

# Human Segmentation by Fusing Visible-light and Thermal Imaginary

Jian Zhao, and Sen-ching S. Cheung  
Center for Visualization and Virtual Environments  
University of Kentucky

Jian.Zhao@uky.edu, cheung@engr.uky.edu

## Abstract

*This paper describes a system for robust segmentation of human in video sequences by fusing the visible-light and thermal imaginary. The system first performs a simple calibration procedure to rectify the two camera views without knowing the cameras' intrinsic characteristics. Then a blob-to-blob homography is learned on-the-fly by estimating the disparity of each blob so that a pixel level registration can be achieved. The multi-modality information is then combined under a two-tier tracking algorithm and a unified background model to attain precise segmentation. Preliminary experimental results shows significant improvements over existing schemes under various difficult scenarios.*

## 1. Introduction

For decades, the problem of segmenting human in video sequence has been a central issue in computer vision. Despite its popularity, it remains to be a challenging problem because visual appearances are subjected to occlusion, illumination change, highlight, shadow and color confusion. Recently, systems using sensors of different modalities have been proposed to improve human video segmentation results. Among them, thermal infrared sensors are particularly popular as human bodies usually present different temperature characteristics from that of the environment.

The introduction of infrared camera provides both opportunities and challenges. On one hand, the extra modality provided by infrared camera offers supplementary information about the human body and thus should potentially improve the classification. On the other hand, the information from the visible-light and thermal cameras are not spatially aligned and the new modality can bring new channels of noises which could further confuse the classifier. In this paper, we tackle the registration problem by learning blob-to-blob homographies according to the disparity of each blob to attain a pixel level registration. The multi-modality information is then combined under a two tier tracking algorithm

and an unified background model to mitigate segmentation noise from either modalities.

Most existing systems solve the registration problem by either optical fusion[14, 13] or image warping[4, 10, 6]. The optical fusion methods use specially-designed optical device to merge the optical axes of the two cameras so that the two cameras can see exactly the same view. Despite its computationally efficiency and registration accuracy, it suffers from high manufacture costs. The image warping method calculates a homography matrix during the calibration procedure by point matching. The homography is used to warp segmentation results from one modality to the other. The same homography matrix is applied to all objects in the scene regardless of their depths. Due to over-simplification from 3D projection to 2D mapping, the performance dwindles when there exists significant variation of objects' depth in the sequences. Some systems such as [6] and [4] adopt additional search procedures to correct the registration error. Those searching algorithm significantly diminish the algorithmic efficiency and will fail when either of two views has defective foreground segmentation.

In [11], instead of performing the warping in image scale, the authors align the two foreground blobs by identifying shape feature points to estimate the homography, through either skeleton or discrete curve evolution. Although our algorithm follows a similar idea, we further exploit the camera model and reduce the number of parameters need to be learned from eight to just one. Furthermore, by including the parameter into a tracker, we make full use of the temporal information to infer the homography so that the registration would still work in noisy frames where no valid observation is available.

Traditional sensor fusion techniques are pervasively used to improve the segmentation from information obtained by multiple sensor. Kumar et al. [10] adopt fuzzy logic to evaluate the confidence from each sensor. Han and Bhanu [6] compares different rules under Bayesian framework, while combined trackers such as Kalman filter and Particle filter are used to fuse the multi-modality observations in [3, 15, 2]. Alternatively, the fusion can be per-

formed from the image perspective. In [14], image segmentation is performed using the output of thermal camera as seeds. Morphological operations are adopted in [4] and [12]. We combine the thermal and color image into a fused non-parametric background model similar to [9]. Here, we do not claim to make contribution to the theoretical fusion problem but argue that a simple two-tier background modeling with adaptive parameters in each tier is good enough to provide efficient and accurate segmentation result.

In summary, the contributions of our system are, 1) By decomposing the homography matrix into rectified domain, we significantly reduce the complexity of parameter estimation; 2) by including the homography parameters as a state of a two-tier tracking system, the registration of multi-modality information becomes more robust with the help of temporal inference.

