GA–SSD–ARC–NLM for Parametric Image Registration

Felix Calderon, Leonardo Romero, and Juan Flores

Universidad Michoacana de San Nicolás de Hidalgo
División de Estudios de Posgrado. Facultad de Ingeniería Eléctrica
Santiago Tapia 403 Centro. Morelia, Michoacán, México. CP 58000
calderon@umich.mx, lromero@umich.mx, juanf@umich.mx

Abstract. We present the GA–SSD–ARC–NLM, a new robust parametric image registration technique based on the non–parametric image registration SSD–ARC algorithm. This new algorithm minimizes a new cost function quite different to the original non-parametric SSD-ARC, which explicitly models outlier punishments, using a combination of a genetic algorithm and the Newton–Levenberg–Marquardt method. The performance of the new method was compared against two robust registration techniques: the Lorentzian Estimator and the RANSAC method. Experimental tests using gray level images with outliers (noise) were done using the three algorithms. The goal was to find an affine transformation to match two images; the new method improves the other methods when noisy images are used.

1 Introduction

The parametric image registration problem [1] consists of finding a parameter set which allows us to match an origin image with a target image. Many algorithms try to minimize the Sum of Squared Differences (SSD) between the origin and target images. Successful SSD applications, including the classical Least Squared method (LS), are presented in [2,3,4]. Nevertheless the SSD based algorithms have poor performances in cases of noisy images and outliers. In particular, the problem with LS is that outliers have a huge weight in the cost function (and gradient vector) and pull the solution towards them; robust methods try to exclude outliers in some way.

Two well known robust methods in the computer vision literature are the Lorentzian Estimator (LE) and the the Random Sample Consensus method (RANSAC). Some authors, instead of using SSD, use the Lorenzian estimator [5] in cases of noisy images, getting good results. The RANSAC method is a stochastic technique which presents good results with outliers, and its application for image mosaics is presented in [6].

Another robust method is called, Sum of Squared Difference with Adaptive Rest Condition (SSD–ARC), which models outliers rejection inside the SSD cost function. An SSD-ARC application for non-parametric image registration was presented in [7] and another application for non-parametric camera calibration

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 227–236, 2006.

is found in [8]. In both articles, the non-parametric SSD–ARC cost function is minimized using a coarse to fine strategy (scale space) and the Richarson Iteration [9].

This paper describes a new robust method named GA–SSD–ARC–NLM that combines the explicit outlier rejection idea from non-parametric SSD–ARC [7] into a new cost function with a searching process in two stages. The first stage is done with a Genetic Algorithm (GA) and the goal is to find an approximate solution inside a bounded parameter space. The second stage refines the solution, reached by GA, using the Newton–Levenberg–Marquardt (NLM) method [9].

An experimental comparison among GA–SSD–ARC–NLM, RANSAC and LE show the robustness pf GA–SSD–ARC–NLM in cases of noisy images.

2 Registration Using an Affine Transformation

The Affine Transformation (AT) [10,11] allows us to compute, at the same time, translation, rotation, scaling, and sharing of images. An AT uses a six-parameter vector Θ, and maps a pixel at position r_i (with integer coordinates $[x_i, y_i]$) to a new position $\widehat{r}_i$ (with real coordinates $[\widehat{x}_i, \widehat{y}_i]$) given by

$$\widehat{r}_i(\Theta) = \begin{bmatrix} \widehat{x}_i \\ \widehat{y}_i \end{bmatrix} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \end{bmatrix} \Theta = M(r_i)\Theta \tag{1}$$

where $M(r_i)$ is the matrix of coordinates and $\Theta = [\theta_0 ... \theta_5]$ is the parameter vector.

