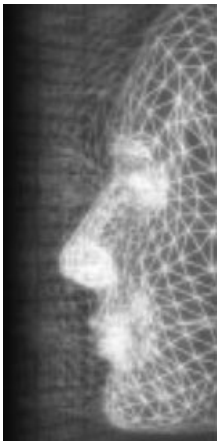
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds 2010; **21**: 509–521
Published online 13 August 2009 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/cav.326

Handling occlusions in video-based augmented reality using depth information



By Jiejie Zhu, Zhigeng Pan, Chao Sun and Wenzhi Chen*

Augmented Reality (AR) composes virtual objects with real scenes in a mixed environment where human–computer interaction has more semantic meanings. To seamlessly merge virtual objects with real scenes, correct occlusion handling is a significant challenge. We present an approach to separate occluded objects in multiple layers by utilizing depth, color, and neighborhood information. Scene depth is obtained by stereo cameras and two Gaussian local kernels are used to represent color, spatial smoothness. These three cues are intelligently fused in a probability framework, where the occlusion information can be safely estimated. We apply our method to handle occlusions in video-based AR where virtual objects are simply overlapped on real scenes. Experiment results show the approach can correctly register virtual and real objects in different depth layers, and provide a spatial-awareness interaction environment.

Received: 15 May 2009; Accepted: 28 May 2009

KEY WORDS: augmented reality; occlusion handling; stereo

Introduction

Augmented Reality (AR) generates a mixed environment that composes virtual objects on real scenes. It provides new human computer interaction strategies by synthesizing real and virtual objects to be one bundled-component that controlled by sensing and display devices.[1] In this sense, AR functions as a bridge linking from real world to virtual world in a way that interaction has no obvious gap. Various applications are booming with this new technology, such as medical surgery, military simulation, manufactory, educational training, entertainment, digital culture, etc.

Video-based Augmented Reality (VAR) featured by its low cost in sensing is most widely applied, now days. VAR captures the real scene by video cameras and composites the scene with virtual objects. Figure 1 shows the structure of a simplified VAR system where computer generated virtual objects and the real scene are mixed.

The quality of semantic meanings of such synthesized scene depends on *how well* the virtual and real objects are aligned. Therefore, one important task of VAR is to *register* virtual objects with real scenes.

Most of VAR simply superimposes virtual objects on video sequence captured by video cameras. This results in real objects often occluded by virtual objects. Such false occlusion relationship leads to confused spatial-awareness when observers are presented. Biologists also find out that long time experience in such environment will cause both eye and motion sickness. We illustrate this problem in Figure 2. It is obvious that the incorrect occlusion among the hand, the cup, and the oranges confuse observers on the spatial relationship.

Occlusion handing in VAR is challenging because the scene is dynamically changing when users are moving. Traditional occlusion handling algorithms depend on features, edges or contour,[2,3] which however, are unstable when the environment is dynamically changed. Recently, researcher are interested in using additional cues to register virtual objects with real scenes, such as affine relationship between sequential frames,[4] depth from stereo cameras,[5] depth with features.[6] Among them, depth is used most widely and successful.

This paper also provides an approach to handle occlusion using depth. Since depth estimated from stereo cameras has noise (such as textureless region), we fuse

*Correspondence to: Z. Pan, State Key Lab of Virtual Reality and Systems, Beihang University, Beijing, 100191, China.
E-mail: zgpan@cad.zju.edu.cn

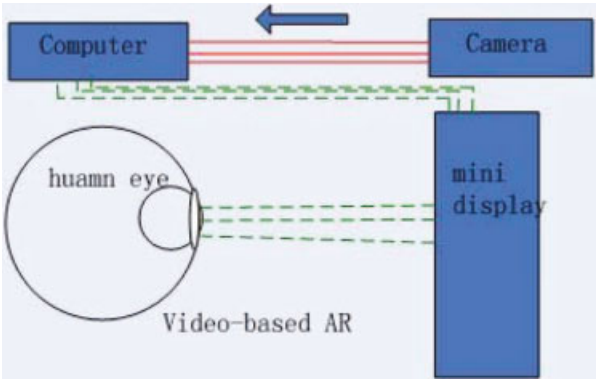


Figure 1. Simplified VAR system. Scene information (red (solid line)) is captured by cameras. By sending videos to computer, synthesized scene (green (dashed line)) is generated and rendered in various displays.

depth with color and neighborhood information instead of using depth alone or with unreliable features as previous methods. Fusion helps to generate complementary results where either depth, color, or neighborhood information does poorly.

