APPLYING HYBRID GENETIC ALGORITHMS TO RECOVER FOCAL LENGTH, CAMERA MOTION AND 3D MODELS FROM SILHOUETTES

[1]AMINE MOUAfi, [1]RACHID BENSLIMANE AND [2]AZIZA EL OUAAZIZI

[1]Laboratoire de Transmission et Traitement d'Information, École Supérieure de Technologie, Université Sidi Mohamed Ben Abdellah, Route d'Imouzzer B.P. 2427, Fez, Morocco.

[2] LIMAO, Faculté Poly-disciplinaire de Taza, Université Sidi Mohamed Ben Abdellah, Route d'Oujda, B.P. 1223, Taza, Morocco.

E-mail: [1]Amine.mouafi@usmba.ac.ma, [1]r.benslimane1@gmail.com, [2]az_elouaazizi@yahoo.fr

ABSTRACT

In this paper, a hybrid method is applied to recover parameters and motion of camera from a set of silhouettes of an object taken under circular motion. Camera parameters can be obtained by maximizing the total coherence between all silhouettes. Two optimization methods, the Powell optimizer (PO) and the Genetic algorithms (GA), are applied to maximize the silhouette coherence and their performances are compared for several experiments. To take advantage of the strengths of the two methods, we developed a hybrid method that combines the genetic algorithm and the Powell optimizer to improve the performances in term of convergence speed and accuracy. The recovered parameters are used for 3D image-based modeling to obtain high fidelity 3D reconstruction.

Keywords: *Hybrid Genetic Algorithms, Powell Optimizer, Silhouette Coherence, Parameters Estimation, Circular Motion*

1. INTRODUCTION

Acquiring 3D information from images has always been a hot research topic in 3D computer vision and recently, it has attracted more and more interest because of its potential applications such as computer games, augmented reality and cultural heritage preservation. In 3D computer vision, it is necessary to know the relationship between the 3D object coordinates and the image coordinates. This transformation is determined in the camera calibration step by recovering the camera intrinsic parameters and the relative pose of the camera.

Recovering camera parameters and motion from image sequences without using any calibration patterns can be classified into two approaches: the feature-based and silhouette-based approaches. In the feature-based approach, structure from motion algorithm [1] determines the camera parameters and the 3D structure of the object simultaneously from the feature correspondences [2], [3]. These methods would therefore be not applicable to smoothed objects with low texture. In addition to feature correspondences, silhouettes also offer important clues for determining both motion and shape. It is especially the case when the object being viewed is composed of non textured smooth surfaces like pottery and sculptures. For this kind of object, silhouettes are the most predominant and stable image feature.

Silhouette-based approaches generally exploit epipolar tangents [4], [5], to locate the images of the frontier points for deriving point correspondences between images. Hernandez et al. [6] considered the problem of recovering both the focal length and the camera motion under circular motion from silhouettes. They extended the idea of exploiting the epipolar tangents [5] to the concept of silhouette coherence, which measures how well a set of silhouettes corresponds to the projections of the visual hull. The author performed camera calibration by maximizing the silhouette coherence in optimization procedure.

The Powell optimizer [7] was able to quickly reach the optimal solution for silhouette coherence maximization. However, we encountered difficulties in robustness when the initial guess for parameters are far away from the optimal solution, and when the desired global maximum was hidden by many local maxima.

Genetic Algorithms (GA) [8, 9], are pseudo-stochastic search methods that derive their fundamental ideas and terminology from the

Darwinian ''Natural selection'' theory, according to which individuals that are better fit to a given environment are more likely to survive. GA are problem-independent and can process information generated at previous stages of a search process. They comprise concepts such as natural selection, quick exploration, and information collection in a design space. In contrast to most of classical optimization methods, GA require no initial guess for parameters and can avoid being trapped in local optimal solutions as shown in our previous work [10]. These characteristics make the GA powerful tools for solving optimization problems.

In this paper, two optimization methods, the Powell optimizer (PO) and a Genetic algorithm (GA), are applied to maximize the silhouette coherence and their performances are compared for several experiments. To take advantage of the strengths of the two methods, we developed a hybrid method that combines the genetic algorithms (GA) and Powell optimizer (PO) to improve the performance of the optimization procedure.

The remainder of this paper is organized as following: in Section 2 we present the circular motion parameterization. In section 3 we present the silhouette coherence measure and its practical implementation. In section 4 and 5, three optimization methods including a Powell optimizer, Genetic Algorithms, and our hybrid GA-PO, are described, applied and compared for several tests in term of convergence and accuracy. In section 6 we build 3D models with the recovered parameters.

