# Robust and Efficient Image Alignment Based on Relative Gradient Matching

Shou-Der Wei and Shang-Hong Lai, *Member, IEEE*

***Abstract*—In this paper, we present a robust image alignment algorithm based on matching of relative gradient maps. This algorithm consists of two stages; namely, a learning-based approximate pattern search and an iterative energy-minimization procedure for matching relative image gradient. The first stage finds some candidate poses of the pattern from the image through a fast nearest-neighbor search of the best match of the relative gradient features computed from training database of feature vectors, which are obtained from the synthesis of the geometrically transformed template image with the transformation parameters uniformly sampled from a given transformation parameter space. Subsequently, the candidate poses are further verified and refined by matching the relative gradient images through an iterative energy- minimization procedure. This approach based on the matching of relative gradients is robust against nonuniform illumination variations. Experimental results on both simulated and real images are shown to demonstrate superior efficiency and robustness of the proposed algorithm over the conventional normalized correlation method.**

***Index Terms*—Energy minimization, illumination variations, image alignment, image matching, industrial inspection, nearest-neighbor search, robust image matching.**

## I. Introduction

IMAGE alignment is a fundamental problem to a number of computer vision and image processing applications, including object recognition, image search, pose estimation, industrial inspection, target tracking, image motion estimation, medical image registration, etc. The requirements for image alignment vary from applications to applications. For example, image alignment with a two-dimensional (2-D) rigid transformation is usually sufficient for many applications in industrial inspection under controlled environment. The main concerns for these applications are high accuracy, fast speed, and robustness against different lighting conditions [1], [2]. For example, fast image alignment with subpixel accuracy is normally required in the first step of many industrial inspection tasks. In addition, the inspection system needs to be robust enough to work under various environments, so the image alignment algorithm should be able to accommodate variations in illumination conditions.

There are two main approaches for image matching; namely, the intensity-based matching [2], [3] and the feature-based (or geometry-based) matching approach [4], [5]. The intensity-based matching has been very popular in the past due to its ease of implementation. Among the intensity-based matching methods, the normalized cross-correlation technique [2], [3] has been used intensively in industrial inspection since it is robust against uniform illumination changes. Considerable efforts have been made to reduce the high computational cost involved in the cross-correlation method when rotation or scaling is required in the search. The coarse-to-fine strategy has been widely employed to improve the search speed of image alignment. In [11], a coarse-to-fine pruning algorithm is presented with the pruning threshold determined from the lower-resolution search space. This search algorithm can be proved to provide the globally optimal image registration. To achieve robustness of image registration under different illumination conditions or partial occlusion, Kaneko *et al.* [12] proposed a selective cross-correlation method that used a mask to exclude the pixels in occlusion and saturation regions from the computation of cross-correlation. Note that this selection mask is determined from the consistency of intensity increase signs at the corresponding locations in the template and search images [12]. Recently, the rotationally invariant moment features have been employed for aligning images with rotation [13]. For example, Choi and Kim proposed a two-stage image matching method that first finds candidates by comparing the vector sums of circular projections, and then these candidates are further matched based on the rotationally invariant Zernike moments. In addition, Wolberg and Zokai [14] presented a hierarchical image registration method. In this method, a log-polar registration module was developed to recover large scale changes and arbitrary rotation angles, followed by a Levenberg-Marquadt nonlinear least squares optimization procedure to achieve subpixel registration accuracy. The mutual information approach [16] is a popular method for multimodal image registration, but the mutual information approach requires to solve a nonconvex minimization problem which normally contains many local minima. To relieve this problem, Haber and Modersitzki [17] proposed to use the normalized gradient in the distance measure between images of different modalities, thus leading to a simpler optimization problem.

In the feature-based approach [4], [5], a geometric feature extraction process is required at the first step to extract some types of geometric features, such as corners or line segments. The performance of the feature-based matching process heavily depends on the consistency of the feature extraction process for the images of the same object acquired under different conditions [4]. Fortunately, a robust feature-matching procedure [5] can relieve this feature extraction inconsistency problem to a certain degree.

Manuscript received April 4, 2005; revised December 23, 2005. This work was supported in part by the MOEA and in part by the National Science Council of Taiwan, R.O.C. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thierry Blu.

The authors are with Department of Computer Science, National Tsing-Hua University, Hsinchu 300, Taiwan (e-mail: dr918308@cs.nthu.edu.tw; lai@cs.nthu.edu.tw).

Color versions of Figs. 5, 6, and 8 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.877500

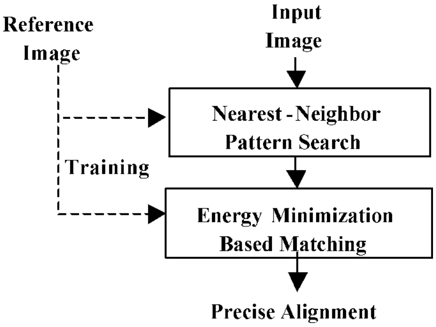

Reference Image
Input Image
Nearest-Neighbor Pattern Search
Training
Energy Minimization Based Matching
Precise Alignment

Fig. 1. Flow diagram of the proposed image alignment algorithm.

In this paper, we propose a new and reliable image-matching algorithm under nonuniform illumination variations based on the relative image gradient matching. The proposed algorithm belongs to the intensity-based matching approach. The main contribution of this paper is the improvement of the robustness of the intensity-based matching approach by matching the corresponding relative gradient maps. In addition, we also proposed a fast two-stage algorithm for accomplishing this image-matching problem. Our image-matching algorithm consists of two stages. Both stages perform image matching based on relative gradients. In the first stage, a fast nearest-neighbor pattern search technique is applied to find a small number of the best candidate poses of the pattern in the image. In the second stage, an iterative energy-minimization procedure is used to refine the candidate poses obtained from the first stage by optimizing the matching between the corresponding relative gradient maps. Experimental results show the robustness of the proposed algorithm under a wide range of illumination variations.