Our approach provides a probability method to efficiently estimate occlusion relationships. Unlike previous methods, we estimate depth using stereo cameras with a statistical analysis of error propagation. In addition, we use color aggregation to improve the depth quality. To accelerate the matching process, we incorporate color quantization. We also introduce mixed Gaussian Kernels to describe the interested objects statistically.

The rest of this paper is organized as follows. First section introduces related works. Next section explains our approach to statistical uncertainty analysis. In the following section, we introduce object recognition method by Gaussian models. Then we present occlusion handling utilizing depth, color, and neighbors followed by our experimental results and finally we conclude our work.

Related Work

Occlusion is an important cue for people understanding the scene in real world. Biological researchers find out that occlusion sensing in vision from 2D images relies on several specific cues. Howard summarized[7] 10 different depth related cues which affects people's occlusion sensing. Among them, Bimber[8] concludes that binocular disparity and binocular convergence, are particularly important for human to generate a scene depth. Binocular convergence depends on physiological functions of human's eye, which is difficult to simulate using devices. Binocular disparity can be created by using two cameras within a baseline (simulating the distance between human eyes), and thus is applied widely in computer vision research for structure reconstruction.

This paper also incorporates stereo cameras to recover depth and then handle occlusions. We group occlusions into two different types in a typical VAR environment: either the real scene is occluded by virtual object; or the virtual object is occluded by real scene. The former is common because scene are often treated as background and virtual objects are overlapped on them. The latter is important for accomplishing interaction tasks, such as operating real devices inside virtual objects.

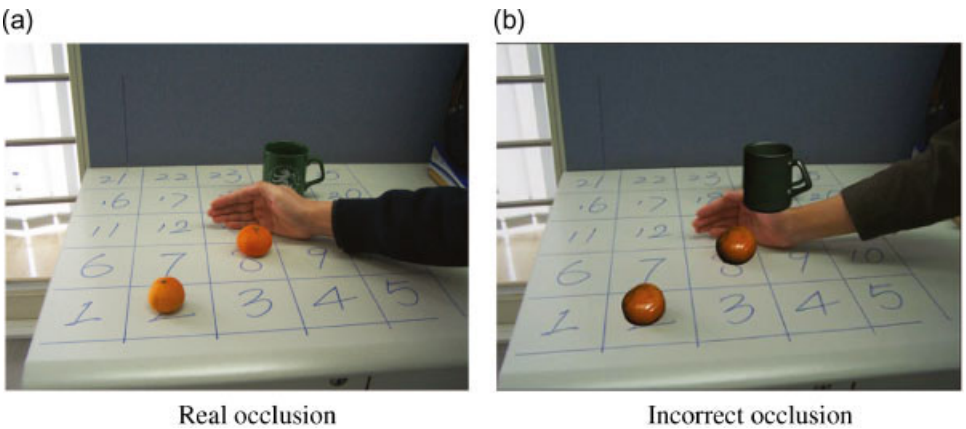


Figure 2. Examples of incorrect occlusion in VAR (a) is a real image captured by camera, (b) is a synthesized image with wrong occlusion relationship. In (b) the oranges and the cup are virtual objects. The hand is falsely occluded by the cup. Given this image, observers lose their spatial awareness not knowing where the hand is actually located.

DOI: 10.1002/cav

Here, we review several approaches of occlusion handling in AR. Fuhrmann[3] utilized 3D models of real objects (such objects actually exist in the real scene) to occlude virtual objects. The approach models real objects and uses that to occlude virtual objects. This method is limited in wide applications because it has to know the content and model the scene ahead.

Yokoya[9] proposed to use scene depth to estimated occluded objects which is similar with our approaches. They estimate the depth map using stereo cameras. To accelerate depth computing, matching region is limited only for virtual objects that will be rendered in the later stage. It is not a dense approach and requires object detection technique to track the interested objects.

