Scaling-Translation Parameter Estimation using Genetic Hough Transform for Background Compensation

Thuy Tuong Nguyen[1], Xuan Dai Pham[2] and Jae Wook Jeon[1]
[1] School of Information and Communication Engineering,
Sungkyunkwan University, Suwon, Gyeonggi 440-746, Korea
[e-mail: ntthuy@skku.edu, jwjeon@yurim.skku.ac.kr]
[2] Saigon Institue of Technology, Ho Chi Minh City 440-746, Vietnam
[e-mail: xuanpd@saigontech.edu.vn]
*Corresponding author: Jae Wook Jeon

Received April 22, 2011; revised June 21, 2011; accepted August 5, 2011; published August 29, 2011

Abstract

Background compensation plays an important role in detecting and isolating object motion in visual tracking. Here, we propose a Genetic Hough Transform, which combines the Hough Transform and Genetic Algorithm, as a method for eliminating background motion. Our method can handle cases in which the background may contain only a few, if any, feature points. These points can be used to estimate the motion between two successive frames. In addition to dealing with featureless backgrounds, our method can successfully handle motion blur. Experimental comparisons of the results obtained using the proposed method with other methods show that the proposed approach yields a satisfactory estimate of background motion.

Keywords: Affine parameter estimation, background compensation, global motion compensation, multi-resolution hough transform, genetic hough transform, genetic algorithm

A preliminary version of this paper appeared in IEEE ICRA 2008, May 19-23, Pasadena, California, USA. This version includes the Multi-resolution Hough Transform used for background compensation. This work was supported by the Ministry of Knowledge Economy and the Korea Institute for Advancement in Technology through the Workforce Development Program in Strategic Technology.

DOI: 10.3837/tiis.2011.08.004

1. Introduction

Motion based tracking via active cameras is associated with several difficulties, including the necessity of compensation for background motion (also known as global motion) and detection of features of object motion. Background motion estimation can also be used as the foundation for powerful video processing and coding. However, background compensation involves a heavy computational burden, for which there are no optimal solutions.

Therefore, several alternative approaches have been proposed to compensate for background motion. In [1], the relationships between pixels representing the same 3-D point in images captured from different camera orientations are used to eliminate background motion. For this approach to function correctly and accurately, knowledge of the values of the pan and tilt angles of the camera are necessary for each captured image. However, a disadvantage of this approach is the difficulty of timming synchronization, which is required for obtaining pan and tilt angles and capturing images. In [2], lines, polygons or other background line structures are matched to compensate for background motion. Background motion is represented by an affine transformation in [3]. Parameters of the affine motion are estimated using the Least Median of Squares (LMedS) method. In this method, a number of feature points in the current frame and their corresponding feature points in the remaining frame are selected, and the resulting pixels are selected from the moving objects' regions and considered to be outliers of the affine parameter estimator. However, choosing feature points in this manner is very sensitive due to motion blur, as feature points can remain outside the background as well as outside the relevant region of the moving object of interest, potentially resulting in inaccurate parameters. In [4], M-estimator like techniques in a multi-resolution framework are used as a parametric motion model estimation algorithm.

In the present paper, we propose a Genetic Hough Transform (GHT) based method to estimate the background motion between two successive images. Previously, the Hough Transform (HT) [5, 6] has been widely applied in motion analysis. Importantly, the advantages of HT involve its robustness to image noise and its ability to detect pattern discontinuities. In [7], a randomized HT is proposed as a means to analyze translational and rotational motions. This HT method randomly chooses pairs of edge pixels from two successive images and it computes the translation between them. Likewise, [8] describes a standard HT applied for achieving fast optimization of linear velocity motion. In [9], HT with projection (HTP), which was designed for dealing with 3-D data, is proposed as a method for estimating the velocities of moving objects. Experimental data in [9] show that utilization of Standard HT might lead to increased noise sensitivity compared with the proposed HTP method. Moreover, the disadvantages associated with Standard HT requiring significant computational power and large storage.

The present paper describes a Genetic Algorithm (GA) combined with HT. GAs [10] have been widely applied in motion estimation. GAs are stochastic search methods based on the principles of natural selection and genetics, and are used to identify approximate solutions of optimization and search problems. A GA in a continuous space (GACS) was proposed previously for matching global motion [11]. Contrary to conventional GAs, the method presented in this study utilizes an actual floating point instead of a series of bits to represent chromosomes. Indeed, in the present study, the chromosomes are formed by the combination of six motion parameters according to the affine model, the experimental results of which show that the GACS method outperforms hierarchical model-based motion estimation

methods [12]. A multi-resolution GA was previously proposed to solve image-related optimization problems including image segmentation, stereo vision, and motion estimation [13]. With respect to the motion estimation problem, these studies demonstrated that the velocities of moving objects can be efficiently estimated even if the objects are in areas that do not contain sufficicent intensity variation. Moreover, using these approaches, the resulting velocity fields can describe the shapes and numbers of moving objects in a given scene. Nevertheless, the processing time required for such GA approaches remains high.

The GHT based method described in the present paper represents part of a continuum of research on developing a novel background compensation method. Our previously published work, which is based on multi-resolution HT [14], has been described in [15] with the goal of achieving high accuracy. In the present study, we combined GA with HT, because GA is extremely effective for solving combinatorial optimization problems. In this method, the first of any two successive gray images is separated into a number of binary images. Then, vertical and horizontal histograms of binary images separated from given gray images are matched to those of binary images separated from preceding gray images, and possible matches accumulate in a Hough space. During this accumulation process, every time the peak of Hough space reaches a threshold, the corresponding parameters at the cell of that peak are defined as new individuals in the GA population. In this manner, new individuals are accumulated and created until the GA has a sufficient number of individuals in the population. Importantly, our Hough based method, which utilizes automatic initialization of a GA population that has an appropriate abundance of guesses, can efficiently reduce execution time. Moreover, because this method involves selecting good candidates for inclusion in the GA population, the accuracy of the affine motion estimator is also increased. In addition to the automatic population initialization that was used to reduce computation time, fitness prediction (FP) is alternately used with fitness evaluation (FE) in the GA to simplify the calculation of the fitness value. Likewise, FP improves computational speed, while FE attempts to correct the inaccurate fitness values predicted by the FP.

The content of this paper is organized as follows. In section 2, the existing background compensation method based on multi-resolution HT (MHT) is reviewed. Section 3 describes the proposed background compensation method using the GHT. Section 4 discusses the computational complexity of the proposed method. Section 5 presents experimental results. Lastly, the conclusions of the paper are described in section 6.

2. Previous Work: Background Compensation using MHT

Motion features of a moving object can be detected using the image difference technique. This technique is widely used for detecting moving objects [1]. The difference image, or inter-frame difference (IFD), is calculated by performing pixel-by-pixel subtraction between two successive images. When the camera rotates and moves, the background of the captured images is altered. Thus, the background motion must be compensated for in order to detect motion features using IFD.

This section reviews the MHT method for estimating background motion between two successive images. Unlike [3], this method does not use pairs of feature points and their corresponding points to determine the parameters of transformation, thereby alleviating the difficulties of selecting feature points and their corresponding points in successive images. **Fig. 1** shows two difficult cases of selecting corresponding feature points in two successive frames. These frames were captured when the camera was rotating and changing zoom values. In **Fig. 1(a)** and **Fig. 1(b)**, the difficulty of selecting feature points and their corresponding points for

estimating parameters in some parts of the background (i.e., the ceiling and the wall) are apparent because the gray intensities of the pixels in those parts are similar. In the remaining case, shown in **Fig. 1(c)** and **Fig. 1(d)**, the two frames were affected by camera motion blur. Similar to the first case, **Fig. 1(a)** and **Fig. 1(b)**, it has low ability of choosing the correct feature and corresponding points in both frames.

(a), **(b)** The intensities of pixels in the ceiling and the wall are similar.

