View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by CWI's Institutional Repository

# Visible and Infrared Image Registration Employing Line-Based Geometric Analysis

Jungong Han, Eric Pauwels, and Paul de Zeeuw

Centrum Wiskunde and Informatica (CWI)
Science Park 123, Amsterdam, The Netherlands

**Abstract.** We present a new method to register a pair of visible (ViS) and infrared (IR) images. Unlike most of existing systems that align interest points of two images, we align lines derived from edge pixels, because the interest points extracted from both images are not always identical, but most major edges detected from one image do appear in another image. To solve feature matching problem, we emphasize the geometric structure alignment of features (lines), instead of descriptor-based individual feature matching. This is due to the fact that image properties and patch statistics of corresponding features might be quite different, especially when one compares ViS image with long wave IR images (thermal information). However, the spatial layout of features for both images always preserves consistency. The last step of our algorithm is to compute the image transform matrix, given minimum 4 pairs of line correspondence. The comparative evaluation for algorithms demonstrates higher accuracy attained by our method when compared to the state-of-the-art approaches.[1]

**Keywords:** Image Registration, line detection, geometric analysis.

## 1 Introduction

Recent advances in imaging, networking, data processing and storage technology have resulted in an explosion in the use of multi-modality images in a variety of fields, including video surveillance, urban monitoring, cultural heritage area protection and many others. The integration of images from multiple channels can provide complementary information and therefore increase the accuracy of the overall decision making process. A fundamental problem in multi-modality image integration is that of aligning images of the same/similar scene taken by different modalities. This problem is known as image registration and the objective is to recover the correspondences between the images. Once such correspondences have been found, all images can be transformed into the same reference, enabling to augment the information in one image with the information from the others.

[1] This work is supported by EU-FP7 FIRESENSE project.

E. Salerno, A.E. Çetin, and O. Salvetti (Eds.): MUSCLE 2011, LNCS 7252, pp. 114–125, 2012.
© Springer-Verlag Berlin Heidelberg 2012

### 1.1 Prior Work on Image Registration

Several related survey papers for image registration have appeared over the years. [1,2,3] have provided a broad overview of over three hundred papers for registering different types of sensors. Following most of literature, we also divide existing techniques into two categories: pixel-based methods and feature-based methods. Pixel-based methods first define a metric, such as the sum of square differences and mutual information [2], which measures the distance of two pixels from different images. The registration problem is then changed to minimize the total distance between all pixels on one image and the corresponding pixels on another image. In feature-based methods, interest points like Harris corners, scale invariant feature transform (SIFT), speed-up robust feature (SURF), etc., are first extracted from images. Afterwards, these features are matched based on the metrics, such as cross correlation and mutual information. Once more than four feature correspondences are obtained, the transform can be computed. In principle, pixel-based method should be better than the feature-based method because the former considers the global minimization of the cost function, but the later one minimizes the cost function locally. In practise, however, feature-based method has better performance for many applications, because the interest point is supposed to be distinctive in a local area, thus leading to the better matching. On the other hand, the pixel-based method is much more expensive than the feature-based algorithm, because every pixel is involved in the computation. Considering both accuracy and efficiency of the algorithm, we adopt the feature-based method in this paper. Therefore, we limit our review to feature-based registration methods, and pay special attention to the work for registering visible and infrared images.

Many approaches have been proposed for automatically registering IR and ViS images. Edge/gradient information is one of the most popular feature as their magnitudes [4] and orientations [5] may match between infrared and visible images. In [6], authors first extract edge segments, which are then grouped to form triangles. The transform can be computed by matching triangles from the source to destination images. Huang *et al.* [7] proposes a contour-based registration algorithm, which integrates the invariant moments with the orientation function of the contours to establish the correspondences of the contours in the two images. Normally it is difficult to obtain accurate registration by using contour-based method, because precisely matching all contours detected from two images is challenging. Moreover, this method drastically increases computation time compared to interest point-based registration. To improve this work, Han *et al.* [8] propose to find correspondences on *moving* contours. They extract silhouettes of moving humans from both images. Matching only the contours of humans significantly improves both the performance and the efficiency of the algorithm. An alternative [9] is to make use of the object moving pathes generated by object tracking algorithm. Finding correspondences between trajectories helps to align images. This type of algorithm works very well when moving objects can be precisely tracked from both channels. Unfortunately, the current tracking algorithm is not satisfactory in many applications.