The image registration problem, try to find the Θ which match an origin or source image I_1 on a target image I_2. However in practical cases both images are corrupted by noise and the problem is to find the best AT to match a transformation of I_1 into I_2. A very well known method to evaluate the match quality is to compute the Sum of Squared Differences (SSD) between the source and target images, pixel by pixel, as in Equation (2).

$$E(\Theta) = \sum_{i=1}^{N} \left[I_1(\widehat{r}_i(\Theta)) - I_2(r_i) \right]^2 = \sum_{i=1}^{N} e_i(\Theta)^2 \tag{2}$$

with a difference vector image e_i given by

$$e_i(\Theta) = I_1(\widehat{r}_i(\Theta)) - I_2(r_i) \tag{3}$$

where $I(r_i)$ is the gray level value of pixel r_i in image I. Using this error measurement, SSD, the registration image task consists of finding Θ^* that makes E reaches a minimum value (Θ^* is the minimizer of Equation (2)). This method is named Least Squared (LS) and some strategies to minimize Equation (2) are presented in [9].

Using the ρ-function, defined by Hampel in [12], the SSD can be defined by Equation (4) with $\rho_{LS}(e_i) = e_i^2$.

$$E_{LS}(\theta) = \sum_{i=1}^{N} \rho_{LS}(e_i(\theta)) \tag{4}$$

The influence function is defined by Hampel in [12], as the derivative of the ρ-function and it helps to see the contribution of the errors to the right solution (see [12]). In the LS case, the influence function is given by (5)

$$\psi_{LS}\left(e_i\right) = 2e_i \tag{5}$$

A robust function presented in [5], is the Lorentzian Estimator (LE), which has a ρ-function and influence function given by Equation (6) and (7)

$$\rho_{LE}(e_i) = \log\left(1 + \frac{e_i^2}{2\sigma^2}\right) \tag{6}$$

$$\psi_{LE}(e_i) = \frac{2e_i}{2\sigma^2 + e_i^2} \tag{7}$$

note, the term $\frac{1}{2\sigma^2+e_i^2}$ in (7), reduces the error contribution on the gradient vector and it is not present in Equation (5). This fact explains why LS is notorious sensitive to outliers. Another new function with similar performance is the parametric SSD–ARC function which is described in the following section.

2.1 Parametric SSD-ARC

The Sum of Squared Differences with Adaptive Rest Condition for non–parametric Image Registration was presented, by Calderon in [7], as the minimization of a quadratic energy function $\widehat{E}_{SSD-ARC}$ with a term $l_i h_i$ to reduce huge error contribution given by Equation (8)

$$\widehat{E}_{SSD-ARC}\left(V, l\right) = \sum_{i=1}^{N}(e_i\left(V_i\right) - l_i h_i)^2 + \mu\sum_{i=1}^{N} l_i^2 + \frac{\tau\mu}{4}\sum_{i=1}^{N}|\nabla V_i|^2 \tag{8}$$

where h_i is an error dependent function, $l_i \in [0, 1]$ is an outlier indicator function under the control of parameter μ, V_i is the displacement vector for each image pixel, and the last term is a homogeneity constrain with regularization parameter τ.

In our case, assuming an AT, $h_i = e_i(\Theta)$, and annulling the homogeneity constraint, a particular parametric SDD-ARC function can be obtained as

$$\widehat{E}_{SSD-ARC}\left(\Theta, l\right) = \sum_{i=1}^{N} e_i^2(\Theta)(1 - l_i)^2 + \mu\sum_{i=1}^{N} l_i^2 \tag{9}$$

For this parametric SDD–ARC function, the term $(1 - l_i)^2$ allows us to discard outliers. The second term in Equation (9) restricts the number of outliers by means of μ. The minimizer l_i^* for Equation (9) can be computed by solving $\frac{\partial\widehat{E}_{SSD-ARC}(\Theta,l)}{\partial l_i} = 0$, so the solution for l_i^* is given by Equation (10). We refer to l_i^* as the outlier field.

$$l_i^* = \frac{e_i^2(\Theta)}{\mu + e_i^2(\Theta)} \tag{10}$$

Replacing the value of l_i^* in Equation (9), we have a new parametric SSD–ARC function $E_{SSD-ARC}(\Theta)$ given by Equation (11), which has an unimodal ρ–function and an influence function given by Equations (12) and (13) respectively.

