Creating a 3D model from a video by a single camera MTAT.03.260

Kristjan Krips
Timo Petmanson
June 21, 2011

1 Introduction

It has been shown by [2] that a 3D model can be constructed from a video made by a stereo camera. However, stereo cameras are yet not widely available and thus in this paper we show how a 3D model could be constructed from a video created by a single camera. As creating a 3D model with a single camera is more difficult than with a stereo camera we assume that the videos used for modelling are taken in strict conditions. We propose two different approaches for solving this problem. The results of this paper show that these methods can be used to create a 3D model. However, the results are visually not comparable to real 3D models, both of these methods have a drawback which is described in the following paragraphs. Thus, the results are a proof of concept which shows that these methods can be used for creating 3D models in certain conditions.

2 General idea

The general idea involves having a video or a stream of images of the object to be reconstructed in 3D making a full turn around a line parallel with z axis and going through the center of the object. The rotation of the object should happen ideally at a constant rate. We do not need all the images of the sequence and choose only n images such that the rotation of the object in two consequent images differs 360/n degrees. Next, we can use any two consequent images, including last and first frame, to derive data about the shape of the object. The two methods used were disparity based and difference based approaches. In the first case, we calculated disparity maps of the image pairs and constructed the resulting 3D model based on the depth of different parts of the object. Difference based method was used to extract the contour of the object and separate it from the background.
3 Getting the input video

The techniques we are using depend on detecting the object that is modelled from a photo. The photos are frames from a video taken by a monocular camera. Thus, the object has to be detected in every frame of the video. This sets strict requirements for the environment. These requirements are described in the following subsection.

3.1 Setting

1. The background has to be of a different color than the object.
2. The background should be static.
3. There should not be light reflection.
4. Light conditions should not change.
5. The object has to rotate with a constant speed.

For these reasons, we placed the object on the centre of a turntable that is rotating around its centre with a constant speed. The camera was set in a fixed position facing the rotating object while being on the same plane with the rotating object. In order to remove distortions that might be caused by the background we built a fixed color wall behind the rotating object. Due to these settings rectification of two consequent frames may not be necessary.

4 Techniques

For 3D construction the object has to be detected in order to get the coordinates for creating the 3D model. We use two consequent frames to detect the object. As the background is static and the only moving objects are the object that we are trying to detect and the plate that creates the rotation, it is possible to use change of position between two frames for object detection. The implementation is done in C++ with the libraries OpenCV [1] and OpenGL.

4.1 Using disparity

The first method is based on creating disparity maps. A disparity map shows the difference of depths of an object. Sampling enough vertical depth vectors makes it possible to construct a 3D model.

Description:

After frames have been extracted from the video, the algorithm runs over all consequent frames. First, the two consequent frames are rectified. For rectification, we detect interesting feature and corner points on the first image by
using OpenCV function `cvGoodFeaturesToTrack`. The found points are further refined by the OpenCV function `cvFindCornerSubPix`. Based on the refined feature points of the first image the feature points of the second image are calculated by using the function `cvCalcOpticalFlowPyrLK`. By now we have found the feature points on both images and we are using these points to find the fundamental matrix that relates corresponding points on these images. For this step we used `cvFindFundamentalMat`. Now it is possible to rectify these two images but as the camera is uncalibrated we have to call the function `cvStereoRectifyUncalibrated` which computes the rectification transformations without the camera parameters. The most problematic step is computing the undistortion and rectification transformation map as the functions that do this work require camera matrix, distortion coefficients and rectification transformation matrix. With frames from a monocular camera these values have to be hardcoded and then used in the `cvInitUndistortRectifyMap` function. As a final step, remapping has to be done to create the rectified output images and for that we used `cvRemap` function.

After rectification, we run the disparity function for all consequent rectified images in order to generate the disparity maps. There are several functions for this that give slightly different results: `cvFindStereoCorrespondenceGC`, `cvFindStereoCorrespondenceBM`, `_stereoBM`.

When we have generated all disparity maps, we take a vertical sample from each of these disparity maps. The sample is taken from the middle of the image, where the middle point of the object should be located. The sample is a vector of floating point values with the size of the height of the image, containing the colors of each pixel on the vertical sample line. These grayscale colors reflect the depths of the objects. The consequent samples are used to create faces for the 3D object by filling the space with triangles. As all the values in the vector represent certain heights, we simply proceeded them by taking two consequent points in both vectors and created two triangles out of them. We continued this process for all vectors describing the consequent disparity maps.