The rest of this paper is organized as follows. We present an overview of the proposed robust image alignment algorithm in Section II. Then, the fast nearest-neighbor pattern search technique is described in Section III. Next, the iterative energy-minimization process for pose refinement is given in Section IV. In Section V, we show some experimental results obtained by applying the proposed new alignment algorithm. Finally, we conclude this paper in Section VI.

## II. Proposed Image-Matching Algorithm

The proposed robust image alignment algorithm consists of a fast nearest-neighbor pattern search procedure and an iterative energy-minimization process. The nearest-neighbor pattern search procedure finds a small number of candidate poses in the image by searching the most similar relative gradient features from a database of such feature vectors corresponding to different sampling in the transformation parameter space. Then, these approximate candidate poses are verified and refined by using an iterative optimization process that minimizes the sum of squared differences between the relative gradient maps.

The flow diagram of the proposed image alignment algorithm is shown in Fig. 1. Note that the solid part is the flow of the execution phase of the proposed image alignment algorithm. The dashed lines in the figure denote the training part of the algorithm.

The novelty of the proposed image alignment is the use of relative image gradient in the matching. This new matching concept is used in both stages of this algorithm. Now, we define the relative image gradient $\overline{G}(x,y)$ as follows:

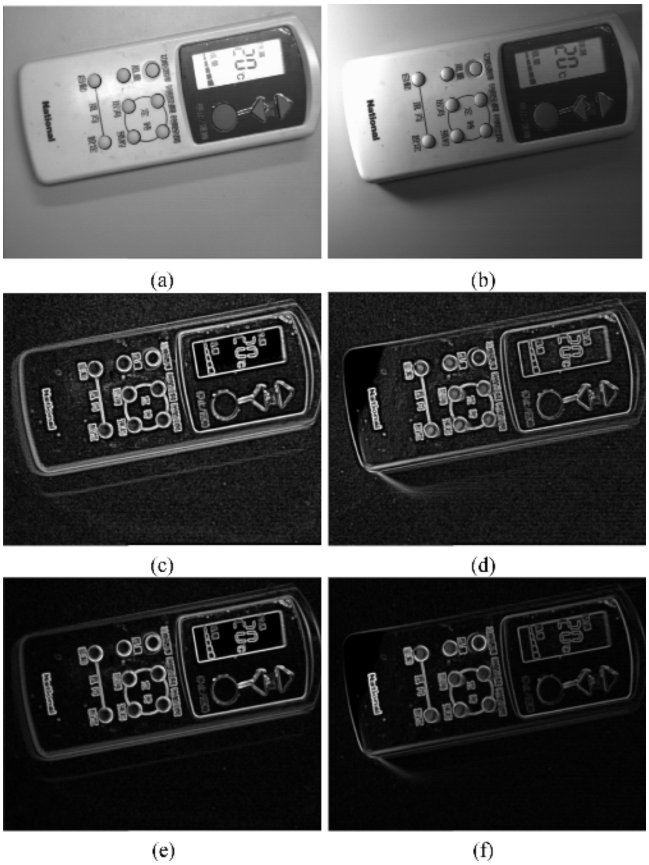


Fig. 2. Images of a remote control acquired under two different lighting conditions are shown in (a) and (b). The corresponding relative gradient images are shown in (c) and (d) with $c = 100$ and (e) and (f) with $c = 1000$, respectively.

$$\overline{G}(x,y) = \frac{|\nabla I(x,y)|}{\max\limits_{(u,v)\in W(x,y)} |\nabla I(u,v)| + c} \tag{1}$$

where $I(x,y)$ is the image intensity function, the notation $\nabla$ denotes the gradient operator that takes the partial differentiation along $x$ and $y$ directions, $W(x,y)$ is a local window centered at the location $(x,y)$, and $c$ is a positive constant used not only to avoid dividing by zero but also to suppress noise effect. To compensate for the effect of lighting variation, we divide the gradient magnitude by its local maximum.

To illustrate the concept of matching the relative gradient function instead of the image brightness function, we show two images of an object acquired under different lighting conditions and the corresponding relative gradient images with different value for the constant c in Fig. 2. From the images, we can see the relative gradient images are very similar for the two differently illuminated images. An appropriate choice for the constant c can suppress noise effect in flat areas and nicely preserve the relative gradient. value under different lighting conditions.

## III. Efficient Nearest-Neighbor Pattern Search

An efficient nearest-neighbor pattern search procedure is used at the first stage of our image alignment algorithm for approximate pattern pose estimation. The results of this procedure provide rough initial guesses for the iterative energy-minimization pose refinement process. This method is based on the learning-from-examples principle [6]. A training database is constructed from synthesizing geometrically transformed images from the reference image with the transformation parameters uniformly sampled from the parameter search space. During the pattern search, the transformation parameter vector of the nearest-neighbor samples searched from the database provides the approximate pose estimates. To improve the efficiency of the search algorithm, we find the best match based on the comparisons between feature vectors computed from the images, instead of comparing directly between images.

There are two phases in this nearest-neighbor pattern search algorithm, i.e., the training phase and execution phase. The training phase involves the generation of image templates from the reference image under different transformation parameter vectors, computation of feature vectors for all the transformed reference images, and training of a fast nearest-neighbor search method. The execution phase consists of the feature generation from the input image and a fast nearest-neighbor search of this feature vector among the training database.

