Image Sequence Stabilization Through Model Based Registration

Marcin Denkowski[1*], Michał Chlebiej[2]

[1]*Faculty of Computer Science, Maria Curie-Sklodowska University,
pl. M. Curie-Sklodowskiej 5, 20-031 Lublin, Poland.*
[2]*Faculty of Mathematics and Computer Science, N. Copernicus University
Chopina 12/18, 87-100 Toruł, Poland.*

Abstract – Acquisition of image series using the digital camera gives a possibility to obtain high resolution/quality animation, much better than while using the digital camcorder. However, there are several problems to deal with when producing animation using such approach. Especially, if motion involves changes in observer position and spatial orientation, the resulting animation may turn out to look choppy and unsmooth. If there is no possibility to provide some hardware based stabilization of the camera during the motion, it is necessary to develop some image processing methods to obtain smooth animation. In this work we deal with the image sequence acquired without stabilization around an object. We propose a method that enables creation of smooth animation using the registration paradigm.

1 Introduction

The creation of high resolution movies is an important task in today's high quality media. When we do not have a possibility to use High Definition camera and a motion stabilization devices such as platforms for rotating objects or camera tracks arranged around photographed object, we have to look for a different solution. One of the methods involves a typical digital camera, tripod and image processing algorithms.

---

*marcin.denkowski@umcs.lublin.pl

In this work we deal with a problem of a single object image sequence stabilization. The image set considered in this paper, consists of several (typically 30-70) photograph images taken around the object (see Fig. 1). This object was inserted in the center of a circle and in every step during the photographic session the camera was being moved along this circle path and pointed onto the center of the circle. Due to inaccuracy of camera orientation towards the photographed object and changes in the distances between consequent shoots the source animation was very choppy and unsmooth.

The idea of the method proposed in this work was to improve the quality of the animation by estimating the optimal position and orientation of the following frames. Our method consists of several stages. In the first stage we generate a simple 3D surface model of the object using 3 photograph images (selected manually) representing left, front and back sides of the object. In the second step parameters of the virtual camera (like distance, field of view and orientation) are chosen that the resulting 3D images of the car model best fit those three original images. We use the standard registration procedure but optimization algorithm searches for appropriate camera parameters instead of geometric transformation. Next, the new 360 images around the simulated object are generated for every one degree. It extends a set of original images and allows us to find the best fitting simulated frame for every image from the source set. The last step is to register selected pairs of images - original and virtual, in order to match their position and orientation. After this step we obtain a software stabilized animation.

Fig. 1. Source set of 42 photographs of different car views.

2 Methodology

The idea of reconstructing 3D model using 2D image series is, in general, a very difficult problem and has been widely discussed in literature [1–3]. In this work we consider a symmetric object which can be realistically reconstructed using only three complementary images. At the beginning we need to segment objects from the photographs and remove background. In this work images have been segmented in the preprocessing step (not described in this article).

The whole process is shown in the diagram in Fig. 2. It starts with the selection of two images representing 2D projections of the front and left sides of the object (Fig. 3 - top). These two images are thresholded to remove background and create binary masks. From these two masks three dimmensional volumes are created by filling them with N copies of each image, where N denotes the width of the perpendicular view image. After that the AND operator is applied to these two 3D volumes to obtain a 3D volume mask. On the basis of this 3D mask, a triangle mesh is created by using marching cubes [4, 5] algorithm. This mesh is filtered by the Laplacian smoothing operator [6] in order to remove common blocky artifacts produced during the contour generation process. The generated 3D surface is shown in the central part of Fig. 3. The last step of 3D model reconstruction is texture mapping. Only three - front, left and back projection images are applied as texture maps. Adding texture to the model makes it look more realistic and increases accuracy of the following stages. Some detail reconstruction errors (like outside mirrors) can be partially hidden using texture operator. Textured model is shown at the bottom in Fig. 3.
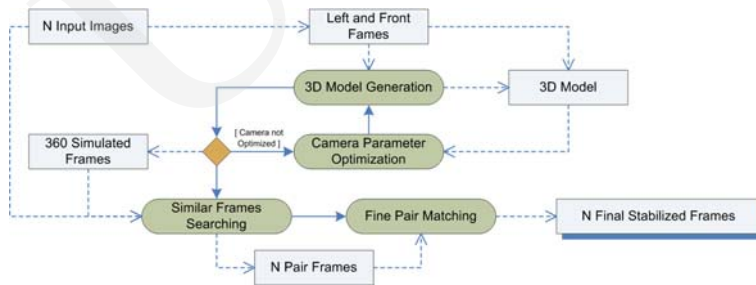

Fig. 2. Software stabilization method diagram.

