Invariant-Based Augmented Reality on Mobile Phones

Jie Shen, Lie Luo
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
Email: zeropoint17@gmail.com

XiaoYu Zheng
Hydro-China Chengdu Engineering Corporation, Chengdu, China
Email: 2004081@chidi.com.cn

Abstract—A calibration-free augmented reality based on affine invariant is firstly formulated by tensor method. This approach does not use the calibration parameters of the camera and the 3D locations of the environment’s object, and can realize the augmentation of virtual objects. Meanwhile, a new approach to resolving occlusion problem in augmented reality is presented. Based on an arm-optimized implementation of the Scale Invariant Feature Transform (SIFT) algorithm developed by David Lowe and Random Sample Consensus (RANSAC) algorithm, the point correspondences in any two views are determined. According to the invariant for two views and these point correspondences, the occluding contours can be transferred to any views, so the occlusion is resolved. Some typical experiments show that the approach of invariant-based augmented reality is feasible on mobile phones.

Index Terms—augmented reality, invariant, occlusion, mobile phones

I. INTRODUCTION

With the advances in web2.0, rich media and smart phones, augmented reality (AR) on mobile phones has become a reality [1][2]. An augmented reality application refers to a live direct or indirect view of a physical real-world environment whose elements are augmented by virtual objects. Web 2.0 and rich media technologies provide more sources of information, but AR allows a richer presentation of this information. Moreover, smart phones based on arm architecture with the low power and high performance is fit for the implementation of augmented reality. However, mobile phones are embedded systems which usually have very strict memory limitations, processor limitations, and speed limitations. Therefore, the algorithms for AR must be computationally feasible on current generation mobile phones, and they can be optimized to run at the desired level of speed, quality, and resource consumption.

In this paper, a more practical AR approach, calibration-free augmented reality based on affine invariant is formulated. The weak perspective projection model is assumed for camera-to-image transformation. This approach does not use the calibration parameters of the camera and the 3D locations of the environment’s object, and can realize the augmentation of virtual objects. Moreover, we present a contour-based approach to resolving occlusion problem in AR. First, the key points of occluding contours between virtual and real objects may be specified interactively according to epipolar and other constraints in the first two frames. Second, The arm-optimized implementation of SIFT [3] and RANSAC algorithm [4] are applied to search for the correct point correspondences in any two views [5]. With these points, the points of the occluding contour C are transferred to any views by the invariant for two views. Finally, it is feasible to track the occluding contours in any views, so the virtual objects can be drawn behind the contour.

II. RELATED WORK

In the past several years, model-based [6][7] and structure-and-motion methods [8] are widely used in augmented reality. However, these methods require a priori knowledge of 3D models or a limitation to motion. With the development of computer vision, the application of AR has been promoted, and the new models have become more practical. These models include that when 3D location and calibration parameters of camera are unknown, several images are used to restore the structure of objects to complete reprojection. Through the recognition of fiducial in the scene and by means of affine representation, the virtual objects produced by computer will be seamlessly synthesized into the video images of the real scene. This method is different from the camera calibration technology [9][10]. In this paper, the weak perspective projection model is assumed firstly for the camera-to-image transformation, and the calibration-free augmented reality based on four points is formulated by the tensor method, and then image augmentation is realized. However, this method is not effective when occlusions exist between virtual and real objects [11]. This occlusion problem could easily be solved on condition that the model of 3D scene is given. Since little is usually known about the real world to be augmented, it becomes challenging to resolve occlusion in augmented reality. Theoretically, a dense map [12] may be inferred from a stereo pair, so the depth between

© 2010 ACADEMY PUBLISHER
doi:10.4304/jmm.5.6.588-595
virtual and real objects can be compared. As a matter of fact, this method lacks accuracy, and is difficult to use. A contour based approach with 3D reconstruction has been developed by Shen [13]. However, four or more fiducial points from frame to frame have to be tracked. The proposed approach in [14] is based on background subtraction. It needs to model the background using two uncalibrated views, and displacements of the camera must be small in order to fit with the initial background model. More recently, an approach using deepness calculation [15] to accelerate processing speed, but it only judge occluding or occluded relation between whole virtual object and whole real object.