The feature vector used in this procedure is a collection of down-sampled relative gradient maps computed from the image inside a fixed template window. Due to the use of relative gradient feature, this pattern search procedure is insensitive to lighting changes. The feature generation involves the following steps. At first, the relative gradient map is computed from the image inside the template window. Then the relative gradient map within the template window is down-sampled to be the feature vector in the pattern search. The down-sampling of the template window to an appropriate size can significantly reduce the computational cost in feature comparisons.

To further reduce the computational complexity in the nearest-neighbor pattern search procedure, we apply a fast nearest-neighbor search algorithm [6]–[8]. This is very useful when the size of the training database is very large. The size of training database depends on the range of search parameter space and the sampling frequency in the parameter space for generating the training samples.

Our fast nearest-neighbor pattern search is a modification of Nene and Nayar's method [8]. Here, we briefly describe their approximate nearest-neighbor search method in the following. They formulate the problem to find the points in the dataset that are within a distance $\varepsilon$ of a query point $\mathbf{q}$ in each dimension. They first find the points that are sandwiched between a pair of parallel planes $(a-\varepsilon)\mathbf{u}_c$ and $(a+\varepsilon)\mathbf{u}_c$, where $a = \mathbf{q} \cdot \mathbf{u}_c$ and $\mathbf{u}_c$ is the unit vector along dimension $\mathrm{c}$, and put them into a candidate list. Then, they recursively trim the candidate list by discarding points that are not located between the other pairs of parallel planes along other dimensions. After the trimming for all dimensions, the remaining points are bounded by a small hypercube. Then, an approximate nearest neighbor to the query can be determined by exhaustive search in the final set of candidates. Note that this is an approximate nearest-neighbor search method because it cannot guarantee the true nearest neighbor can always stay in the candidate list during the trimming process. This problem is more pronounced when the dimension of the feature vector is large. Therefore, this algorithm is suitable for nearest-neighbor search in a lower dimensional space.

Although Nene and Nayar's approximate nearest neighbor search algorithm [8] is conceptually very simple, there are two critical issues related to this method. One is the order of the trimming sequence and the other is the determination of the trimming boundary parameter $\varepsilon$. For the first problem, the authors argued that the optimal trimming sequence is dependent on the number within the range $\varepsilon$ of the query point in each dimension. Therefore, the trimming sequence is determined when the query point is given. Our modified nearest-neighbor search algorithm provides a way to determine the trimming sequence in the training stage and resolve the aforementioned problem of the nearest-neighbor search in a high-dimensional space. To resolve these two problems, we apply the principal component analysis (PCA) on the feature vectors of the search database to reduce the dimension of feature vector, and then the trimming can be done in the PCA space with the trimming sequence directly determined by the order of the associated eigenvalues. This is because the eigenvalue is directly related to the variance of the projections of all the feature vectors in the database on the associated principle component. Thus, by using PCA, we reduce the dimension of the trimming space and determine the order of trimming in the training stage.

The other issue with Nene and Nayar's nearest-neighbor search approach [8] is how to determine an appropriate trimming parameter $\varepsilon$. This parameter is very critical to the efficiency of the search algorithm. Setting $\varepsilon$ too large may result in a large set of final remaining vectors, while setting $\varepsilon$ too small may result in an empty set of remaining vectors after trimming. Nene and Nayar showed that if the distribution of the point set of the search database around the query point $\mathbf{q}$ is given, then the best value for $\varepsilon$ can be determined. They justify it through experiments on search databases with uniform and normal distributions. However, it may not be feasible to approximate the distribution of the search database by some simple parametric distribution functions in many applications. In this work, we assume that the query point is close to its nearest neighbor in the search database and the samples in the database are well representative such that the distance of the query vector to the nearest neighbor along a principal component direction is less than the maximum of those of all the samples in the database to their nearest-neighbor samples. Thus, we can compute a suitable value for the trimming parameter $\varepsilon$ in each principal component direction during the training phase.

To be more specific, we determine the parameter $\varepsilon$ for each trimming principal component direction during the training stage with the following procedure. First, we find the nearest neighbor sample of each sample in the database to obtain the pairs $\{(\mathbf{p}_i, \mathbf{p}_{nn(i)}) \,|\, i = 1, \ldots, n\}$, where all the $\mathbf{p}_i$s, for $\mathrm{i} = 1, \ldots, \mathrm{n}$, form the database $\mathrm{D}$ and $\mathbf{p}_{nn(i)}$ is the nearest neighbor of $\mathbf{p}_i$, i.e.,

$$nn(i) = \arg \min_{1 \le j \le n, j \ne i} \|\mathbf{p}_i - \mathbf{p}_j\| . \tag{2}$$

The difference vector $\mathbf{d}_i$ for each nearest-neighbor pair $(\mathbf{p}_i, \mathbf{p}_{nn(i)})$ is computed as follows:

$$\mathbf{d}_i = \mathbf{p}_i - \mathbf{p}_{nn(i)}. \tag{3}$$

Then, we compute the maximum value of all the displacement vectors projected onto the k most dominant principal component direction $\mathbf{v}_k$ to be the trimming parameter $\varepsilon_k$ for this direction, i.e.,

$$\varepsilon_k = \max_{1 \le i \le n} \left| \mathbf{d}_i \cdot \mathbf{v}_k \right| . \tag{4}$$

The proposed nearest-neighbor search algorithm via improved PCA-based trimming is given as follows.

*Algorithm 1: The fast nearest-neighbor search algorithm via improved PCA-based trimming*

- *$D = \{\mathbf{p}_i | i = 1, \ldots, n\}$ is the search dataset.*
- ***q** is a query vector.*
- *The goal is to find the nearest vector of **q** in $D$*

***Training Stage:***

*1) Apply PCA on all the vectors in $D$ to find the most dominant PC vectors* $\{\mathbf{v}_k | k = 1, \ldots, m\}$. Note that the PC vectors are ordered such that the corresponding eigenvalues are in a nonincreasing order.

