bottom left image* is the background subtraction from thermal camera, and the *top right* is the segmentation result from visible-light camera. The *bottom right image* is the final result by removing the person on the left and obfuscating the person in the middle, both benefited from an accurate fused segmentation result



Fig. 15 The *left image* is the original and the *right image* shows the occlusion handling of our algorithm. The segmentation between the two persons is quite accurate except for the *lower left* edge of the person in the back is slightly enlarged to include part of the person in front

8 Conclusions

In this paper, we have presented a robust human segmentation system by fusing visible-light and thermal imagery. After a simple calibration procedure, blob-wise registration can be achieved by estimating the disparity of each corresponding blob-pair in real-time. The estimation of registration parameters is further improved by a two-tier tracking algorithm. The segmentation under a fused tracking and background subtraction system shows significant improvements over that of using either modality alone. In our current implementation, the temporal inference of the disparity is performed using a simple weighted averaging together with a gating process. A more sophisticated tracker such as particle filter may be used to estimate the disparity under a probabilistic framework. Finally, our current implementation separates overlapping blobs during occlusion based only on texture information. If the two objects have different depths, the distribution of the measured disparities should be multi-modal. One can conceivably use a statistical test to determine the number of modes and then use the different modes to identify separate homographies. Such approach should produce a more accurate registration than the current approach.

Acknowledgements We would like to thank the anonymous reviewers and the guest editors for their valuable comments.

Appendix

A Proof of Theorem 31

Since the homography matrix H' is up to scale, we can assume it is in the form of

$$H' = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}$$

According to the definition of image rectification, epipoles of the two image is at infinity and in the form of $[1\ 0\ 0]^T$ and $[a\ 0\ 0]^T$ also subject to the homography. Plug them in (3) we have

$$a_{11} = a$$

$$a_{21} = 0$$

$$a_{31} = 0$$

since

$$y'_2 = \frac{a_{22}y'_1 + a_{23}}{a_{32}y'_1 + 1} = y'_1$$

the following equation will always hold,

$$a_{32}y_1^2 - (a_{22} - 1)y_1 - a_{23} = 0$$

Therefore, all the coefficients for different order have to be zero. We have $a_{32} = 0$, $a_{22} = 1$, $a_{23} = 0$.

References

1. Beyan C, Yigit A, Temizel A (2011) Fusion of thermal-and visible-band video for abandoned object detection. *J Electron Imaging* 20:033,001
2. Bouguet JY (2005) Matlab camera calibration toolbox. Online at http://www.vision.caltech.edu/bouguetj/calib_doc/
3. Bradski G, Kaehler A (2008) *Learning openCV*. O'Reilly Media Press
4. Brown D (1966) Decentering distortion of lenses. *Photogramm Eng* 32(3):444–462
5. Bunyak F, Palaniappan K, Nath S, Seetharaman G (2007) Geodesic active contour based fusion of visible and infrared video for persistent object tracking. In: *IEEE workshop on applications of computer vision, WACV'07*. IEEE, pp 35–35
6. Cevher V, Sankaranarayanan A, McClellan J, Chellappa R (2007) Target tracking using a joint acoustic video system. *IEEE Trans Multimedia* 9(4):715–727
7. Chen S, Zhu W, Leung H (2008) Thermo-visual video fusion using probabilistic graphical model for human tracking. In: *IEEE International Symposium on Circuits and systems, ISCAS 2008*. IEEE, pp 1926–1929
8. Chen X, Davis J, Slusallek P (2000) Wide area camera calibration using virtual calibration objects. In: *Conference on computer vision and pattern recognition, vol 2*. IEEE, pp 520–527
9. Chen Y, Han C (2008) Night-time pedestrian detection by visual-infrared video fusion. In: *7th World congress on intelligent control and automation, WCICA 2008*. IEEE, pp 5079–5084
10. Conaire C, O'Connor N, Smeaton A (2008) Thermo-visual feature fusion for object tracking using multiple spatiogram trackers. *Mach Vis Appl* 19(5):483–494
11. Conaire CO, Cooke E, O'Connor N, Murphy N, Smeaton AF (2005) Fusion of infrared and visible spectrum video for indoor surveillance. In: *Proc. of international workshop on image analysis for multimedia interactive services*. Montreux, Switzerland
12. Cramer H, Scheunert U, Wanielik C (2003) Multi sensor data fusion using a generalized feature model applied to different types of extended road objects. In: *6th international conference of information fusion, vol 1*, pp 2–10
13. Davis J, Sharma V (2007) Background-subtraction using contour-based fusion of thermal and visible imagery. *Comput Vis Image Underst* 106(2):162–182
14. Davis JW, Sharma V (2005) Fusion-based background-subtraction using contour saliency. In: *CVPR '05: proceedings of the 2005 IEEE computer society conference on Computer Vision and Pattern Recognition (CVPR'05)—workshops*. IEEE Computer Society, Washington, DC, p 11. doi:10.1109/CVPR.2005.462