$$E_{SSD-ARC}(\Theta) = \sum_{i=1}^{N} \frac{\mu e_i^2(\Theta)}{\mu + e_i^2(\Theta)} \tag{11}$$

$$\rho_{SSD-ARC}(e_i) = \frac{\mu e_i^2}{\mu + e_i^2} \tag{12}$$

$$\psi_{SSD-ARC}(e_i) = \frac{2\mu^2 e_i}{(\mu + e_i^2)^2} \tag{13}$$

Note, the parametric SSD-ARC influence function exhibits a behavior similar to the Lorentzian Estimator influence function. In both functions, large differences give derivatives values near to zero, as can see in Equations (13) and (7) respectively. The parametric SSD–ARC influence function has a maximum value located at $\widehat{e} = \sqrt{\mu/3}$ and values greater than $2\widehat{e}$ will have a derivative value near to zero. Nevertheless, there is not a gradient-based algorithm capable to reach the minimum, if the initial value gives an error greater than $2\widehat{e}$. For this reason, a minimization method in two steps for the parametric SSD–ARC error function is proposed.

3 Algorithm GA–SSD–ARC–NLM

We propose to begin with a stochastic-based search, as Genetic Algorithm, and then to refine the results using a gradient-based algorithm, the Newton Levenberg–Marquardt (NLM). The following names are used in order to distinguish between the ways to minimize the parametric SSD–ARC Equation. If only the NLM algorithm is used, the minimization process is named SSD–ARC–NLM (see Algorithm 2.); if we use GA, the minimization procedure will be named GA–SSD–ARC (see Algorithm 1.), and when we use a combination, performing first GA and then NLM, the process will be called GA–SSD–ARC–NLM. A similar convention is used with LE.

3.1 GA–SSD–ARC

Haupt and Haupt in [13], describe the steps to minimize a continuous parameter function cost using GA. In our case, the parametric SSD–ARC given by (11) is optimized and the six vector parameter Θ will be the chromosome for each individual and the jth parameter is randomly computed by

$$\theta_j = (\theta_j^{\max} - \theta_j^{\min}) * \alpha + \theta_j^{\min} \tag{14}$$

where $\theta_j^{\max}$ and $\theta_j^{\min}$ are the upper and lower bounds and α is a random number in $[0, 1]$.

Algorithm 1. GA-SSD-ARC

Given μ, $I_1(r_i)$, $I_2(r_i)$ and N_{pop} then
1. Compute a initial population
   - For $k = 1$ to N_{pop}
     Randomly compute $\Theta^{(k)}$ using Equation (14),
     For each $\Theta^{(k)}$ compute $I_1(\widehat{r}_i(\Theta^{(k)}))$ and $e_i(\Theta^{(k)})$, by Equation(3)
     Compute the error $E_{SSD-ARC}(\Theta^{(k)})$ using Equation (11)
   - Select the half population with least $E_{SSD-ARC}(\Theta^{(k)})$; $N_{pop} \leftarrow N_{pop}/2.0$
2. Matting.
   - Randomly select couples over the best population half given more matting probability those elements with least $E_{SSD-ARC}(\Theta^{(k)})$
3. Reproduce
   - Replace the worst population half by the offsprings created by Equation (15)
   - and then, compute the Error $E_{SSD-ARC}(\Theta^{(off)})$ (Equation (11)) for each offspring.
4. Mutation
   - Randomly select the $k-th$ population member and the $j-th$ parameter, and replace it by a new parameter computed by Equation (14)
   - Never mutate the best population member.
5. Repeat steps 2, 3 and 4 until the population do not reach convergence.
6. For the best population member $\Theta^{(0)}$, compute $e_i(\Theta^{(0)})$ using Equation (3), and then, the outliers field l_i by Equation (10)

At each generation, a fitness-based selection process indicates which individuals from the population will mate and reproduce, yielding new offsprings. Once we have selected two individuals $\Theta^{(f)}$, and $\Theta^{(m)}$ for mating, cross-over is accomplished according to the following formulae

$$\begin{aligned} \theta_j^{(k)} &= \theta_j^{(m)} - \beta(\theta_j^{(m)} - \theta_j^{(f)}) \\ \theta_j^{(k+1)} &= \theta_j^{(f)} + \beta(\theta_j^{(m)} - \theta_j^{(f)}) \end{aligned} \tag{15}$$

where β is a random number between zero and one, and $\theta_j^{(k)}$ denotes the $j-th$ parameter of the vector parameter $\Theta^{(k)}$. Newly born offsprings ($\Theta^{(k)}$ and $\Theta^{(k+1)}$) are incorporated to the population replacing the worst elements and their fitness is computed by Equation (11). The final GA-SSD-ARC is presented in the Algorithm 1.