The second stage of our reconstruction method involves the 3D scene arrangement. The whole stabilization process is very sensitive to the similarity of the 3D model and the original set. For this reason, we have to find the optimal settings of the virtual camera such as distance to the object, elevation, yaw and focal length to ensure that the simulated images are the most similar to the original ones. The method proposed in this work strongly depends on the registration paradigm. The idea is to minimize the difference between the simulated front and the left views by minimizing the following equation:

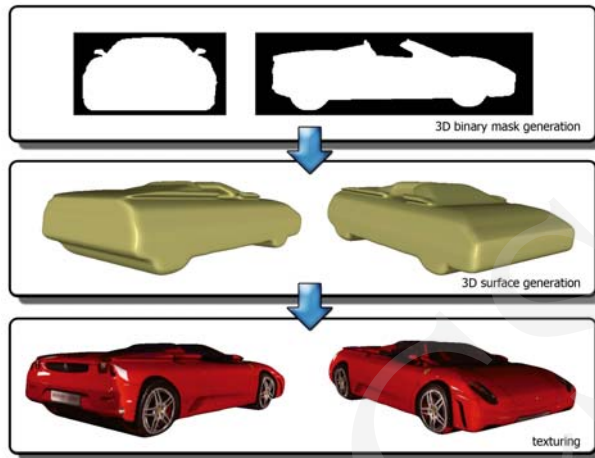$$E(T) = E_{front}(T) + E_{left}(T), \tag{1}$$

Fig. 3. Flow diagram of 3D model reconstruction algorithm.

where E_{front} and E_{left} are the similarity measures between the simulated images and original ones, and T is the 4-parameter vector of the camera settings which are being optimized.

In all registration procedures the MSD (mean square difference) similarity function has been used:

$$E_{MSD}(FI, RI, T) = \frac{1}{N} \sum \sum \left(I_{RI}(p) - I_{FI}(T(p))(i)\right)^2, \qquad (2)$$

where I_{RI} represents the reference image intensities (calculated as the weighed sum of RGB channels - $0.299R + 0.587G + 0.114B$), I_{FI} represents the corresponding transformed intensities of the floating image, p represents the single pixel and N is the total number of overlapping pixels.

For the minimization of E function we use Powell's [7, 8] optimization method. The examplepary results of camera settings optimization are presented in Fig. 4.

The third stage of stabilization method begins with generation of 360 frames of simulated smooth animation around the car. The goal of this step is to find the most similar frames for all originals out of all simulated 360 pictures. To obtain this we propose the method of searching candidates using energy map paths. At first, similarity (using Equation 2) between all possible pairs of original and simulated images is calculated. To make sense of this calculation we match the centers of corresponding objects just by matching their bounding boxes. After that a similarity function behavior map is generated to enable analysis of the best fitting candidates (see Fig. 5). It is clearly seen that the minimum values for the consequent original frames are situated along distinct valleys. The deepest valley represents the optimal path. The second visible parallel valley lies in a distance of 180 frames (degrees) and represents the local minimum path (which matches the front to the back and the left side to the right side of the car). Next, the best fitting pair selection starts with the left

Fig. 4. The results of camera calibration process through image registration.

projection and shifts along the minimum path just by comparing the function values for 10 nearest neighbours from the simulated dataset. Using this approach approximation corresponding frames are selected. The last stage of our method is the orientation stabilization. We apply matching of selected images with fine tuning of both position and rotation parameters. As a result smooth and stabilized animation is obtained.

3   Results

The proposed method has been implemented in C++ language using Trolltech QT and Kitware VTK libraries for visualization purposes. For the testing procedure we have prepared 5 high resolution image dataset containing 32 – 42 photograph images taken around an object. Images were photographed using typical DSLR camera and preprocessed manually.