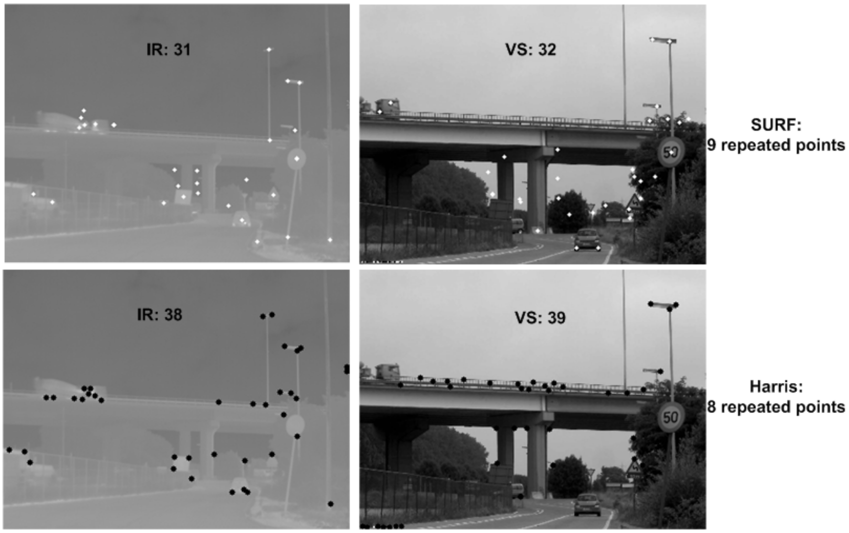


**Fig. 1.** Interest point detection for IR and ViS images. Two different methods: SURF and Harris corner detection are used.

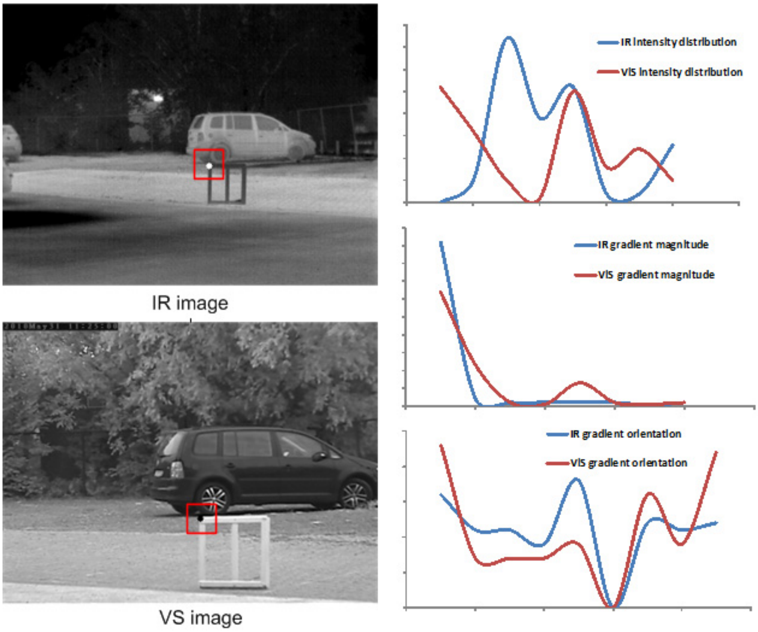


**Fig. 2.** Left: Statistics of corresponding points (within red square). Right, top: Distributions (normalized histogram) of intensity value. Middle: Distributions of gradient magnitude. Bottom: Distributions of gradient orientation. The x-axis represents the bin of histogram, and the y-axis refers to the number of pixels casted to the bin.