Berger and Lepetit[2,10] presented a contour-based method to do occlusion handling. The approach is fast because of no request for 3D reconstruction. The difficulty is that contours varies under different views and cannot be easily detected or tracked. Therefore, their approach has to define some key frames manually, which is tedious if given a long video sequence.

Ohta[11] designed a client/server architecture to accelerate depth computing. The server can estimate scene depth fastly and the client is responsible for rendering the mixed environment. The challenge in the approach is how to synchronize the view points between clients and the server. Inconsistent scene rendering causes motion gap.

A real-time stereo system is used to assist 3D detection and tracking for various tasks in Reference [5]. Both the range and intensity information are combined to help tracking features (such as fingertip and pen) in different scenes. The limitation of the approach is that the stereo cameras are pre-calibrated and the working volume is restricted around 60 cm.

In Reference [12], a virtual annotation approach is presented by utilizing depth of field, aerial perspective, and temporal distance coding. The basic idea is combining depth and color cues to remove perception ambiguity. The approach also claims that occlusion relationship is conveyable by varying the visual attributes of the edges and/or surfaces of the entity.

In Reference [13], various scene features (such as vertical and horizontal shadow planes, color encodings of relative depth) assist user's depth perception for far field outdoor mobile AR system. By utilizing such information, the approach can correctly help user detect the 3D cursor with physical objects at various distances.

Fortin[14] introduced a metric division of the distance and provides two methods, Static Scenery and Dynamic Scenery, to handle occlusion in near and far distance, respectively. Reference [15] proposed an approach of measuring egocentric depth judgements in AR. Their experiments range from 3 to 45 m, which is the longest distance (to our best knowledge) reported in the literature. Their results show that the egocentric depth of AR objects is underestimated, but to a lesser degree than has previously been found for Virtual Reality environments.

Although several approaches have been proposed for far-field depth sensing, due to large noise and instability of features, the results are not robust. This paper restricts to handle occlusion in near-field range, approximated from 1 to 2 m. The most similar approach like ours is provided by Hayashi.[16] We both use stereo cameras to recover dense scene depth and utilize it to do occlusion handling. The difference is that we provide a more general framework to fuse depth, color, and neighbors to estimate depth. Results show this method is more robust for dynamic scenes.

Matching with Uncertainty

We first introduce an error propagation method to robustly estimate the fundamental matrix. A set of sparsely matched features computed from this fundamental matrix will guide the matching process to generate a dense depth map. By analyzing propagation of the uncertainty (probability of an image corners), our approach works well on matching pixels in two cameras. A nice feature of this approach is that it can be directly applied for dynamic stereo cameras (for example, users move together with stereo cameras) because the fundamental matrix is invariant to scene structure.

Fundamental Matrix

Given a rectified stereo pair, the epipolar constraint defines the matched pixel on a horizontal scan line which reduces the matching searching space from 2D to 1D. This constraint can be represented by a 3×3 matrix called fundamental matrix. It describes the internal relationship between camera's intrinsic parameters and its posture in a world (3D) space.

Given a set of matched pixels, the fundamental matrix maps one pixel from one view to another on the epipolar line. We illustrate their relationship in Figure 3, where e_1 and e_2 are two epipolor points.

Mathematically, this constraint can be formed as: $\mathbf{X}'\mathbf{F}\mathbf{X} = 0$. $\mathbf{X}' \leftrightarrow \mathbf{X}$ represents matched pixels. $\mathbf{F}$ is the fundamental matrix which can be realigned by columns