The rest of the paper is organized as follows. In Section 2, we explain our model for blob based registration and our algorithm to estimate the model parameters on the fly. Our whole system including robust tracking scheme and fused background modeling is shown in Section 3. Preliminary experimental results are shown in 4 and we conclude our paper in Section 5

## 2. Cross Camera registration

### 2.1. Camera model

We assume the ‘‘pinhole’’ camera model for both visible-light and thermal cameras. Under this model, any 3D point observed in one camera can be anywhere along the epipolar line in the other camera view. Therefore, there is no precise point-to-point mapping between the camera views. However, if we assume the points on the same human as co-plane, there does exist a bijective mapping between the points in correspondent blobs in the two cameras. That is, for any correspondent points  $\mathbf{X}_1$  and  $\mathbf{X}_2$  on the correspondent blobs in the two images, there is a linear mapping in the 2D homogeneous coordinate [7, ch.2] as follows

$$\mathbf{X}_1 = H\mathbf{X}_2 \quad (1)$$

where  $H$  is a  $3 \times 3$  matrix with eight degrees of freedom known as homography matrix. Unfortunately, the homography is variable with respect to the plane’s depth and pose [7, ch. 13.1]. Therefore, the homography matrix has to be adaptively estimated.

In 3D computer vision, it is common to rectify the camera views before estimating the scene structure so that the search for correspondent points can be significantly expedited. The rectification process finds linear mappings in homogeneous coordinate 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. Denote the rectification matrices for the two cameras

as  $H_1$  and  $H_2$ , the points after rectification in two images as  $\mathbf{X}'_1$  and  $\mathbf{X}'_2$ , and the homography between the two rectified image planes as  $H'$ , we have,

$$\begin{aligned} \mathbf{X}'_1 &= H_1\mathbf{X}_1 \\ \mathbf{X}'_2 &= H_2\mathbf{X}_2 \\ \mathbf{X}'_2 &= H'\mathbf{X}'_1 \end{aligned} \quad (2)$$

From Equation (1) and (2) we have,

$$H = H_1^{-1}H'H_2 \quad (3)$$

Theorem 2.1 presents the form of homography matrix in rectified domain. We can see that after decomposing the homography matrix  $H$  into rectified image plane, the number of parameters need to be estimated on line reduces from eight to three.

**Theorem 2.1 (Homography in rectified images)** *The homography  $H'$  in rectified image domain is in the form*

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

The proof of Theorem can be found in Appendix A.

If the person is not too close to the camera, we can further assume all parts of the same person’s silhouette have the same depth to the camera. This is a valid assumption because in common surveillance scenarios, the person is usually several meters away from the cameras, and the depth variation of the different parts of human body is within several centimeters, which is very small portion of the overall depth. From [5, ch. 11.1.1] We can see our constant depth assumption induces constant disparity for correspondent blobs. Based on this assumption, we have  $x'_2 - x'_1 = d$  for each correspondent blob. Combining with the homography matrix  $H'$  in Theorem 2.1, we have

$$(a_{11} - 1)x'_1 + a_{12}y'_1 + a_{13} - d = 0$$

which will always hold regardless the value of  $x'_1, y'_1$ . Thus all the coefficients for all variables must be identically zero and we have  $a_{11} = 1, a_{12} = 0$  and  $a_{13} = d$ . The homographies between correspondent blobs becomes a family depends only on one coefficient  $a_{13}$ , which is the disparity in rectified image.

### 2.2. Calibration procedure

In stereo vision, one needs to build a large 3D target or collect multiple shots of a plenary chessboard pattern to estimate the intrinsic and extrinsic parameters for the cameras. Due to the phenomenological differences of objects

in color and thermal images, this is not easy to implement. Alternatively, we carry out our calibration by only collecting the correspondent point pairs and use them to infer the geometry constraints. In fact, This is much simpler as we bypass the explicit estimation of cameras' intrinsic and extrinsic parameters such as focal length, aspect ratio, translation and rotation matrices, and directly obtain the rectification matrices from correspondent points. We stick an color tag to a round metal slice as a calibration object. As shown in Figure 1, the tag will be visible when heated. A color classifier using HSV color space based on Mixture of Gaussian model is trained to identify the tag in the video camera. Both cameras use a least square ellipse fitting algorithm to detect the center of the color tag and use them as the feature points for calibration. This is by no means the only method for calibration. Any tools which can provide unique correspondence between the two views can be used, such as an incandescent lighting bulb.