*2) Find the nearest neighbor sample $\mathbf{p}_{nn(i)}$ for each sample $\mathbf{p}_i$ and compute their difference vector $\mathbf{d}_i$, for $i = 1, \ldots, n$.*

*3) Determine the* trimming parameter $\varepsilon_k$ *for the $k$ PC direction to be the maximal absolute projection of all the difference vectors onto this PC vector.*

*4. Build the necessary data structure of the PCA subspace projected search database for the trimming-based nearest-neighbor search, i.e., the order lists for all the PC directions.*

***Execute Stage:***

*1) Set $k = 1$ and candidate list to be the set of all search samples.*

*2) Project the query vector $\mathbf{q}$ onto the $k$ PC direction, i.e., $a = \mathbf{q} \cdot \mathbf{v}_k$.*

*3) Perform a binary search on the order list associated with the $k$ PC such that the corresponding PC projection values inside the interval $[a - \varepsilon_k,\ a + \varepsilon_k]$ remain in the candidate list.*

*4) Set $k = k + 1$. If not converged, return to step 2).*

*5) Perform exhaustive search of the nearest neighbor of $\mathbf{q}$ from the final candidate list.*

Note that the convergence criterion of the proposed algorithm is chosen such that the iteration number k is greater than a fixed number, which can be determined from the eigenvalue distribution in the PCA computation, or the cardinal number in the candidate list is less than a threshold.

## IV. Energy-Minimization-Based Matching

After the proposed nearest-neighbor search algorithm provides an approximate image alignment result, we refine the image alignment by using an iterative energy-minimization procedure. Without loss of generality, we consider 2-D rigid transformation with size changes in the following derivation of this energy minimization algorithm. The scaled rigid transformation $T_{(\Delta x, \Delta y, \theta, \sigma)}$ transforms the location $(x, y)$ to the new location $(u, v)$ as follows:

$$\begin{aligned} \begin{pmatrix} u \\ v \end{pmatrix} &= T_{(\Delta x, \Delta y, \theta, \sigma)}(x, y) \\ &= \sigma \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} + \begin{pmatrix} x_c + \Delta x \\ y_c + \Delta y \end{pmatrix} \end{aligned} \tag{5}$$

where $(\Delta x, \Delta y)$ is the translation vector, $\theta$ is the rotation angle, $\sigma$ is the size scaling factor, and $(x_c, y_c)$ is the center of rotation. Since there are infinitely possible combinations of the translation vector, rotation angle and rotation center for any 2-D rigid transformation, we choose the rotation center to be the same as the center of the template throughout this paper without loss of generality.

The energy function to be minimized in this algorithm is a sum of squared differences between the corresponding relative gradient values given as follows:

$$E(\Delta x, \Delta y, \theta, \sigma) = \sum_i w_i (\overline{G}_0(u_i, v_i) - \vec{G}_1(x_i, y_i))^2 \tag{6}$$

where the location $(u_i, v_i)$ is related to $(x_i, y_i)$ by a scaled 2-D rigid transformation as given in (5), and $w_i$ is the weight associated with the $i$ constraint. Note that the weight $w_i$ is determined from the image grayscale $I_1(x_i, y_i)$ to assign small weights for very dark or very bright grayscales. The selection of the weight function is similar to that described in [9]. This is used to alleviate the problem of relative gradient matching in the shadow or brightness saturation regions.

When the transformation is small, we can take the first-order Taylor series approximation for the function $\overline{G}_0(u_i, v_i)$ to obtain the following new energy function:

$$\begin{aligned} &E'(\Delta x, \Delta y, \theta, \sigma) \\ &\quad = \sum_i w_i (\overline{G}_{x,i}(\Delta x + x_c - x_i) + \overline{G}_{y,i}(\Delta y + y_c - y_i) \\ &\qquad + \sigma(f_i \cos\theta + g_i \sin\theta) + \overline{G}_0(x_i, y_i) - \vec{G}_1(x_i, y_i))^2 \end{aligned} \tag{7}$$

where $\overline{G}_{x,i}$ and $\overline{G}_{y,i}$ are the partial derivatives of the $\overline{G}_0(x, y)$ at the location $(x_i, y_i)$ along $x$ and $y$ directions respectively, $f_i = \overline{G}_{x,i}(x_i - x_c) + \overline{G}_{y,i}(y_i - y_c)$, and $g_i = -\overline{G}_{x,i}(y_i - y_c) + \overline{G}_{y,i}(x_i - x_c)$. This energy minimization is a nonlinear least square minimization problem. When a good initial guess is available, we can employ Newton method to solve this minimization problem very efficiently [9], [10]. In general, when an initial guess for the transformation parameters is given as $(\Delta x^{(k)}, \Delta y^{(k)}, \theta^{(k)}, \sigma^{(k)})$, we can update the transformation parameters by minimizing the following energy function:

$$\begin{aligned} &E^{(k)}(\Delta x, \Delta y, \theta, \sigma) \\ &\quad = \sum_i w_i (\overline{G}_{x,i}(\Delta x + x_c - x_i) + \overline{G}_{y,i}(\Delta y + y_c - y_i) \\ &\qquad + \sigma(f_i \cos\theta + g_i \sin\theta) + \overline{G}_0(x_i, y_i) \\ &\qquad - \overline{G}_0(T^{-1}_{(\Delta x^{(k)}, \Delta y^{(k)}, \theta^{(k)}, \sigma^{(k)},)}(x_i, y_i)))^2. \end{aligned} \tag{8}$$

Note that the inverse transformation $T^{-1}_{(\Delta x,\Delta y,\theta,\sigma)}$ can be proved to be given by

$$T^{-1}_{(\Delta x,\Delta y,\theta,\sigma)}(x,y) = T_{(\Delta x',\Delta y',-\theta,1/\sigma)}(x,y) \tag{9}$$

with $(\Delta x',\Delta y')^T = -(1/\sigma)R_{-\theta}(\Delta x,\Delta y)^T$, where $R_{-\theta}$ is a rotation matrix with rotation angle $-\theta$. When applying Newton method to minimize the dynamic twice-differentiable energy function in (8) with respect to the transformation parameters, we need to compute the function values $\overline{G}_0, \overline{G}_1$, and their partial derivatives at subpixel image coordinates. This is accomplished by using the bilinear interpolation technique and central difference method.

