Fast and Accurate Registration of Visible and Infrared Videos

Socheat Sonn, Guillaume-Alexandre Bilodeau, Philippe Galinier
Department of computer and software engineering, École Polytechnique de Montréal
Montréal, QC, Canada
{socheat.sonn, gabilodeau, philippe.galinier}@polymtl.ca

Abstract

In this work, we propose a novel fast and accurate method based on keypoints and temporal information to solve the registration problem on planar scenes with moving objects for infrared-visible stereo pairs. A keypoint descriptor and a temporal buffer (reservoir) filled with matched keypoints are used in order to find the homography transformation for the registration. Inside a given frame, the problem of registration is formulated as correspondences between noisy polygon vertices. Sections of polygons are matched locally to find the corresponding vertices inside a frame. These correspondences are then accumulated temporally using a reservoir of matches for homography calculation. Results show that our method outperforms two recent state-of-the-art global registration methods by a large margin in almost all tested videos.

1. Introduction

Nowadays, the field of image registration is rapidly expanding in computer vision community as new sensors become available. Because it is a well-known domain, the focus is to accelerate the computing time [8] and to improve the precision of the registration [7]. However, registration is still very challenging for image modalities that capture very different information, as for example, visible and infrared. Even if difficult, combining two complementary sensors such as infrared and visible is a good solution to extract more information about the targets in the scene. Many applications such as human detection [4, 14] or tracking system as well as medical imaging to monitor a patient and its temperature, benefit from that kind of combination. Finding an accurate transformation that maps objects from an image to another allows us to clarify their boundaries and to apply information fusion (appearance, shape, etc.).

Since images captured by infrared and visible cameras come from different phenomena [12], finding the correspondences (i.e. registration) between both sources is quite challenging. For example, object's texture from visible image is often missing in the infrared image because it does not have a big influence on the heat emitted by the object. To solve this problem, we propose a new fast and accurate method for finding correspondences between pairs of visible and infrared videos. Since image regions are very different in both sources, we decided to focus on the boundaries of binary silhouettes and to formulate the problem as finding correspondence of vertices of noisy polygons. As such, our method more specifically addresses video surveillance applications aimed toward detecting and tracking humans. We also focus on finding the homography of persons in planar scenes. We use an adapted implementation of Discrete Curve Evolution (DCE) [1] to extract keypoints on the contours. A reservoir (temporal buffer) that contains a temporal set of matches is used to find the best global transformation (planar homography) for a scene. Our assumption about the stereo pair configuration is that the cameras are co-located and roughly parallel. The contributions of the paper are:

1. We formulated the registration problem as finding corresponding vertices between noisy polygons. Vertices are matched using the local shape formed by three consecutives vertices;

2. We included temporal information to add candidate pairs for the homography computations by using a reservoir (temporal buffer) of matched pairs.

We compared our method with DCE keypoints-based [1] and trajectory-based [18] registration methods in terms of precision. Our results show that our global registration method outperforms the state-of-the-art.

2. Related works

Previous works have considered image region correlation or mutual information [12, 13, 15] to register visible-infrared images or videos. The problem is that texture information in both sources is quite different, thus correlation is hard to find in the entire image. Local Self-Similarity (LSS) over regions seems promising, but computing LSS

over large regions is slow [17]. Because features on boundaries such as orientations and magnitudes are similar in both sources [3, 5, 11], using edges or connected edges is the most popular solution [3, 5]. With the same idea, skeleton or DCE keypoints are used in [1] to identify shape feature and to estimate the homography. A method that combines edges and feature points can also be used like the one described in [10]. Another solution is to add a temporal process such as blobs tracking [19] or trajectory computation [2, 9, 18] in order to find correspondences.

Usually, methods based on feature points are simple, fast and give reasonable results. But, they are not very accurate because they are often applied one frame at a time. Thus, the number of matching points is not high enough to obtain accurate results. On the other hand, trajectory-based methods use a temporal component to have more matches, but it is very hard to have stable trajectory points based on the centroid or the point at the top of silhouettes. To benefit from both approaches, our proposed method combines contour keypoints (feature points) and temporal information in the form of a reservoir of keypoint matches to achieve accurate and fast registration.