2. CIRCULAR MOTION PARAMETERIZATION

2.1 Camera Model

We consider a pinhole camera model. The geometry of a pinhole camera model is illustrated in Fig. 1. Let *M = (x, y, z)* be a 3D point in an object frame and *m = (u, v)* the corresponding image point in the image frame. The central projection of a 3D scene point *M* onto its 2D image point *m* can be written with the following linear equation using homogeneous coordinates:

$$m \approx PM \approx K[R|T]M \quad (1)$$

Where

$$K = \begin{pmatrix} fx & 0 & u_0 \\ 0 & fy & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

The projection matrix P is a 3×4 matrix defined up to a scalar factor that captures both the extrinsic and intrinsic camera parameters. R and T representing the rotation and translation between the world coordinate system and the camera coordinate system respectively. K is the camera calibration matrix. The parameters *fy* and *fx* represent the focal lengths measured in pixel units, with the aspect ratio defined as r = *fy/fx*, *(u_0,v_0)* represents the coordinates of the principal point.

In this paper, the aspect ratio is assumed to be one (r=1), the principal point (*u_0,v_0*) is considered to be the center of the image. The only intrinsic parameter that we consider is the focal length *f*.

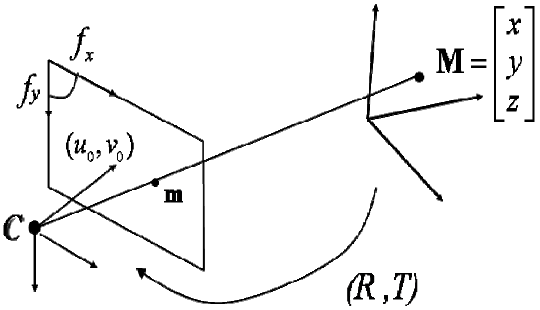

Figure 1: The geometry of a pinhole camera model

2.2 Circular Motion Parameterization

Circular motion is a practical setup for image-based modeling. A circular motion image sequence can be obtained equivalently in two ways. The most common, and the one used in our real image experiments, is the case of a static camera viewing an object rotating on a turntable. A second method is that of a camera rotating around a fixed axis and pointing at a static object. Figure 2 shows the 3D geometry of circular motion. The camera matrix P_1 of the first view can be written as:

$$P_1 = K[R_1|t_1] \ (2)$$

where K is the camera calibration matrix, R_1 and t_1 are the rotation and translation that transform the world coordinate system to the camera coordinate of the first view.

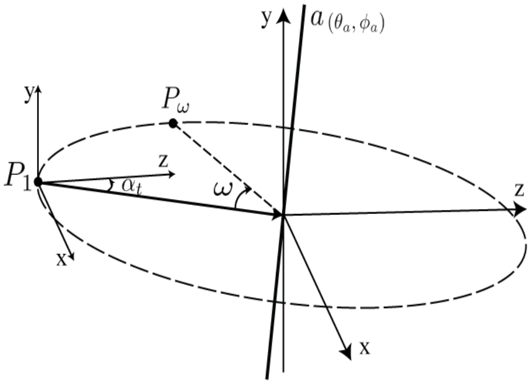

Figure 2: Circular Motion Parameterization

After rotating by ω about the axis $\boldsymbol{a}(\theta_a, \varphi_a)$, the camera matrix P_ω of the second view can be achieved by post-multiplying $[R_1|t_1]$ with $R_a(\omega)$:

$$P_\omega = K[R_1|t_1]R_a(\omega)\,(3)$$

Suppose that the circular motion image sequence consists of *n* views and the camera matrices for each view is denoted by P_i $i=1,\ldots,n$, from (2) and (3) we have

$$P_i = P_1 R_a(\omega_i)\ \ (4)$$

where ω_i denotes the rotation angle between the i^{th} and the first view, the rotation matrix $R_a(\omega_i)$ is written as a function of ω_i and the axis $\boldsymbol{a}(\theta_a, \varphi_a)$ as follow:

$$R(\omega_i)=(1-\cos\omega_i)\begin{bmatrix} a_x^2 & a_xa_y & a_xa_z \\ a_xa_y & a_y^2 & a_ya_z \\ a_xa_z & a_ya_z & a_z^2 \end{bmatrix}+\begin{bmatrix} \cos\omega_i & -a_z\sin\omega_i & a_y\sin\omega_i \\ a_z\sin\omega_i & \cos\omega_i & -a_x\sin\omega_i \\ -a_y\sin\omega_i & a_x\sin\omega_i & \cos\omega_i \end{bmatrix}$$