15. Denman S, Lamb T, Fookes C, Chandran V, Sridharan S (2010) Multi-spectral fusion for surveillance systems. *Comput Electr Eng* 36(4):643–663
16. Elmenreich W (2002) Sensor fusion in time-triggered systems. Ph.D. thesis, Vienna University of Technology
17. Forsyth DA, Ponce J (2002) *Computer vision: a modern approach*. Prentice Hall. <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0130851981>
18. Goubet E, Katz J, Porikli F (2006) Pedestrian tracking using thermal infrared imaging. Mitsubishi Electric Research Laboratories, Technical Report, TR2005-126
19. Hall DL, McMullen SAH (2004) *Mathematical techniques in multisensor data fusion* (Artech House Information Warfare Library). Artech House, Inc., Norwood, MA, USA
20. Han J, Bhanu B (2007) Fusion of color and infrared video for moving human detection. *Pattern Recogn* 40(6):1771–1784. doi:10.1016/j.patcog.2006.11.010
21. Hartley R, Reid I (2004) *Multiple view geometry in computer vision*. Cambridge University Press
22. Hartley RI (1999) Theory and practice of projective rectification. *Int J Comput Vis* 35(2):115–127. doi:10.1023/A:1008115206617
23. Johnson M, Bajcsy P (2008) Integration of thermal and visible imagery for robust foreground detection in tele-immersive spaces. In: 11th international conference on information fusion, 2008. IEEE, pp 1–8
24. Kim K, Chalidabhongse TH, Harwood D, Davis L (2005) Real-time foreground-background segmentation using codebook model. *Real-Time Imaging* 11(3):172–185. doi:10.1016/j.rti.2004.12.004. <http://www.sciencedirect.com/science/article/B6WPR-4FV362T-1/2/64a99673b255f07c51631846435c3ba5>. Special issue on video object processing
25. Kolmogorov V, Zabih R (2001) Computing visual correspondence with occlusions via graph cuts. Tech. rep., Cornell University, Ithaca, NY, USA
26. Krotosky S, Trivedi M (2006) Multimodal stereo image registration for pedestrian detection. In: Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, pp 109–114
27. Kumar P, Mittal A, Kumar P (2006) Fusion of thermal infrared and visible spectrum video for robust surveillance. In: ICCVGP06, pp 528–539
28. Lee S, McHenry K, Kooper R, Bajcsy P (2009) Characterizing human subjects in real-time and three-dimensional spaces by integrating thermal-infrared and visible spectrum cameras. In: IEEE International Conference on Multimedia and Expo, ICME 2009. IEEE, pp 1708–1711
29. Leykin A, Hammoud R (2010) Pedestrian tracking by fusion of thermal-visible surveillance videos. *Mach Vis Appl* 21(4):587–595
30. Llinas J, Bowman C, Rogova G, Steinberg A, Waltz E, White F (2004) Revisiting the JDL data fusion model II. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.2996>
31. St-Laurent L, Maldague X, Prévost D (2007) Combination of colour and thermal sensors for enhanced object detection. In: 10th international conference on information fusion, 2007. IEEE, pp 1–8
32. St Onge P, Bilodeau G (2007) Visible and infrared sensors fusion by matching feature points of foreground blobs. In: ISVC07, pp II: 1–10
33. Steinberg AN, Bowman CL (2004) Rethinking the JDL data fusion levels. In: NSSDF conference proceedings. JHAPL
34. Svoboda T, Martinec D, Pajdla T (2005) A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments* 14(4):407–422
35. Torresan H, Turgeon B, Ibarra-Castanedo C, Hebert P, Maldague XP (2004) Advanced surveillance systems: combining video and thermal imagery for pedestrian detection. In: Burleigh DD, Cramer KE, Peacock GR (eds) *Thermosense XXVI*, vol 5405. SPIE, pp 506–515. doi:10.1117/12.548359. <http://link.aip.org/link/?PSI/5405/506/1>
36. Ulusoy I, Yuruk H (2011) New method for the fusion of complementary information from infrared and visual images for object detection. *IET Image Process* 5(1):36–48
37. Venkatesh MV, Cheung SC, Zhao J (2008) Efficient object-based video inpainting. *Pattern Recogn Lett: Special issue on video-based object and event analysis*. doi:10.1016/j.patrec.2008.03.011
38. Venkatesh MV, Zhao J, Profitt L, Cheung SCS (2009) Audio-visual privacy protection for video conference. In: Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, ICME'09. IEEE, Piscataway, NJ, pp 1574–1575. <http://portal.acm.org/citation.cfm?id=1698924.1699317>