3. Methodology

3.1. Overview of the method

The first step of our method is to perform a simple background subtraction like the one described in [16] that detects the foreground using the temporal average of the intensity of each pixel and a threshold. From the foreground blobs, we use the adapted implementation of the DCE described in [1] to detect significant keypoints on a contour. Then, those keypoints, viewed as polygon vertices, are described and used for the matching process in order to make correspondences. The correspondences of each frame are saved in a reservoir that has a fixed size, which is the temporal extent of the keypoint matches that will be considered for finding the homography. If the reservoir is full, we delete the first entry and add the last one, like a sliding temporal window. All the keypoints in the reservoir are used to compute the global transformation matrix. That matrix is obtained by a standard RANSAC-based algorithm [6]. Then, we apply the matrix on the current infrared frame to evaluate the overlap ratio between the transformed infrared and the visible frame. At each iteration, the overlap ratio is compared with the last ratio and the best one is saved, as well as the transformation matrix. At the end of the algorithm, we obtain the best transformation matrix for the video.

3.2. Keypoint extraction and description

The adapted DCE algorithm of [1] is used on the infrared-visible foreground images to detect and keep the most significant keypoints of the contours. It is essentially

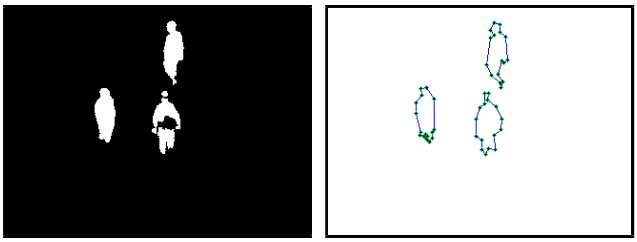

Figure 1. Polygonal approximation and keypoints found. Left : The binary foreground. Right : The contours. The dots (vertices) represent the keypoints found using [1].

a polygonal approximation of each foreground blob, where it is possible to control the number of vertices we want to keep in the final contours. To make the matching process easier and faster, we elected to eliminate all the possible holes inside the contours. Each contour ends with only 16 significant vertices. These contours form polygons that estimate the human shape. Figure 1 gives an example of the foreground, the polygons and DCE keypoints found in each image. To describe a DCE keypoint, we describe the local shape of the polygon at each vertex. The polygons are too noisy to allow matching them globally. We use a feature vector with two components (c, θ). Note that this feature vector is different from what was previously proposed by [1]. Ours focuses on the polygon vertices local shape, instead of global segment poses in the image. Suppose that we have three consecutive keypoints (P_1, P_2 and P_3) on the contour (polygon) in a clockwise order.

- c : is the convexity of the polygon at each keypoint. For example, to find the convexity of the polygon at keypoint P_2, all we need to do is to compute a simple cross product,

$$\vec{n} = \vec{P_{12}} \times \vec{P_{23}} \, , \tag{1}$$

  where $\vec{n}$ is the normal vector, $\vec{P_{12}}$ is a vector from P_1 to P_2 and $\vec{P_{23}}$ is a vector from P_2 to P_3. In this case, we suppose that each keypoint vector in equation 1 is in three dimensional coordinate $(x, y, 0)$. After the cross product, $\vec{n}$ will have a value in the z coordinate. If that value is greater than 0, the polygon at keypoint P_2 is concave, otherwise, it is convex. This is true only if the keypoints of the polygon are placed in a clockwise order. In a counter clockwise order, it will be the opposite.

- θ : is the angle of the polygon at each keypoint. For example, the angle of the polygon at keypoint P_2 is the angle between $\vec{P_{21}}$ and $\vec{P_{23}}$. To find that angle, we use standard trigonometry and calculate

$$\theta = \cos^{-1}\left(\frac{\left|\vec{P_{21}}\right|^2 + \left|\vec{P_{23}}\right|^2 - \left|\vec{P_{13}}\right|^2}{2 * \left|\vec{P_{21}}\right| * \left|\vec{P_{23}}\right|}\right) . \tag{2}$$

3.3. Matching process

After describing all of the infrared and visible keypoints, we have to define some metrics for the matching process. These metrics are :

- E_d : The euclidean distance between two keypoints.

$$E_d = |P_I - P_V| \ , \tag{3}$$

  where P_I and P_V are the position of a keypoint from an infrared and a visible image respectively.

- E_θ : The difference between two keypoint angles.

$$E_\theta = |\theta_I - \theta_V| \ , \tag{4}$$

  where θ_I and θ_V are the angles of a polygon at a keypoint from an infrared and a visible image respectively.