(5)

the rotation axis ***a*** is written in function of spherical coordinates (θ_a, φ_a):

$$a = (\sin\theta_a \cos\phi_a, \sin\theta_a \sin\phi_a, \cos\theta_a)$$

the translation is written in function of an angle α_t (the angle formed between the camera viewing direction and the z-axis see Fig. 2) as follow:

$$t_1 = (\sin\alpha_t, 0, \cos\alpha_a)$$

For *n* views, we parameterize the circular motion with *n+3* parameters: the focal length *f*, the translation direction angle α_t, the rotation axis coordinates (θ_a, φ_a), the *n-1* camera angle steps ω_i. In this paper, our goal is to recover the projection matrices P_i of a set of silhouettes S_i of an object taken under circular motion as the set of n+3 parameters:

$$v = (\theta_a, \varphi_a, \alpha_t, \omega_i, f), i = 1, \ldots, n-1$$

3. SILHOUETTE COHERENCE

Given a set of silhouettes S_i, $i = 1,\ldots,n$ of a same 3D object taken from different points of view, and the corresponding set of camera projection matrices P_i. Let *Vh* denote the reconstructed visual hull[1] using the set of silhouettes S_i, and S^v_i denote the reconstructed visual hull silhouettes. We would like to evaluate the coherence between the silhouette S_i and all the other silhouettes $S_{j\neq i}$ that contributed to the reconstructed visual hull *Vh*.

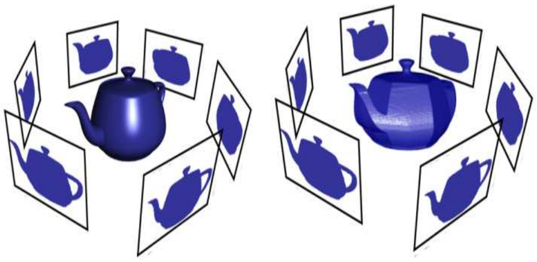

Figure 3: Visual Hull Reconstruction From A Set Of Silhouettes. Left: Silhouettes Obtained By Projecting The Original Object Back Into Cameras. Right: The Reconstructed Visual Hull Using These Silhouettes.

We assume that the silhouettes segmentation and the projection matrices are exact. We say that the silhouette S_i is coherent with all the other silhouettes $S_{j\neq i}$ if the reconstructed visual hull silhouettes S^v_i and the original silhouette S_i are exactly the same ($S_i = S^v_i$). Two examples of coherent and non-coherent silhouettes are shown in Fig.4 and Fig. 5.

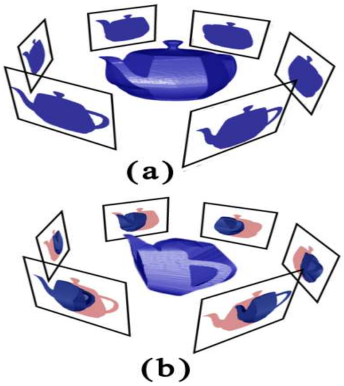

Figure 4: Two Examples Of Different Silhouette Coherence. (A) Perfect Coherent Silhouette. (B) Low Silhouette Coherence.

[1] visual hull is an outer approximation of the observed solid, constructed as the intersection of the visual cones associated with all input cameras projection matrices P_i, see Fig. 3

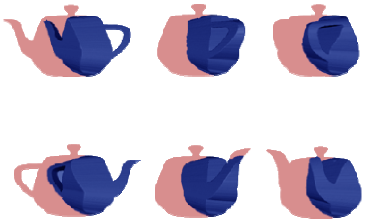

Figure 5: The Original Silhouettes S_i Superposed With The Visual Hull Silhouettes $S^v{}_i$. The Red Region Indicates Non Coherent Silhouette Pixels.

3.1 Measure Of Silhouette Coherence

To evaluate the coherence between silhouettes, some kind of similarity measurement between the original silhouette S_i and the reconstructed visual hull silhouettes $S^v{}_i$ is needed. Hernandez [6] defines this measure of coherence as the ratio of the silhouette contour lengths:

$$C(S_i, S_i^v) = \frac{\int (\partial_i^v \cap \partial_i)}{\int \partial_i} \in [0,1] \quad (6)$$

where ∂_i denote the contour of the original silhouette Si and $\partial^v{}_i$ the contour of the reconstructed visual hull silhouette $S^v{}_i$.