### 1.2 Problem Statement

Most existing publications for image registration dedicate to solving four problems: 1) an efficient way to extract feature points, which guarantees the majority of features on both images is identical; 2) a better feature descriptor; 3) a suitable metric to measure the distance of two feature descriptors; 4) a proper transform model. Among these four problems the detection of repeatable features and also the feature matching are more challenging when dealing with visual and infrared cameras. The main reason is that the electromagnetic wavelengths of ViS sensor and IR sensor are quite different. Normally, the wavelength of IR sensor is from 4 to 12 microns, while the wavelength of ViS sensor roughly lies between 0.4 to 0.7 microns. This leads to the fact that IR images have noticeably less texture in the area where temperatures are more homogeneous. However, the texture information is very important for both interest point detection and feature matching. In Fig. 1, we extract equivalent number of interest points from both IR and ViS images exploring two popular algorithms, where SURF method enables a scale- and rotation-invariant interest point detection but Harris method focuses on detecting corner points on the single scale. Seen from the results, the majority of extracted interest points is unfortunately not repeatable. To explain the feature matching problem, we show statistics (see Fig. 2) of image patches ($15 \times 15$) surrounding two corresponding points. We compute the distribution (normalized histogram) of intensity value within the image patch, the distribution of gradient magnitude, and also the distribution of gradient orientation, respectively. Those are all feature descriptors widely used for IR and ViS image registration. To obtain a good feature matching result, we expect the signal distributions of two corresponding points to be similar to each other. Unfortunately, none of them is capable of measuring the correlation between two points in this case, though the gradient orientation is clearly better than the others. This example illustrates that comparing image patches may not be a reliable way to correlate features between IR and ViS images.

### 1.3 Our Contributions

In order to address two problems mentioned above, we propose a new algorithm here, which differentiates with existing work in two aspects. First, we do not use interest point as the feature to align the image. Instead, we try to align images based on lines derived from edges of the images. These lines strongly relate to the boundaries of objects, which always appear on both images though IR sensor and ViS sensor have significantly different properties. Secondly, our algorithm enables a one-to-many matching based on a simple feature descriptor, which allows one feature on one image to have more potential correspondences in another image. This ensures that the majority of the initial matching is correct. A central point of our feature-matching scheme is that it relies more on the geometric structure checking of features, which gives much better matching results. The last feature of our work is that we prove the traditional point-to-point transform can be directly computed, given minimum four pairs of line correspondence.

In the sequel, we first present our mathematical model in Section 2, which introduces how we compute the transformation matrix by employing lines. In Section 3, we describe several key algorithms, such as line reorganization, line initial matching and line-configuration computing. The experimental results are provided in Section 4. Finally, Section 5 draws conclusions and addresses our future research.

## 2 The Mathematical Model

The goal of image registration is to match two or more images so that identical coordinate points in these images correspond to the same physical region of the scene being imaged. To make our explanation simple, we assume that there are only two images involved in the registration, each representing one plane. In fact, the registration is to find a mathematical transformation model between these two images, which minimizes the *energy* function of image matching. This optimization procedure can be described mathematically

$$\tilde{\mathbf{H}} = arg \min_{\mathbf{H}} \sum_{i} E(p_i, \mathbf{H}p'_i). \tag{1}$$

Here, $p_i$ is the $i^{th}$ pixel in the image $I$ and $p'_i$ is its corresponding pixel in the image $I'$. The energy function is to measure the *distance* between $I$ and the transformed version of $I'$ based on $\mathbf{H}$. This transformation helps to establish a plane-to-plane mapping, transforming a position $p$ in one plane to the coordinate $p'$ on another plane. The point $p = (u, v, w)^T$ in image coordinates corresponds to Euclidean coordinates $(u/w, v/w)$. In our paper, we assume a 2D perspective transformation. Writing positions as homogeneous coordinates, the transformation $p = \mathbf{H}p'$ equals

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}. \tag{2}$$

Homogeneous coordinates are scaling invariant, reducing the degrees of freedom for the matrix H to only eight. In order to determine the eight parameters, at least four point-correspondences between two images have to be found. In the literature, most publications employ interest (corner) points for establishing point-correspondences. Normally, this matrix $\mathbf{H}$ is related to the camera model, that is, the matrix $\mathbf{H}$ can be further decomposed into camera intrinsic and extrinsic parameters.