During the matching process, we first compare the convexity of a pair of keypoints (from infrared and visible images). If it is the same, it is a possible match, otherwise we ignore it and we continue with another pair. If the convexity is the same, we then compute E_d (equation 3) and E_θ (equation 4). To improve the algorithm accuracy by eliminating false matches, we define two thresholds :

- E_{dMax} : The maximum euclidean distance. This corresponds to maximum expected disparity, a standard threshold in all registration methods.

- $E_{\theta Max}$: The maximum angle error. This is to enforce a minimum level of similarity between keypoints.

If $E_d \leq E_{dMax}$ and $E_\theta \leq E_{\theta Max}$, then, a keypoint pair may be a possible match. Otherwise, we ignore it and we continue with another pair. If all conditions are met, we save the match temporarily, because it is possible that for some keypoints in the infrared image, there is more than one match in the visible image. If there is only one possible match, this is the best match. Otherwise, to select the best match, the following score is minimized :

$$S = \frac{\alpha E_d}{E_{dMax}} + \frac{E_\theta}{E_{\theta Max}}. \tag{5}$$

We used $\alpha = 2$ because our experimentations show that the distance is more important than the angle for determining the quality of a match. All possible pairs are considered during the matching process. At the end, we obtain a set of matched keypoints for a given pair of frames.

3.4. Finding the best global transformation matrix

With only one set of matched keypoints from one frame, it is not possible to find an accurate global transformation matrix for all videos, particularly when the detected foregrounds are noisy. We do not have enough good keypoint pairs. We can solve this problem by saving our matches from each frame in a match reservoir (temporal buffer of keypoint matches). But, if the length of the video is one hour, we cannot save all the frames, it would take too much memory and the computation time would be too long. It is more appropriate to limit the number of frames that we save in the reservoir. When the reservoir is full, we can simply erase the first saved matches and add the new ones in the fashion of a temporal sliding window. With this method, we use only a certain number of the last frames to compute the homography matrix for the current frame. The homography matrix is more accurate in this way, because those last frames in the reservoir are more similar to the current frame. Using the matches in the reservoir, we use a RANSAC-based algorithm [6] to filter all the matches and to find the homography matrix. We use the standard function for finding homography with RANSAC in OpenCV. That matrix is saved and applied on the infrared foreground frame. We then compute the following overlapping ratio (R) to select the best homography matrix dynamically:

$$R = \frac{A_I \cup A_V}{A_V}, \tag{6}$$

where A_I and A_V are the foreground regions of the transformed infrared and the visible blobs respectively. At the end, to select the best matrix, we keep the one that gives us the ratio closest to 1. Algorithm 1 shows all the steps of our method.

4. Experimentation and results

4.1. Overview of the experimentation

We validate our registration method with several planar scenes. We used a publicly available dataset that contains 9 video sequences, which is the LITIV dataset [18]. The persons are viewed from afar, the silhouettes are small and the scenes are assumed to be planar. We performed global image registration with our proposed method to find the correspondences between the foreground blobs in infrared and visible images. We selected the best matrix for each video using the ratio calculation (see Eq. 6). To compare our best result for each sequence, we used the ground-truth matrices in the LITIV dataset [18] as a reference to transform the infrared silhouettes. The mean euclidean distance (E) between the centroids of the transformed infrared silhouettes from the ground-truth and the ones from our method is computed. We also compared our results with two state-of-the-art methods. The first one uses described DCE keypoints (Bilodeau *et al.* [1]), but does not use any temporal information, and the second use a trajectory point matching method (Torabi *et al.* [18]). For each sequence, we perform two tests. The first with 30 frames in our reservoir (size of *Reservoir* in algorithm 1), and the second, with 100 frames.

```
Reservoir = empty;
foreach frame do
    foreach video (IR and Visible) do
        - Apply background subtraction [16] ;
        - Extract DCE keypoints (kpt) [1];
        foreach keypoint do
            - Calculate the convexity (c, Eq. 1);
            - Calculate the angle (θ, Eq. 2);
        end
    end
    foreach described IR keypoint do
        foreach described Visible keypoint do
            if IR kpt convexity = Visible kpt convexity
            then
                - Compute the distance between the
                two keypoint positions (E_d, Eq. 3);
                - Compute the difference between the
                two keypoints angles (E_θ, Eq. 4);
                if (E_d ≤ E_dMax) and (E_θ ≤ E_θMax)
                then
                    - Save the Visible keypoint index;
                    - Save the distance error (E_d);
                    - Save the angle error (E_θ);
                end
            end
        end
        - Compute the min error with Eq. 5;
        - Save the match with min error;
    end
    if Reservoir is full then
        - Erase the first saved matches from the
        Reservoir;
    end
    - Add current frame matches in Reservoir;
    - Use RANSAC [6] with the keypoints from
    Reservoir to find the homography matrix;
    - Apply the homography matrix on the IR frame;
    - Compute the blob ratio (R) with Eq. 6;
    if The new blob ratio (R) is better than the last
    iteration then
        - Save the new blob ratio (R);
        - Save the new homography matrix;
    end
end
- Use the last saved homography matrix;
```