DOI: 10.1002/cav

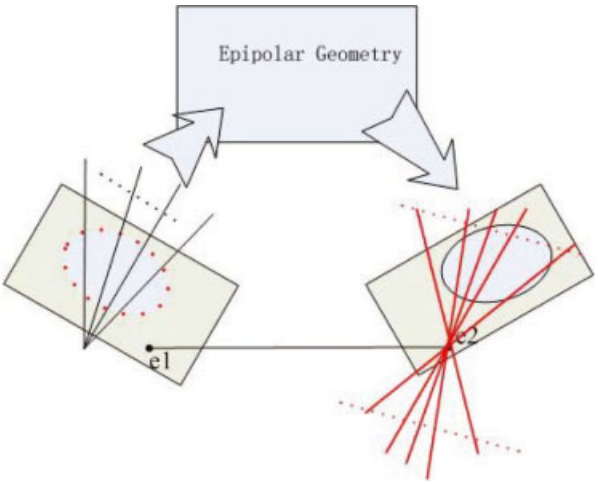


Figure 3. Stereo matching using epipolar constraint. One pixel in first view is mapped to a line in the second view by a fundamental matrix.

to be a vector with nine elements. The algebraic version of this equation can be reformulated as

$$\Phi(X_i) = \begin{bmatrix} x_i'x_i & x_i'y_i & x_i' & y_i'x_i & y_i'y_i & x_i & y_i & 1 \end{bmatrix} \quad (1)$$

By adding a noise function to matched pixels, we compute the fundamental matrix from nonlinear regression.[17] This problem can be formed as

$$f(X_i, \beta) = \Phi(X_i)\theta \quad (2)$$

where $\Phi(X_i)$ is a nonlinear function of measurements and θ is a vector of parameters (the fundamental matrix). By assuming $\|\theta\| = 1$, we follow Bogdan's[18] approach using a general eigenvalue

$$\begin{cases} s(\theta)\theta = \lambda c(\theta)\theta \\ s(\theta) = \sum_{i=1}^{n} z(x_i) C^{+}_{f|(x_i)} z(x_i)^T \\ c(\theta) = \sum_{i=1}^{n} (\eta_i \otimes I_p) C_{\Phi(x_i)} (\eta_i \otimes I_p)^T \end{cases} \quad (3)$$

where $s(\theta)$ is a weighted scatter matrix and $c(\theta)$ is a weighted covariance matrix. This equation can be solved using generalized eigenvectors. Compared with other methods, such as Normalized Total Least Square, Renormalization, and Fundamental Numerical Schemes, this approach generates more robust estimates.

Uncertainty Analysis

The basic idea of optimization matching is to incorporate uncertainty from corners. We detect image corners using Harris Corner Detection method.[19] An image corner is defined as

$$M = G(p) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \quad (4)$$

where I_x and I_y, respectively, are the gradient direction of x and y, $G(p)$ is the Gaussian kernel and k is a constant, normally set to 0.04. The probability of a pixel to be the corner is computed as $p = \det(M) - k \times \text{trace}^2(M)$. We represent the uncertainty of an image corner using the second term of the right side of Equation (4)

$$\mathbf{U} = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \quad (5)$$

Its geometrical property can be represented by Principle Component Analysis (PCA). Two eigenvectors of U represent long and short axis of an ellipse, whose minor axis shows the maximum changing of the gradient. Figure 4 gives an example.

Given a pair of matched image corners, we assume their uncertainties are independent, therefore we can formulate the uncertainty of a matched pair as a 4×4 matrix

$$\mathbf{U_2} = \begin{pmatrix} I_x^2 & I_x I_y & 0 & 0 \\ I_x I_y & I_y^2 & 0 & 0 \\ 0 & 0 & I_x'^2 & I_x' I_y' \\ 0 & 0 & I_x' I_y' & I_y'^2 \end{pmatrix} \quad (6)$$

Optimization with Uncertainty

We use a nonlinear forward error propagation method to calculate the uncertainty propagation in $\Phi(X_i)$. If $\mathbf{v}$ is a random vector in R_m with mean $\overline{\mathbf{v}}$ and covariance matrix $\sum$, and suppose that $f : R_m \to R_n$ is differentiable in the neighborhood of $\overline{\mathbf{v}}$. Then, up to a first order approximation, $f(\mathbf{v})$ is a random variable with mean $f(\overline{\mathbf{v}})$ and covariance matrix $\mathbf{J} \sum \mathbf{J^T}$, where $\mathbf{J}$ is the Jacobin matrix of f evaluated at $\overline{\mathbf{v}}$. From Equation (1) we have

$$\mathbf{J_i} = \begin{pmatrix} x_i' & 0 & 0 & y_i' & 0 & 0 & 1 & 0 & 0 \\ 0 & x_i' & 0 & y_i' & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 & 0 & 0 & 0 \\ x_i & y_i & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (7)$$