After the minimum solution $(\Delta x',\Delta y',\theta',\sigma')$ is obtained from minimizing the energy function in (8), we can update the transformation parameters as follows:

$$\begin{aligned}\sigma^{(k+1)} &= \sigma'\sigma^{(k)}, \theta^{(k+1)} = \theta'\theta^{(k)}, (\Delta x^{(k+1)}, \Delta y^{(k+1)})^T \\ &= (\Delta x',\Delta y')^T + \sigma' R_{\theta'}(\Delta x^{(k)},\Delta y^{(k)})^T.\end{aligned} \tag{10}$$

An iterative energy minimization algorithm that includes Newton update has been used to obtain the least square solution for minimizing the dynamic energy function in (8) very efficiently, followed by updating the transformation parameters with (10). This process is repeated until the solution is converged.

## V. Experimental Results

Our experimental results contain two parts. The first part of experiments is to show the quantitative accuracy and the robustness of the proposed image alignment algorithm under different simulated illumination conditions and different noisy level. In the second part of our experiments, we use real images acquired from a CCD camera with lighting variations to validate the robustness of the proposed algorithm through visual inspection. In both parts of our experiments, we compare our image alignment algorithm with the standard normalized correlation method to demonstrate its superior performance under illumination variations.

### A. Testing on Synthesized Images

To demonstrate the robustness of the proposed image alignment algorithm, we tested it on images synthesized with random translation and rotation as well as different lighting conditions. The testing platform is a normal PC equipped with P4–2.4 G CPU and 512 mb RAM. In the following experiments, we applied our alignment algorithm on several image datasets synthesized from real images with geometrical transformation parameters and different illumination variations. Each dataset contains 125 images for testing the image alignment algorithms to provide the corresponding quantitative accuracies.

We used a simple lighting model that is similar to the Phong model [15] in computer graphics to synthesize images under different lighting conditions from a real image. The lighting synthesis model contains diffuse and specular components of the Phong model. As depicted in Fig. 3, the radiance for the specular component at each pixel is determined by the angle $\theta$ between the light direction and the local surface, i.e., the specular radiance is propotional to $\cos(90^\circ - \theta)$.

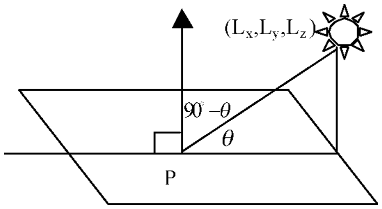


Fig. 3. Simple lighting synthesis model used to synthesize testing images with different illumination conditions in our experiments.

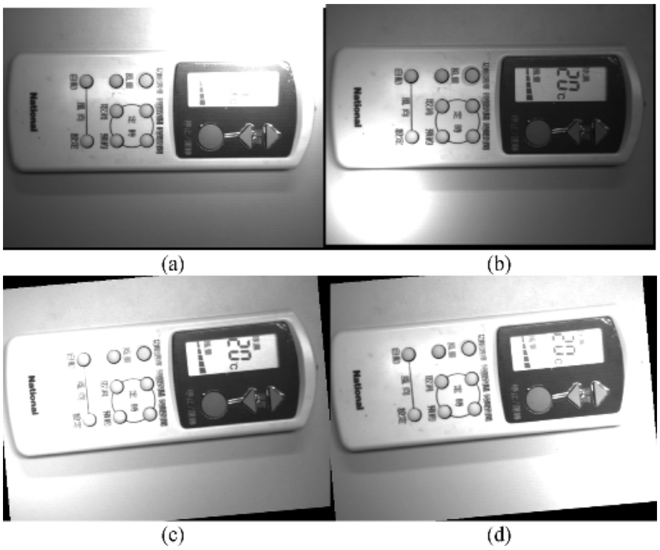


Fig. 4. Example images simulated from a real image with different lighting model parameters associated with the datasets with the parameter settings given in Table I.

The lighting simulation model used for image synthesis with different illumination conditions is given by

$$S'(x,y) = S(x,y)\, L(x,y) \tag{11}$$

where $S(x,y)$ is the image brightness function under normal ambient lighing condition, $L(x,y)$ is a simulated illumination function, and $S'(x,y)$ is the simulated image function under nonuniform illumination condition. In our experiment, we employ the following illumination function to simulate images with different lighting conditions for image alignment

$$\begin{aligned}L(x,y) &= a_0 \times \cos(90-\theta) + a_1 \\ &= a_0 \times \frac{L_z}{\left(\sqrt{(x-L_x)^2+(y-L_y)^2+L_z^2}\right)} + a_1\end{aligned} \tag{12}$$

where $(\mathrm{L}_x,\mathrm{L}_y,\mathrm{L}_z)$ is the position of the spot light, $\mathbf{a}_0$ is the specular reflectance due to spot light, and $\mathbf{a}_1$ is the diffuse reflectance due to ambient light. Different reflectance parameter values are used to synthesize images under a wide range of different lighting conditions from a real image.