III. AUGMENTED REALITY BASED ON AFFINE INVARIANTS

A. Affine Structure Based on Four Point

Based on the results of Koenderink and van Doorn [16], the affine representations of virtual objects can be formulated by the tensor method. As known, the affine frame is constructed by four noncoplanar points. Let \( O, P_1, P_2, P_3 \) be four noncoplanar points in the 3D world, and \( O', P'_1, P'_2, P'_3 \) the corresponding coordinates from the second camera position. An object point of interest \( P \) with respect to the basis \( OP_1, OP_2, OP_3 \) is shown as follows:

\[
P = b' \ g_i.
\]  

Where \( g_i \) is \( OP_i \), and \( b'_i \) is the affine coordinate of the point \( P \). In order to make the deduction convenient, the two-dimensional question is shown by Greek letters such as \( \alpha, \beta \), and the three-dimensional question by Latin letters such as \( i, j \).

Under parallel projection, the viewing transformation between two scene-cameras can be represented by an arbitrary affine transformation, i.e. \( OP'=T(OP) \), where \( T \) is a linear transformation. Therefore, the coordinates \( b'_i \) of \( P \) remain fixed under the viewing transformation. At the second camera position, the corresponding \( P' \) of the point \( P \) may be written as follows:

\[
T (P)=T (b'_i, g_i).
\]

i.e.: 

\[
P' = b'_i \ g'_i.
\]  

Since the depth is lost under affine transformation, we have a similar relation in image coordinates (using lower case):

\[
p = b'_i \ g_i,
\]

\[
p' = b'_i \ g'_i.
\]

Let \( g_i \) and \( p \) be divided in image coordinate, and substitute them into equation (3):

\[
p = b'_i u_{ai}.
\]  

According to equation (5), if the coordinates of the point \( P \) in two images are known, the affine coordinates of the point \( P \) can be derived from the affine basis. Finally, the position of the point \( P \) in any frame may be reprojected according to equation (4).

B. Virtual Object Rendering

Under affine coordinate, the locations of virtual objects are determined by the affine basis. At the same time, the rendering operation like Z-buffering is allowed, and thus the hidden-surfaces may be processed. Let \( \xi \) be the direction of optical axis, which is given by the cross product:

\[
\xi = \phi \times \psi
\]

where \( \phi, \psi \) are the corresponding vector of row value of basis vector. Thus, the transformation of the point \( P \) can be achieved:

\[
P' = [\phi, \psi, \xi] \ b'^T.
\]

Where \( P=[p^1, p^2, p^3] \) is the reprojection of the point \( P \) and depth value, and \( [\phi, \psi, \xi] \) is the mix product.

C. Location of Virtual Object

Before virtual objects can be augmented into a three-dimensional environment, the geometrical relationship between these objects and the environment must be established. From the result of stereo vision the three-dimensional location of a point in the environment can be derived from two images taken by the different locations of a camera. The main questions are how many point projections need to be specified, and how the user specifies the projections of these points. According to the result of affine geometry, if \( y_1, y_2, y_3, y_4 \) are the coordinates of four noncoplanar points on virtual object expressed in the object’s coordinate frame and \( y_1', y_2', y_3', y_4' \) are their corresponding coordinates in the affine frame. There is an invertible and homogeneous object-to-affine transformation \( A \) such that:

\[
[y_1', y_2', y_3', y_4'] = A [y_1, y_2, y_3, y_4]
\]

Therefore, the affine coordinate of any point on the objects can be determined, when the affine coordinates of four points on the objects are known, and the transformation \( A \) can be derived from the affine coordinates of four points.