DOI: 10.1002/cav

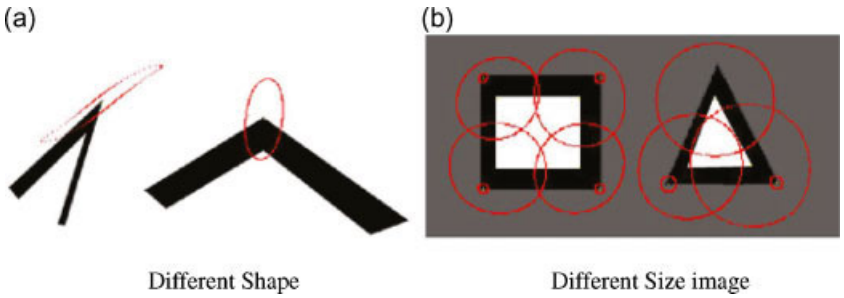


Figure 4. Geometrical explanation of uncertainty (a) shows the maximum direction changing of gradient that is consistent with the minor axis of uncertainty ellipse and (b) shows different gradients of intensity near corners with different uncertainties.

thus the covariance matrix of uncertainty is

$$C_{\Phi(X_i)} = \mathbf{J}^{\mathbf{T}} U_2 \mathbf{J} \tag{8}$$

Interested Object Recognition

To reduce the computing power of pixel matching, we limit the matching process only to the objects that we are interested in the scene. We represent objects by different features and using templates to recognize them. Dense depth are estimated for these objects and occlusion relationships are detected among them. We also propose a background subtraction algorithm by color quantization and mixed Gaussian Kernels.

A common cue for background subtraction is the color information. By computing local color histogram (which is invariant to image spatial transformation), background scene can be separated from foreground. The disadvantages of color histogram are: (1) Computing complex if the color bit is large. If color is represented by 16 bits, clustering 2^{16} colors requires more computing power. (2) Lack of spatial information. Two different images may share the same color histogram; however the actual color and gradient are different, which confuses the histogram-based separation. (3) Sensitive to image noise. The image corner's uncertainty may cause pixel in the same histogram falsely clustered in different groups. We resolve above problems by utilizing color quantization, image blocks and global difference.

Color Gaussian Kernel

By color quantization, some researchers have shown that it can efficiently reduce color bits and therefore accelerate the histogram computation.[20] In this paper, we quantize colors in UV channels into 256, and we select n_r types of most often appeared colors. n_r can be pre-determined by capturing several images in the environment and clustering all the colors after quantization.

With quantization, we describe the pixel similarity using a Color Gaussian Kernel (CGK)

$$G_j^C = \frac{1}{2\pi\sigma_c} \exp\left(-\frac{d_u^2 + d_v^2}{2\sigma_c^2}\right) \tag{9}$$

if u_k, v_k is the current pixel, and u_j, v_j are the neighbor pixels, $d_u = C(u_j) - C(u_k)$ and $d_v = C(v_j) - C(v_k)$ are the difference of colors, σ_c represents the standard deviation of color Gaussian kernel. It controls the shape of the kernel function. We set σ_c to 2.

Spatial Gaussian Kernels

CGK computes the color difference of two pixels. In a dynamic background or considering the camera is dithering, the position of pixels with the same color will change locally. We include the geometric relationship of the pixels (the Spatial Gaussian Kernels (SGK)) to compensate CGK to jointly indicate the similarity of two pixels. Besides, we incorporate multi-scale technique to preserve long range relationship considering CGK and SGK are both local kernels. We separate images into $n \times n$ blocks, each block is overlapped with its neighbors. Thus, each pixel belongs to $\frac{n}{x}^2$ blocks. In such blocks, we calculate the weight of each pixel as its distance to the center pixel by a Gaussian distribution

$$G_j^S = \frac{1}{2\pi\sigma_s} \exp\left(-\frac{d_u^2 + d_v^2}{2\sigma_s^2}\right) \tag{10}$$

if u_k, v_k is the current pixel, and u_j, v_j are the neighbor pixels; $d_u = u_k - v_j$ and $d_v = u_k - v_j$ are the difference