As we mentioned before, our work wants to align images based on line correspondences. However, our objective is to compute a point-to-point transform matrix $\mathbf{H}$. Therefore, the central issues are whether and how we can obtain $\mathbf{H}$ based on line correspondences.

Let us now denote two lines ($l$ and $l'$) on both image coordinates as:

$$au + bv + c = 0 \qquad and \qquad a'u' + b'v' + c' = 0. \tag{3}$$

We can rewrite the line equation to

$$(a, b, c)\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \quad and \quad (a', b', c')\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0. \tag{4}$$

If we left multiply $(a, b, c)$ to both sides of eqn. (2), it will become

$$(a, b, c)\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = (a, b, c)\mathbf{H}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0. \tag{5}$$

Comparing (4) and (5), and taking into account that the line coefficients $(a', b', c')$ for a given line are essentially unique (up to an arbitrary scaling factor), we can deduce that $(a, b, c)\mathbf{H} = \lambda(a', b', c')$. Here, $\lambda$ is a scaling factor. If we write it in a formal way, we will obtain:

$$\mathbf{AH} = \mathbf{\Lambda A}', \tag{6}$$

where $\mathbf{A}$ is a matrix, encoding the parameters of lines on one image coordinates, while $\mathbf{A}'$ is its corresponding matrix on another image coordinates. And, $\mathbf{\Lambda}$ encodes scaling factors for all lines. (5) turns out that it is possible to compute $\mathbf{H}$ directly from lines, given a number of pairs of corresponding lines. Next, we need to know how we compute the parameters of $\mathbf{H}$. Suppose that we have lines $l : (a, b, c)$ and $l' : (a', b', c')$ from different image coordinates, which correspond to each other. The associated scaling factor is $\lambda_1$. Therefore, we can get three equations, which are

$$\begin{aligned} ah_{11} + bh_{21} + ch_{31} &= \lambda_1 a' \\ ah_{12} + bh_{22} + ch_{32} &= \lambda_1 b' \\ ah_{13} + bh_{23} + ch_{33} &= \lambda_1 c'. \end{aligned} \tag{7}$$

If we divide the first equation by the third equation, and divide the second equation also by the third equation, we can remove the parameter $\lambda$, thereby achieving two linear equations:

$$\begin{aligned} ac'h_{11} + 0h_{12} - aa'h_{13} + bc'h_{21} + 0h_{22} - ba'h_{23} + cc'h_{31} + 0h_{32} &= ca'h_{33} \\ 0h_{11} + ac'h_{12} - ab'h_{13} + 0h_{21} + bc'h_{22} - bb'h_{23} + 0h_{31} + cc'h_{32} &= cb'h_{33}. \end{aligned} \tag{8}$$

Normally, we force $h_{33}$ to 1. Therefore, we have 8 parameters to compute, and each pair of lines provides two equations related to $\mathbf{H}$. To have a complete matrix, at least four pairs of lines are required. The above deductions prove that it is possible to compute a point-to-point transform based on line correspondences.

## 3 Algorithm Implementation

In this section we will introduce algorithm implementations of two key modules in more details, which are line generation and line matching. The line generation module consists of line detection, line duplication deletion, line label and line sort. The line matching module includes initial matching and geometric matching of line composition. All steps are designed with the goal of constructing an efficient system.

### 3.1 Line Generation

RANSAC-based Hough transform from our previous work [10] is used to detect lines in the image. The output of our previous work is the start point and also the end point of a line. More precisely, it returns a line *segment*. The first step of our algorithm is to filter out some shorter line segments. For the rest of line segments, we will extend this segment until it goes through the entire image space, thereby leading to a *real* line. The reason for the first step is that line segments extracted from both images vary dramatically, but most *major* segments with sufficient length are identical on both images.