To determine the locations of a point on the virtual object in two images, interactive method may be adopted. If the locations of a point \( Y \) are determined in two images, its affine coordinate can be computed, and thus reprojection can be realized. The locations of point \( Y \) in two images are constrained each other. If the location of point \( Y \) in first image is determined, the corresponding location in second image is certain to be determined through epipolar and other constraints.

As shown in Fig. 1, \( Y \) is a point of 3D space, and \( X_1, X_2, X_3 \) are coplanar with images \( x_1, x_2, x_3 \) and \( x'_1, x'_2, x'_3 \).
in the first and second images respectively. The epipolar plane defined by the optical centers O, O’ and point Y intersects the plane II which is determined by the lines YO and YO’ respectively. From the result of affine geometry, three points determine a plane affine transformation such that \( x^i = T x_i \), \( i = 1, 2, 3 \). This transformation is used to transfer the point \( y \) to \( y^i = Ty \), and so a point \( y^i \) on the epipolar line can be determined. Under affine transformation, all epipolar lines are parallel. The direction of any epipolar line is simply determined by making the global affine coordinate frame aligned with the first image [17].

A simple solution for \( S \) is:

\[
S = \begin{bmatrix}
A_1^{-1} & -A_1^{-1}d_1 \\
0 & 0 & 1
\end{bmatrix}
\]

Where \( A_1 \) is 2x2 and \( d_1 \) is 2x1. If \( B \) is divided into \( B = \begin{bmatrix} A_1 & d_1 \end{bmatrix} \), and \( B = \begin{bmatrix} A_i & d_i \end{bmatrix} \), \( d_i \) is the direction of the epipolar line. To get \( d_i \), we must make each projection matrix post-multiplying matrix \( S \).

Once a point and the direction of epipolar line are determined, the epipolar line can be drawn in images. For a point \( Y \) of 3D space, not only is there epipolar constraint, but also collinearity or coplanarity constraint. Through these constraints the location of point \( Y \) in the second image can be determined.

IV RESOLVING OCCLUSION BASED ON INVARIANT FOR TWO VIEWS

A. Invariant for Two Views

From the result of stereo vision, suppose we have found a set of matched points \( u_i \leftrightarrow u_r \), and the fundamental matrix is defined by the equation

\[
u_i^T F u_r = 0
\]

for the pair of matching points. In particular, writing \( u_i = (x_i, y_i, 1)^T \) and \( u_r = (x_r, y_r, 1)^T \), the above equation will be

\[
u_i^T F u_r = \begin{bmatrix}
f_{11} & f_{12} & f_{13} \\
f_{21} & f_{22} & f_{23} \\
f_{31} & f_{32} & f_{33}
\end{bmatrix}
\begin{bmatrix}
x, y, 1
\end{bmatrix}^T ;\ (10)
\]

This equation is written by a dot product, \( p \cdot f \), where \( p = (x_1y_2, x_2y_1, x_1y_2, y_1x_2, y_1x_2, x_1y_2, y_1x_2, y_1x_2, 1)^T \) and \( f = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T \).

Given nine points correspondences in two images can be constructed as follows,

\[
P = \begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
\ldots \\
p_9
\end{bmatrix}.
\]

From the nine points correspondences, we obtain a set of linear equations of the form

\[
P f = 0.
\]

To avoid a trivial solution \( f \), the determinant of \( P \) must be identically zero. Since the condition \( |P| = 0 \) is satisfied in any position of the cameras, the \( |P| = 0 \) is an invariant for any two views. This invariant can be used to transfer any ninth point.