In this experiment, we synthesized five datasets from a remote controller image based on different parameter values in the above lighting model each with an example image depicted in Fig. 4. The lighting synthesis parameters for all the testing

TABLE I
PARAMETER VALUES OF THE LIGHTING MODEL FOR THE TESTING DATASETS

| | a0 | a1 | $L_x$ | $L_y$ | $L_z$ |
|---|---|---|---|---|---|
| DataSet1 | 5.0 | 0.1 | 500 | 400 | 70 |
| DataSet2 | 5.0 | 0.1 | 100 | 100 | 40 |
| DataSet3 | 2.0 | 0.1 | 32 | 120 | 200 |
| DataSet4 | 3.0 | 0.3 | 640 | 120 | 150 |

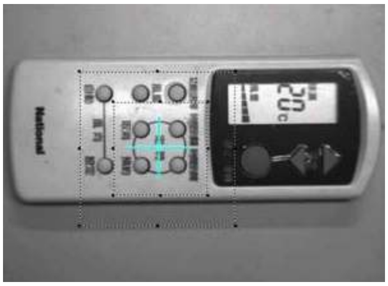


Fig. 5. Training image with the inner dash window representing the template window and the outer one being the search window that specifies the search range. The cross sign denotes the reference location of the template.

TABLE II
RESULTS OF APPLYING OUR ALIGNMENT METHOD ON THE SYNTHESIZED DATASET

| | MEAN | | | STD | | |
|---|---|---|---|---|---|---|
| | ErrAngle | ErrDx | ErrDy | ErrAngle | ErrDx | ErrDy |
| DataSet1 | -0.17426 | -0.21095 | -0.13609 | 0.043354 | 0.075237 | 0.025329 |
| DataSet2 | 0.045631 | 0.01563 | -0.05825 | 0.053462 | 0.074214 | 0.014704 |
| DataSet3 | -0.16132 | -0.03673 | -0.04957 | 0.075187 | 0.081636 | 0.013363 |
| DataSet4 | -0.01868 | -0.1607 | -0.15987 | 0.04424 | 0.071071 | 0.024815 |

datasets are shown in Table I. For each dataset, we simulate 125 images with different translation and rotation parameter values.

Fig. 5 shows a training image that contains two dash windows. The inner window is the template of size 150 × 150, and the outer one is the search window of size 250 × 250.

The results of applying the proposed algorithm on all the datasets synthesized from the remote control image are shown in Table II. It is obvious that the proposed algorithm can provide very accurate image alignment under a wide variety of illumination variations. For comparison, we also applied the normalized correlation method and the image gradient matching on the same datasets. The results of the normalized correlation method and the image gradient-matching method are shown in Tables III and IV, respectively. It is evident that the proposed algorithm significantly outperforms these two methods in this experiment of image alignment on different datasets with different illumination variations. In this experiment, the average time of image alignment by our algorithm takes about 50 ms.

TABLE III
RESULTS OF APPLYING NORMALIZED CORRELATION ON THE SYNTHESIZED DATASET

| | MEAN | | | STD | | |
|---|---|---|---|---|---|---|
| | ErrAngle | ErrDx | ErrDy | ErrAngle | ErrDx | ErrDy |
| DataSet1 | 4.669436 | -0.21051 | 0.92 | 2.938378 | 5.901963 | 3.395773 |
| DataSet2 | 3.685787 | -4.14431 | -1.52 | 1.942157 | 4.010099 | 2.483885 |
| DataSet3 | 4.679366 | -2.20986 | 0.08 | 6.904229 | 10.66075 | 2.85018 |
| DataSet4 | 3.810377 | -0.51338 | 0.76 | 2.402204 | 5.205956 | 3.19071 |

TABLE IV
RESULTS OF APPLYING GRADIENT MATCHING ON THE SYNTHESIZED DATASET

| | MEAN | | | STD | | |
|---|---|---|---|---|---|---|
| | ErrAngle | ErrDx | ErrDy | ErrAngle | ErrDx | ErrDy |
| DataSet1 | 0.448937 | -0.30653 | 0.162593 | 1.623522 | 1.763189 | 1.3729 |
| DataSet2 | 0.564524 | -0.86654 | -0.1461 | 1.087046 | 2.537813 | 1.602871 |
| DataSet3 | 0.462139 | -0.37879 | 0.41878 | 1.56056 | 1.79292 | 1.172546 |
| DataSet4 | 2.319036 | -1.87264 | 0.241884 | 2.880786 | 2.976796 | 2.090353 |

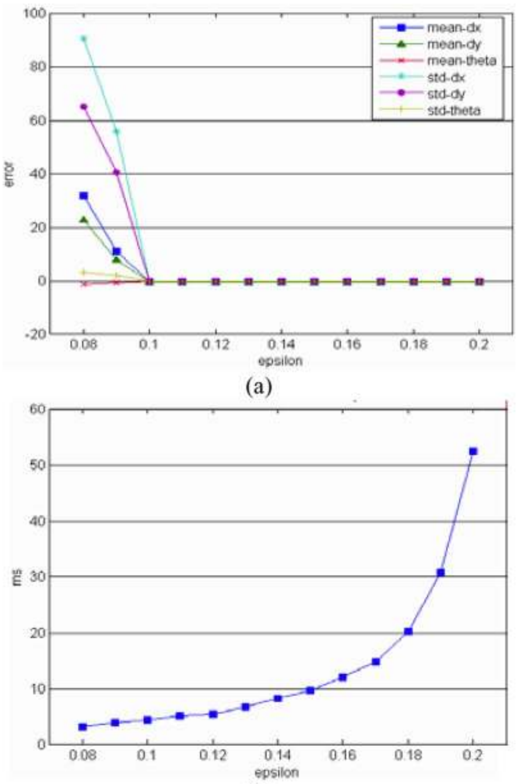
TABLE V
AVERAGE TIME (IN MSEC) IN THE FIRST STAGE OF THE PROPOSED IMAGE ALIGNMENT HIERARCHY WITH DIFFERENT NEAREST-NEIGHBOR SEARCH METHODS

| Method | Exhaustive | KD-tree | Nene and Nayar's (with optimal epsilon) | Our Method |
|---|---|---|---|---|
| search time (ms) | 77.9624 | 712.1936 | 4.3280 | 4.0256 |