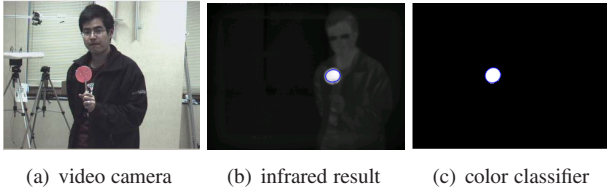


Figure 1. The pink tag in 1(a) is used as calibration object. 1(b) shows the fitting result in the thermal camera and 1(c) shows the classification and fitting result in the visible-light camera

Once the correspondent points are collected, we use Hartley's method for rectifying uncalibrated cameras [8]. The method first performs a RANSAC algorithm to remove the outliers, then uses the points correspondence to estimate the fundamental matrix. After finding the epipoles by decomposing the fundamental matrix, an iterative method is used to find two homographies which map the epipoles into infinity and minimize the mapping error given by the point correspondences. Note that after rectification, the rectified images will have the same resolutions regardless their original resolutions.

After performing the rectification, the disparity range is measured by projecting the calibration points into the rectified domain and finding the minimum and maximum of the difference between the  $x$  coordinate of the correspondent points. This will be used to help identifying the outliers during the on-the-fly registration stage.

### 2.3. Blob-wise Registration

Unlike most stereo vision systems, which use texture information to estimate the depth of the object, the thermal-visible camera pair does not share any similarity in terms of the texture captured. Under the assumption of constant

depth, Algorithm 1 estimates the disparity between corresponding blobs, based on the mode of the measured disparities between a large set of corresponding pairs of pixels. The algorithm uses the contours of human blobs as the pool of correspondences and utilizes the constraints obtained from previous calibration process to boost the estimation of registration parameters. The algorithm works as follows,

**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  $\bar{d}$  ;  
        Obtain  $H'$  by setting  $a_{13} = \bar{d}$  ;  
        Obtain  $H$  by Equation(3) ;  
    **else**  
        return failure;  
    **end**  
**end**

Algorithm 1: On line registration algorithm

Two processes are used to efficiently filter out the massive outliers. Firstly, when a scan line have unequal number of points in two views, it is discarded without calculating any disparity. This process helps to rule out some difficult situations due to occlusion or defective segmentation. As shown in Figure 2, the green disparity scanned by the green line is recorded in the histogram while the red line is not counted. Secondly, disparities out of the disparity range due to false point match are simply discarded. In Figure 2, the rightmost points of each image alone the blue line is a false match and is likely to be filtered out by disparity range.

### 3. Robust fusion via tracking and background modeling

In reality, there can be multiple blobs in the views and due to the inaccuracy in blob segmentation, it may not be easy to find correspondent blobs between the views. Sometimes, it is simply too hard to obtain a good estimation of the blob-wise disparity due to occlusion or defective segmen-

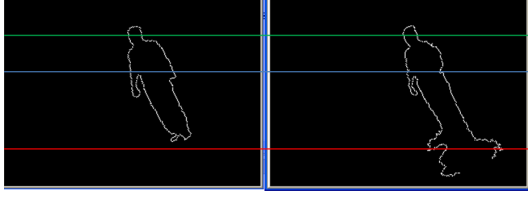


Figure 2. Simple scan line for ruling out some of the false correspondent points. Green line(upper) gives good match while red line (bottom) contains uneven number of points due to the shadow, which will be ruled out by our algorithm. Along the blue line (middle), the rightmost points of each image is a false match and is likely to be filtered out by disparity range.

tation. We handle these problems by designing a two-tier tracking scheme together with a joint background subtraction.

Firstly, background subtraction is performed individually to extract the blobs from each camera. These blob information are fed into individual trackers to detect long-existing objects and filter out possible false positives. A combined tracker is then used to match objects between the two camera views, calculate and track of the disparity of each object. Using the disparity estimated from combined tracker, the homography matrix can be calculated using Equation (3) so that matched object can be aligned to perform a joint background subtraction. Finally, the improved segmentations from the joint background subtraction are fed back to the trackers to improve estimation of the state of each tracker. The system flow chart is shown in Figure 3.