If the ninth vector \( p_9 \) is written by the notation \( p(u_i, u_r) \), \( P \) between views 1 and 2 can be expressed as,

\[
P = \begin{bmatrix}
p_1 \\
p_2 \\
\ldots \\
p_9
\end{bmatrix}
\]

where \( u_i \) denotes the position of the ninth point in view \( i \). The invariant \( |P| = 0 \) leads to a linear expression about the coordinates of \( u_i \). That is

\[
\alpha x_i + \beta y_i + \gamma = 0.
\]

If eight point correspondences are given between 1 and \( i \), the position of \( u_i \) lies on a line in image \( i \). Similarly, we can determine another line by this invariant between views 2 and \( i \). Thus, the position of the ninth point in view \( i \) is given by the intersection of these two lines.
B. Feature Detection and Matching

Epipolar geometry defines the geometry between the two cameras creating a stereoscopic system, but all the epipolar geometry is contained in the so called fundamental matrix. Hartley did a lot of work about fundamental matrix [18]. However, in the computation of the fundamental matrix, outliers typically arising from gross errors such as correspondence mismatch result in estimation error. In this paper, a method based on the RANSAC algorithm for fundamental matrix estimation is put forward. Thus, the estimation error due to the mismatching points is eliminated. At the same time, the matching points are obtained. The flow scheme based on robust RANSAC method is as follows.

Step 1. Find interesting points in scale space.

In this work we use Lowe’s [3] SIFT features, which are geometrically invariant under similarity transforms and invariant under affine changes in intensity. The SIFT algorithm may be decomposed into four stages: feature point detection, feature point localization, orientation assignment and feature descriptor generation. The resulting 128 element feature vectors are called SIFT descriptors.

However, the traditional SIFT computation and matching algorithms do not work very well on low-end embedded platforms. Some improvements have been proposed to speed up the SIFT computation. The SIFT’s runtime is dominated by the computation of the Gaussian pyramid and descriptors. First, based on the experimental results in [3], the number of orientations and size of the descriptor can be used to vary the complexity of the SIFT computation. We use a 3x3 descriptor with 4 orientations, resulting in feature vectors with 36 dimensions, which lead to only ~10 percent worse than the best variant with 128 dimensions. Second, a 2D gaussian kernel is a separable convolution. In other words, it may be decomposed into two successive 1D blurs. This means that instead of a RxR(R is the radius of the kernel) convolution kernel, you have a 1xR kernel and a Rx1 kernel. Obviously this is quite a improvement. Finally, the memory access optimizing techniques are applied in our system. Since ARM compiler uses four registers, namely R0-R3 to pass parameter to a function, we limit the parameters of a function no more than four to eliminate unnecessary memory accesses. At the same time, the C library functions memset and memcpy are designed for all type data size, and thus they include a number of checks for different cases and sizes. However, these checks will cause processing time increasing. We can customize these functions to exactly suit specific requirements. Due to space limitations, we will only discuss these optimization approaches.

Step 2. Find feature matching and its consistent set.

Now all features have been detected and described, this step is to match them. The original SIFT uses a k-d Tree with the BBF strategy. This paper adopts the hybrid SP-Tree search approach [19] which has been shown to outperform other existing approaches. Since this approach may properly set an overlapping buffer, the computational load of back-tracking can be greatly reduced. Next, we use RANSAC to reject additional outliers.

Step 3. Do J++

Step 3.1 Choose a set of seven points from candidate matches.

Step 3.2 Compute fundamental matrix using seven points method.

Step 3.3 Compute median $M_j$. The $M_j$ is given by

$$M_j = \text{median}, |r^2|$$

$$r^2_i = d(x_i^T, F x_i) + d(x_i^T, F^T x_i) = x^T F x + (x^T F)_i + (F x)_i$$

Where $r^2_i$ is the distance from matches point to its epipolar lines.

Step 3.4 Compute weight function and number of inliers.

The weight function is proposed by Torr [20] is the following:

$$w_i = \begin{cases} 1, & r^2_i < T^2 \\ 0, & r^2_i \geq T^2 \end{cases}$$

Where $n$ is the number of data. Once the $w_i$ is obtained, the number of inliers is given by

$$I_j = \sum w_i$$

Step 3.5 until $J > n$

Step 3.6 Calculate the number of inliers for each $F$, and the chosen $F$ is the one that maximizes it. Once the inliers are obtained, $F$ is recalculated using the normalized 8-point algorithm.