(c), **(d)** Images are affected by motion blur.

Fig. 1. Difficult cases for selecting feature points and corresponding points in successive frames.

The MHT based background compensation algorithm is summarized in a block diagram shown in **Fig. 2**. First, binary images are created from two successive gray images that are used to estimate background motion. Next, the binary vertical and horizontal histograms of the created binary images are matched, and all possible best matches accumulate in a Hough space. For the purposes of this section, we assume a multi-resolution Hough space. Finally, the affine motion parameters are estimated using the cell with the greatest accumulating value in the multi-resolution Hough space. We compensate for background motion by using these affine motion parameters.

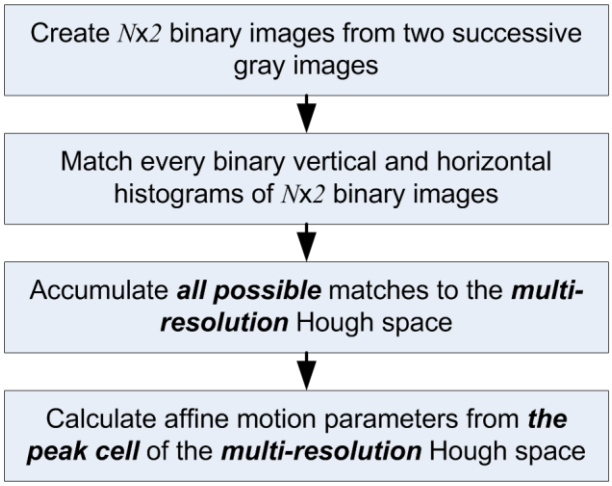


Fig. 2. Diagram of the MHT based background compensation method.

Let $I(t-1)$ and $I(t)$ be two images at time index $t-1$ (the previous image) and t (the current image), respectively. Given the aforementioned assumptions, the non-proportional scaling transformation followed by a translation is geometric. This 2-D transformation is used to transform $I(t-1)$ to $I(t)$, where each point $(x, y)^T$ in $I(t-1)$ is mapped to a corresponding point $(x', y')^T$ in $I(t)$ using the geometric transformation:

$$\begin{pmatrix} x' & y' \end{pmatrix}^T = \mathbf{A}\begin{pmatrix} x & y \end{pmatrix}^T + \mathbf{b}, \tag{1}$$

where $\mathbf{A} = (a_{11} \;\; 0; \; 0 \;\; a_{22})$; $\mathbf{b} = (b_1 \;\; b_2)^T$; $a_{11}, a_{22} > 0$. It is noted that camera motion might not necessarily contain rotation and shear motion; therefore, only scale and translation motion are considered in this paper. **Fig. 3** shows the resulting transformation mapping the point $\mathbf{x} = (x, y)^T$ in $I(t-1)$ to the point $\mathbf{x}' = (x', y')^T$ in $I(t)$.

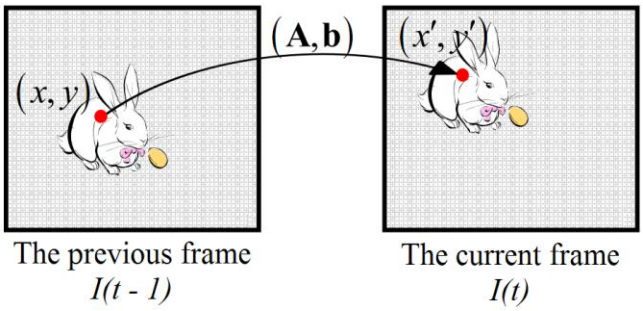


Fig. 3. Motion represented by a scaling transformation followed by a translation.

The IFD applied background compensation between two successive images $I(t-1)$ and $I(t)$ is defined by $IFD(t, \mathbf{x}') = | I(t, \mathbf{x}') - I(t-1, \mathbf{x}) |$. At first, the previous gray scale image $I(t-1)$ is separated into N binary images, $I^l(t-1, \mathbf{x})$ where $0 \le l \le N-1$. In the original gray image $I(t-1)$, every pixel with an intensity within the range $[255l / N; 255(l+1) / N]$ becomes a white pixel in the corresponding binary image $I^l(t-1)$; otherwise, it becomes a black pixel. Similarly, the current gray scale image $I(t)$ is also separated into N binary images, $I^l(t, \mathbf{x}')$ where $0 \le l \le N-1$. Next, we match the binary vertical and horizontal histograms of $N \times 2$ binary images. Let

$H^V(I)$ be the vertical histogram of the image I, giving

$$H^V(I) = \left\{ h_j^V(I) : j = 0,1,...,W-1 \right\}, \tag{2}$$

where W is the number of columns; $h_j^V(I)$ is the binary histogram or the number of white pixels in column j. Similarly, $H^H(I)$ is defined as the horizontal histogram. All points having the same x should have the same $x' = a_{11}x + b_1$ after mapping; and all points having the same y should have the same $y' = a_{22}y + b_2$. Indeed, column j is mapped onto column j', and row i is mapped onto row i', where $j' = a_{11}j + b_1$ and $i' = a_{22}i + b_2$. Likewise, when column j' in $I^l(t, \mathbf{x}')$ is obtained from the mapping of column j in $I^l(t-1, \mathbf{x})$, the number of white pixels in column j' is approximately equal to that in column j:

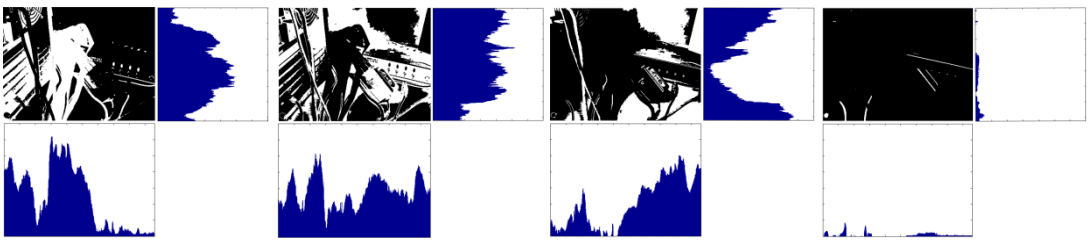
$$h_j^V\left[I^l(t-1)\right] \approx h_{j'}^V\left[I^l(t)\right]. \tag{3}$$

A threshold T^V is used to measure the similarity between $h_j^V\left[I^l(t-1)\right]$ and $h_{j'}^V\left[I^l(t)\right]$, where $h_j^V\left[I^l(t-1)\right] = 0$ implies column j does not correspond to any column j':

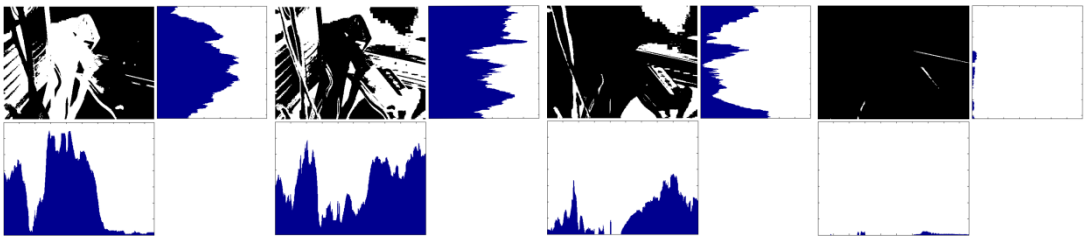
$$\left| \frac{h_j^V\left[I^l(t-1)\right] - h_{j'}^V\left[I^l(t)\right]}{h_j^V\left[I^l(t-1)\right]} \right| < T^V. \tag{4}$$

Similarly, a threshold T^H is used to measure the similarity between $h_i^H\left[I^l(t-1)\right]$ and $h_{i'}^H\left[I^l(t)\right]$. **Fig. 4** shows an example of two gray images and their corresponding binary images. In this case, each gray image is separated into $N = 4$ binary images. Indeed, the white pixels in **Fig. 4(c)** and **4(g)** represent the pixels in gray images that have values within [0, 63], and the white pixels in **Fig. 4(d)** and **4(h)** represent the gray values within [64, 127], and the same goes for **Fig. 4(e)**, **4(f)**, **4(i)** and **4(j)**. In all of the cases described above, each binary image is illustrated with its vertical histogram (on the down side) and horizontal histogram (on the right side).