Algorithm 1: Fast and accurate registration algorithm

4.2. Results

Table 1 shows the mean registration errors for the best frames for each sequence (1 to 9) in a planar homography scenario. For sequences 1 to 4, we compare our results with the DCE keypoints method proposed by Bilodeau *et al.* [1]. These results are also the minimum errors that they found for each sequence. For sequences 1 to 9, we compare our method with a trajectory point matching method (Torabi *et al.* [18]). In this case, a matrix selection method is used to find the best homography matrix. With a 30-frame reservoir the best selected frames for sequences 1 to 9 are the following : {110, 193, 902, 113, 290, 76, 393, N.A., 180}. With 100-frame reservoir, the best selected frames are : {98, 157, 892, 125, 319, 108, 806, N.A., 183}.

Table 1. Global Registration errors for the LITIV dataset (Seq. 1-9, videos from LITIV dataset (dataset 01)[18]). $E = \sqrt{{E_x}^2 + {E_y}^2}$: Mean registration error compared to the ground-truth.

Seq.	Method	E
1	Our method (30-frame reservoir)	1.06
	Our method (100-frame reservoir)	**0.94**
	Bilodeau *et al.* [1]	4.34
	Torabi *et al.* [18]	2.27
2	Our method (30-frame reservoir)	4.63
	Our method (100-frame reservoir)	**1.05**
	Bilodeau *et al.* [1]	8.79
	Torabi *et al.* [18]	5.34
3	Our method (30-frame reservoir)	**1.19**
	Our method (100-frame reservoir)	1.76
	Bilodeau *et al.* [1]	13.33
	Torabi *et al.* [18]	3.95
4	Our method (30-frame reservoir)	4.56
	Our method (100-frame reservoir)	**1.55**
	Bilodeau *et al.* [1]	4.71
	Torabi *et al.* [18]	4.82
5	Our method (30-frame reservoir)	2.94
	Our method (100-frame reservoir)	**2.74**
	Torabi *et al.* [18]	4.20
6	Our method (30-frame reservoir)	**0.53**
	Our method (100-frame reservoir)	1.61
	Torabi *et al.* [18]	6.69
7	Our method (30-frame reservoir)	**1.83**
	Our method (100-frame reservoir)	4.25
	Torabi *et al.* [18]	5.68
8	Our method (30-frame reservoir)	N.A.
	Our method (100-frame reservoir)	N.A.
	Torabi *et al.* [18]	**3.77**
9	Our method (30-frame reservoir)	4.25
	Our method (100-frame reservoir)	**2.49**
	Torabi *et al.* [18]	7.44

The results from table 1 are obtained with $E_{dMax} = 65$ pixels and $E_{\theta Max} = 40$ degrees. We used $E_{dMax} = 65$ because the viewpoint of the 9 sequences is from afar, so, the disparity tends to be large for each sequence. Because the silhouettes are small, the angle of each vertex on the

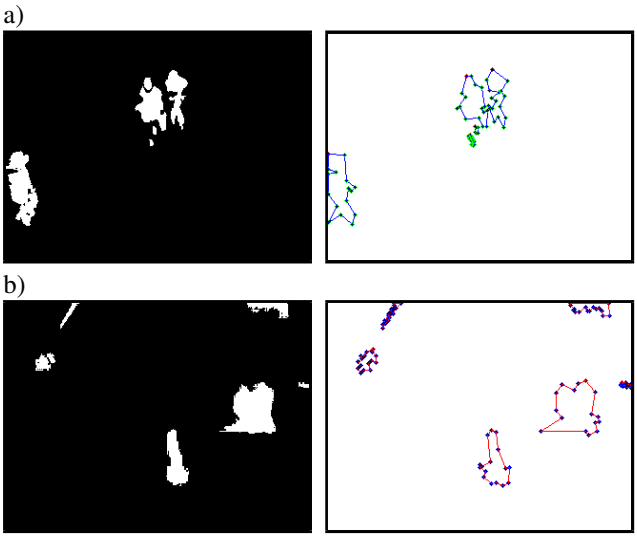

Figure 2. Planar homography scenario (Sequence 8, frame no. 330). a) Infrared foreground and contours. b) Visible foreground and contours. The disparity and the difference between the polygons from both images (infrared-visible) are too large to find good matches.

