Quantitative Evaluation Method for Model-based Tracking of 3D Rigid Curved Objects

Marina Atsumi Oikawa*
Nara Institute of Science and Technology

Goshiro Yamamoto†
Nara Institute of Science and Technology

Makoto Fujisawa‡
Tsukuba University

Toshiyuki Amano§
Yamagata University

Jun Miyazaki¶
Nara Institute of Science and Technology

Hirokazu Kato‖
Nara Institute of Science and Technology

Abstract

In this paper we describe a methodology developed for quantitative evaluation of a model-based tracking framework targeting 3D rigid curved objects. In particular, this framework considers the use of sparse polygonal meshes for tracking curved objects in order to solve the trade-off between the computational time and tracking accuracy. A simulator was designed considering how to compare objective results from the standard edge based tracking with the new proposed framework. Experimental results using synthetic data are showed.

Keywords: model-based tracking, rigid curved objects, quantitative evaluation

Index Terms: G.1.2 [Numerical Analysis]: Approximation—Approximation of surfaces and contours; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking.

1 Introduction

In computer vision, tracking the 3D pose of a known object is a common task, being important in applications such as augmented reality, robotic manipulation, gesture recognition, among others. For rigid objects, this means to continuously recover 6 DOF parameters representing the object position and orientation relative to the camera while it moves around the scene [4].

For model-based tracking, a common approach represents the target object using a polygonal mesh, whose outlines are matched with detected edges in the video image. This matching is achieved by looking for strong gradients in the video image using an initial estimation of the pose and performing edge normal search of projected edges. The final pose is then obtained after an optimization process [1], [2]. However, the approaches mentioned above are applied mainly to polyhedral objects with flat faces.

When dealing with smooth objects, dense meshes are required to accurately recover the object shape, creating a trade-off between the computational time and the tracking accuracy as exemplified in Fig.1.a: a sparse mesh (light green lines) is overlaid on the target object and the red lines represent the edges from the patches on the contour from where normal search will be performed to find nearby edges. If the edges from the mesh are used directly, it is possible to notice a large distance d between the sample points on the model and the detected edge points.

*e-mail: marina-o@is.naist.jp
†e-mail: goshiro@is.naist.jp
‡e-mail: fujis@slis.tsukuba.ac.jp
§e-mail: amano@yz.yamagata-u.ac.jp
¶e-mail: miyazaki@is.naist.jp
‖e-mail: kato@is.naist.jp

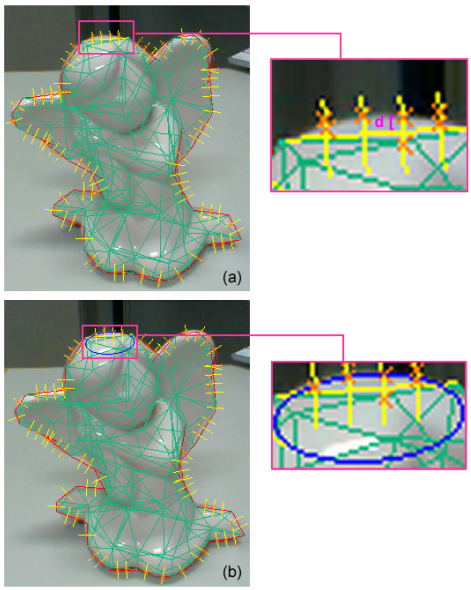

Figure 1: A sparse polygonal mesh is overlaid on the target object and a comparison between the distance evaluation of (a) the standard tracking and (b) the tracking framework to be evaluated using conics is showed.

The tracking framework evaluated in this paper tries to solve this problem by using *quadric equations*, calculated for each patch in the mesh, to give local approximations of the object shape. Hence, instead of using the original edges from the mesh for matching with detected edges in the video image, curves representing the quadrics projection in the current viewpoint are used for distance evaluation as can be seen in Fig.1.b. This will be referenced in the remaining parts of this paper as Conics Tracking (CT).

For quantitative evaluation of tracking algorithms, numerical results that measure and quantify them are necessary. Usually three main conditions are evaluated: robustness, computational time and accuracy. In this paper we present the simulator we designed to allow a fair comparison of these conditions, considering a model-based tracking scenario having polygonal meshes with different levels of quality. The main goal is to obtain an objective comparison between the CT approach and a standard edge based tracking using both sparse and dense meshes.

Different parameters can be controlled by the user in this simulator, such as its initial pose, number of attempts, the object movement and also the noise to be added in order to disturb the system and create a more realistic scenario. The next section describes in more details the simulator and experimental results using two objects in four different poses are given in section 3 followed by the conclusions in section 4.