(a), **(b)** Gray images of **Fig. 1(c)** and **1(d)**



(c), **(d)**, **(e)**, **(f)** Binary images of **(a)** and their corresponding vertical and horizontal histograms



(g), **(h)**, **(i)**, **(j)** Binary images of **(b)** and their corresponding vertical and horizontal histograms

Fig. 4. An example of two gray images and their corresponding binary images.

To determine the values of a_{11} and b_1, vertical histograms of the binary previous and current images are used. For each value of l, where $0 \le l \le N-1$, all possible pairs of (j, j') that satisfy (4) are searched. Similarly, the values of a_{22} and b_2 are determined based on horizontal histograms of the two binary images. Equations $j' = a_{11}j + b_1$ and $i' = a_{22}i + b_2$ show that the relationships between j, i and j', $i'$are linear. Therefore, the affine motion parameters can be determined using HT. Importantly, while general HT has the disadvantages of requiring high computation power and large memory resources, MHT [14] can be applied to reduce computation time and storage requirements based on multi-resolution images and accumulator arrays. Logarithmic range reduction is used for achieving faster convergence. To determine a_{11} and b_1, all points (j, j') are transformed using the equation expressed in polar coordinates $\rho = j\cos\theta + j'\sin\theta$ to obtain ρ as θ changes successively in the parameter space. Having the values (ρ,θ), the HT algorithm then looks for accumulator cells into which the parameters fall. Likewise, the HT then increases the values of those cells. The most dominant global motion, that we have to estimate, is represented by the highest value in the parameter space. If we let the values of ρ and θ at the peak cell be ρ^V and θ^V, the result is $a_{11} = -\cos\theta^V / \sin\theta^V$ and and $b_1 = \rho^V / \sin\theta^V$. Similarly, $a_{22} = -\cos\theta^H / \sin\theta^H$ and and $b_2 = \rho^H / \sin\theta^H$ for cases where ρ^H and θ^H are the ρ and θ of the peak cell in the horizontal Hough space. We can reduce the search range of θ by limiting the scale factor changes. In this analysis, scale parameters a_{11} and a_{22} must be greater than zero, and should be limited by the range $[a_{\min}, a_{\max}]$ that contains 1 (1 implies no scale). The range of θ is defined as follows:

$$\arctan\left(-1/a_{min}\right) \le \theta \le \arctan\left(-1/a_{max}\right). \tag{5}$$

In the MHT algorithm, if the number of iterations L is large, then the computation time required for the HT is reduced. However, the multi-resolution based algorithm is unstable when L is increased, because the smallest image used in the first iteration lacks distinct features that are required for detecting the transformation parameters. This situation implies that the smallest images resulting from L down-sampling reductions of previous and current images are too similar to use to determine the parameters of transformation. In this method, L was set to 2.

3. Background Compensation using Genetic Hough Transform

This section describes the Genetic HT based method for background compensation. This work is intended to improve the accuracy and processing time of the MHT based method that was presented in the previous section. The HT is robust with respect to image noise; however, the discretization of a low resolution Hough space might lead to large errors in the final results.

Here, GA is used to evolve and reduce the error coming from results of the HT.

Our method is summarized as follows: 1) two successive gray images are both separated into a number of binary images, and vertical and horizontal histograms of the separated binary images are matched in the same way as shown in the two first steps of the algorithm in section 2, where 2) possible matches accumulate in a Hough space to initialize the GA population, and 3) affine motion parameters are determined according to the best individuals in the most recent GA population when the GA iterations are finished. **Fig. 5** presents a block diagram of the proposed method. This method differs from MHT based methods in its second and third steps, because it replaces the accumulation of all match pairs, the identification of the final peak based on the stepwise accumulation of a number of match pairs, and the choice of the peak (not the final peak) if the peak value satisfies a threshold. This method saves time when it does not scan through the Hough space. Moreover, the GA selects candidate peaks as good guesses for inclusion in its population. Next, this stochastic search technique, based on the principles of natural selection and genetics, is used to identify an optimal solution. The second and third steps of the GHT based method are described below.

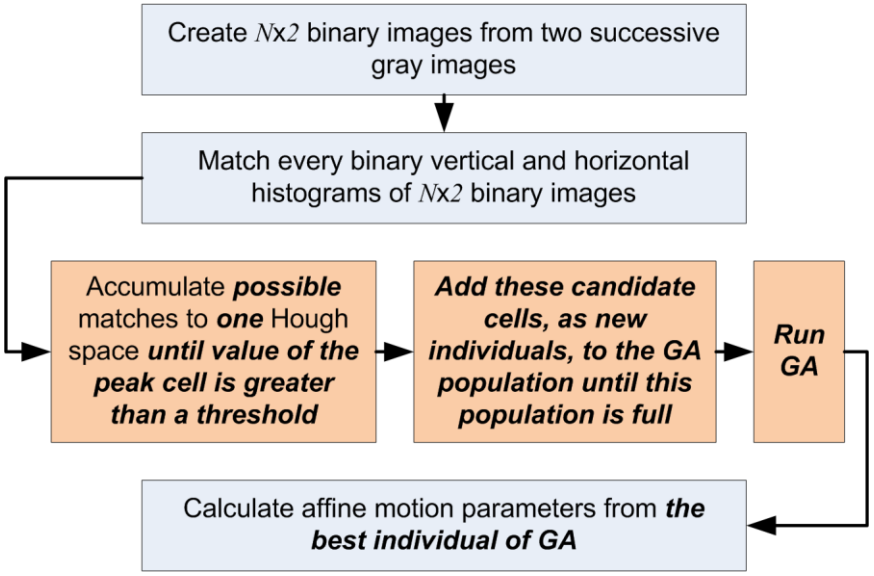


Fig. 5. Diagram of the GHT based background compensation method.

3.1 Using HT to initialize the GA population

To minimize computational requirements, the HT based method used to initialize the GA population proceeds as follows. A new match pair, or a new feature point, is randomly selected for voting. After making a vote, the current peak is tested to determine whether or not it exceeds the threshold T_v. The test of the peak requires a single comparison to the threshold per accumulator cell update. When a likely global motion is detected, the supporting feature points retract their votes. This step is similar to emptying the Hough space. However, the supporting points are not removed from the set of all possible points, and the process continues with other random selections. The threshold T_v is set by assuming that all match pairs (feature points) are caused by noise, which is the worst-case assumption. However, this assumption is also valid if a great deal of motion is present in the two successive images because there a fraction of feature points still belong to the most likely global motion. This threshold is manually defined

in this paper.

The HT algorithm has several interesting properties. Candidate or likely global motion is detected as soon as the content of the Hough space allows a decision to be made. Furthermore, it can be interrupted and still yield useful results. A full HT, which requires significant amounts of computational time, is not performed in such cases. Indeed, depending on the data, only a small fraction of match pairs are allowed to vote, the rest being used to support the next selection to initialize the GA population. If constraints are given, e.g., the minimum likely global motion, then a stopping rule can be tested before selecting a new match pair for voting.