To compute the total coherence between all the silhouettes, we compute the mean coherence of each silhouette with the (n-1) other silhouette [6].

$$C_{SC}(S_i,\ldots,S_j) = \frac{1}{n}\sum_{i=1}^{n} C(S_i, S_i^v) \quad (7)$$

If the silhouettes segmentation and the projection matrices are exact then:

$$C_{SC}(S_i,\ldots,S_j) = 1 \quad (8)$$

3.2 Silhouette coherence implementation

The simplest implementation of silhouette coherence, would be the following: i) compute the reconstructed visual hull defined by the silhouettes, ii) project the reconstructed visual hull back into the cameras, iii) compare the reconstructed visual hull silhouettes to the original ones. The major drawback of this approach is the computation time. Since the reconstruction of the visual hall will take several minutes, and if we want accurate projected silhouettes, we need a high resolution 3D model of the visual hull, which is computationally very expensive and not appropriate for an iterative optimization process. In addition, we are not interested in 3D model in itself but in the comparison between its silhouettes with the original silhouettes. Therefore, it is a waste of time to build the visual hull completely when only some views of it are required. The imaged-based visual hull (IBVH) technique [11] does not compute a 3D representation of the reconstructed visual hull but only 2D views of it. To take advantage of epipolar geometry, 3D ray intersections required to construct the VH are reduced to 2D line intersections, which simplify the computation.

The implementation of silhouette coherence will be as following: for each 2D point in contour, compute the intersection between its corresponding optic ray and the visual hull by a ray-casting approach, this is equivalent to (see Figure 6):

1. project the optic ray into each silhouette
2. compute the 2D intersection intervals between the projected ray and each silhouette
3. back project all the 2D intervals onto the original 3D optic ray,
4. Compute the intersection on the 3D optic ray of all the intervals of all the silhouettes

For any optic ray, we have a set of remaining depth intervals, possibly empty, which represent the intersection between the optic rays and the implicit visual hull.

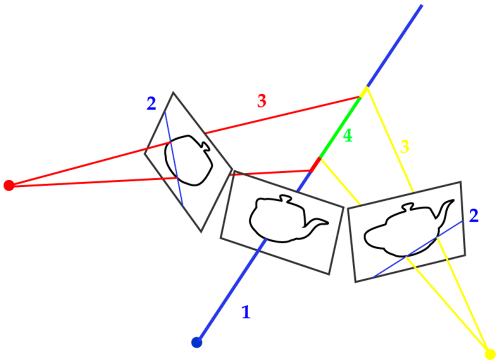

Figure 6: 2D Computation Steps, (1) Project The 3D Optic Ray, (2) Compute The 2D Intersection Intervals Between The 3D Optic Ray And Each Silhouette, (3) Back Project All 2D Intervals Onto The 3D Optic Ray, (4) The Intersection On The 3D Optic Ray (Green Segment).

4 OPTIMIZATION ROUTINES

In order to exploit silhouette coherence for recovering camera parameters and motion under circular motion, the idea is to use the silhouette coherence measure C_{SC} as the cost in an optimization procedure. In this section, three optimization methods, including a Powell optimizer, Genetic Algorithms, and a hybrid GA-PO, are applied to maximize C_{SC} (Eq. 7) and compared for several tests.

4.1 Powell Optimizer (PO)

The first optimization method used here is the Powell direction set method. The Powell optimizer applied in this paper is the version described in [12, 13] in which starting points and a set of independent search directions are provided to the program. In each iteration the method serially performs a sequence of line minimizations along the various directions in the space of parameters. At the end of each iteration the method replaces one of the original directions with the line joining the starting and ending points. A special care is taken to ensure that the directions remain linearly independent. This version of the Powell optimizer is applied to the Silhouette coherence maximization problem. Although there are many other implementations of the PO such as described in [14, 15], the current work does not intend to include a comparative study of the merits of each of these implementations.

4.2 Genetic Algorithms (GA)

When solving an optimization problem using GA, each solution is usually coded as an alphabet string of finite length called chromosome. Each string or chromosome is considered as an individual. A collection of N individuals is called population. GA start with a randomly generated population of size N, in each iteration of the algorithm, a new population of the same size is generated from the current population by applying operators, termed selection, crossover and mutation [16], that mimic the corresponding processes of natural selection. Following nature's example the probability p_m of applying the mutation operator is very low compared to the probability of applying the crossover operator p_c.

To improve the search process of the global optimum, an additional operator, elitism, was implemented. The aim of the elitist strategy is to carry the best chromosome from the previous generation into the next. We have implemented this strategy in the following way:

- step 1: Copy the best individual ind_0 of the initial population pop_0 in a separate location.
- step 2: Perform selection, crossover and mutation operations to obtain a new population pop_1.
- step 3: Compare the worst individual ind_1 in p_1 with ind_0 in terms of their fitness values. If ind_1 is found to be worse than ind_0, then replace ind_1 by ind_0.
- step 4: Find the best individual ind_2 in pop_1 and replace ind_0 by ind_2.