Figure 4 is a snapshot of this process. In the previous time instant shown in Fig. 4(a), there is only one object in each view. However, in the next time instance in Fig. 4(b), due to the split of the shadow with the human body, the visible-light camera has two blobs after background subtraction and the individual tracker mistakenly takes the shadow blob as the new observation. In the combined tracker, there is no observation of disparity because all the point pairs are filtered out by Algorithm 1. However, thanks to temporal inferencing, the disparity estimated by the tracker is still good. By a joint background subtraction, we are able to get much better segmentation shown in Fig. 4(c). The new information is passed back to the individual and combined trackers to improve the estimation of their states.

### 3.1. Robust tracking

Each tier of the tracking process consists of simple trackers at two different levels — the individual level and combined level. The individual tracker tracks the objects' bounding box and velocity. The velocity is updated at a fixed adaption rate  $\alpha$ .

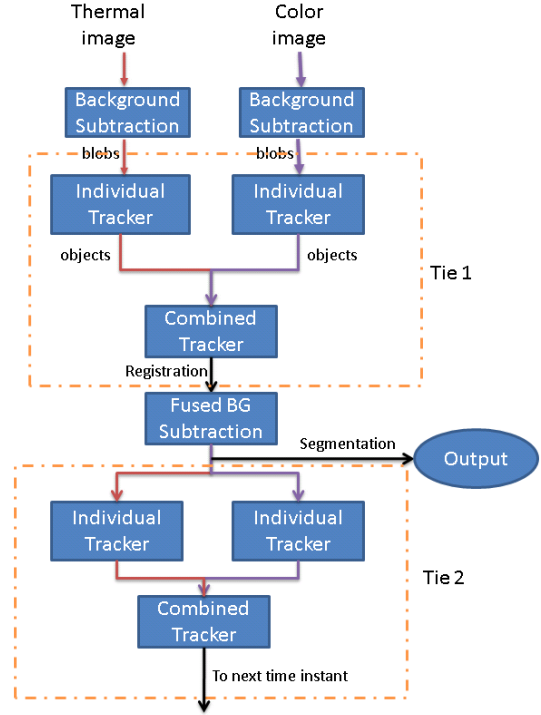


Figure 3. Block diagram

$$v_t = \alpha v_{t-1} + (1 - \alpha) \hat{v}_t \quad (4)$$

where  $v_{t-1}$  is the previous velocity,  $\hat{v}_t$  is the current observed velocity. The tracker also records the number of the times the object has been observed or missed to deal with new object emergence, occlusion and noisy observation. An object will be regarded as a new object only if it has been observed more than a number of times in successive frames; an object will be deleted from the list only if it has been lost observation in a number of successive frames.

The combined tracker attempts to infer the disparity of the object using observation from both camera views, which is calculated by Algorithm 1. The state of the combined tracker is calculated by

$$z = \frac{1}{D - \bar{d}} \quad (5)$$

where the  $\bar{d}$  is the disparity output from Algorithm 1 and  $D$  is the largest positive disparity during calibration. From [5, ch. 11.1.1]  $z$  is linearly proportional to the depth of the object to the rectified image plane. Since the observation of the disparity is much noisier than what of the individual sensor, we apply a “gating” process to rule out the apparent false estimation. If  $\|\hat{z}_t - z_{t-1}\| > \epsilon_1$ , the observation is discarded and  $z$  is updated with  $z_t = z_{t-1}$ , where  $\epsilon_1$  is a design parameter. When the observation is valid, the state  $z$  is updated similar to Equation(4).