DOI: 10.1002/cav

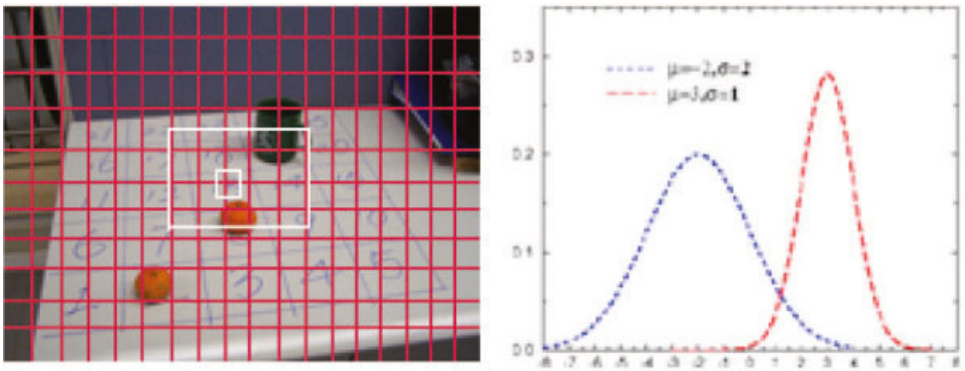


Figure 5. Color and spatial Gaussian kernels in multi-scales. The pixel which has large color difference and is close to the square center contributes the highest value of two Gaussian distributions.

of geometric relationship, σ_s is the standard deviation of the spatial Gaussian kernel and also controls the shape of the kernel function, which is set to 1.2.

By fusing CGK with SGK, we are able to describe histograms using both color and spatial information

$$h_j = \sum_{k;k\in N(j)} G_j^s G_j^c \tag{11}$$

where $N(j)$ indicates all neighbors of j.

In practical, we compute the probability of a pixel belongs to background or foreground using Bhattacharyya Distance

$$P_j = \sum_{k;k\in N(j)} \sqrt{h_k^b h_k^f} \tag{12}$$

where h_j^b and h_j^f, are the histograms of background and foreground from initial frames.

We can see the probability of a pixel belongs to background is higher if its distance to foreground pixel is large and it is close to neighbor's background center. Figure 5 shows an example of utilizing CGK and SGK in multi-scale approach.

If the foreground object has similar colors with background, local information using CGK and SGK is not adequate to recognize foreground object, such as the cup in Figure 6 cannot be correctly detected.

We introduce a global term to compensate color loss in quantization. Basically, a weight is assigned to indicate the confidence of the global difference to the dynamic movement caused by background moving or camera dithering. The probability of being a foreground pixel is calculated using

$$w = \frac{(2^n - \Delta(1 - w_d))}{2^n} \tag{13}$$

where n is the color bit number; Δ indicates the color difference, and w_d is the dynamic weight value showing the degree of background moving or camera shaking. When the degree is large, it is close to 1 and

Figure 6. Local background subtraction. The cup cannot be extracted by only using local method. The left image is the background image. Most selected colors are quantized in color bins with value of (19,20) or (20,20). The center image is the foreground image, where the color of the cup is also quantized into a bin valued (19,20), thus the cup is falsely regarded as the background and cannot be subtracted in right image.