3.2 Using a GA to find an optimal solution

A chromosome in a GA can be identified using the array formed by parameters of the motion model. In affine motion models (1), in this paper the chromosome is formed by combining four motion parameters. The chromosomes representing four motion parameters a_{11}, a_{22}, b_1 and b_2 should be actual floating point vectors instead of a series of bits. Although the series of bits are very widely used, they have shortcomings in the linkage between different solution parameters and quantization [11]. Moreover, the chromosomes formed by the concatenation of the four parameters are simpler but no less robust than those formed by the concatenation of six parameters. Thus, they might lead the GA to the convergence point faster.

In our GA model, first, a population with a given number of chromosomes is created by the HT described in the previous section. The initialization step is not critical, provided that the initial population narrows or spans the range of variable settings. If the explicit information of the system is provided in advance, then this information can be included in the initial population (e.g., our method creates good guesses for the population). In the second step of the GA, the fitness of each individual is evaluated. The goal of the fitness evaluation is to numerically encode the performances of chromosomes. For motion compensation applications, the fitness function is defined as:

$$f_E\left(C_j^i\right) = \frac{1}{\eta} SAD\left((a_{11})_j^i, (a_{22})_j^i, (b_1)_j^i, (b_2)_j^i\right)$$
$$= \frac{1}{\eta}\sum_{\mathbf{x}} \left| I(\mathbf{x}, t) - Aff\left(I(\mathbf{x}, t-1)\right)\right|, \quad (6)$$

where $C_j^i = \left((a_{11})_j^i, (a_{22})_j^i, (b_1)_j^i, (b_2)_j^i\right)$ is chromosome j (j = 1, 2..., N_p, where N_p is the number of individuals in the population) at the ith generation; SAD represents the sum of absolute differences; $I(\mathbf{x}, t)$ and $I(\mathbf{x}, t-1)$ are the pixel values of the point $\mathbf{x} = (x, y)$ in the frame t and $t-1$, respectively; Aff is the affine transformation function that utilizes $\left((a_{11})_j^i, (a_{22})_j^i, (b_1)_j^i, (b_2)_j^i\right)$; η is the number of common points in images $I(t)$ and $I(t-1)$. The third step is the natural selection step, which is implicitly coupled with the replacement step. In fact, when a new individual enters a population, another individual must leave. The process of going from the current population to the next population constitutes a generation of the GA. The fourth step is applying crossover and mutation. Two chromosomes (parents) from the current population are randomly selected to be mated. The chromosomes that are not selected to mate pass on unchanged to the next generation. One-point, n-point, uniform crossover, or multiple crossovers per couple [17] can be used for recombination. Likewise, mutations result int small changes to a single gene in order to maintain a way of exploring the search space. Gene-wise mutation is primarily used at this stage. At this point, all of the above steps are then

repeated for each generation until a termination condition is met.

In order to speed up the process of the GA search, we propose a method of alternating fitness evaluation (FE) and fitness prediction (FP). There are a number of methods intended to reduce computational complexity such as hybridizing [18], compacting GA [19], predicting [17, 20], inheriting [21], imitating [22] or partially evaluating fitness [23]. Although these GA approaches can yield optimal solutions, they are still imperfected because high accuracy and fast computation cannot be obtained at once. The user should have some knowledge about the nonlinearities of real-world applications in order to define trade-offs between accuracy and computation speed. For instance, a method of fitness prediction is provided in [17] to reduce the execution time of GA in optimizing switched reluctance motor parameters. The proposed method can find an optimum solution faster than conventional methods, while it yields acceptable results in terms of accuracy.

In this paper, however, we focus only on a simple method that alternately employs FE and FP. FE is not completely replaced by FP in all generations, but only in a few. FP is simpler than FE in terms of computation, and thus it improves computation speed, whereas FE attempts to correct the inexact fitness values predicted by FP. The fitness predicted by FP may be exactly equal, or approximately equal, to the fitness evaluated by FE. The predicted fitness of chromosome *j* at the *i*th generation is defined as:

$$f_P\left(C_j^i\right) = w_I f\left(C_I^{i-1}\right) + w_{II} f\left(C_{II}^{i-1}\right), \tag{7}$$

where $f\left(C_I^{i-1}\right)$ and $f\left(C_{II}^{i-1}\right)$ are the fitness values of the two individuals who are most similar to individual C_j^i of the previous generation. These fitness values can be estimated by FE or FP. The w_I and w_{II} are the corresponding weights of the two most similar chromosomes in the (i–1)th generation. The weight w_k is calculated by using the similarity function $s\left(C_j^i, C_k^{i-1}\right)$ between individual C_j^i and N individuals C_k^{i-1} (k = 1, 2..., N) from the most recent generation [17]. The similarity function is measured using the Euclidean distance, and the weight is as follows:

$$w_k = \frac{s\left(C_j^i, C_k^{i-1}\right)}{\sum_{j'=1}^{N} s\left(C_j^i, C_{j'}^{i-1}\right)}. \tag{8}$$

The FP might negatively affect the performance of the GA methods. Therefore, FE is manipulated to improve performance. **Fig. 6** shows the fitness values when applying GA based background compensation, with the alternation of two FEs and one FP, to two successive images in the Flower Garden sequence, as is shown in **Fig. 8(a)**. Ten generations were made in this case. In **Fig. 6**, two FEs are performed on the first run, and one FP is then executed; two FEs then continue the generation, and so on. FP may degrade the next generation because it sometimes does not yield exact fitness values. This leads us to define three rules as follows:

- *Rule 1*: The fitness calculation of the first generation must be a FE. This rule makes the basis for the later FP.
- *Rule 2*: The fitness calculation of the last generation should be a FE. If FP is performed on the last generation, then it may degrade the final result. Therefore, FE is utilized in order to ensure the best result.

- *Rule 3*: The more FP is used, the greater the obtained error, and less time is consumed.

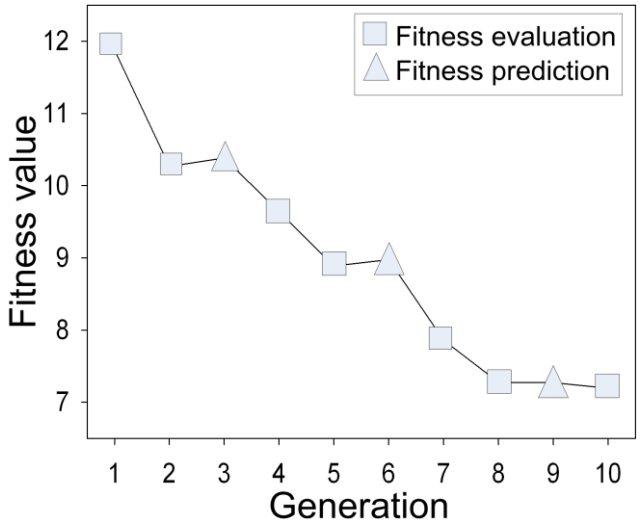
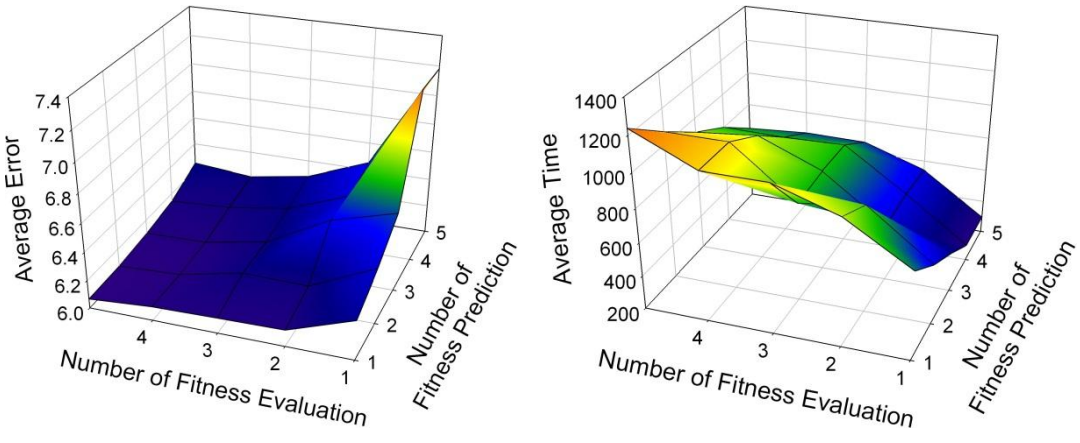


Fig. 6. Fitness values obtained by altering two FEs and one FP.