3.2 SSD–ARC–NLM

Equation (16) gives the iterative steps to find the minimum value using the Newton Levenberg–Marquardt NLM [9], and the strategy for computing $\lambda^{(k)}$ is given by Algorithm 2.

$$\Theta^{(k+1)} = \Theta^{(k)} - \left[H\left(\Theta^{(k)}\right) + \lambda^{(k)} I\right]^{-1} \nabla E\left(\Theta^{(k)}\right) \tag{16}$$

Algorithm 2. SSD–ARC–NLM

Given the μ, $I_1(r_i)$, $I_2(r_i)$, $\lambda^{(0)}$ and $\Theta^{(0)}$ then:
1. Set $k = 0$ and compute $E(\Theta^{(0)})$ by Equation (11)
2. Do
   Compute $H(\Theta^{(k)})$ and $\nabla(\Theta^{(k)})$ by Equations (17) and (18)
   Do
      Update $H(\Theta^{(k)}) \leftarrow H(\Theta^{(k)}) + \lambda^{(k)} I$
      Compute $\Theta^{(k+1)}$ by Equation (16) and $E_{SSD-ARC}(\Theta^{(k+1)})$ by Equation (11),
      If $(E_{SSD-ARC}(\Theta^{(k+1)} > E_{SSD-ARC}(\Theta^{(k)})$ $\lambda^{(k+1)} = 10\lambda^{(k)}$ else $\lambda^{(k+1)} = \lambda^{(k)}$
   While $((E_{SSD-ARC}(\Theta^{(k+1)}) > E_{SSD-ARC}(\Theta^{(k)}))$ and $(\lambda^{(k+1)} < \lambda_{max})$
   If $(\lambda^{(k)} > \lambda_{min})$ then $\lambda^{(k)} \leftarrow \lambda^{(k)}/10$
   If $(E_{SSD-ARC}(\Theta^{(k+1)} > E_{SSD-ARC}(\Theta^{(k)}))$ then $\Theta^{(k+1)} = \Theta^{(k)}$
   Set $k \leftarrow k + 1$
  While $((E_{SSD-ARC}(\Theta^{(k)}) < E_{SSD-ARC}(\Theta^{(k-1)}))$
3. For the final Θ^*, compute $e_i(\Theta^*)$ using Equation (3), and then l_i by Equation (10)

where $\nabla E(\Theta)$ is the gradient vector and $H(\Theta)$ is the Hessian matriz at each iteration and they are computed by Equations (17) and (18)

$$\nabla E(\Theta) = 2 \sum_{i=0}^{N-1} J(\widehat{r}_i(\Theta)) \frac{\mu e_i^2}{\mu + e_i^2} \tag{17}$$

$$H(\Theta) = 2 \sum_{i=0}^{N-1} J^T(\widehat{r}_i(\Theta)) J(\widehat{r}_i(\Theta)) \frac{2\mu^2 \left(\mu - 3e_i^2\right)}{\left(\mu + e_i^2\right)^3} \tag{18}$$

$$J(\widehat{r}_i(\Theta)) = \left[A^{-T} \nabla I_n(r_i)\right]^T M(r_i)$$

where $J(\widehat{r}_i(\Theta))$ is the Jacobian matrix, $M(r_i)$ is defined by Equation(1), $I_n(r_i) = I_1(\widehat{r}_i(\Theta))$ and $A = \begin{pmatrix} \theta_0 & \theta_1 \\ \theta_3 & \theta_4 \end{pmatrix}$.

4 Experiments

We compare GA–SSD–ARC–NLM, LE and RANSAC methods using pairs of synthetic and real images. For LE, a minimization scheme similar to GA–SSD–ARC–NLM (using GA and NLM) is used in order to give similar minimization conditions (only replacing $E_{SSD-ARC}$ by E_{LE} in Algorithms 1. and 2.). Since in experiments with pairs of synthetic images, we know the right parameter vector $\widehat{\Theta}$, so the Euclidean distance $|\Delta\Theta|$ between the known parameter vector $\widehat{\Theta}$ and the estimated parameter vector Θ, is used as a proximity measure.