The Hough transform has the disadvantage that thick lines in the input image usually result in a bundle of detected lines, which all lie close together. In practice, we do not need so many lines, which are similar and close to each other. We expect to have only one representative line within certain area. To solve this problem, we introduce a line duplication deletion step after the Hough transform. Let a line obtained from the Hough transform be parameterized by its normal $\mathbf{n} = (n_x, n_y)^T$ with $||\mathbf{n}|| = 1$ and the distance to the origin $d$. Two lines $l_1$, $l_2$ are considered equal if the angle between both is small, such as $\mathbf{n_1}^T\mathbf{n_2} > \cos(1.5°)$), and their distance is also small ($|d_1 - d_2| < 3$). The whole duplicate deletion process is repeated until the number of lines remains stable, which is usually after only three iterations.

Next, lines are labeled as either horizontal line or vertical line by

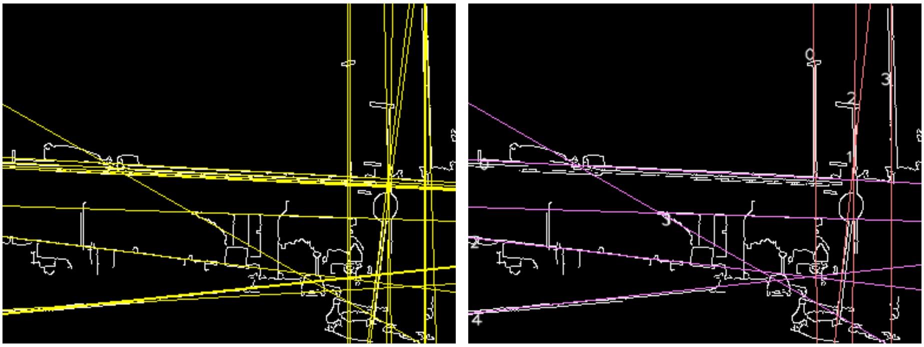
$$L_{hv} = \begin{cases} 1 & \text{if } |x_{end} - x_{start}| \geq |y_{end} - y_{start}|, \\ 0 & \text{otherwise}, \end{cases} \tag{9}$$

where $x_{start}, y_{start}$ and $x_{end}, y_{end}$ refer to $x$ and $y$ coordinates of start point and end point of a line, respectively. After labeling lines, the set of vertical lines are ordered left to right, the set of horizontal lines top to bottom. Later, when we will search for correspondences between images, we will put the constraint on the assignment that the order must be preserved. This constraint is likely valid in case that our transform is either affine transform or perspective transform.

Finally, the line is modeled by three parameters, which are $L_{hv}$, $sp$ and $os$. If a line is labeled as a horizontal line ($L_{hv} = 1$), $sp$ is defined as the angle of the line to the $x-$axis, and $os$ means the offset of the line on the $y-$axis. The definitions for $sp$ and $os$ are just inverse if line is a vertical line. Fig. 3 shows the results after each step mentioned above. For this case, the number of lines detected by Hough transform is 20, but it reduces to 9 after processing.

### 3.2 Line Matching

As we can see from the problem statement part, feature initial matching schemes used by existing systems are in general not accurate enough. The main reason is that two images captured by different modalities are quite different at the pixel level. To solve this problem, our system enables a sort of one-to-many feature matching, which allows a line in one image to have several corresponding lines



(a) (b)

**Fig. 3.** Line generation. (a) Lines detected by Hough transform. (b) Lines after duplication deletion, HV labeling and HV sort. Horizontal lines and vertical lines are marked with different colors, and the number indicates the order of HV lines.