Following rule 3, a comparison of the results obtained from alternating the number of FEs (n_{fe}) and the number of FPs (n_{fp}) is shown in **Fig. 7**, where n_{fe}, $n_{fp} \in [1...5]$. The result obtained in this figure is also from the Flower Garden sequence. As we see in **Fig. 7(a)**, the case ($n_{fe} = 5$, $n_{fp} = 1$) yields the best error (lowest fitness value). Conversely, the worst error is obtained in the case ($n_{fe} = 1$, $n_{fp} = 5$). However, the best error case takes the longest time (1237 ms), while 265 ms is required for the worst error case (**Fig. 7(b)**). Processing time was calculated using parameter sets that will be given later in the experimental results section.



(a) (b)

Fig. 7. Average error (a) and time results (b) obtained from various alternations of FE and FP.

At this point, a definition of the trade-off between accuracy and speed (i.e., the number of FEs and the number of FPs) is required. Based on experiments in which the number of

generations was set to 10, we chose $n_{fe} = 2$, $n_{fp} = 1$ as the best trade-off for the proposed GA model. This trade-off can be used for any data set and for any application.

4. Computational Complexity of the GHT

The MHT was compared with the SHT and the Adaptive HT (AHT) [16] in [14]. The MHT outperforms the SHT and AHT with respect to computational complexity. We do not consider the memory requirements of these algorithms because requirements are continually addressed by advances in technology. Thus, the consideration of computational cost outweighs concern for memory savings [14]. The computational complexity of MHT is determined by

$$C^M = a_{size}^M n_p^M + \frac{a_{size}^M}{\sigma} \cdot \frac{n_p^M}{\sigma^2} + \ldots + \frac{a_{size}^M}{\sigma^{n_i^M - 1}} \cdot \frac{n_p^M}{(\sigma^2)^{n_i^M - 1}}, \tag{9}$$

where a_{size}^M is the size of the accumulator array; n_p^M is the total number of feature points (match pairs); σ is the reduction factor; and n_i^M is the number of iterations required for MHT. In section 2, the existing algorithm stops at the number of iterations $L = n_i^M = 2$; thus, the complexity in (9) becomes:

$$C^M = a_{size}^M n_p^M + \frac{a_{size}^M}{\sigma} \cdot \frac{n_p^M}{\sigma^2}. \tag{10}$$

The computational complexity of the HT used to create the GA population is

$$C^G = a_{size}^G n_p^G, \tag{11}$$

where a_{size}^G is the accumulator array size and n_p^G is a small number of feature points in the HT. In the accumulation process, every time the peak reaches T_v, there are a specific number of feature points $n_p^G < n_p^M$. Certainly, the vote threshold T_v must be smaller than the threshold used in the SHT or MHT. From (10) and (11), we have

$$\frac{C^M}{C^G} = \frac{a_{size}^M n_p^M \left(1 + \frac{1}{\sigma^3}\right)}{a_{size}^G n_p^G}. \tag{12}$$

Assuming $a_{size}^M = a_{size}^G$ and having $\sigma > 0$, we simplify (10) to:

$$\frac{C^M}{C^G} = \frac{n_p^M \left(1 + \frac{1}{\sigma^3}\right)}{n_p^G} > 1. \tag{13}$$

In MHT, all feature points are found by scanning all possible match pairs in the binary histograms; therefore, the complexity of finding all points is:

$$C_{n_p}^M = I_{size} w_{size}, \tag{14}$$

where I_{size} is the width (in case of vertical histograms) or height (in case of horizontal

histograms) of the image; and w_{size} is the size of the window where the neighborhoods of the estimated values are searched. The Hough Transform used to create the GA population has a complexity of randomly finding a number of feature points:

$$C_{n_p}^{G} = I'_{size} w'_{size}, \tag{15}$$

where $I'_{size} << I_{size}$ and $w'_{size} \leq w_{size}$. Therefore, $C_{n_p}^{G} << C_{n_p}^{M}$, and from (13), we have:

$$C^{G} << C^{M}. \tag{16}$$

In addition, in order to obtain a smaller computational complexity of the proposed GHT based method as compared to the multi-resolution based method, a condition should be given:

$$\frac{C^{M}}{N_p C^{G} + C^{GA}} > 1, \tag{17}$$

where N_p is the total number of individuals in the GA population; and C^{GA} is the computational complexity of the GA process. The experimental results show that the proposed GHT outperforms the MHT based method. Moreover, (16) and (17) are satisfied.

5. Experimental Results

Four image sequences were used in the experiments: Flower Garden, OSU-2, Table Tennis (these three sequences are available at: *http://sampl.ece.ohio-state.edu/data/motion/*), and Tsukuba (*http://vision.middlebury.edu/stereo/data/scenes2001/*). The Flower Garden sequence was used to test the estimation of small motion changes, while the Tsukuba sequence was utilized for large motion changes. The Table Tennis sequence was used to test scaling, while OSU-2 contained small and large motion changes during which the camera was panned and tilted. We extracted two sample frames (frame #0-#1 of Flower Garden, #10-#11 of OSU-2, #30-#31 of Table Tennis and #0-#1 of Tsukuba) from each image sequence for the first test scenario, which was a comparison between the proposed method and other GA methods, as well as other non-GA methods. The second test scenario used the whole image sequence to compare the proposed method with other non-GA methods. **Fig. 8** shows the sample frames of the test sequences.

(a) **(b)** **(c)** **(d)**

Fig. 8. Test images. (a) Frame #0 of Flower Garden (resolution of 352×240); (b) frame #10 of OSU-2 (320×240); (c) frame #30 of Table Tennis (352×240); (d) frame #0 of Tsukuba (384×288).

The PC used in this paper was an AMD Athlon 64 X2 Dual Core Processor 3800+ 2.00 GHz with 2 GB RAM. Parameters and operators used in the GA based method of the first scenario are presented in **Table 1**. The parameters of GA methods include population size N_p, crossover

rate p_c, mutation rate p_m and number of generations n_{gen}. As described in section 3.2, n_{fe} = 2 and n_{fp} = 1 were applied when n_{gen} = 10 was used. The number of iterations for the whole GA process (n_{iter}) was set to 100. The scale range was defined as ±10%, and the translation range was [−30; 30]. **Table 2** shows the results of error and time comparisons of the proposed method, the classical GA, and the GA proposed in [11] and [17]. While the proposed method and the classical GA used uniform crossover and gene-wise mutation, [11] is based on a random variable uniformly distributed on a unit interval and a Gaussian random variable to take form the crossover and mutation. Conversely, crossover in [17] involves multiple crossovers per couple (MCPCP). In this type of crossover, FE is replaced by FP after childbirth. Two parents are mated and replaced by the two individuals with the highest predicted values generated from C crossovers. The proposed GHT yields the best e_{best} and $\bar{e}$ in most cases. In addition, the time required for the proposed GHT is lower than those required by the other three methods. The reasons why the proposed model in [17] has the highest error and time are: 1) selecting the two children with the highest predicted values might spoil the replacement of the two parents because FP itself is not perfectly accurate, and 2) only the latest generation is used to predict offspring fitness. Because the search range is limited using the HT, the GHT method can reduce the outliers that affect average error.