Note that an individual ind_1 is said to be better than another individual ind_2 if the fitness value of ind_2 is less that of ind_1, since the problem under consideration is a maximization problem.

To adapt the GA to the camera parameters estimation problem, the real-valued Coding GA is utilized. The C_{SC} of the chosen Silhouettes is taken to be the objective function. The camera parameters are encoded as a chromosome string of n+3 genes as shown in Fig. 6. The alleles of each gene are constrained to a bound of values to ensure only feasible solution are adopted for evolution.

There exists no criterion in the literature [17], which ensures the convergence of GA to an optimal solution. But usually, two stopping criteria are used in Genetic Algorithms: In the first, the process is executed for a fixed number of iterations and the best individual obtained is taken to be the optimal one. In the second, the algorithm is terminated if no improvement in the fitness value of the best individual for a fixed number of iterations, and the best chromosome is taken to be the optimal one. We have adopted the second stopping criteria with elitist strategy [18].

θ_a	ϕ_a	α_t	$\Delta\omega_1$	…	$\Delta\omega_{n-1}$	f

Fig. 6 Encoding of camera parameters as chromosome string

4.3 Hybrid approach GA-PO

Although GA can quickly locate the region in which the optimal solution exists, it takes a relatively long time to converge to the optimal solution [10]. On the other hand, the Powell optimizer is known for its fast convergence speed but the correctness of solution is very dependent on the quality of the initial guess. Therefore, we exploited in the C_{SC} maximization problem the benefit of combining the Powell Optimizer (PO) and the GA. The proposed hybrid method consists of two steps. We start the search for good initial parameter values using GA followed by the refining process using PO in order to get more accurate solution.

In our Hybrid method, the Genetic Algorithm is responsible to provide good initial values of camera parameters, while the Powell optimizer is responsible for the quality of parameters estimation. The advantage of this hybridization is not only reduces the search space greatly, but also avoids premature convergence of Genetic Algorithm to some extent.

5 EXPERIMENTAL RESULTS

In order to test the performance of our hybrid method, several parameters of the GA operators need to be determined. As recommended in [9] the crossover probability p_c was set to 0.6 and the mutation probability p_m was set to the inverse of the population size N. The determination of the population size N depends on the number of parameters to optimize. For a simple case (3 parameters), such as the estimation of focal length *f* and rotation axis coordinates *(θ_a, φ_a)*, a population with N=50 is sufficient. However, for full motion estimation (n+3 parameters) a small population size can drive the GA to converge to a local maximum. To facilitate the implementation of our algorithm, we have used the GAlib version 2.4 developed by MIT.

5.1 Comparison between PO and GA

In this experiment, we have used a synthetic Teapot sequence of 18 exact silhouettes shown in Fig. 7. We conducted systematic comparisons between the PO described in [8] and of the GA to estimate the focal length *f* and the rotation axis coordinates *(θ_a, φ_a)*. We found that in this case the PO performs better than the GA, as shown by comparison of the convergence histories in Fig. 8. Both the PO and the basic GA converge correctly to the optimal solution. However, the PO converges to the optimal solution more rapidly than the GA.

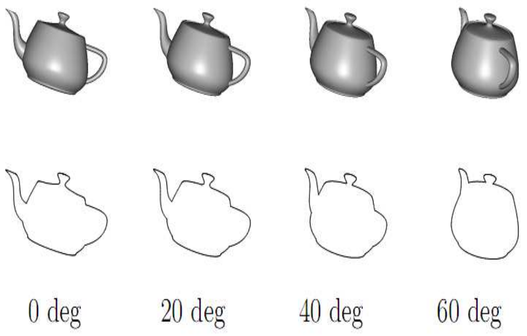

Figure 7: Some Views Of Synthetic Teapot Sequence With Their Corresponding Exact Silhouettes And Their Absolute Camera Angles

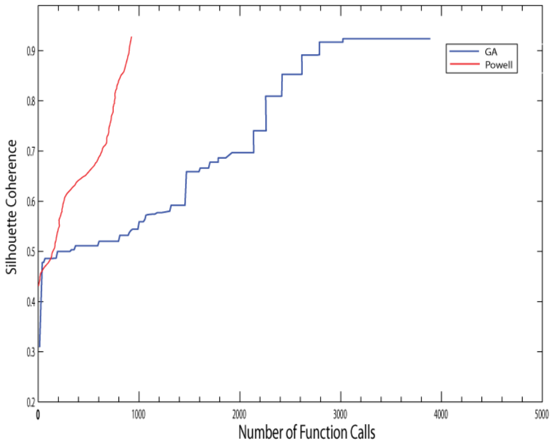

Figure 8: Comparison Of The Convergence Histories Of PO And GA For Focal Length And Rotation Axis Estimation.