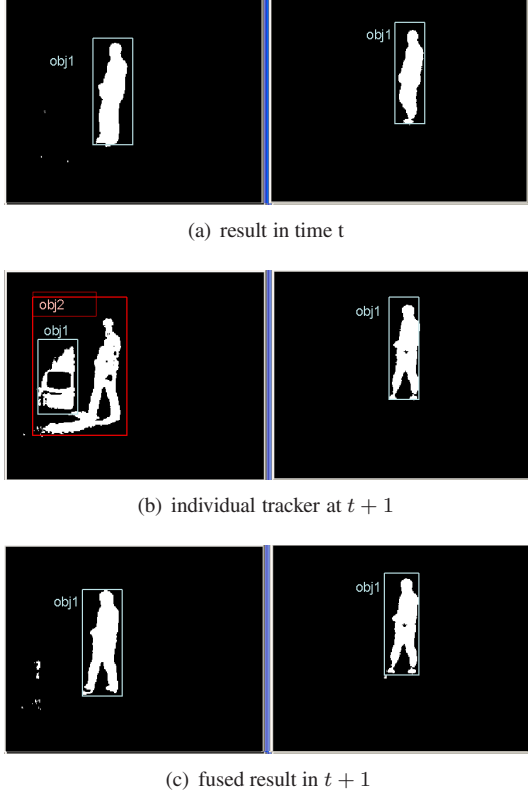


Figure 4. Snapshot of segmentation result in successive frames. The color bounding box shows the state in individual tracker, we see how the second tier of the tracking correct the wrong estimation from individual tracker in the first tier

The two tiers of tracking basically adopt the same process. However, the result of first tier is only used to provide an estimation of the registration between two camera views. After obtaining the fused segmentation result, the state are restored to previous time instance and the second tier of tracking is used estimate the state with higher accuracy.

### 3.2. Background modeling

There are three different background modeling processes in our system, two of which are performed individually in each camera view. Due to the significant temperature difference between the environment and human body, the detection of human in thermal image is relatively easy. Therefore, a static Gaussian model is used to model each pixel in the background. By collecting a fixed amount of background frames, the mean and variance ( $\mu, \sigma^2$ ) are calculated to model each background pixel. By applying this model to an incoming image, a probability map can be generated for foreground detection. The label for each pixel  $\mathbf{x} = (x, y)$

in thermal image is determined by,

$$\text{foreground label } l_{\mathbf{x}} = \begin{cases} 1 & (T(\mathbf{x}) - \mu_{\mathbf{x}})^2 > \epsilon_2 \cdot \sigma_{\mathbf{x}}^2 \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Where  $T(x, y)$  is the pixel intensity in thermal image,  $\epsilon_2$  is a fixed threshold.

For color image, we adopt a recent non-parametric adaptive background modeling algorithm [9]. In this model, background pixel is traced in a list of code word including a color vector  $\mathbf{v}_m$  and a brightness range  $(\check{I}, \hat{I})$ . For a test pixel vector  $C_{\mathbf{x}} = (R_{\mathbf{x}}, G_{\mathbf{x}}, B_{\mathbf{x}})$ , if it matched any code in the codebook, it is classified as background. A match is defined if

- Color distortion:  $colorDist(C_{\mathbf{x}}, \mathbf{v}_m) \leq \epsilon_3$ , where the color distortion function is defined as

$$\sqrt{\|C_{\mathbf{x}}\|^2 - \frac{\langle \mathbf{v}_m, C_{\mathbf{x}} \rangle^2}{\|\mathbf{v}_m\|^2}}$$

where  $\langle \cdot \rangle$  is the inner product and  $\epsilon_3$  is a design parameter.

- Brightness:  $\check{I} \leq I(\mathbf{x}) \leq \hat{I}$

When registered information from both cameras are available, we adapt the codebook background model for fused background subtraction based on two observations:

1. The infrared camera generally gives more confident classification. Specifically, in Tier 2, we can increase  $\epsilon_2$  to achieve close to zero false positive rate. Note that decreasing the false positive will decrease the detection rate as well. This is why we don't use it in Tier 1 because it will make the contour of the objects less recognizable.
2. Shadows and high lights are major source of noise in indoor surveillance for background subtraction in regular cameras, which usually have a high brightness variation but not much color distortion

Therefore, our fused background subtraction algorithm basically tightens the threshold for thermal image and uses it as a key reference and enlarges the range of each code word in order to include the shadow and high light, as shown in Algorithm 2

