AUGMENTED REALITY WITH OCCLUSION RENDERING USING BACKGROUND-FOREGROUND SEGMENTATION AND TRIFOCAL TENSORS

Wang Hee Lin¹
¹National University of Singapore
10 Kent Ridge Crescent
Singapore 119260
Dept. of ECE, VIP lab.
enGP1622@nus.edu.sg

Kuntal Sengupta²
²Advanced Interfaces, Inc.
403 S. Allen St.
State College, PA 16801, USA
ksengupta@advancedinterfaces.com

Rajeev Sharma²,³
³Department of CSE
317 Pond Laboratory
Penn State University, PA 16802
rsharma@psu.edu
rsharma@advancedinterfaces.com

ABSTRACT

Over the recent years, video communication has emerged as one of the most popular form of communication. Developing the technology for merging the real and virtual image streams and 3D models for better communication is an active research topic. In this paper, we propose a novel method of real time augmentation of 3D objects in 3D scenes using trifocal tensors. The augmentation is robust to arbitrary camera motion. The proposed technique does not require camera calibration, use of fiducials and prior knowledge of structural model of the scene. Also, using background-foreground segmentation, the occlusion issue in augmented reality is addressed. We show several results of the successful working of our algorithm in real life situations. The technique works on a real time video from a USB camera, Creative webcam III on a P IV 1.6GHz system without any special hardware support.

1. INTRODUCTION

Augmented Reality (AR) is an active field of research with a wide range of applications in bioengineering, military, entertainment, etc. An AR system allows the user to see the real world, with virtual objects superimposed upon or composed with it. It supplements reality with valuable and easy to understand information. Ideally, AR allows the user to preserve both virtual and real objects coexisting seamlessly in the same space. Azuma[1] expanded the definition to systems which have the following characteristic features:

1. Combines real and virtual objects in a real environment
2. Runs interactively, and in real time; and
3. Registers (aligns) real and virtual objects with each other

Recently, AR has been used for real-time augmentation of broadcast videos, primarily to enhance sporting events and to insert or replace advertisements in a scene. With sophisticated computer vision algorithms advertisers can embellish broadcast video with virtual ads and product placements using AR augmentations. In both systems, the environments are carefully modeled ahead of time, and the cameras are calibrated and precisely tracked. Significant improvements in computer vision techniques have led to real time and interactive imaging of complex bioengineering systems [2].

Most of the previous work in object registration utilizes 3D image data, obtained from a scanning laser rangefinder and searches their best match with 3D model data sets by using a least square minimization of distances between data sets [3]. Uenohara and Kanade developed a real time 2D image overlay system [4] using vision algorithms. The system runs at frame rate using multiple DSPs, and the features for tracking have to be manually picked. Their system utilizes 2D intensity images and detects feature points by template matching. The change of intensity pattern due to view change is compensated by skewing reference images with computed object pose parameters.

The system employed in this paper is a monitor based monocular system and satisfies the criterion for augmented reality given by Azuma[1]. It is basically meant to enhance visual communication for several applications. Our system does not require careful preparation of the background (like blue screen) or elaborate calibration of the camera, save for a preset number of background frames to model the background. The system runs in real time using Intel OpenCV libraries and employs no special hardware support. This system will find pertinence in media broadcast and communication industries.

We propose a novel method of augmenting 3D objects in 3D scenes using trifocal tensors. We begin by introducing the system overview for 3D augmentation. We review feature point detection and tracking methods involved, as well as algorithms used to obtain accurate and reliable point correspondences across different frames. We then describe our technique for reprojecting the augmentation of a 3D object onto the video scene. The augmentation matches the geometry of the scene and hence seems to be a part of the scene and not an artificial imposition. Also, the occlusion problem inherent in most
AR systems is addressed using foreground-background segmentation. Finally we present the results of our method that demonstrate the practicality of the system.