2 SIMULATOR

2.1 Overview

This simulator takes as input three models of the target object in different levels of quality: a very dense mesh, a dense mesh and a sparse mesh. While the actual evaluation is done between the tracking using sparse mesh and the tracking using dense mesh, the very dense mesh is used only to create a synthetic representation of the real object.

Figure 2.a shows the very dense mesh being used to create a realistic outline of the target object and in Figure 2.b and Figure 2.c, the dense mesh and the sparse mesh, respectively, are represented by the white lines and overlaid on the target object area.

The simulation consists of a series of different trials, each one representing the object in a different initial position, manually initialized. Each trial has a number n of runs specified by the user and in the beginning of each run, the test model moves back to its initial position. Then, a new pose is produced by the simulator using Gaussian noise, whose values are the same for all approaches being compared.

Considering the pose parameters vector is represented using the notation $s = (r_x, r_y, r_z, t_x, t_y, t_z)^T$, the noise values are calculated for each of the rotation (r_x, r_y, r_z) and translation (t_x, t_y, t_z) components, according to a standard deviation value also specified by the user. The bigger these values are, the further the test model moves. In Figure 2, this new pose is represented by the blue lines and the camera displacement between the initial pose and this new pose will give the values for quantitative comparison between the approaches.

Another parameter that can be also controlled by the user is the presence of noise in the edge detection process. This noise aims to create a simulation closer to a real scenario, which can be disturbed by edge points which does not necessarily belongs to the object being tracked.

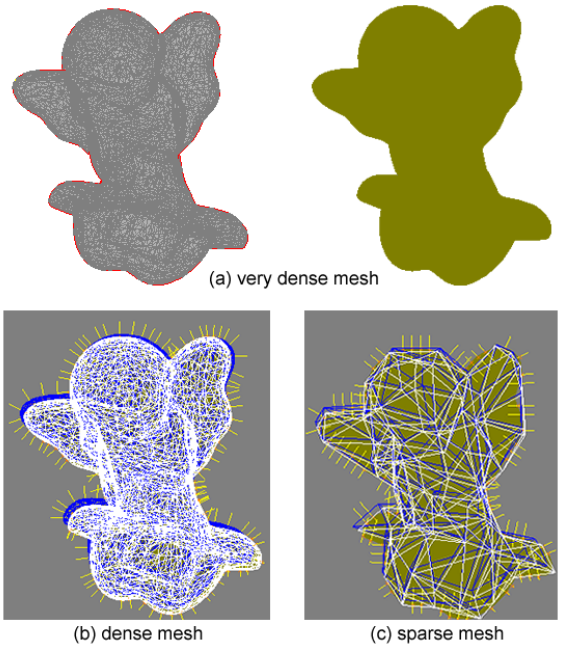

Figure 2: The (a) very dense mesh creates a synthetic representation of the target object and its outline. Then, (b) dense mesh and (c) sparse mesh can be evaluated and their values can be compared.

2.2 Evaluation

As mentioned earlier, three conditions are evaluated in this simulation: robustness, computational time and accuracy.

Robustness is defined as the success rate of the tracking within the n runs. Considering the error values are stored in a vector d_e, during the optimization step, the following cost equation is minimized:

$$e_c(s) = \sum_{i=1}^{n} d_e(X_i, \varphi(c_c(c_w(s)))) \quad (1)$$

The terms in the equation above represent:

- c_w: contour parameters in world coordinates.
- $(c_c(c_w(s)))$: contour parameters in camera coordinates parameterized by the pose parameter vector s.
- $\varphi(c_c(c_w(s)))$: projection of the contour in camera coordinates to the 2D image plane.
- X_i: detected points in image coordinates.
- d_e: vector containing the distance between the contour edges and the detected points in image coordinates.

An error threshold is set in order to define the success rate: if the error at the end of the optimization step is smaller than this threshold, the tracking succeeds; otherwise, the pose is computed again until the error falls below this threshold or a maximum number of iterations is reached. If the error value is still above an acceptable value, it is considered a failure.

Computational time represents the average processing time and *accuracy* is evaluated by analyzing the mean squared error (MSE) and the standard deviation of the estimated error values for the angle and the distance components of the pose parameter vector.

3 EXPERIMENT RESULTS

The experiments considered two different objects with no texture information available in four different poses, showed in Figure 3. Each pose was evaluated in n=1000 trials and the approximate number of patches used for each of the objects can be seen in Table 1.