**Problems with this method:**

If a detailed disparity map is used then there are no major problems. However, it is not trivial to create a detailed disparity map by using a single uncalibrated camera. The idea is to simulate the stereo camera by using two consequent frames by a single camera. In order to create the disparity map, it is important to know the properties of the camera that is used for recording. By using a single camera the calibration has to be done manually, which decreases the quality of the disparity maps. In order to solve this problem, the camera should be calibrated. For this the setting of the recording should be changed by fixing the object, adding calibration features and moving the camera around the object with constant speed. This kind of setting is described in [3]. Unfortunately we were not able to build such setting and therefore could not test the 3D reconstruction method described by [3].
4.2 Difference method

The second method is based on detecting image differences on three consequent frames. Due to the changes of the object position it is possible to detect the contours of the object. A vector of contour coordinates is used for creating the 3D model of the object.

Figure 1: A snapshot of a video with a basket with the contour obtained by image difference method.

Description:

First we blurred the input frames with Gaussian kernel to decrease the effect of noise and camera vibration. Kernels with size up to 15x15 were used with more diverse backgrounds where smaller kernel was used for more homogeneous backgrounds. Second, we used three consequent frames with a fixed frame index step from original video to detect the boundaries of the object. The boundaries were detected by calculating the absolute differences between first and second and second and third frame and taking the average of those two. As a result, the moving parts of the object were easily recognizable, while the background remained close to zero (black). Next, we applied erosion to the difference image to eliminate less brighter narrow lines and then dilated the image several times to achieve more connected components in the image. Then, we used binary threshold to cut all the less significant features in the images and only left most significant parts in the foreground. An example of all above steps till binary thresholding step is given in Figure 1.

Last, we used OpenCV function findContours to detect the largest contour in the image and assumed it to be the contour of the object we are reconstructing in 3D. Then, we divided the contour into two parts, splitting by the middle vertical line in the center of the image and used the right part of contour to build the profile of the object from the specified angle. We repeated all these steps for all angles. The color for the points on profile were derived from image contour looking at the object from position 90 degrees anti-clockwise, thus lessening the illumination and distorts of the color. The colors were derived from the other images by taking the points y-coordinate on the contour and middle x coordinate on the other picture. The faces of the 3D model were constructed based on two consequent profile contours. The points on contour were sorted
such that the topmost point that opened the contour from the middle of the image were in the beginning and other points were in the order of traversing the contour. Then, similarly to disparity map method, we created triangles between two consequent points on both contours. In case a contour had more points than the other, the triangles were drawn to the last point of the shorter contour.

Problems with this method:
The method can distort the object, making thinner parts of the object thick. This happens, because an outstanding part of the object can make a thinner part appear more thick - the contour is drawn as a profile on plane intersecting the middle of the object, thus ignoring the fact that outstanding objects may not lie on the plane.

5 Experimental results
Here we show the results of creating a 3D model from simpler symmetric objects and more complex objects.

5.1 Disparity based results
It turned out that the the disparity maps are not precise enough. In order to make the transition of colors in the disparity map smoother we used blurring on the disparity images. However, this was not enough to create a model that resembles the original. An example of a disparity image is shown on Figure 5.1.

Figure 2: A sample of an unblurred disparity map
Figure 3: The 3D reconstruction of a basket using image difference based profile contour method.

Figure 4: The 3D reconstruction of a basketball using image difference based profile contour method.
Figure 5: 3D reconstruction of a dragon (top), a male figure (middle) and another male figure (bottom). The difference based profile contour method distorts more complex and unsymmetric objects. The two bottom objects are examples of distortions mainly caused by the object not perfectly rotating around the z axis.

5.2 Difference based results

Image difference based method works slightly better on symmetric and round objects with most parts of the object in equal distance to the object center. In Figures 3, 4, we see examples of a basket and basketball that both are such objects. However, there are also some artifacts caused by the turntable the
objects are standing on. Another small issue is also demonstrated here: the mapping of colors to vertices is slightly wrong due to different sizes of contours in basketball images. Part of the green color of the ball is wrongly rendered onto the turntable. With basket this does not turn out so badly as the baskets body is much more contrast than the background and the turntable, making it easier to separate it from all the rest.

In Figure 5, we see the results of more complex objects we tried to reconstruct in 3D from the image sequences. We see that the bodies of the all objects are distorted. Thin parts are appearing more fat due to the nature the contour reconstruction algorithm works: it places outstanding objects perfectly, when they are viewed full in profile, but with slightly different angles these parts introduce errors to surrounding parts, making them also look like stretched out. Two bottom objects in Figure 5 actually are not as complex objects as the dragon, but they are an example what happens if the object does not rotate perfectly around the axis.

6 Conclusion

This paper proposes methods for 3D model creation and describes the advantages and disadvantages of each approach. Difference based approach distorts the 3D model of complex objects and is therefore worse than the disparity based method. However, with symmetric objects like a ball it can give results with sufficient quality. Contrary to the difference method, disparity method will give low quality or bad results on symmetric objects and better results on more complex objects.

References