2. SYSTEM SETUP

To have a realistic augmentation, the foreground objects in the scene should occlude the augmented object, which in turn occludes the background regions of the scene. This is possible by identifying the foreground segmentation. Using Stauffer and Grimson's[7] method of background modeling, where pixels are actively modeled with bimodal Gaussian distributions, the system learns the background of the 3D scene in a preset number of background frames. Strong feature points are identified and their image coordinates are stored. Without moving the camera, the user then places the 3D object that is to be augmented into the scene. Another image of the scene is then captured. The user then uses a mouse to draw a bounding box around the object of augmentation to limit the search space for it to speed up subsequent computation. Using foreground-background segmentation within the box, the foreground pixels that make up the 3D object are identified.

The background model of initial view is learnt. Strong tracking points are identified. The 3D scene both with and without the physical object for augmentation are imaged from two model views. Dense correspondence of object between model views are computed.

Preparation Stage

The camera is then moved in a straight direction perpendicular to the optical axis. This translation movement is typically equivalent to around 20 pixels for an augmentation object 50cm away from the camera, with the feature points being continuously tracked throughout the translation. With the camera stationary in this 2nd position, the image coordinates of the tracked points are once again stored. The 3D object is then taken away. Using global dynamic programming, the dense point correspondences of the object taken from the two different camera positions are computed.

With the preparatory work completed, the user may now shift the camera arbitrarily to any neighboring rotation and translation. This view constitutes the third view with respect to the first two camera views, which are known as the model views. At every frame, the features points in the scene are continuously tracked. views: the 3D augmenting object can be reprojected in real-time into the present video frame. This gives viewers the impression the object is actually present at the scene, thus in effect augmenting the 3D scene with the object.

3. FEATURE DETECTION AND TRACKING

In order to augment the scene with a 3D object in a 3D scene for a moving camera, our methods require that some suitable transformation be found to transfer points from a reference model view to a new view, or in other words find a transformation $T_{r}(u, v) = (u', v')$ with $(u, v), (u', v')$ being a point correspondences between the views. This transfer process is known as reprojection and is responsible for reprojecting the augmenting object from the model views to a present new view, thus augmenting the present view. The transformation is equivalent to a trifocal tensor $T$ for the 3D case and a minimal of 7 point correspondences between 3 frames is required to compute it. Therefore there is a need to identify and track strong features points like corners throughout the entire video sequence.

3.1. Feature Detection

To avoid the use of fiducials, it is necessary to identify good features in the background for tracking. The minimal eigenvalue of every pixel is calculated [5] and pixels whose eigenvalue fall below a certain adjustable threshold are rejected. The remaining pixels are then rejected if they fall within a certain Euclidean distance of each other and their eigenvalue magnitudes are smaller relative to their neighbors. The eigenvalue threshold is adjusted automatically to take into account local lighting and texture conditions.
3.2. Feature Tracking and Management

The tracking as proposed by Lucas and Kanade[6] was used to track the feature points from frame to frame. Due to the large number of points being tracked while the camera moves, the \( \text{Tr} \) calculated at every frame is theoretically very accurate. However spurious feature points sometimes arise as a result of depth discontinuities, reflective surfaces as well as occluded points, and lead to incorrect tracking results.

To detect such outlier points, the \( \text{Tr} \) is initially computed by using all the point matches. Next, it is used to estimate the matches between frames. If the Euclidean distances between any corresponding pair of predicted coordinates and actual coordinates exceeds a certain low threshold, it is marked invalid and a new \( \text{Tr} \) is calculated from the remaining valid tracking points.

4. REPROJECTION THEORY

To augment a 3D object in a 3D scene, the traditional means has been to recover either the depth map or structure of the object and scene. However such attempts are highly susceptible to noise, especially for faraway objects and situations where the model views have small base-line. Two recent works have used different approaches to avoid depth recovery while reprojecting an object in a novel third view. Avidan and Shashua[9] used the trifocal tensor while Lei and Hendriks[10] used a process of rectification, interpolation followed by derectification.

Our method is based on using the trifocal tensor as a means of capturing the geometry of a three camera situation in order to augment a 3D scene. In essence, the trifocal tensor encapsulates the information provided by the projection matrices of the three cameras. The key idea behind using trifocal tensors to augment is that given the image coordinates of a 3D point in two model views, the trifocal tensor can reproject the 3D point as image coordinates in a third novel view.