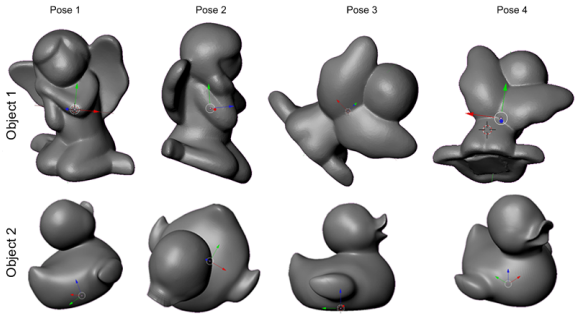

Figure 3: Object poses

Table 1: Approximate number of patches used in the experiments

	Object 1	Object 2
Very dense mesh	10000 patches	5000 patches
Dense mesh	5000 patches	1000 patches
Sparse mesh	500 patches	100 patches

Object 1	Pose 1		Pose 2		Pose 3		Pose 4	
	Success	Comp. Time	Success	Comp. Time	Success	Comp. Time	Success	Comp. Time
CT	1000	0.088831 sec	1000	0.084855 sec	999	0.106549 sec	1000	0.095312 sec
SMT	999	0.115570 sec	1000	0.097858 sec	999	0.108616 sec	1000	0.098430 sec
DMT	1000	1.431688 sec	1000	1.186602 sec	1000	1.276849 sec	999	1.232331 sec

Object 2	Pose 1		Pose 2		Pose 3		Pose 4	
	Success	Comp. Time	Success	Comp. Time	Success	Comp. Time	Success	Comp. Time
CT	1000	0.038128 sec	1000	0.029863	1000	0.053283 sec	997	0.049721 sec
SMT	1000	0.040000 sec	1000	0.034693	1000	0.047287 sec	999	0.052782 sec
DMT	999	0.116087 sec	1000	0.121856	999	0.137381 sec	1000	0.131301 sec

Figure 4: Robustness and computational time results

Object 1

Pose 1	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	0.876622	0.835693	0.0124081	0.00916952
SMT	0.908676	2.46856	0.0107216	0.0159188
DMT	0.269952	0.946225	0.00467812	0.00690348

Pose 2	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	0.403887	0.919724	0.006259	0.00745
SMT	0.664057	2.35575	0.017147	0.015899
DMT	0.255037	0.670227	0.003992	0.004242

Pose 3	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	0.335	0.654014	0.00344031	0.00472019
SMT	0.722188	1.43217	0.00577582	0.0107705
DMT	0.196981	0.703172	0.00271571	0.00345929

Pose 4	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	0.261151	0.853827	0.006798	0.012201
SMT	0.550688	1.31514	0.013219	0.018748
DMT	0.217302	0.924496	0.013712	0.012878

Object 2

Pose 1	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	1.4844	2.65818	0.0339053	0.0616411
SMT	4.32007	2.83655	0.0552055	0.0536052
DMT	1.36455	2.62508	0.038761	0.0701283

Pose 2	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	1.69288	2.14973	0.03109	0.02878
SMT	3.0853	3.46102	0.027695	0.037822
DMT	0.590922	0.89609	0.011951	0.013371

Pose 3	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	3.21394	2.46584	0.0726296	0.057316
SMT	3.60407	3.87971	0.0532213	0.0657333
DMT	3.1109	2.41052	0.0691495	0.0552911

Pose 4	MSE		Standard deviation	
	Angle	Distance	Angle	Distance
CT	1.58083	1.23309	0.043988	0.040248
SMT	2.31727	4.91812	0.033102	0.033299
DMT	0.59282	1.12427	0.031934	0.026293

Figure 5: Accuracy results

Remeshing of the models was obtained by the software QSlim provided in [3] and the sparse mesh was created such that it has approximately 10% of the total number of patches of the dense mesh.

The standard deviation for the noise in the rotation component was set to 5.0 degrees and in the translation component was set to 3.0 cm per frame, considering a frame rate of 30fps and the hypothetical situation of the user holding the object and moving it in a distance of 90.0 cm.

Results are showed comparing CT, Sparse Mesh Tracking (SMT) and Dense Mesh Tracking (DMT). The main hypothesis while evaluating these algorithms is that CT should perform better than SMT and present accuracy closer to results given by DMT, but with better performance.

Figure 4 shows the success rate and the computational time for each of the objects and the respective poses. It is possible to see the success rate was high for all methods tried, with small variations among them. On the other hand, CT showed better results for the computational time in most of the cases (which can be also visualized in the graph of Figure 6). Furthermore, in Figure 5, CT also presented better results than SMT and results very close to DMT, validating the hypothesis of the tracking framework being evaluated. A better comparison of these numerical results can be visualized in the graphs of Figure 7.

4 Conclusion

This paper has presented a methodology for quantitative evaluation of model-based tracking of 3D rigid curved objects, when comparison between the use of sparse and dense polygonal meshes are necessary. The evaluation considered poses of the object in distinctive views and with parameters which tried to approximate possible movements in the real world.