C. Resolving Occlusion Based on Invariant

When a virtual object is added into the scene, the user needs to specify the relationship between them. For instance, we want to add the virtual house A behind the real house B (Fig. 1). In order to resolve occlusion, if it is feasible to track the contours consisted of key points 1, 2, 3, 4, the virtual house A can be drawn behind the contour. If the scene is complicated, we may label each contour point as being “behind” or “in front of”, this idea depending on whether it is in front of or behind the virtual object.

According to the above invariant, if position of a point on occluding contours is determined in stereo images, the position of the point in any images may be calculated using the intersection of two lines. The main question is, after the position of a point in the first frame is...
determined, how to fix its position in the second frame. As known, if projection of point P in a 3D world is specified in one image, its projection in the second image must lie on a line satisfying the epipolar constraint. At the same time, through point collinearity or coplanarity constraint, the position of any point in the second frame can be determined.

![Figure 2. The occlusion between real and virtual objects.](image)

V. INTERACTIVE ALGORITHM

Step 1. Select four fiducial points in the first image to establish the affine basis, and four noncoplanar points on the virtual object s as well as the key points of occluding contours between virtual object s and the real scene.

Step 2. Select four corresponding fiducial points in the second image to obtain the projection matrix, and compute the epipolar lines of four points on the virtual objects and the key points of the occluding contours in the second image.

Step 3. Specify the locations of four points on the virtual objects and the key points of the occluding contours in the second image according to epipolar and other constraints.

Step 4. Compute the affine coordinates of four points on the virtual objects from equation (5) as well as the object-to-affine transformation matrix A in equation (8), and determine the affine coordinate of any point on the virtual objects.

Step 5. Detect and track the four fiducial points in any images to construct affine basis, and implement the reprojection of the virtual objects. At the same time, based on the invariant for two views, the occluding contours are tracked in any images, and so the virtual objects can be drawn behind the occluding contours.

VI. EXPERIMENTS

In order to demonstrate the effectiveness of this method, two typical examples were selected and briefly described in the following. The target device at client is a ASUS mobile phone with an Intel XSCALE CPU works at 520 MHz.

A. Experiment 1

The first experiment is to fuse the virtual building into real 3D scene which is based on affine structure for calibration-free augmented reality. Fig. 2 is the experiment environment. Now, suppose that a high building will be constructed. In Fig. 2(a), a set of salient points are extracted using SIFT algorithm. The black lines show all correct correspondences based on RANSAC. In Fig. 2(b) and (c), the affine frame is constructed from the image coordinates of cross-center in Fig. 2(b), which are acquired by interactive method. Similarly, the image coordinates of key points on the occluding contour are obtained in the first frame (see the four small black squares in Fig. 2(b)). The epipolar lines of these points are computed and drawn in the second image (see the four black lines in Fig. 2(c)). Then, according to epipolar and collinearity or coplanarity constraints, the image coordinates of key points on the occluding contour in Fig. 2(c) are obtained with interactive method. Furthermore, according to the invariant for two views and point correspondences, the positions of the occluding contour in other frames are determined. In Fig. 2(d) and (e) the virtual building is fused into the 3D scene very well.

B. Experiment 2

The second experiment in Fig. 3 is to fuse building group into a scenic spot. The process of resolving the occlusion problem is similar to that in experiment 1. The difference between these two experiments is that this occluding contour between virtual and real objects (see the two black circles in Fig. 3(b)) is relatively complicated. Of course, the more accurate the occluding contours tracked are, the more key points on contours which should be interactively marked in the first two frames are needed.