To demonstrate the efficiency of the proposed nearest-neighbor search in the first stage of the proposed image alignment algorithm, we compare the execution time required in the nearest neighbor search at the first stage of the proposed image alignment approach by using the proposed PCA-based trimming method, Nene and Nayar's method [8], exhaustive search and the kd-tree method, respectively. The average time required for the nearest-neighbor searches by using these different methods in one image alignment task from the experiments on dataset 1 is given in Table V. It is obvious that the proposed PCA-based trimming method is much more efficient than the other two standard nearest-neighbor search methods and it is as efficient as Nene and Nayar'e method with the optimal trimming parameter setting. The efficiency of the kd-tree method is poor because kd-tree is suitable only for the nearest-neighbor search in a low-dimensional space.

As shown in Table V, Nene and Nayar's nearest neighbor search method [8] is very efficient if the trimming parameter is selected appropriately, but it is still a big problem how to automatically determine an appropriate trimming parameter for each image alignment task. Fig. 6 shows the result of applying Nene and Nayar's nearest neighbor search method in the first stage of our image alignment system with different trimming parameter values on the dataset 1. We can see that the optimal parameter value is 0.1 from Fig. 6. Our proposed nearest neighbor search method can achieve similar accuracy and speed to those of Nene and Nayer's method with optimal parameter setting, but our method does not require human intervention for optimal parameter selection.

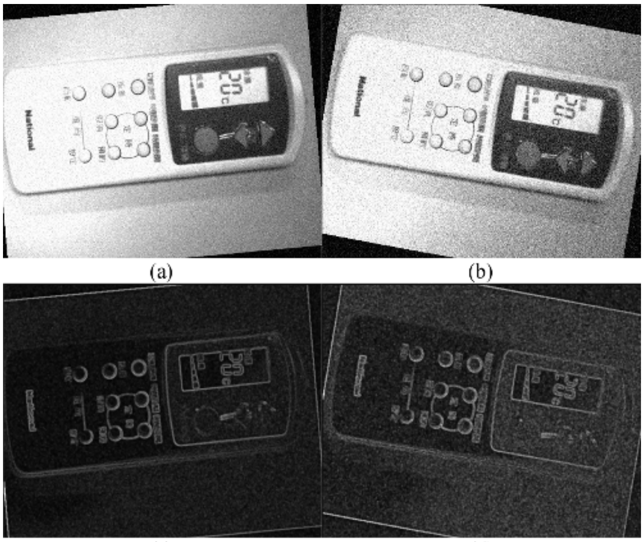
In the following, we show the performance of the proposed image alignment method under different levels of random additive Gaussian noises. Some examples of noisy image are shown in Fig. 7(a) and (b) with different noise standard deviations. The constant $c$ can be used to suppress the noise effect, but it is hard to determine the optimal value for $c$. In this experiment, we select $c = 1500$ empirically. The results of applying the proposed image alignment algorithm on these noisy datasets are given in Table VI. It is evident that the proposed algorithm with appropriate choice of the parameter $c$ value is quite robust to random



(a)

(b)

Fig. 6. Accuracy and speed of applying Nene and Nayar's nearest neighbor search method [8] in the first stage of the proposed image alignment system on dataset 1 with different trimming parameter values are shown in (a) and (b), respectively. We can see the optimal parameter value for this image alignment task is 0.1.



(a) (b)

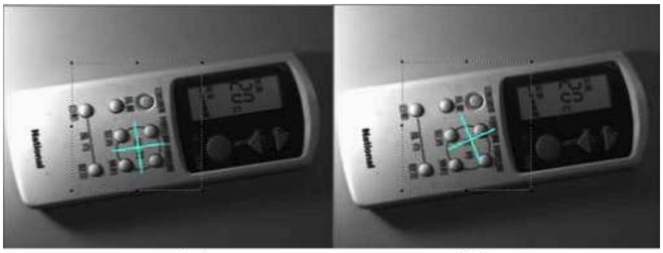
(c) (d)

Fig. 7. Examples of noisy images with additive Gaussian noises with zero mean and standard deviation (a) 18 and (b) 31, respectively, and the corresponding relative gradient maps computed with $c = 1500$ are given in (c) and (d), respectively.

noises. Fig. 7(c) and (d) depicts the corresponding relative gradient maps.

TABLE VI
RESULTS OF APPLYING THE PROPOSED IMAGE ALIGNMENT METHOD ON THE NOISY IMAGES. THE NOISE TYPE IS ADDITIVE GUASSION NOISE WITH ZERO MEAN

| standard | MEAN | | | STD | | |
|---|---|---|---|---|---|---|
| deviation | ErrAngle | ErrDx | ErrDy | ErrAngle | ErrDx | ErrDy |
| 0 | -0.01868 | -0.1607 | -0.15987 | 0.04424 | 0.071071 | 0.024815 |
| 18 | -0.01384 | 0.007048 | 0.075806 | 0.082742 | 0.070913 | 0.022109 |
| 26 | -0.00198 | -0.02474 | 0.076448 | 0.081302 | 0.081065 | 0.036922 |
| 31 | 0.011596 | -0.04803 | 0.080058 | 0.082315 | 0.089693 | 0.04706 |



(a) (b)

Fig. 8. (a) Example of image alignment on real images acquired under different lighting conditions by using the proposed algorithm. The cross signs indicate the estimated results. (b) The results of image alignment on the same images as (a) by using the normalized correlation method.