39. Volfson L (2006) Visible, night vision and ir sensor fusion. In: 9th international conference on information fusion, pp 10–13:1–4
40. White F (1988) A model for data fusion. In: 1st national symposium on sensor fusion
41. Wolfram Research I (2010) Mathematica edition: version 8.0. Champaign, IL
42. Wu Q, Boulanger P, Bischof WF (2008) Bi-layer video segmentation with foreground and background infrared illumination. In: MM '08: Proceeding of the 16th ACM international conference on multimedia. ACM, New York, NY, pp 1025–1026. doi:[10.1145/1459359.1459562](https://doi.org/10.1145/1459359.1459562)
43. Zhao J (2011) Camera planning and fusion in a heterogeneous camera network. Ph.D. thesis, University of Kentucky
44. Zhao J, Cheung SC (2009) Human segmentation by fusing visible-light and thermal imagery. In: International Conference on Computer Vision workshops (ICCV workshops). IEEE, p 1185
45. Zhou H, Taj M (2008) Cavallaro: target detection and tracking with heterogeneous sensors. *IEEE J Sel Topics Signal Process* 2(4):503–513



Jian Zhao received the B.S. degree in electric engineering in Zhejiang University, Hangzhou, China, in 2005, and the Ph.D. degree from University of Kentucky, in 2011. He has been a software developer in camera and imaging team in Microsoft Corporation since 2011. His areas of interest include camera network, statistical optimization, computer vision, machine learning and image processing.



Sen-ching S. Cheung is an associate professor from the Department of Electrical and Computer Engineering of the University of Kentucky (UKY). He also has a joint appointment with the UKY Center of Visualization and Virtual Environments. Before joining UKY in 2004, he was a computer scientist in the Scientific Data Mining group at Lawrence Livermore National Laboratory. Samson got his Ph.D. from University of California, Berkeley in 2002. His work spans a number of different areas in multimedia including video copy detection, data mining, video surveillance, privacy protection, encrypted-domain signal processing, and computational multimedia for therapy. He is an associated editor of *IEEE Transactions of Multimedia*, *Signal Processing: Image Communications*, *Statistical Analysis and Data Mining*, and *EURASIP Journal on Information Security*. He is a senior member of both IEEE and ACM.