Table 1. Parameters of the proposed method and other GA models

	Proposed method	**Classical GA**	**GA Moscheni [11]**	**GA Mutoh [17]**
Np	50			
Selection	Tournament			
Crossover	Uniform	Uniform	Using a λ random variable	MCPCP
p_c	0.5			
Mutation	Gene-wise	Gene-wise	Using a Gaussian random variable	Gene-wise
p_m	0.05			
n_{gen}	10			
n_{iter}	100			

Table 2. Error and time comparison results of the proposed method and other GA models (in case of two successive images)

		Flower Garden (#0-#1)	OSU-2 (#10-#11)	Table Tennis (#30-#31)	Tsukuba (#0-#1)
Proposed method	e_{best}	*7.534*	*1.897*	*4.208*	5.577
	$\bar{e}$	**8.43**	**2.094**	**5.318**	*5.926*
	t (ms)	**951.92**	**879.29**	**993.71**	**1046.05**
Classical GA	e_{best}	7.534	1.901	4.22	*5.469*
	$\bar{e}$	*9.49*	2.446	*5.323*	**5.9**
	t (ms)	1289.94	1165.63	1330.72	1506.99
GA Moscheni [11]	e_{best}	**7.533**	**1.895**	**4.143**	**5.447**
	$\bar{e}$	9.683	*2.36*	5.382	5.998
	t (ms)	*1283.56*	*1151.28*	1347.08	*1504.57*
GA Mutoh [17]	e_{best}	7.647	1.973	4.416	5.504
	$\bar{e}$	12.848	3.754	6.241	7.216
	t (ms)	1308.77	1199.65	*1284.65*	1438.96

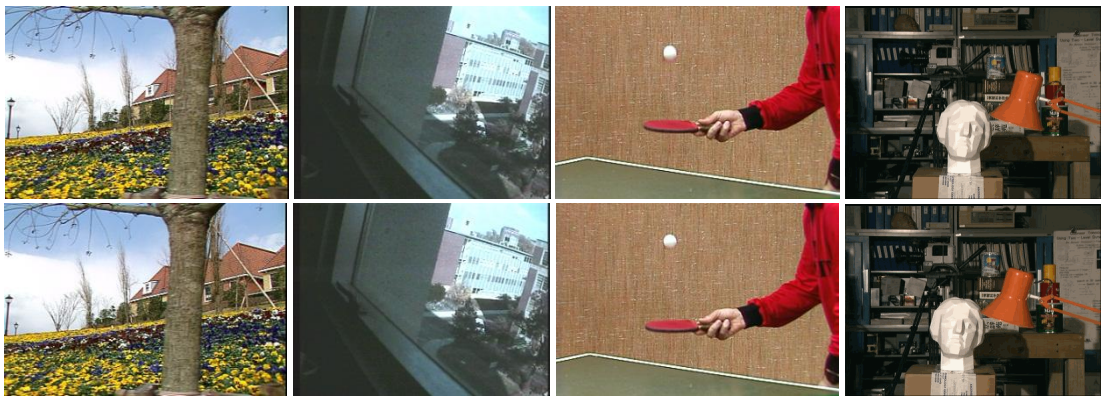
In addition, because the proposed method is based on the HT, its Hough-related parameters

were the same as those in the MHT based method [15]: the number of binary images N was 32, the tolerance value T was 0.1, the scale range was [0.9, 1.1] ($\theta \in [-48.04^{o}, -42.30^{o}]$), and the resolution was $\Delta\theta = \Delta\rho = 0.0003$. Moreover, for the MHT, the number of iterations was 2, and the reduction factor $\sigma = 2$, $\mu = 1$; and for the proposed GHT, the vote threshold T_v was 10. **Table 3** shows the error and processing time comparison of the proposed method and other non-GA methods for two successive images. In the RANSAC algorithm [24], the performance depends on the number of iterations. In our study, the number of iterations was 1000. The proposed GHT method outperforms other methods in terms of accuracy and processing time.

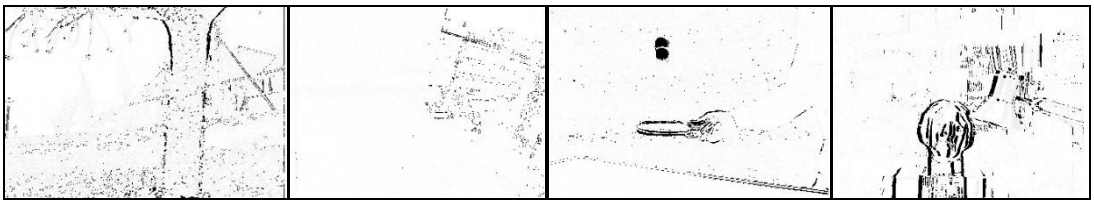
Table 3. Error and time comparison of the proposed method and other methods (in case of two successive images)

		Flower Garden (#0-#1)	OSU-2 (#10-#11)	Table Tennis (#30-#31)	Tsukuba (#0-#1)
Proposed method	*e*	**7.534**	**1.897**	**4.208**	**5.577**
	t	**963**	**887**	**982**	**1031**
RANSAC [24]	*e*	10.878	4.061	7.507	6.71
	t	15531	5140	12500	7656
MHT based method [15]	*e*	10.261	2.426	5.939	6.63
	t	1828	1110	1078	1453

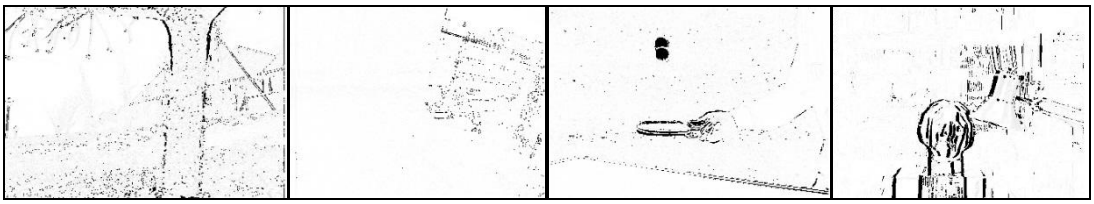
Fig. 9 shows the errors of four pairs of images. Fewer black pixels in results indicate smaller errors. The error, calculated using (6), was measured by intensity per common pixels. The results of the proposed method and other three GA methods were very similar in the intuitive view, because their errors had only small differences. **Fig. 10** presents an average error and processing time comparison between the proposed GHT and other methods for two successive images. Along with the classical GA and GA Moscheni [11], GHT was the best method in terms of accuracy.



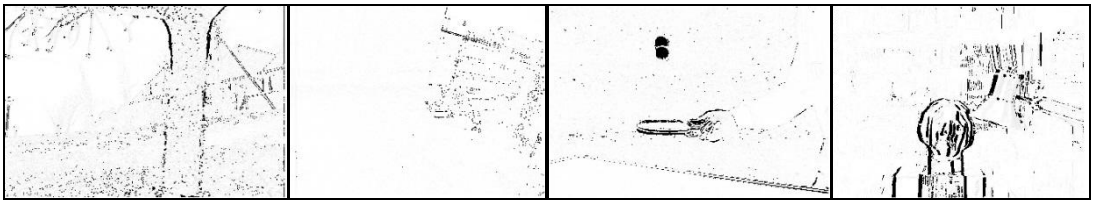
(a) Pairs of input successive images. First row: the previous images (#0, #10, #30, #0); second row: the current images (#1, #11, #31, #1) of the corresponding sequences