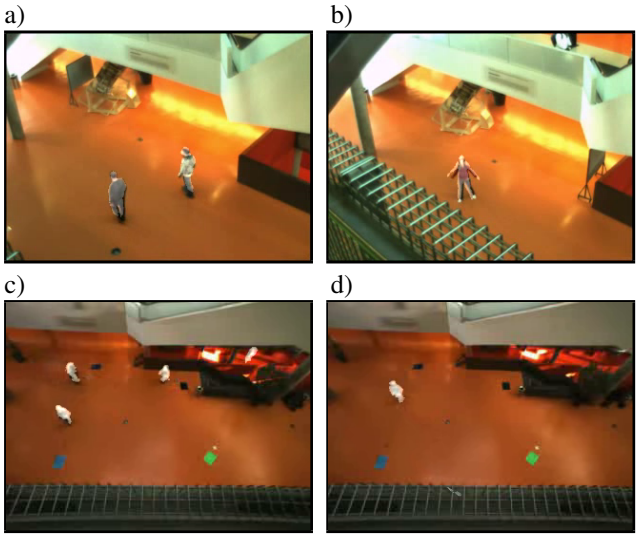

Figure 3. Planar homography scenario. Samples of our registration method. a) Sequence 1, frame no. 98, 100 frames in reservoir. b) Sequence 2, frame no. 157, 100 frames in reservoir. c) Sequence 5, frame no. 319, 100 frames in reservoir. d) Sequence 6, frame no. 76, 30 frames in reservoir. See table 1 for errors comparison.

contours is very different between an infrared and a visible image. This is the reason why $E_{\theta Max}$ is big (40). In any case, our score function 5 can filter the matches and keep the best.

Table 1 shows that our method outperforms the other methods almost all the time, except for sequence 8. Sequence 8 is indeed very complex, because the disparity between infrared and visible images is too high. The polygons between the two images are also very different. Figure 2 shows the differences between infrared and visible foreground and contours for sequence 8. We tried to change E_{dMax} to put a higher value because of the high disparity, but it cannot solve the problem, because of a poor background subtraction, there is a lot of noise in the visible image. This noise produces bad matches. Torabi *et al.* [18] performs better in sequence 8 because they do not need to deal with the shapes of polygons, but only with trajectories. Even if there is noise in the scene, only the moving blobs are tracked and considered for the registration.

The good point is that except for sequence 8, the errors for all the other sequences are always lower than the state-of-the-art by a large margin. The errors vary between 0.5 and 5 pixels. This is possible because our matching process is very strict. For each frame, only few best matches are saved in the reservoir. Figure 3 shows four samples of our results from table 1.

Our experimentation shows that the more we have matches to work with, better the result will be. This is exactly what table 1 shows. Most of the times, we have fewer errors with 100 frames in the reservoir than with 30 frames. The reason is that even if we have some noise (bad matches) in the reservoir, we have better chance to keep track of more good matches in the previous frames than noise. If this is the case, RANSAC algorithm will eliminate all the noise. With a smaller reservoir, it is possible that sometimes in the video, it contains more noises than good matches. In that case, RANSAC algorithm will eliminate good matches and keep the noisy matches as good matches. However, because the reservoir works like a sliding window, when the noisy matches exit the temporal window, everything will work again in the next frames. Thus, it is not always necessary to have a larger reservoir.

Moreover, the problem with a larger reservoir is that computing time is longer. Table 2 shows the average computing time for one frame in each sequence. These computing times include all the steps from background subtraction to matrix computation and selection. If an application needs a faster result, a smaller reservoir can be used without radically degrading the results, but if it needs a better accuracy, it is worth to have a larger reservoir. In short, depending on the application, reservoir size should be adapted. Even if 100 frames in the reservoir is better than 30 most of the time, the results for 30 frames in the reservoir are also good and better than the other methods in table 1.