on another image. By doing so, we can ensure that several matching candidates must include the correct one. The basic idea for this initial matching is to check and compare three parameters of two lines located in two images. The first parameter is $L_{hv}$. We assume that two corresponding lines should have same label, which means the horizontal/vertical line in one image should correspond to a horizontal/vertical line on another image. The assumption is valid for most applications, where modalities are mounted on the same platform. The second parameter is $sp$, where we assume that corresponding lines have similar slope to the axis. The last parameter is used to compare distributions of the edge pixel surrounding the line. The surrounding area is the zone between two border lines, which have the same slope with the candidate line but with $\pm\ \epsilon$ offset shift, respectively. The distribution of the edge pixel within this area can be simply specified by the edge pixel percentage $pec_{edge}$ of that area, equaling to $N_{edge}/N_{total}$. Here, $N_{edge}$ refers to the number of edge pixels within that area, while $N_{total}$ means the total number of pixels within that area. If we denote the parameters of two candidate lines as $(L_{hv}, sp, pec_{edge})$ and $(\tilde{L}_{hv}, \tilde{sp}, \tilde{pec}_{edge})$, our matching score $S$ can thus be formulated as:

$$S = (L_{hv} == \tilde{L}_{hv}) \cdot K(\frac{sp - \tilde{sp}}{\sigma_{sp}}) \cdot K(\frac{pec_{edge} - \tilde{pec}_{edge}}{\sigma_{pec}}), \tag{10}$$

where the first term of the left side $(L_{hv} == \tilde{L}_{hv})$ returns *true* if they are the same; otherwise it returns *false*. The rest two terms follow the same manner, in which $K(\cdot)$ is the Epanechnikov kernel function and $\sigma$ indicates the width of the kernel, which can be set manually. The kernel function is specified by

$$K(y) = \begin{cases} 1 - |y|^2 & \text{for } |y|^2 \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

We compute the matching scores between a given line and all candidate lines. Instead of selecting the best one, we allow one line to have multiple correspondences. The criterion is that we keep only one candidate if the matching score of this candidate is much higher than others. With the similar idea, we can adaptively assign up to three correspondences to a line.

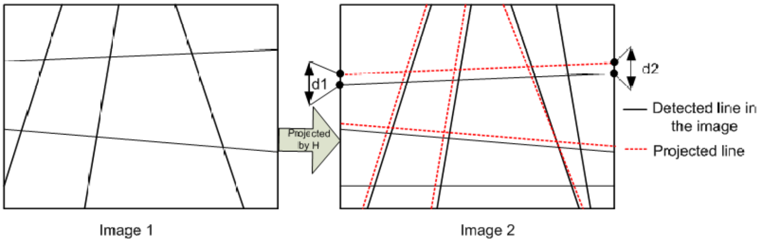


**Fig. 4.** An example for best line-configuration computing

After the step of line initial matching, we will process the geometric structure alignment of line compositions. The reason is that the matching between individual lines is not reliable due to the significant statistics difference between two image signals. However, the geometric structure (layout) of lines always remains consistency between two images. This observation motivates us to align the images by measuring the distance between two geometric structures formed by lines. The basic idea is that we randomly choose four lines from the first image to form a mini-configuration. Depending on the initial matching result, we will have several corresponding mini-configurations in the second image. This configuration-correspondence allows to compute the parameters of our eight-parameters perspective transform by solving a linear equation system according to formulas in Section 2. Using the obtained geometry transformation, we project one image onto the other image. The match between two images is evaluated by counting the total distance of the line to its closest projected line. We search for the transform parameters that provide the shortest distance by iterating over all configurations. This idea can be explained by a illustrated figure (Fig. 4), on which we transform the *image* 1 to the *image* 2. The black solid line represents the detected line on the image, and five red dash lines indicate the projected lines of the *image* 1 onto the *image* 2. From the mathematical point of view, finding the best configuration match equals to minimize a matching error $M_e$

$$M_e = \sum_{l \in \Phi} min(\|l', \mathbf{H}l\|_2, e_m), \tag{12}$$

where $\Phi$ the collection of lines in the *image* 1 and $l'$ is the closest line of the projected line $\mathbf{H}l$ in the *image* 2. The metric $\|, \|_2$ denotes the Euclidean distance between the two lines, and the error for a line is bounded by a maximum value $e_m$.

The distance between two lines can be computed by summing up the distance of two *start* points $d_1$ and the distance of two *end* points $d_2$, which are illustrated in Fig. 4. Note that the start point and end point refer to the start and end point of a line on the image.