In the second test we have increased the complexity of the optimization problem, we have taken 9 views spaced of 20 degrees and we have computed the full circular motion (translation direction α_t and rotation axis coordinates *(θ_a, φ_a)*, camera angles $\Delta\omega_i$) by keeping the focal length to its calibrated value. Fig 9 shows the convergence histories for the basic GA and the PO for this case. As shown in Fig 9 the GA converges correctly to the optimal solution while PO has been trapped in local optimal solution due to bad initialization of starting points. It is important to note that there are different implementations of the PO. Although the version applied in this paper fails to find the global maximum, there may be other versions of the PO that can improve the result. Though, seeking the best version of the Powell optimizer is not the intent of this paper.

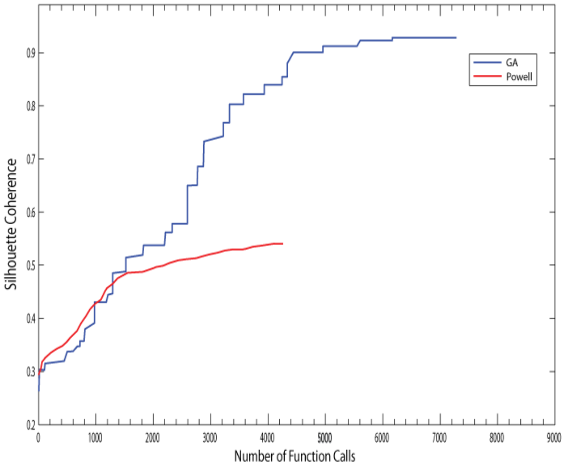

Figure 9: Comparison Of The Convergence Histories Of Powell And GA For Full Motion Estimation

5.2 Comparison between the GA and Hybrid GA-PO

To demonstrate the advantage of the hybrid method over the previous one (basic GA), we applied it to recover camera parameters and motion from real silhouettes. In this experiment, we have used the Hannover dinosaur sequence shown in Fig 10. The dinosaur sequence (36 images) is binarized by a segmentation algorithm, and then the contours are extracted from silhouettes using a GVF snake [19].

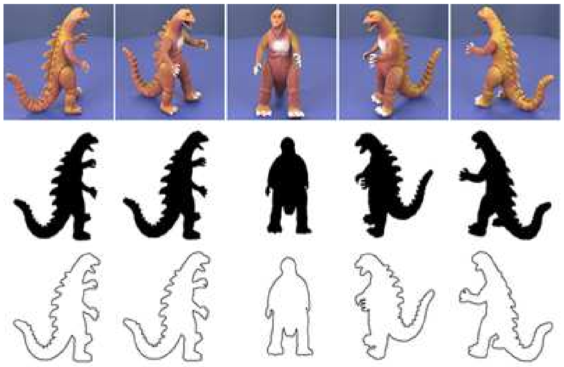

Figure 10: Same Images Of Hannover Dinosaur Sequence. From Top To Bottom: Color Images, Binarized Silhouettes And Contours Extracted From Silhouettes.

Table 1 gives the ranges of values for each parameter that we set for this experiment and the estimation of the rotation axis coordinates (θ_a, φ_a), the translation direction α_t and the focal length f by the GA and the Hybrid method. The results are good for both methods. The Hybrid method outperforms the basic GA when computing the rotation axis and the translation direction. The comparison of convergence histories between the both methods is shown in Fig. 11. It is seen that strong improvement is obtained when the PO is launched after 50 generations (1000 function calls). The hybrid method maximizes the C_{SC} faster than the basic GA.