## 4. Experimental result

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 in Figure 5. The system runs on a Shuttle computer with 2GB memory and Athlon Dual core 3800+

**Input:** Registered thermal image  $T$ , color image  $I$ ,  $\mathbf{X}$ , codebook background  $\mathbf{v}_m, \tilde{I}, \hat{I}$ , thermal background parameter  $(\mu, \sigma)$ , thresholding parameter  $\epsilon_2, \epsilon_3$ , relaxed parameter  $\epsilon'_2, \tilde{I}', \hat{I}'$

**Output:** foreground label  $l$

```

foreach pixel in video camera do
  if  $(T(\mathbf{x}) - \mu_{\mathbf{x}})^2 > \epsilon'_2 \sigma_{\mathbf{x}}^2$  in thermal image then
    |  $l_{\mathbf{x}} = 1$ ;
  else
    |  $l_{\mathbf{x}} = 1$ ;
    foreach code  $\mathbf{v}_m, \tilde{I}', \hat{I}'$  in the code book do
      | if  $\text{colorDist}(C_{\mathbf{x}}, \mathbf{v}_m) \leq \epsilon_3$  and
      |  $\tilde{I}' \leq I(\mathbf{x}) \leq \hat{I}'$  then
        | |  $l_{\mathbf{x}} = 0$ ;
        | | break;
      | end
    end
  end
end

```

**Algorithm 2:** fused background subtraction algorithm



Figure 5. System setup

CPU at 2.0GHz. Both cameras capture images at resolution  $320 \times 240$ . Our single-thread unoptimized code runs at 12.8 fps, comparing with 7.5 fps in [11].

In the first experiment, we show that most of pervasively adopted image warping methods are not suitable for indoor surveillance application, where the depth of the target object varies in the scene. We can see from Figure 6, our method described in Algorithm 1 clearly outperform image warping. In the first row of Figure 6, we can see both methods work equally well when the calibration points are at the same depth of the object. However, when the depth of the object changed in the second row of Figure 6, the single homography registration in image warping is no longer accurate and the two blobs do not align. On the contrary, our registration algorithm can successfully register object regardless of its depth variation.

In the second experiment, we show the effectiveness of our combined tracker over the segmentation using the two modality separately. We adopted the OPENCV[1] library for codebook implementation and use its default parameters. In background subtraction in thermal image, the threshold  $\epsilon_2$  is set to 2 to get roughly the best visual segmentation. In the combined tracker,  $\epsilon_1 = 10$ ,  $\epsilon'_2 = 3$ ,  $\tilde{I}' = 20$ , and  $\hat{I}' = 10$ . Figure 7 is a snapshot of the tracking result. Comparing between Figure 7(c) and 7(d), the thermal image gives much better segmentation but still has some part missing due to occlusion and low temperature appurtenance, while the code book background subtraction in visible camera suffer from illumination changes and shadows. All of these problem can be solved in the fused tracker in Figure 7(e).

## 5. Conclusion

In this paper, we provide a robust human segmentation system by fusing video and thermal imagery. After a simple calibration procedure, a blob wise registration is achieved by estimating the disparity of each correspondent blobs on the fly. The estimation of registration parameters is further improved by temporal inferencing via a two-tier tracking algorithm. The segmentation under a fused background subtraction shows significant improvement over that of using either modality alone.

Currently, the background subtraction in fused image is only a union of individual modality with tightened thresholds. Further improvement can be obtained by fusing the two modalities under a specific human body model. Also, the inference of disparity using temporal information is performed by a simple weighted averaging together with a gating process. A more sophisticate tracker such as particle filter may be used to estimate the disparity under a probabilistic framework. Last but not least, our system can only segment the human bodies out of the background, it is interesting to see how to obtain separate segments when there is occlusion between multiple human blobs.