## 4 Experimental Results

We have tested our algorithm with 6 pairs of IR and ViS images, where 4 of them are outdoor scenarios[2] and 2 of them are describing indoor scenarios[3]. We show original images of both IR and ViS channels in Fig. 5, where the last three pairs are more challenging in terms of the focal length difference of two cameras.

We have registered these images by using our algorithm. A key parameter of our algorithm is the minimum length of the accepted line, for which we set 40 pixels. In general, our algorithm can register all pairs of images except the last one. The failure is caused by the fact that we cannot extract sufficient lines for geometric matching. To evaluate our registration algorithm, we measure and report the transform errors in Table 1. The transform error is measured by the distance between one point and its transformed corresponding point. More specifically, we randomly choose 5 salient points on IR image, and transform these 5 points to ViS image by using computed transform model. We manually label the corresponding points of those 5 points. The distance between the labeled point and the transformed point is proportional to the transform error.

**Table 1.** The measurement for transform errors

| | pair 1 | pair 2 | pair 3 | pair 4 | pair 5 |
|---|---|---|---|---|---|
| transform error | 1.8 pixels | 7.8 pixels | 2.2 pixels | 3.6 pixels | 16.2 pixels |

We also compared our line-based algorithm with algorithms based on interest point matching. Since gradient magnitude [4] and orientation [5] are widely used for IR and ViS image registration, our implementation explores statistics of gradient magnitude and orientation to describe the feature point, respectively. Afterwards, nearest neighbor approach is applied for feature matching. Next, RANSAC is used for rejecting some outliers. Finally, perspective transform matrix is computed based on a number of point correspondences between two images. We have tested these two feature descriptors for the same dataset. The gradient magnitude-based descriptor failed for all the pairs, and gradient orientation-based descriptor only succeeded in registering pair 2. We show the warped images in Fig. 6, where we warp the IR image to ViS image based on computed transform matrix. The results reveal that the registrations for pair 1, pair 3 and pair 4 are accurate. The registration for pair 2 is accepted for most

[2] Videos and images are provided by XenICs NV (Belgium).

[3] Images can be downloaded via
`http://www.dgp.toronto.edu/~nmorris/data/IRData/`