4.1. Finding Dense Correspondences

From the above description of the trifocal tensor, reprojection an augmenting object in a third novel view requires the dense point correspondences of the object pixels in the two model views. Scharstein and Szeliski [11] have recently conducted a comprehensive survey on various dense correspondence methods, where it is assumed that the methods “are given a pair of rectified images as input.” In our experiments, the camera is translated between model views in a direction parallel to the image plane, which circumvents the need for rectification. However our method works equally well should the need for rectification arise.

Using the code supplied by Scharstein and Szeliski [11], various dense correspondence algorithms are evaluated on typical examples of model views. Dynamic programming, which is classified under [11] as a global optimization algorithm, is utilized because it strikes the right balance between performance and real-timeliness as it can “find the global minimum for independent scanlines in polynomial time.”

4.2. Trifocal Tensor and Reprojection

With the dense correspondences between 2 model views computed, the only other thing required for reprojection of the augmenting object will be the computation of the trifocal tensor relating each different view with the model views. Prior to calculating the trifocal tensor, a separate translation is applied to each image such that the centroid of the point correspondences is at the origin, and then a scaling is applied so that the average (RMS) distance of the points from the origin is \( \sqrt{2} \). This ensures numerical stability during the solving of linear equations [8].

Now let \( n \) number of scene points (where \( n \geq 7 \) for a linear solution) be tracked throughout the entire video sequence. Then the homogenous coordinates of any particular scene point over any 3 arbitrarily chosen frames can be denoted by

\[
(x^i, x^j, x^k), (x'^i, x'^j, x'^k), (x''i, x''j, x''k)
\]

where the prime superscript denotes the frame index and the homogenous coordinates are represented in a way such that \( (x^i, x^j, x^k) \) are set to 1.

For any of the \( n \) point correspondences over the 3 frames, 4 independent equations can be derived from the equation

\[
A(x^i x'^j x''k) - x'^i T_{ij} + x''i T_{ik} = 0 \quad (1)
\]

for the four different choices of \( i, l = 1,2 \) where \( k \) is summed through \( k = 1,2,3 \) for each of these choices.

With \( n \) triple point correspondences, writing out the linear equations result in the matrix equation \( A t = 0 \) where \( A \) is a \((4n x 27)\) matrix and \( t \) is a \((27 x 1)\) column vector denoting the entries of the trifocal tensor. Using SVD, the trifocal tensor can be obtained as the eigenvector corresponding to the least eigenvalue [8].

4.3. Reprojection

Given any pair of point correspondences \( (x^i, x^j) \) in the model views, its correspondence \( (x'^i, x'^j) \) in the novel view can be determined by eqn(1) via substitution. Though backward-mapping is usually more preferable for reprojection, however in this case
backward-mapping requires a non-linear solution that need not necessarily be unique. Therefore forward-mapping, whereby pixel intensity values of the augmenting object from the reference frame are copied to the computed image coordinates in the novel view, is employed. However the problem of “holes” arises when a sparse number of pixels, owing to quantization, are not mapped to where it should. Considering the small number of “holes”, an averaging filter has been used to fill them up by applying the average value of the surrounding augmented pixels to the “holes”.

5. RESULTS AND CONCLUSION

For the experimental setup, a Creative webcam3 was used to feed in a real-time video stream to the program. The 3D object to be augmented in the experiment was a pink piglet plush toy. In the example above, the background model of the reference view in fig 2.a was learnt. Fig (2.a, 2.b) were the views in which the model views were taken, albeit without the foreground object. The dense correspondence between the two model views computed next. Fig (2.c) to (2.h) illustrate the augmented views as the camera moves around. Note that the foreground object occludes the augmented bunny. Traditional AR systems (like the AR toolkit) fails to render such occlusion. The processing rate is around 8 fps.

In this paper we present a method for augmenting 3D objects in 3D scenes. The augmentation is continuously rendered in real-time under arbitrary camera motion in such a way that gives the impression that the augmentation is physically present. The techniques do not require multiple cameras, camera calibration, use of fiducials nor structural model of the scene to work. From experiments, the methods are demonstrated to be practical, real-time and successful. In the future, we plan to investigate ways to incorporate parts that are not present in both model views as well as take into account the occlusion effects caused by true foreground objects.

6. REFERENCES