Table 1: Camera Parameter Estimated By Ga And The Proposed Hybrid Method

Parameters	Rotation		Translation	Focal leght
	θa	***φa***	***αt***	***f***
Ground Truth	92.663	2.261	2.735	3217
Lower range	-180	-90	-5	3000
Upper range	180	90	5	3400
Recovered by GA	92.405	2.272	2.774	3232
Recovered by Hybrid	92.658	2.257	2.743	3246

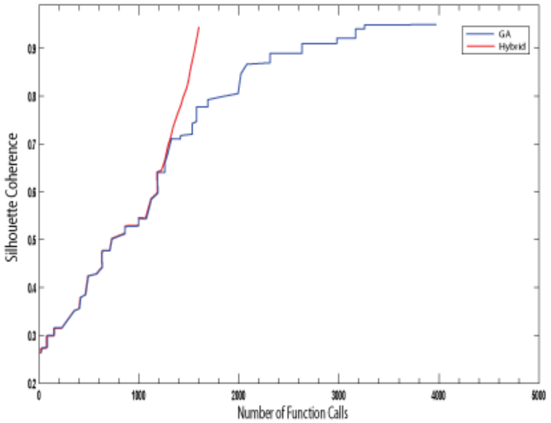

Figure 11: Comparison Of The Convergence Histories Of GA And Our Hybrid Method

6 3D MODEL RECOSTRUCTION

The 3D object surface is determined by an octree based algorithm: We dispose of a set of 36 silhouettes (dinosaur sequence in Fig. 10) and their corresponding projection matrices Pi recovered by our hybrid GA-PO method. The algorithm needs two additional input data: the level of detail (the size of the voxel), and an initial bounding box. Starting from the bounding box, the octree approach subdivides a cube into 8 children whenever it is on the isosurface, and iterates the process recursively until the maximum level of depth is attained. To evaluate a given cube, we project it into all the silhouettes to assign it one of the 3 labels [20] depending on whether it lies entirely inside (in), entirely outside (out), or partially intersects the silhouette (on). If the cube is on and the maximum depth is not still reached, we subdivide it and recursively test its children. At the end, only the cubes that are on surface have been subdivided. We can see the result of this step for different levels of resolution in Fig. 13.

Once the octree is constructed, the next step is to mesh it. Marching cubes algorithm [21] provides an initial consistent surface which is then smoothed using a decimation algorithm. Examples are shown in Fig. 14.

7 CONCLUSION

In this paper we developed a hybrid method GA-PO that combines the genetic algorithms and the Powell optimizer to maximize the silhouette coherence between a set of silhouettes in circular motion. The Comparison between the GA and the PO shows that both methods converged correctly to the optimal solution. However, the GA was slower than the PO. Important improvements ware obtained with the hybrid method in term of convergence speed and parameters accuracy. The hybrid method can correctly find the optimal parameters without the need of initial values and successfully avoid to be trapped in local maxima. These characteristics will make the silhouette coherence concept more efficient and powerful to work in general motion instead of circular motion.

ACKNOWLEDGMENT

The dinosaur images used here were provided by Wolfgang Niem at the University of Hannover.

REFRENCES:

[1] R. Hartley and A. Zisserman, Multiple View Geometry, Cambridge University Press, 2000.

[2] A. W. Fitzgibbon, G. Cross, and A. Zisserman, “Automatic 3D model construction for turn-table sequences,” in 3D SMILE, June 1998, pp. 155–170.

[3] G. Jiang, H. Tsui, L. Quan, and A. Zisserman, “Single axis geometry by fitting conics,” in ECCV, vol. 1, 2002, pp. 537–550.

[4] Wong, K.-Y. K. and Cipolla, R. (2001). Structure and motion from silhouettes. In 8th IEEE International Conference on Computer Vision, volume II, pages 217-222, Vancouver, Canada.

[5] Paulo R. S. Mendonca, Kwan-Yee K. Wong, and Roberto Cipolla. Epipolar geometry from profiles under circular motion. IEEE Trans. Pattern Anal. Mach. Intell., 23(6): 604616, 2001.

[6] Carlos Hernndez, Francis Schmitt, and Roberto Cipolla. Silhouette coherence for camera calibration under circular motion. PAMI, 29(2):343349, February 2007.

[7] M. Powell, ”An efficient method for finding the minimum of a function of several variables without calculating derivatives”, Computer Journal, vol. 17, pp. 155162, 1964.

[8] J. H. Holland, Genetic algorithms, Scientif. Am. 44 (1975).

[9] D. E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning (Addison-Wesley, Reading, MA, 1989).

[10] A. Mouafi, R. Benslimane, A. El Ouaazizi. A Genetic Algorithm for Recovering Camera Parameters and Motion from Silhouettes. Telecommunications (ICT), 20th International Conference on, 6-8 May 2013.

[11] Matusik, W., Buehler, C., Raskar, R., Gortler, S., and McMillan, L. Image-based visual hulls. SIGGRAPH 2000, pages 369{374.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes, 2nd ed. (Cambrige Univ. Press, Cambridge, UK, 1992).

[13] 18. F. S. Acton, Numerical Methods That Work (Mathematical Association of America, Washington, DC, 1970) p. 464. [1990 corrected edition].