4.1 Experiments with Synthetic Images

In synthetic experiments, the NLM uses the vector $\Theta = [1, 0, 0, 0, 1, 0]^T$ as the initial value, and the stop criterion was $1e - 5$ or 1000 iterations. For GA, a

population of 3000 individuals and 100 generations were used. In order to accelerate the convergence procedure, in some cases the error function was evaluated only on 20% of the image pixels, all these parameters were handpicked in order to have a good performance. The GA search boundaries, for each of the affine transformation parameters, are $\{0.5, 1.5\}$, $\{-0.5, 0.5\}$, $\{-10, 10\}$, $\{-0.5, 0.5\}$, $\{0.5, 1.5\}$, $\{-10, 10\}$. The parameters μ for parametric SSD–ARC and σ for LE were 20 and 25 respectively, in order to give the better performance for both algorithms, and they are the same in all experiments.

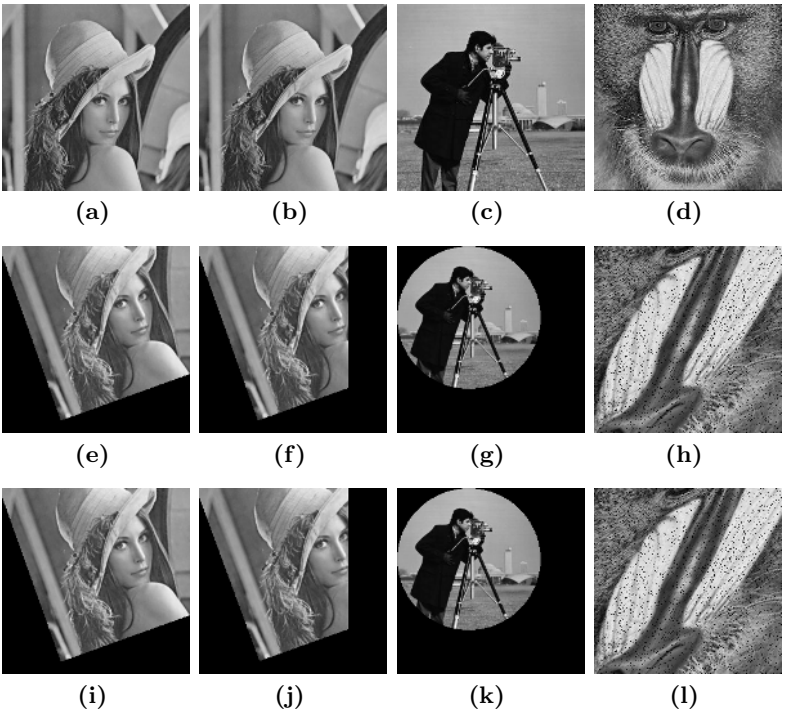


Fig. 1. Experiments with synthetic images. Origin images (a, b, c and d), target images (e, f, g and h), and resulting images using GN–SSD–ARC–NLM (i,j,k and l)

In the first experiment, an affine transformation given by $\widehat{\Theta} = [0.9396, -0.3420, 3.0000, 0.3420, 0.9396, 3.0000]$ is applied to the Lena image (Figure 1(a)) and the target images is shown in Figure 1(e). In the second experiment, in contrast with the previous one, 20% of the image pixels were set to black color (zero value) in order to simulate a regular outlier field. Using the same AT as the experiment one, the target image is shown in Figure 1(f). In the third experiment the complement of a circular-shaped outlier field and an affine transformation, given by $\widehat{\Theta} = [1.3, 0, 0, 0, 1.3, 0]$ are applied to the Cameramen Image (Figure 1(c)), yielding the picture in Figure 1(g). In the fourth experiment we use the

Baboon image (Figure 1(d)), an affine transformation given by $\widehat{\Theta}$ =[0.7, 0.3, 0, 0.3, 0.7, 0] and a random outlier field are applied to the baboon image, the target image is shown in Figure 1(h).