### B. Real Image Alignment Experiment

We also tested our image alignment algorithm on real images acquired from a CCD camera with different lighting condition. Since we cannot compute the accuracy of the image alignment without knowing the ground truth in this experiment, we show some results of image alignment in Fig. 8. The reference image is shown in Fig. 5 with the template and search windows illustrated in the same figure. The cross sign in this figure specifies the reference location of the template window. Fig. 8(a) depicts the image of the translated and rotated remote control acquired under different lighting conditions. The position and orientation of the template window estimated by the proposed algorithm is represented by the reference cross sign for each of these three examples as shown in the figure. The proposed image alignment algorithm can accurately locate the template from these real images acquired under different lighting conditions. For comparison, we also show the image alignment results of the normalized correlation method on the same images in Fig. 8(b). Obviously, the normalized correlation method fails to provide satisfactory image alignment in these two examples.

## VI. CONCLUSION

In this paper, we presented a new robust image alignment algorithm based on matching of relative image gradients. This image alignment algorithm consists of a fast nearest-neighbor pattern search procedure and an iterative energy minimization alignment process. Both of these two procedures are based on the new concept of matching relative image gradients. The fast nearest-neighbor pattern search finds approximate nearest-neighbor examples from a database of training samples obtained by synthesizing transformed images from a reference template image. The poses of the searched nearest-neighbor examples are verified and refined through the iterative energy minimization process. Experimental results on synthetic and real images with different illumination variations were shown to demonstrate the superior accuracy and robustness of the proposed alignment algorithm over the traditional normalized correlation method.

## REFERENCES

[1] R. T. Chin, "Automatic visual inspection: 1981 to 1987," *Comput. Vis., Graph., Image Process.*, vol. 41, no. 3, pp. 346–381, 1988.

[2] T. S. Newman and A. K. Jain, "A survey of automatic visual inspection," *Comput. Vis. Image Understand.*, vol. 61, no. 2, pp. 231–262, 1995.

[3] S. Manickam, S. D. Roth, and T. Bushman, "Intelligent and optimal normalized correlation for high-speed pattern matching," Tech. Rep., Datacube, Inc.

[4] S. Ando, "Image field categorization and edge/corner detection from gradient covariance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 179–190, Feb. 2000.

[5] H. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," *IEEE Comput. Sci. Eng. Mag.*, vol. 4, no. 4, pp. 10–21, Oct. 1997.

[6] S.-H. Lai and M. Fang, "A hybrid image alignment system for fast and precise pattern localization," *Real-Time Imag.*, vol. 10, no. 1, pp. 359–366, 2001.

[7] J. McNames, "A fast nearest-neighbor algorithm based on a principal axis search tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 9, pp. 964–976, Sep. 2001.

[8] S. A. Nene and S. K. Nayar, "A simple algorithm for nearest-neighbor search in high dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 9, pp. 989–1003, Sep. 1997.

[9] S.-H. Lai, "Robust image matching under partial occlusion and spatially varying illumination change," *Comput. Vis. Image Understand.*, vol. 78, pp. 84–98, 2000.

[10] G. A. F. Seber and C. J. Wild, *Nonlinear Regression*. New York: Wiley, 1989.

[11] M. Gharavi-Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 526–533, Apr. 2001.

[12] S. Kaneko, Y. Satoh, and S. Igarashi, "Using selective correlation coefficient for robust image registration," *Pattern Recognit.*, vol. 36, no. 5, pp. 1165–1173, May 2003.

[13] M.-S. Choi and W.-Y. Kim, "A novel two stage template matching method for rotation and illumination invariance," *Pattern Recognit.*, vol. 35, no. 1, pp. 119–129, 2002.

[14] G. Wolberg and S. Zokai, "Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1422–1434, Oct. 2005.

[15] A. Watt, *3D Computer Graphics*, 3rd ed. Reading, MA: Addison-Wesley, 2000.

[16] P. A. Viola, "Alignment by maximization of mutual information," Ph.D. dissertation, Artif. Intell. Lab., Mass. Inst. Technol., Cambridge, 1995.

[17] E. Haber and J. Modersitzki, "Intensity gradient based registration and fusion of multi-modal images," Tech. Rep., Dept. Math. Comp. Sci., Emory Univ., Atlanta, 2004.

**Shou-Der Wei** was born in Taiwan, R.O.C., in 1973. He received the B.S. degree in computer science from Chung Hua University, Hsinchu, Taiwan, in 1995, and the M.S. degree in computer science from the National Tsing-Hua University, Hsinchu, in 2002, where he is currently pursuing the Ph.D degree in computer science.

He was an Engineer at Holtek Semiconductor, Inc., from 1997 to 2000.

**Shang-Hong Lai** (M'95–M'06) received the B.S. and M.S. degrees in electrical engineering from the National Tsing-Hua University (NTHU), Hsinchu, Taiwan, R.O.C., and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 1986, 1988, and 1995, respectively.

He joined Siemens Corporate Research, Princeton, NJ, as a member of technical staff in 1995. Since 1999, he became a faculty member in the Department of Computer Science, NTHU, where he is currently an Associate Professor in the same department. In 2004, he was a Visiting Scholar with the Department of Electrical Engineering, Princeton University. He has authored more than 80 papers published in related international journals and conferences. He holds ten U.S. patents for inventions related to computer vision and medical image analysis. His research interests include computer vision, visual computing, pattern recognition, medical imaging, and multimedia signal processing.