## A. Proof of Theorem 2.1

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

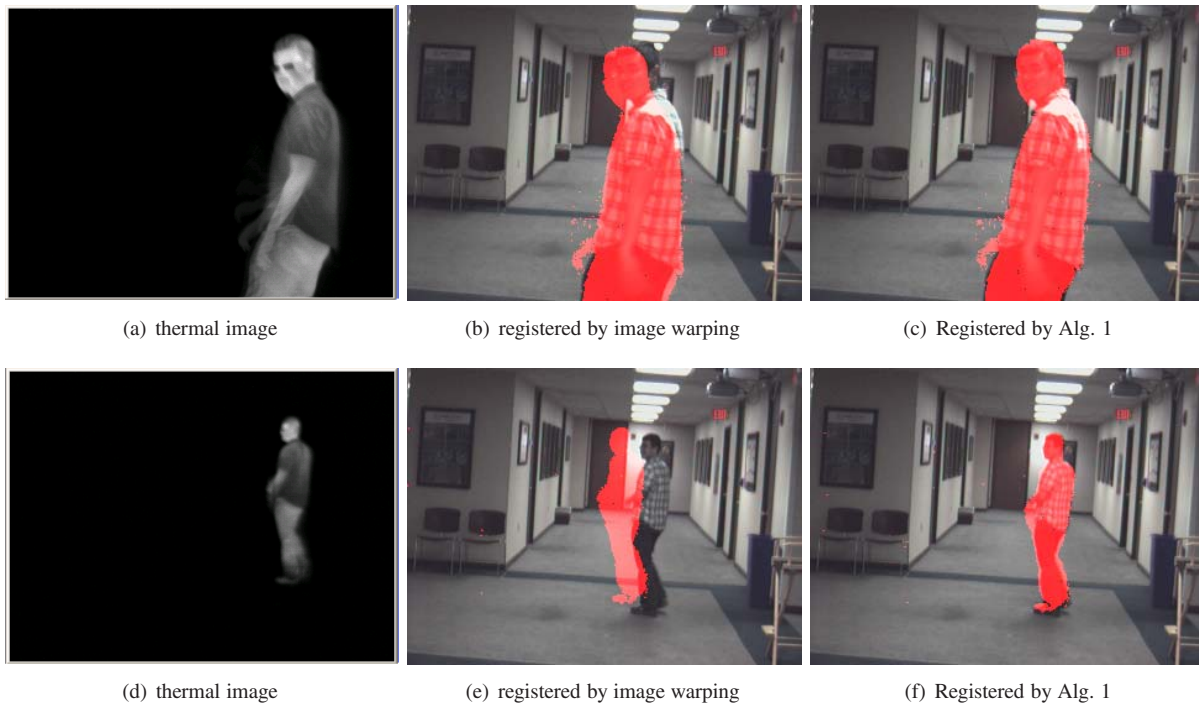


Figure 6. In the first row, we can see the registration between thermal image and regular image are both fine by using image warping and our method, shown in the red blob. However, in the second row, the image warping method fails when there is a depth variation