The results for LS, SSD–ARC–NLM, GA-SSD-ARC, GA–SSD–ARC–NLM, LE–NLM, GA–LE, GA–LE–NLM and RANSAC are presented in Table 1. These results show that GA–SSD–ARC–NLM outperforms the other methods, specially in the cameraman and Baboon images. The final images computed by GA–SSD–ARC–NLM in the four experiments are presented in Figures 1(i), 1(j), 1(k) and 1(l). Note the transformed origin images are very close to target images. You can note the bad performance for the same parametric SSD–ARC function when this is minimized using only NLM.

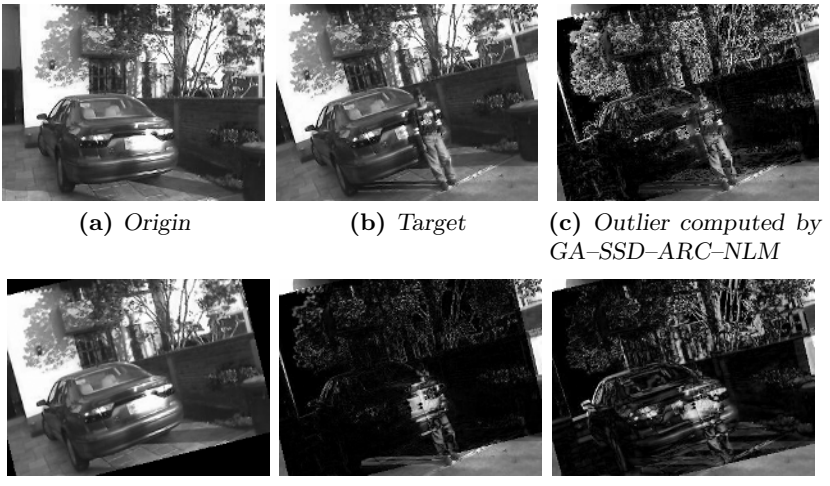
Table 1. Comparative results for parametric SSD-ARC, LE and RANSAC for synthetic experiments

	Lena	**Lena**	**Cameraman**	**Babbon**
Algorithm	$\|\boldsymbol{\Delta\Theta}\|$	$\|\boldsymbol{\Delta\Theta}\|$	$\|\boldsymbol{\Delta\Theta}\|$	$\|\boldsymbol{\Delta\Theta}\|$
LS	0.00010	51.4019	10.0451	1.3056
SSD-ARC-NLM	7.04200	4.2710	8.1507	12.5695
GA-SSD-ARC	0.96880	2.8368	1.3841	1.0910
GA-SSD-ARC-NLM	**0.00010**	**0.0431**	**0.0002**	**0.0000**
LE-NLM	19.60010	262.6573	19.6643	0.0006
GA–LE	0.42630	0.5471	6.7396	0.1371
GA-LE-NLM	0.00014	0.2089	6.3691	0.0006
RANSAC	0.12590	0.5376	0.4082	94.0000

4.2 Experiment with Real Images

This experiment use the origin and target images shown in Figure 2. These images show some differences that can be modelled using an affine transformation. Additionally, the target image has a boy in front of the car (which does not appear in the origin image), in order to introduce more complexity in the outliers, and the camera was rotated. The goal is to obtain the boy image as part of the outlier field and the affine transformation introduced by the camera rotation.

The transformation computed by GA–SSD–ARC–NLM was Θ = [0.9700, -0.2280, 42.7323, 0.2413, 0.9768, -20.6006] and by RANSAC was Θ = [0.9166, -0.2473, 47.1152, 0.2151, 0.9249, -15.3065]. In this case there are differences between the parameter vectors computed by both algorithms but these values do not allow us to conclude which one is the best or produces the nearest image to the target image. Using Equation (10), the outlier field can be computed and its image is presented in Figure 2(c), note in this image the contour of the boy in front of the car. Figure 2(d) shows the resulting image computed by GA–SSD–ARC–NLM. Finally the Figures 2(e) and 2(f) give a clear idea of the accuracy of both AF computed. In both Figures the absolute value of the difference between the target image and the computed image by GA–SSD–ARC–NLM and RANSAC were computed; dark areas correspond to low difference. Note the