An examplary frame of the original data and the corresponding frame representing the virtual 3D model are presented in the first column in Fig. 7 for all five cases. The second column in Fig. 7 presents the motion picture of the original data generated from all frames on the basis of maximum intensities. The third column shows the model motion and stabilized motion picture. It is clearly seen that the original motion pictures look more irregular and less symmetric than the stabilized ones. Similarity between the stabilized and model based motion pictures shows accuracy of the presented method.

In order to compare smoothness of a source animation and a stabilized image sequence we have decided to calculate the MSD measure (equation 2) between every two subsequent frames. We have presented variability of this similarity measure in the graphs for all five cases (see Fig. 7 - fourth column). In every graph there are four plots:

- source - representing smoothness of an original animation,
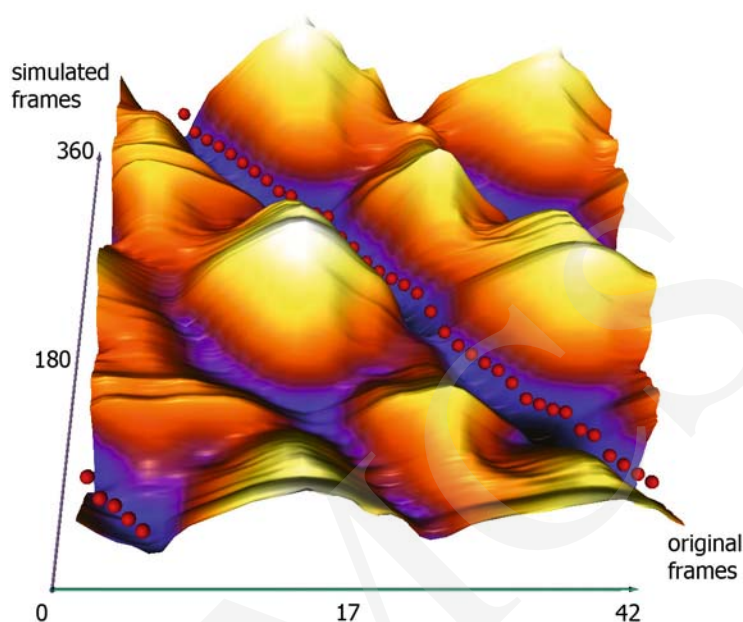- model - representing reconstructed model based "perfect" animation,

Fig. 5. The visualization of the similarity function behaviour calculated for
all pairs from both simulated and original datasets (with matched bounding
boxes). Red dots situated in the deepest valleys represent the most similar
pairs selected automatically for the final registration procedure.

- first match - representing smoothness after boundary matching of model and
  source frames selected using a frame similarity graph (see Fig. 5),
- final - representing the fine-tuned animation resulting from a rigid registration
  procedure.

Most of the frames were stabilized correctly, but in a few cases (1 frame in the first
case, 2 frames in the third case, and 3 frames in the fifth case) were slightly mismatched.
In three situations the simulated frames were selected improperly (5 degree error) and
in the case of three other frames the result of object rotation matching was incorrect
(3 degree error). Visual presentation of the examplary frames has been shown in Fig.
6 using the difference images of simulated and final, matched set of images.

It is clearly visible that for all cases smoothness increases (smaller function values)
during the stabilization procedure. In the graph representing the fifth case it is visible
that the original animation was the most stable of all cases (original, model and final
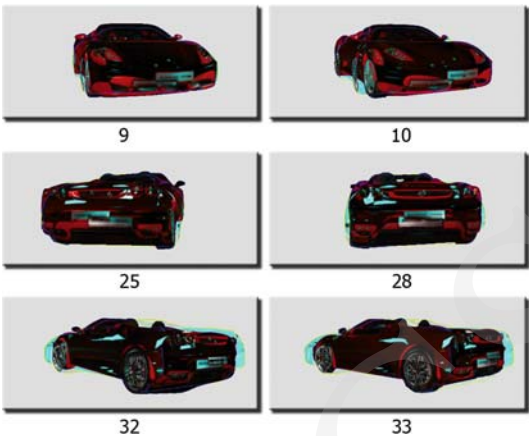plots are not very distant).