One improvement to be done on the proposed simulator includes improvement on the algorithm that generates the noise used in the simulated environment. Furthermore, if the tracking framework includes some prediction model, a method to compare results using and not using this prediction model can be necessary, which is yet to be implemented.

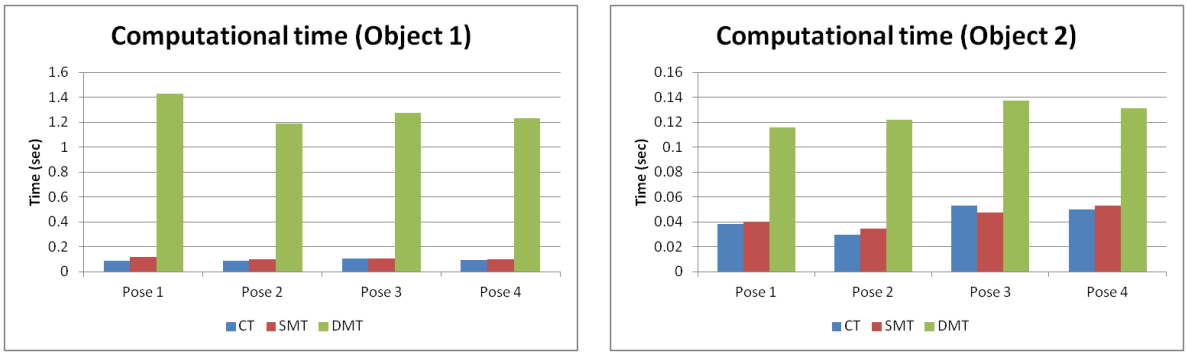

Figure 6: Graphs showing computational time results for object 1 and object 2

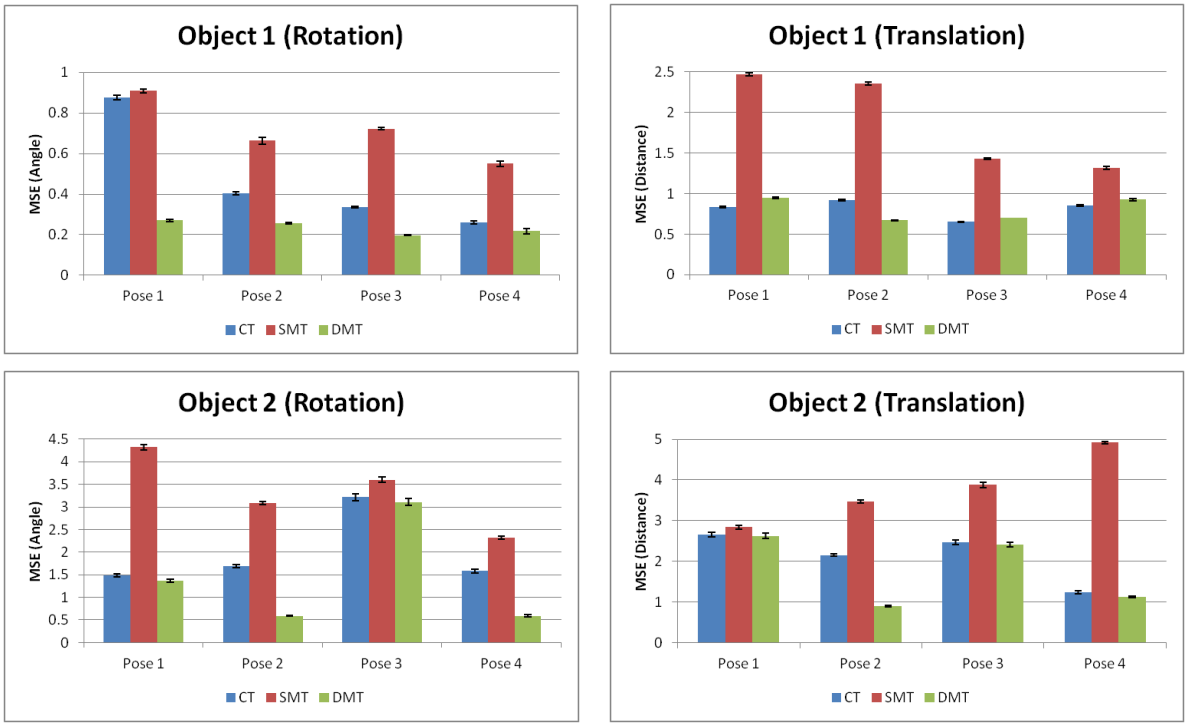

Figure 7: Graphs showing accuracy results for object 1 and object 2

REFERENCES

[1] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: The virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628, 2006.

[2] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002.

[3] M. Garland. *Quadric-Based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University, 1999.

[4] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 2005.