(a) *Origin* **(b)** *Target* **(c)** *Outlier computed by GA–SSD–ARC–NLM*

(d) *Final Image computing by GA–SSD–ARC–NLM* **(e)** *Difference between GA–SSD–ARC–NLM and target images* **(f)** *Difference between RANSAC and target images*

Fig. 2. Car park image registration

quality for the AT computed by GA–SSD–ARC–NLM, shown in Figure 2(e), most static objects like the car or walls are almost perfectly matched; only the boy, and the leaves of trees do not match. In the RANSAC difference image (Figure 2(f)) even the car is not fully matched.

5 Conclusions

In this paper, we presented GA–SSD–ARC–NLM, an algorithm for parametric image registration, based on the non–parametric SSD–ARC algorithm. The Objective function is minimized in two steps, using GA at the beginning and then the NLM to refine the solution found by GA. The final algorithm improved the solution using only GA or using only NLM and it is robust when the images are corrupted by noise. A comparison of GA–SSD–ARC–NLM with other image registration algorithms, such as RANSAC and LE, was presented in order to provide experimental proof of the robustness of GA–SSD–ARC–NLM. We tested GA–SSD–ARC–NLM using different kinds of images and outlier fields. In all these tests, GA–SSD–ARC–NLM improved the results of the RANSAC and LE methods. Our method is similar to GA–LE–NLM but it is less sensitive to the particular parameter σ, and it is easier to find better solutions. Additionally the GA–SSD–ARC–NLM provides an explicit way to compute the outliers and does not need an extra image processing.

With synthetic images, we tested the robustness of GA–SSD–ARC–NLM and presented how the two minimization steps improved the solution using only NLM for the parametric SSD–ARC function. In case of real images, the comparison

was done using only the final parameter vector, computed by GA–SSD–ARC–NLM and RANSAC.

Furthermore, GA–SSD–ARC-NLM has the advantage of computing the outliers with accuracy even in case of a random outlier field (as shown in the experiments). In contrast with RANSAC, GA–SSD–ARC–NLM computes the outliers for the whole image using a simple Equation. This outlier Equation is implicit on the parametric SSD–ARC function and the outlier field is computed when the algorithm converges.

References

1. Zitova, B., Flusser, J.: Image registration methods: A survey. Image and vision computing **21** (2003) 977–1000
2. Szeliski, R., Coughlan, J.: Spline-based image registration. Technical Report 94/1, Harvard University, Department of Physics, Cambridge, Ma 02138 (1994)
3. Szeliski, R.: Video mosaics for virtual environments. In: IEEE Computer Graphics and Applications. Volume 16,2., IEEE (1996) 22–30
4. Vemuri, B.C., Shuangying, Huang, Sahni, S., Leonard, C.M., Mohr, C., Gilmore, R., Fitzsimmons, J.: An efficient motion estimator with application to medical image registration. Medical Image Analysis **2** (1998) 79–98
5. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. International Journal of Computer Vision **19** (1996) 57–92
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, University of Oxford, UK (2003)
7. Calderon, F., Marroquin, J.L.: Un nuevo algoritmo para el calculo de flujo optico y su aplicacion al registro de imagenes. Computacion y Sistemas **6** (2003) 213–226
8. Calderon, F., Romero, L.: Non-parametric image registration as a way to obtain an accurate camera calibration. In Spriger, ed.: Advances in Artificial Intelligence, Mexico City, MICAI (2004) 584–591
9. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equiations. Society for Industrial and Applied Mathematics. siam, Philadelphia (1996)
10. Hearn, D., Baker, M.P.: Computer Graphics C Version. second edn. Prentice Hall, Inc., Simon and Schuster / A Viacom Upper Saddle River, New Jersey 07458 (1997)
11. Watt, A.: 3D Computer Graphics. 3rd edn. Pearson Education Limited (2000)
12. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust Statistics: The Approach Based on Influence Functions. John Wiley and Sons, New York, N.Y (1986)
13. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms. John Wiley and Sons, New York, N.Y (1998)