VII. SHADOW ALGORITHM

A. Shadow Algorithm

Since shadows to provide what is essentially a second view of an object convey a large amount of information, it is necessary to model lighting conditions and shadows. However, they are difficult to compute in most augmented reality environments on mobile phones. Currently, no algorithm is known that renders physically correct and dynamically updated shadows in arbitrary scenes, and so shadow algorithm requires simplifications to meet the physical limitation needs of mobile phones. In our augmented reality system, a single algorithm is chosen for shadow calculations. As depicted in figure 5, a light source at infinity is modeled. Because of the light being from infinity, we may assume that all the rays reaching the object are parallel. According to the geometric relationship of the light source and polygons, each polygon’s projection on z=0 plane is calculated. Given that the point-vectors of light direction and vertex point are \((x_l, y_l, z_l)\) and \((x_p, y_p, z_p)\) respectively, the shadow point \((x_s, y_s, z_s)\) is calculated as follows:

\[
S = P \cdot M_{\text{shadow}}
\]

(20)
Figure 3. Resolving occlusion with a simple occluding contour

(a) Extracted salient feature-points and correct correspondences
(b) The first view of the scene
(c) The second view of the scene
(d) Augmented virtual building
(e) Augmented virtual building

Figure 4. Relatively complicated occluding contour

(a) The view of real scene
(b) Augmented virtual building group
Where

\[
M_{\text{shadow}} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
-x_1/z_1 & -y_1/z_1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Figure 5. Lighting model

B. Experiment

This example for demonstrating our shadow algorithm is to fuse a virtual building with shadow into a real scene. As depicted in figure 6, the process added is similar to the experiment 1. After the virtual building is added into the real scene (Fig. 6(a) and (b)), we can know how the proposed new virtual building with shadow would change people’s view and environment.

VIII. CONCLUSION

This paper proposed a contour-based approach without 3D reconstruction which can be used to calculate correct occlusion. The two experiments in the paper have demonstrated that this algorithm is feasible for augmented reality. We compare our approach to previous work by Zhu [15] and Shahid [11], and the contribution of this work includes that there is not the occluding or occluded limit-relation between virtual object and real object, and there is not any marker. Moreover, virtual objects can be mobile or even deformable. However, our approach is not able to deal with partial occlusion problem when there are some movable real objects on occluding contour, which is still the important future work. To our knowledge, no methods can successfully handle all the cases of occlusion problem in augmented reality. The algorithm in this article has greatly enriched the tools for resolving the occlusion problem.

At the same time, the augmented reality system that is based on affine representation is presented in this paper. Using this system, users only need a mobile phone to realize a richer presentation of information. Through these examples, it has been demonstrated that augmented reality is a powerful tool for visualization. It is easy to construct a photo-quality virtual world, and can present more realistic results. Since there is no need to model all the scenes, thus save more time than the traditional VR and raise the efficiency greatly.

ACKNOWLEDGMENT

This work has been partially funded by the National Key Technology R&D Program under Grant No. 2008BAH25B02, and Major National Science and Technology Special Projects under Grant No. 2009ZX01039-002-001-05.

Figure 6. The building with shadow fused into a 3D real scene
REFERENCES


Jie Shen was born in 1970. He received his post-doctorate from University of Electronic Science and Technology, Chengdu, China in 2005. He received his PhD degree in engineering from Sichuan University, Chengdu, China in 2001. He is currently an associate professor of computer science & engineering in the University of Electronic Science and Technology. His research interests include mobile multimedia communication, image/video coding, and augmented reality. Jie Shen has been the Principal-Investigator of 3 research projects supported by National Key Technology R&D Program and Major National Science and Technology Special Projects.

Lei Luo received the BSc and MSc degree in 1987 and 1992 from University of Electronic Science and Technology (UESTC), Chengdu, China. Currently, she is a professor of UESTC. Her research interests include industry-oriented embedded software platform and embedded real-time operating System and development tools. Prof. Luo has been the Principal-Investigator of 8 research projects supported by National Key Technology R&D Program and Major National Science and Technology Special Projects. She is the Member of IEEE.

XiaoYu Zheng was born in 1979. She received her BSc and MSc degree in 2001 and 2004 from SiChuan University, Chengdu, China. She is currently a engineer in Hydro-China Chengdu Engineering Corporation.Her research interests include scientific visualization and finite element simulation.