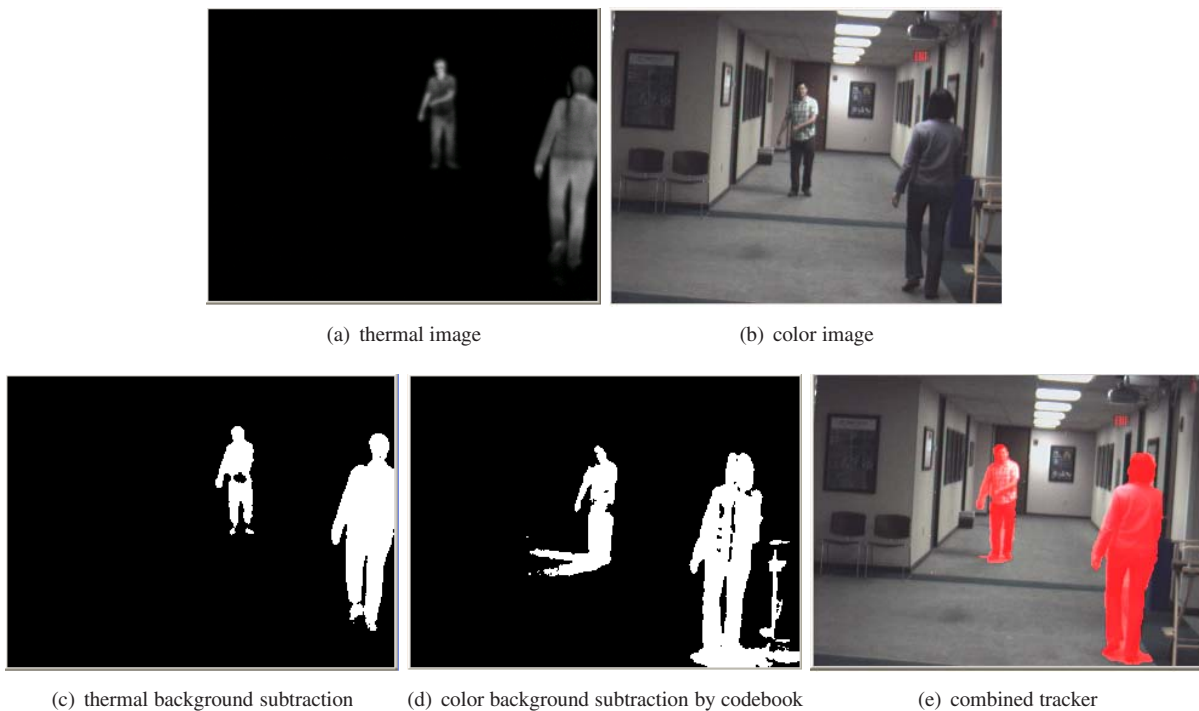


Figure 7. Comparison of our fused system with any single camera system. Our result in 7(e) shows best results over using thermal camera alone in 7(c) or video camera alone in 7(d)

Equation(2) we have

$$\begin{aligned}a_{11} &= a \\ a_{21} &= 0 \\ a_{31} &= 0\end{aligned}$$

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  $a_{32} = 0$ ,  $a_{22} = 1$ ,  $a_{23} = 0$ . Q.E.D.

## References

- [1] G. Bradski. The opencv library. 2006.
- [2] V. Cevher, A. Sankaranarayanan, J. McClellan, and R. Chelappa. Target tracking using a joint acoustic video system. *IEEE Transaction on Multimedia*, 9(4):715–727, 2007.
- [3] H. Cramer, U. Scheunert, and C. Wanielik. Visible, night vision and ir sensor fusion. In *Sixth International Conference of Information Fusion*, pages 1:2–10, 2003.
- [4] J. W. Davis and V. Sharma. 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*, page 11, Washington, DC, USA, 2005. IEEE Computer Society.
- [5] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.
- [6] J. Han and B. Bhanu. Fusion of color and infrared video for moving human detection. *Pattern Recogn.*, 40(6):1771–1784, 2007.
- [7] R. Hartley and I. Reid. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [8] R. I. Hartley. Theory and practice of projective rectification. *Int. J. Comput. Vision*, 35(2):115–127, 1999.
- [9] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using code-book model. *Real-Time Imaging*, 11(3):172 – 185, 2005. Special Issue on Video Object Processing.
- [10] P. Kumar, A. Mittal, and P. Kumar. Fusion of thermal infrared and visible spectrum video for robustsurveillance. In *ICCVGIP06*, pages 528–539, 2006.
- [11] P. St Onge and G. Bilodeau. Visible and infrared sensors fusion by matching feature points of foreground blobs. In *ISVC07*, pages II: 1–10, 2007.
- [12] H. Torresan, B. Turgeon, C. Ibarra-Castanedo, P. Hebert, and X. P. Maldague. Advanced surveillance systems: combining video and thermal imagery for pedestrian detection. In D. D. Burleigh, K. E. Cramer, and G. R. Peacock, editors, *Thermosense XXVI*, volume 5405, pages 506–515. SPIE, 2004.
- [13] L. Volfson. Visible, night vision and ir sensor fusion. In *9th International Conference on Information Fusion*, pages 10–13: 1–4, 2006.
- [14] Q. Wu, P. Boulanger, and W. F. Bischof. Bi-layer video segmentation with foreground and background infrared illumination. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 1025–1026, New York, NY, USA, 2008. ACM.
- [15] H. Zhou, M. Taj, and Cavallaro. Target detection and tracking with heterogeneous sensors. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):503–513, 2008.