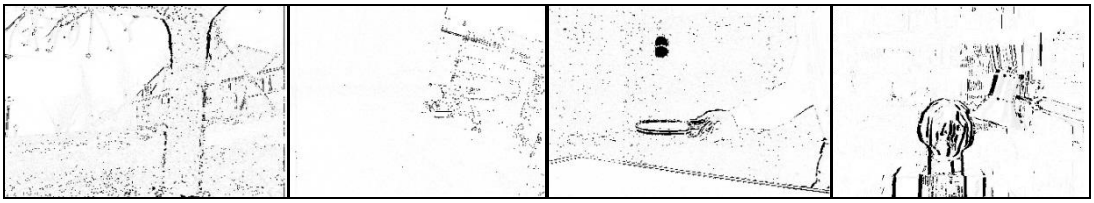
(b) Results of the proposed method



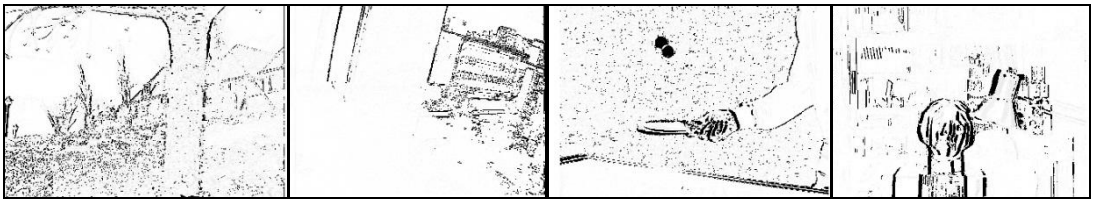
(c) Results of the classical GA based method



(d) Results of GA Moscheni [11]



(e) Results of GA Mutoh [17]



(f) Results of RANSAC [24] based method

(g) Results of MHT [15] based method

Fig. 9. Results of background compensation methods for four image pairs: Flower Garden (#0-#1), OSU-2 (#10-#11), Table Tennis (#30-#31) and Tsukuba (#0-#1).

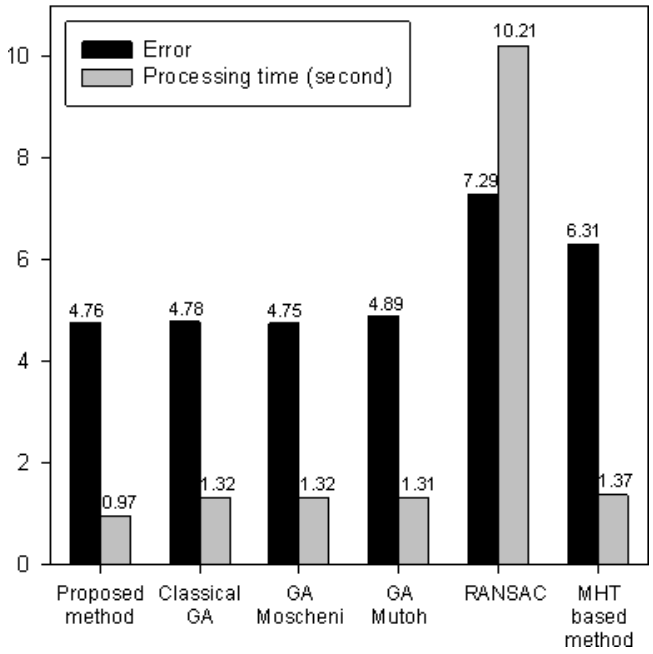


Fig. 10. Comparison of average error and processing time for two successive images.

In the second test scenario, four image sequences were used to compare the proposed method with RANSAC and the MHT based method. The results of error and time comparisons of these methods are presented in **Table 4** and **Fig. 11**. One hundred frames were used in the Flower Garden image sequence, 100 frames were used in the OSU-2 sequence, 50 frames were used in the Table Tennis sequence, and 8 frames were used in the Tsukuba sequence. The GHT method outperforms RANSAC and MHT based methods in terms of both accuracy and processing time. Additionally, the robustness of our method was evaluated in terms of critical situations, namely, featureless background and motion blur. Three sample image results are shown in **Fig. 12** to illustrate for the critical situations. In these cases, it is difficult to select feature points in the three input images because of the dark region shown in **Fig. 12(a)** and the mirror shown in **Fig. 12(b)**. In **Fig. 12(c)**, there are no feature points on the wall and table; moreover, motion blur occurs around the white ball and moving hands. **Table 5** presents the error comparison to featureless background and blurring effect. In this table, the average errors of all experimental images are categorized into two types: all critical situations considered, and no critical situation considered. Our method can handle critical cases as well as no critical cases, and the increase in error between these cases was small.

Table 4. Error and time comparison of the proposed method and other methods (for the whole image sequences)

		Flower Garden (100 frames)	OSU-2 (100 frames)	Table Tennis (50 frames)	Tsukuba (8 frames)
Proposed method	$\bar{e}$	**7.629**	**3.824**	**3.979**	**6.011**
	$\bar{t}$	**906.586**	**872.939**	**1110.245**	**1237.286**
RANSAC [24]	$\bar{e}$	17.209	6.479	8.205	8.992
	$\bar{t}$	19477	5351.85	12471.3	7932.86
MHT based method [15]	$\bar{e}$	9.852	5.321	5.538	6.782
	$\bar{t}$	2168.64	990.566	1930.35	1388.43

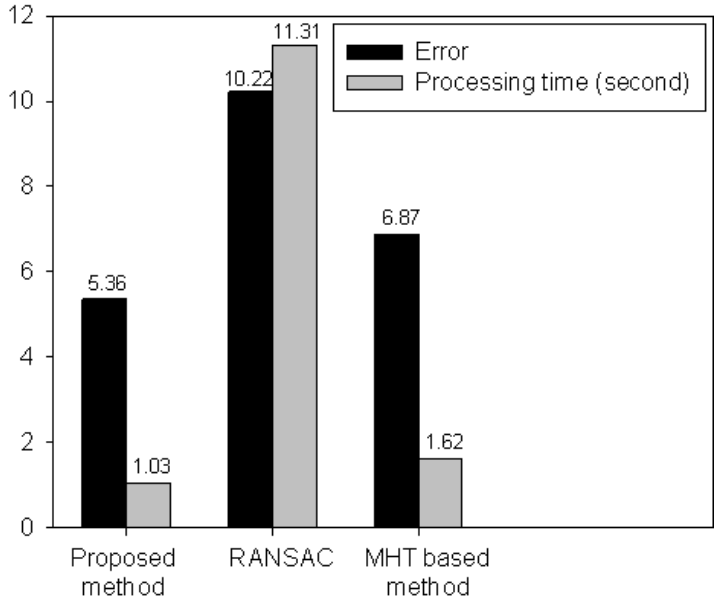
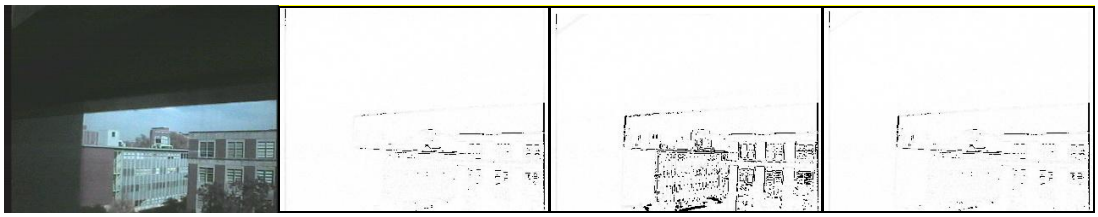
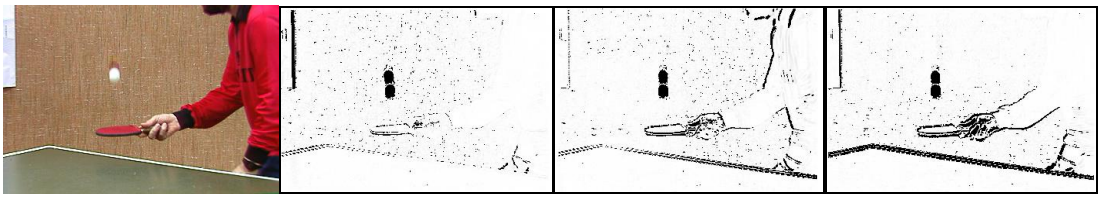


Fig. 11. Comparison of average error and processing time for whole image sequences.