Fig. 6. Selected frames of final matching quality - difference of simulated images and aligned original photographs with the tuned parameters of position and orientation. Frames 10 and 32 are slightly mismatched.
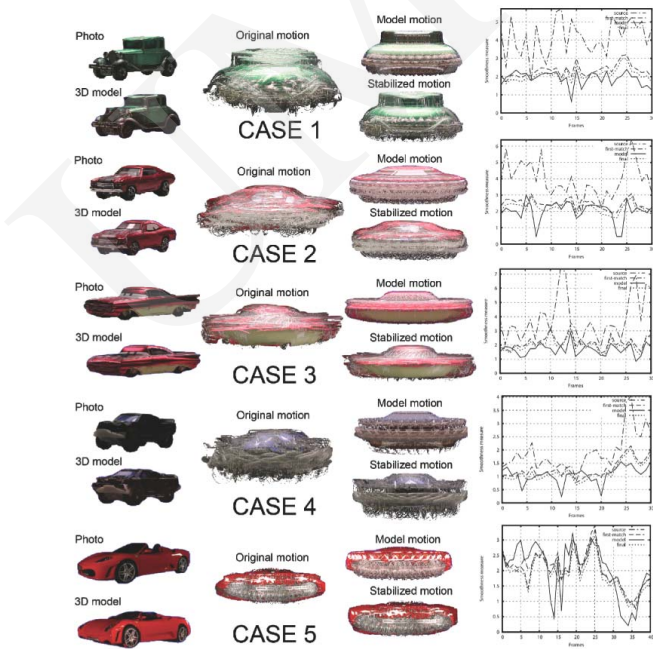

Fig. 7. Examples of the stabilization procedure for all five cases.

4   Conclusions and Prospective Directions

In this paper we have presented the method of image sequence stabilization, tests and preliminary results using five datasets consisting of high resolution and colour images.

The presented results seem to be very promising. The method seems to work well in most cases, but there are several problems that have to be resolved. The first one is the virtual camera parameters optimization. It is the most crucial part of the method determining whether the simulated frames will be properly selected or not. In the near future we would like to simulate the image deformations produced by real lenses so that we could reconstruct virtual scenec in a more realistic way. The second crucial part of our algorithm is the selection of similarity measure. During our research we have tried several functions including mutual information [9], cross correlation and contour distance [10] but the errors generated using these measures were similar to these using a simple sum of squared differences. In the near future we would like to extend evaluation of similarity measure of images by calculation of the object boundary parameters such as position, scale and rotation invariants [11, 12]. We also need to perform more tests on different image sets to value objectively the quality of our stabilization method.

References

[1] Pollefeys, M., Self-calibration and metric 3D reconstruction from uncalibrated image sequences, PhD thesis, Katholieke Universiteit Leuven (unpublished, 1999).

[2] Stamos, I., Allen, P., 3-D model construction using range and image data, Computer Vision & Pattern Recognition Conf. (CVPR) (2000): 531–536.

[3] Baillard, C., Zisserman A., A plane-sweep strategy for the 3d reconstruction of buildings from multiple images, International Archives of Photogrammetry and Remote Sensing 32(2) (2000): 56–62.

[4] Lorensen W. E., Cline H. E., Marching cubes: a high resolution 3D surface construction algorithm, Computer Graphics 21(4) (1987): 163–169.

[5] Durst, M. J., Additional reference to marching cubes, Computer Graphics 22(2) (1988): 72–73.

[6] Hansbo P., Generalized Laplacian smoothing of unstructured grids, Communications in Numerical Methods in Engineering 11 (1995): 455–464.

[7] Powell M. J. D., On efficient method for finding the minimum of a function of several variables without calculating derivatives, Comput. J. 7 (1964): 155–162.

[8] Press W. H., Flannery B. P., Teukolsky S. A., Vetterling W. T., Numerical Recipes in C (second edition) (Cambridge Univeristy Press, 1992).

[9] Shannon C. E., A mathematical theory of communication, Bell System Technical Journal 27 (1948): 379–423, 623–656.

[10] Veltkamp R., Hagedoorn M., State-of-the-art in shape matching (Technical report UU-CS-1999-27) (Utrecht University, 1999).

[11] Tuytelaars T., Local, invariant features for registration and recognition, PhD thesis, K. U. Leuven (unpublished, 2000).

[12] Vijayakumar B., Kriegman D., Ponce J., Invariant-based recognition of complex curved 3-D objects from image contours, Computer Vision and Image Understanding (1998): 287–303.