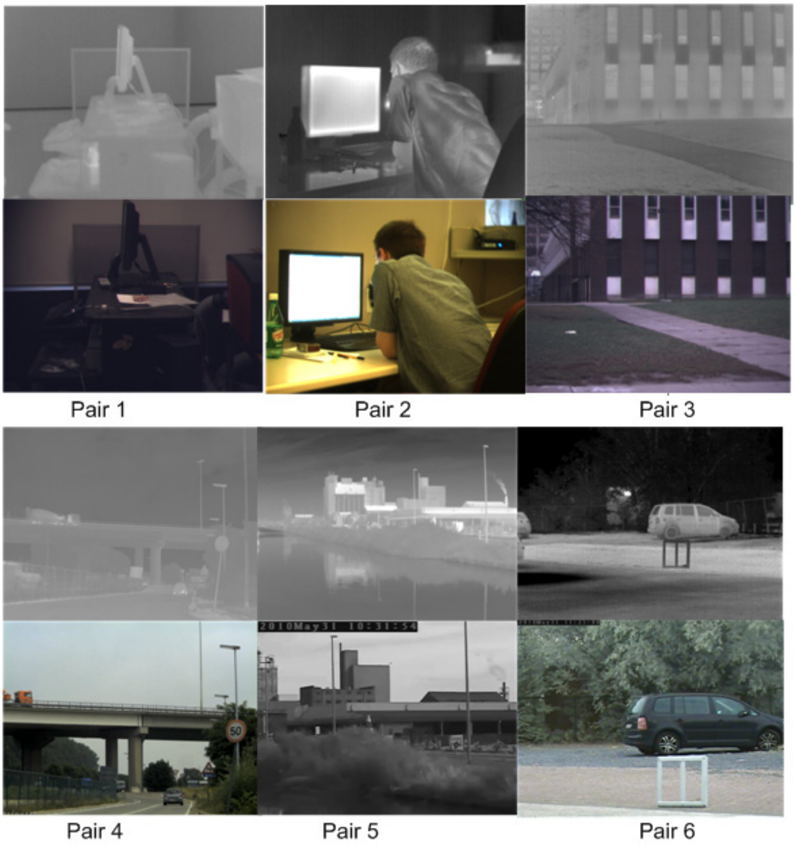


**Fig. 5.** Original images for the experiment

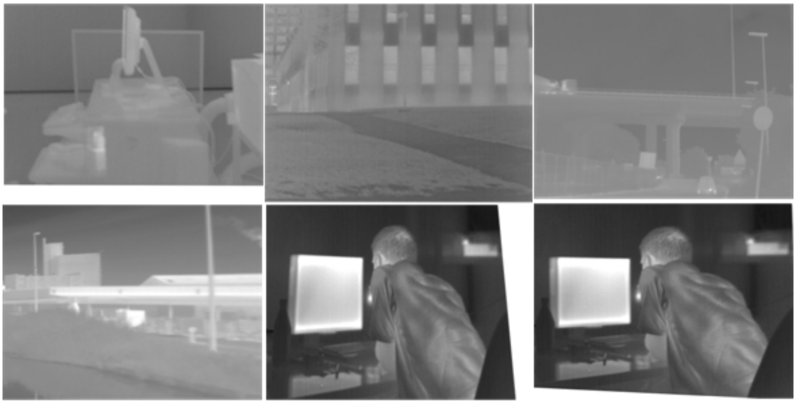


**Fig. 6.** Warped images. The last one is generated by using gradient orientation-based feature matching.

parts, except for the right-upper corner of the image. The result for pair 5 is not good, but it is encouraging in the sense that two images are significantly different. In this figure, we also show the warped image of pair 2 by using the statistic of the gradient orientation as the feature descriptor.

## 5 Conclusion

In this paper, we have examined the use of line-correspondence for registering IR (long wavelength) and ViS images. Comparing with the interest point, line derived from edge pixels well represents the boundary of the object, which are always repeatable on images captured by different modalities. The feature matching module of our new method relies more on aligning the geometric structure of features, rather than matching individual feature only. Our new algorithm provides significant advantages over state-of-the-art approaches. The future work is the combination of global transform model used by this paper and the local transform model in order to further refine the registration locally.

## References

1. Brown, L.: A Survey of Image Registration Techniques. ACM Computing Surveys 24(4), 325–376 (1992)
2. Zitova, B., Flusser, J.: Image Registration Methods: A Survey. Image and Vision Computing 21, 977–1000 (2003)
3. Xiong, Z., Zhang, Y.: A Critical Review of Image Registration Methods. Int. J. Image and Data Fusion 1(2), 137–158 (2010)
4. Lee, J., Kim, Y., Lee, D., Kang, D., Ra, J.: Robust CCD and IR Image Registration Using Gradient-Based Statistical Information. IEEE Signal Processing Letter 17(4), 347–350 (2010)
5. Kim, Y., Lee, J., Ra, J.: Multi-Sensor Image Registration Based on Intensity and Edge Orientation information. Pattern Recognition 41, 3356–3365 (2008)
6. Coiras, E., Santamaria, J., Miravet, C.: Segment-Based Registration Technique for Visual-Infrared Images. Optical Engineering 39, 282–289 (2000)
7. Huang, X., Chen, Z.: A Wavelet-Based Multisensor Image Registration Algorithm. In: Proc. ICSP, pp. 773–776 (2002)
8. Han, J., Bhanu, B.: Fusion of Color and Infrared Video for Moving Human Detection. Pattern Recognition 40, 1771–1784 (2007)
9. Caspi, Y., Simakov, D., Irani, M.: Feature-Based Sequence to Sequence Matching. Int. J. Comput. Vision 68(1), 53–64 (2006)
10. Han, J., Farin, D., de With, P.: Broadcast Court-Net Sports Video Analysis Using Fast 3-D Camera Modeling. IEEE Trans. Circuits Syst. Video Techn. 18(11), 1628–1638 (2008)