5. Summary and conclusions

We have presented an alternative to region-based, frame-by-frame keypoints-based and trajectory-based registration methods that works for visible and infrared stereo pairs. It

Table 2. Average computing time for one frame

Seq.	30-frames reservoir Comp. time (s)	100-frames reservoir Comp. time (s)
1	0.114	0.167
2	0.157	0.211
3	0.119	0.141
4	0.165	0.230
5	0.143	0.231
6	0.122	0.189
7	0.153	0.185
8	0.123	0.184
9	0.190	0.341

Computer used : Windows 7 (64-bits), Intel Core i5 CPU, 2.4 GHz, 6 Go RAM.

uses a simple contour keypoint descriptor and a temporal buffer (reservoir) filled with matched keypoints. We confirmed the accuracy of our global registration method with planar scenes from a publicly available dataset. The results show that our method outperforms two recent methods from the state-of-the-art by a large margin, for almost every tested sequence. However, it doesn't work well when the disparity and the difference between the polygons in both infrared-visible images are too high.

Future work: Although the results with our global registration method are very good for a planar scene, it is not perfectly adapted to non-planar scenes, because in that case, it is possible to have more than one depth plane in the same image. So, we cannot only apply one global transformation matrix for all the silhouettes in the same image. We need to consider a local approach. To improve our registration method, it would be useful to add after a first global transformation, a local transformation for each silhouette, for example, by a silhouette tracking method to get their matching points.

References

[1] G.-A. Bilodeau, P.-L. St-Onge, and R. Garnier. Silhouette-based features for visible-infrared registration. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 68 –73, jun. 2011. 1, 2, 4

[2] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *Int. J. Comput. Vision*, 68(1):53–64, 2006. 2

[3] E. Coiras, J. Santamaria, and C. Miravet. Segment-based registration technique for visual-infrared images. *Optical Engineering*, 39:282–289, jan 2000. 2

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, 2005. 1

[5] M. I. Elbakary and M. K. Sundareshan. Multi-modal image registration using local frequency representation and computer-aided design (cad) models. *Image Vision Comput.*, 25(5):663–670, 2007. 2

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 2, 3, 4

[7] D. Gallup, J. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1418 –1425, 2010. 1

[8] S. Gehrig and C. Rabe. Real-time semi-global matching on the cpu. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 85 –92, jun. 2010. 1

[9] J. Han and B. Bhanu. Fusion of color and infrared video for moving human detection. *Pattern Recognition*, 40(6):1771–1784, 2007. 2

[10] J. Han, E. J. Pauwels, and P. De Zeeuw. Visible and infrared image registration in man-made environments employing hybrid visual features. *Pattern Recogn. Lett.*, 34(1):42–51, Jan. 2013. 2

[11] X. Huang and Z. Chen. A wavelet-based multisensor image registration algorithm. In *Signal Processing, 2002 6th International Conference on*, volume 1, pages 773–776 vol.1, 2002. 2

[12] S. J. Krotosky and M. M. Trivedi. Mutual information based registration of multimodal stereo videos for person tracking. *Comput. Vis. Image Underst.*, 106(2-3):270–287, 2007. 1

[13] S. K. Kyoung, H. L. Jae, and B. R. Jong. Robust multi-sensor image registration by enhancing statistical correlation. In *Information Fusion, 2005 8th International Conference on*, volume 1, page 7, 2005. 1

[14] O. Oreifej, R. Mehran, and M. Shah. Human identity recognition in aerial images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 709 –716, jun. 2010. 1

[15] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The correlation ratio as a new similarity measure for multimodal image registration. pages 1115–1124. Springer Verlag, 1998. 1

[16] B. Shoushtarian and H. Bez. A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking. *Pattern Recognition Letters*, 26(1):5–26, Jan. 2005. 2, 4

[17] A. Torabi and G.-A. Bilodeau. Local self-similarity-based registration of human rois in pairs of stereo thermal-visible videos. *Pattern Recogn.*, 46(2):578–589, Feb. 2013. 2

[18] A. Torabi, G. Massé, and G.-A. Bilodeau. An iterative integrated framework for thermal-visible image registration, sensor fusion, and people tracking for video surveillance applications. *Comput. Vis. Image Underst.*, 116(2):210–221, Feb. 2012. 1, 2, 3, 4, 5

[19] J. Zhao and S. Cheung. Human segmentation by fusing visible-light and thermal imaginary. pages 1185–1192, 2009. 2