(a) Result of images #69 and #70 from OSU-2

(b) Result of images #98 and #99 from OSU-2



(c) Result of images #38 and #39 from Table Tennis

Fig. 12. Results of our method, the RANSAC based method, and the MHT based method (from left to right) in critical situtation. The images shown in the first column represent two successive images.

Table 5. Robustness in critical situations: featureless background and motion blur

	Proposed method	RANSAC [24]	MHT based method [15]
Average error in case critical situations excluded	5.107	8.995	6.416

Average error in case critical situations included	5.361	10.221	6.873
Error increase	0.254	1.226	0.457

6. Conclusions

In this paper we describe a novel GHT method for background motion compensation. The proposed method contains two major components: 1) the HT that is used to initialize the GA population, and 2) the GA that is then applied to find an optimal solution. In HT, the possible matches obtained by matching binary vertical and horizontal histograms accumulate into the Hough space. Every time the peak of the Hough space reaches a threshold, the corresponding parameters at the cell of that peak are utilized as a new individual of the GA population. Based on appropriate estimates, the GHT outperforms both conventional GA and other non-GA methods. Furthermore, in GA, FP is used to interpolate the new fitness of the current generation based on the fitness of the past generation. FP reduces the computational complexity of fitness calculation. FE and FP are alternately utilized to increase speed and to maintain accuracy of the GHT method. By experimentally comparing the results of the proposed method with those of other GA models and other non-GA methods, we found that the proposed approach yielded satisfactory estimates of background motion.

References

[1] D. Murray, A. Basu, "Motion Tracking with an Active Camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 449-459, 1994. Article (CrossRef Link)

[2] Q. Cai, A. Mitiche, J. K. Aggarwal, "Tracking Human Motion in an Indoor Environment," in *Proc. Int. Conf. Image Process.*, pp. 215-218, 1995. Article (CrossRef Link)

[3] S. Araki, T. Matsuoka, N. Yokoya, H. Takemura, "Real-time Tracking of Multiple Moving Object Contours in a Moving Camera Image Sequence," *IEICE Trans. Inform. Syst.*, vol. E83-D, no. 7, pp. 1583-1591, 2000. Article (CrossRef Link)

[4] J. Odobez, P. Bouthemy, P. Temis, "Robust Multi-resolution Estimation of Parametric Motion Models in Complex Image Sequences," *J. Vis. Commun. Image Represent.*, vol. 6, pp. 348-365, 1994. Article (CrossRef Link)

[5] P.V.C. Hough, "Method and Means for Recognizing Complex Patterns," *U.S. Patent 3069654*, 1961. Article (CrossRef Link)

[6] R.O. Duda, P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun. ACM*, vol. 15, pp. 11-15, 1972. Article (CrossRef Link)

[7] H. Kalviainen, E. Oja, L. Xu, "Randomized Hough Transform applied to Translational and Rotational Motion Analysis," in *Proc. Int. Conf. Pattern Recognit.*, pp. 672-675, 1992. Article (CrossRef Link)

[8] Y. Pnueli, N. Kiryati, A.M. Bruckstein, "Hough Techniques for Fast Optimization of Linear Constant Velocity Motion in Moving Influence Fields," *Pattern Recognit. Lett.*, vol. 15, no. 4, pp. 329-336, 1994. Article (CrossRef Link)

[9] G. Mostafaoui, C. Achard, M. Milgram, "A Hough Transform with Projection for Velocity Estimation," *Mach. Vis. Appl.*, 2008. Article (CrossRef Link)

[10] J. H. Holland, "Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence," *University of Michigan Press*, Ann Arbor, 1975. Article (CrossRef Link)

[11] F. Moscheni and J. Vesin, "A Genetic Algorithm for Motion Estimation," in *Proc. 15eme Colloque sur le Traitement des Signaux et Images*, Juan-les-Pins, France, pp. 825-828, 1995. Article (CrossRef Link)

[12] J. Bergen, P. Anandan, K. Hanna, R. Hingorani, "Hierarchical Model-Based Motion Estimation,"

in *Proc. Euro. Conf. on Comput. Vis.*, pp. 237-252, 1992. Article (CrossRef Link)
[13] M. Gong, Y.H. Yang, "Quadtree-Based Genetic Algorithm and Its Applications to Computer Vision," *Pattern Recognit.*, vol. 37, no. 8, pp. 1723-1733, 2004. Article (CrossRef Link)
[14] M. Atiquzzaman, "Multi-resolution Hough Transform - An Efficient Method of Detecting Patterns in Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 11, pp. 1090-1095, 1992. Article (CrossRef Link)
[15] X.D. Pham, J.U. Cho, J.W. Jeon, "Background Compensation using Hough Transformation," in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 2392-2397, 2008. Article (CrossRef Link)
[16] J. Illingworth, J. Kittler, "The Adaptive Hough Transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 690-698, 1987. Article (CrossRef Link)
[17] A. Mutoh, T. Nakamura, S. Kato, H. Itoh, "Reducing Execution Time on Genetic Algorithm in Real-World Applications using Fitness Prediction: Parameter Optimization of SRM Control," in *Proc. IEEE Congress Evol. Comput.*, pp. 552-559, 2003. Article (CrossRef Link)
[18] V. Rodehorst, O. Hellwich, "Genetic Algorithm Sample Consensus (GASAC) - A Parallel Strategy for Robust Parameter Estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshop*, pp. 103-110, 2006. Article (CrossRef Link)
[19] G.R. Harik, F.G. Lobo, D.E. Goldberg, "The Compact Genetic Algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 6, pp. 287-297, 1999. Article (CrossRef Link)
[20] M.D. Schmidt, H. Lipson, "Coevolution of Fitness Predictors," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 736-749, 2008. Article (CrossRef Link)
[21] J.H. Chen, D.E. Goldberg, S.Y. Ho, K. Sastry, "Fitness Inheritance in Multi-Objective Optimization," in *Proc. Genetic Evol. Comput. Conf.*, pp. 319-326, 2002. Article (CrossRef Link)
[22] Y. Jin, B. Sendhoff, "Reducing Fitness Evaluations using Clustering Techniques and Neural Network Ensembles," in *Proc. Genetic Evol. Comput. Conf.*, pp. 688-699, 2004. Article (CrossRef Link)
[23] A.A.O. Rodriguez, M.R.S. Ortiz, "Partial Evaluation in Genetic Algorithms," in *Proc. Int. Conf. Ind. Eng. Appl. Artif. Intell. Expert Syst.*, pp. 217-222, 1997. Article (CrossRef Link)
[24] D.A. Forsyth, J. Ponce, "Computer Vision - A Modern Approach," *Englewood Cliffs*, NJ: Prentice-Hall, 2003. Article (CrossRef Link)

Thuy Tuong Nguyen received the B.S. (magna cum laude) degree in computer science from Nong Lam University, Ho Chi Minh City, Vietnam, in 2006 and the M.S. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, Korea, in 2009. He is currently working toward the Ph.D. degree with the School of Information and Communication Engineering, Sungkyunkwan University. His research interests include computer vision, image processing, and graphics processing unit computing.

Xuan Dai Pham received the B.S. degree in computer engineering from the University of Technology, Ho Chi Minh City, Vietnam, in 1994, the M.S. degree in computer science from the University of Natural Sciences, Ho Chi Minh City, in 2001, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, Korea, in 2008. Since 2008, he has been with the Information Technology Faculty, Saigon Institute of Technology, Ho Chi Minh City, as a Lecturer and a Researcher. His research interests include computer vision and image processing.

Jae Wook Jeon received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1990. From 1990 to 1994, he was a Senior Researcher with Samsung Electronics, Suwon, Korea. Since 1994, he has been with the School of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, as an Assistant Professor, where he is currently a Professor. His research interests include robotics, embedded systems, and factory automation.