[14] R. P.Brent, in Algorithms for Minimization without Derivatives (Prentice Hall International, Englewood Cliffs, NJ, 1973), Chap. 7.

[15] J. E. Dennis, Jr. and R. B. Schnable, Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Prentice Hall International, Englewood Cliffs, NJ, 1983)

[16]Z. Michalewiez,”Genetic Algorithms + Data Structure = Evolution Programs”, Springer, Berlin, 1992.

[17] A. El ouaazizi, M. Zaim and R. Benslimane, A Genetic Algorithm for Motion Estimation, International Journal of Computer Science and Network Security, VOL.11 No.4, April 2011

[18] A. El ouaazizi, R. Ouremchi and R. Benslimane, Reconstruction of gray-level image by genetic algorithm , Proceeding of 4th International Conference on Quality Control by Artificial Vision, Japan 1998.

[19] Xu, C. and Prince, J. L. (1998). Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing, pages 359-369.

[20] R. Szeliski, Rapid octree construction from image sequence. CVGIP, 58(1):23-32, July 1993.

[21] William E. Lorensen, Harvey E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, ACM Computer Graphics, 21(24): 163-169, July 1987.

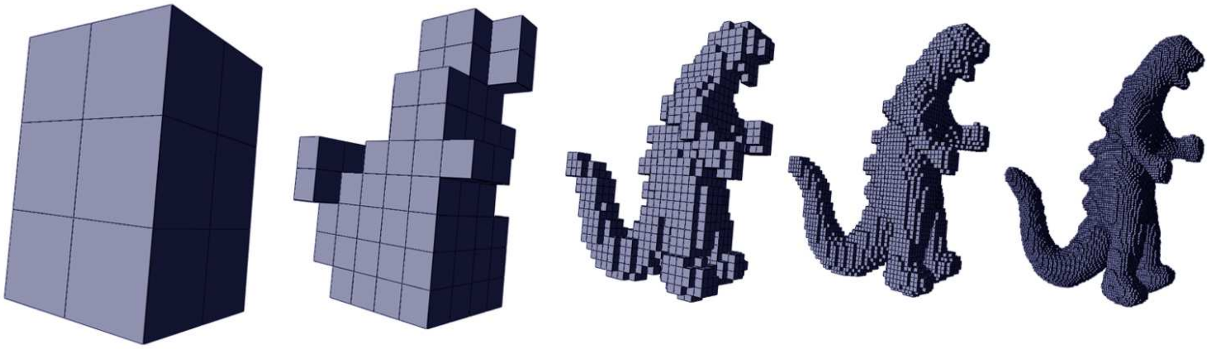

Figure 13: Octree Generation: The Dinosaur Octree Is Carved From A Single Bounding Box Given 36 Images. From Left To Right: Bounding Box, Level 3, 5, 6 And 7 Of Subdivision.

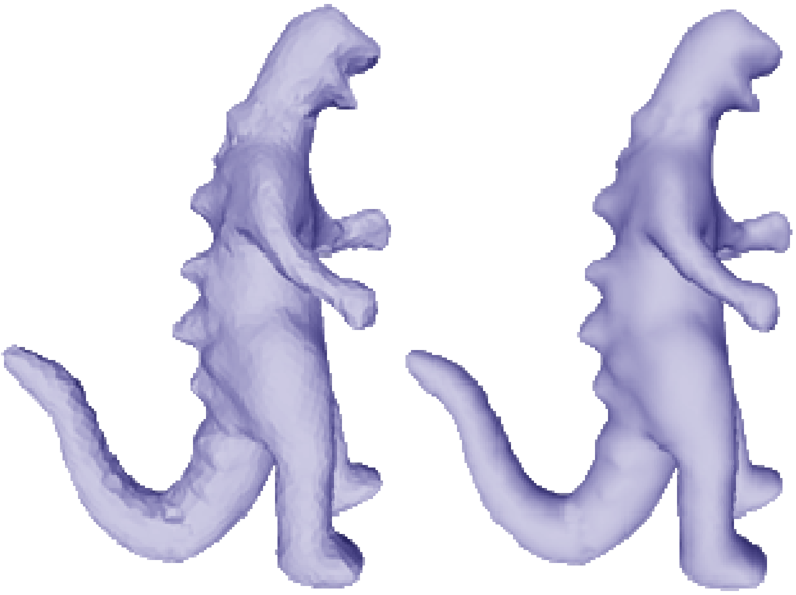

Figure 14: Dinosaur Model: Marching Cubes Meshing And Decimation Steps.