DOI: 10.1002/cav

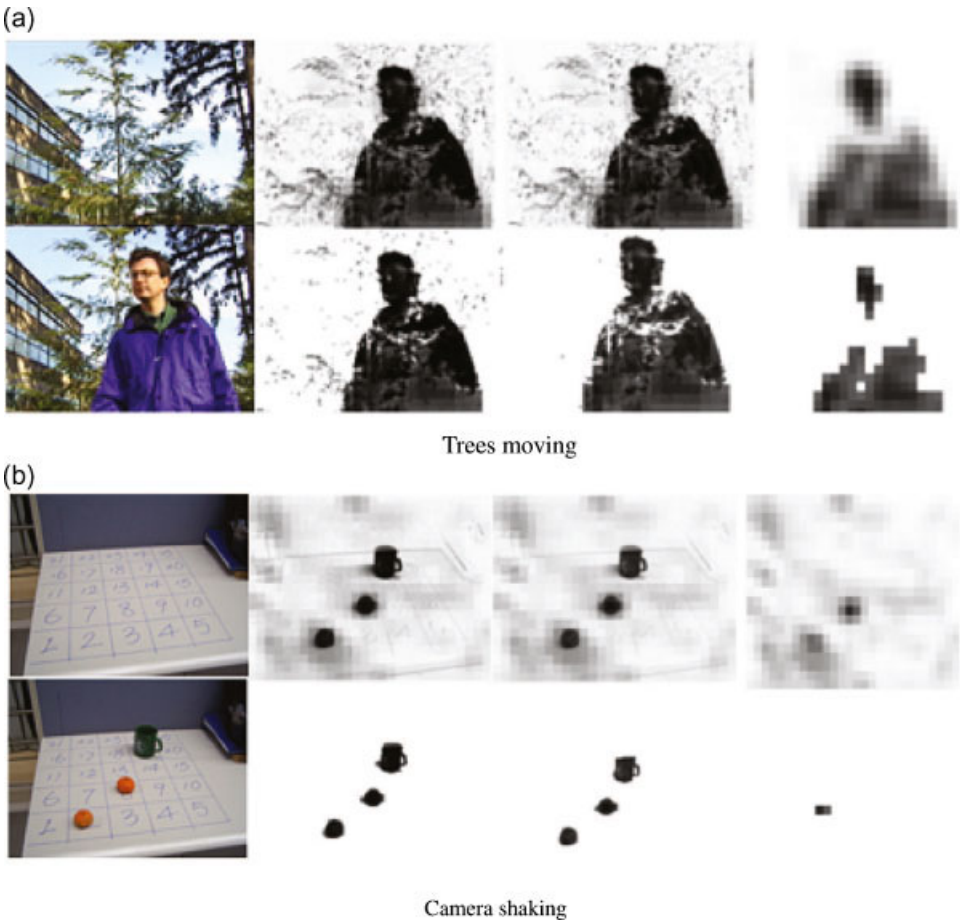


Figure 7. Experiment results of object recognition. From left to right: reference image, threshold = 0, threshold = 0.4 and threshold = 1. (a) Tree moving. (b) Camera shaking.

the global difference is not confident. Otherwise, it is close to 0.

Given this global term, the equation of calculating pixel's probability of being a background can be rewritten as

$$P' = w^{\lambda} \times P \tag{14}$$

where λ is used to reduce the sensitivity of global difference to noise. We set σ_c to 5. Figure 7 shows two examples of background subtraction results.

Occlusion Handling

In this section, we introduce our approach of occlusion handling using both depth and color information. To accelerate depth computation, we simply estimate depth of interested objects. This help to locate its disparity in the another view quickly. By incorporating shape priors, our approach first recognizes interested objects and then using depth information to detect their occlusion relationships.

Dense Depth Generation

We compute the pixel dissimilarity for stereo cameras first using Birthfield method[21] and optimized it with color aggregation process.[22] To weight on both smooth and discontinuous regions, an appropriate window should be selected during aggregation. In a sense, the window should be large enough to cover sufficient area in textureless regions, while small enough to avoid crossing regions with depth discontinuities. In our implementation, we incorporate color weighted aggregation to obtain this reliable correlation volume, which is based on the adaptive weight aggregation strategy.

Comp. Anim. Virtual Worlds 2010; **21**: 509–521
DOI: 10.1002/cav

We compute the weights using both color and spatial proximity to the central pixel of the support window. The color difference in this support window is expressed in *RGB* color space as

$$\Delta C(x, y) = \sum_{c \in R,G,B} |I_c(x) - I_c(y)| \tag{15}$$

where I_c is the intensity of the color channel c. The weight of pixel x in the support window of y (or *vice versa*) is then determined using both its color and spatial difference as

$$w_{xy} = e^{-\left(\frac{\Delta C(x,y)}{\gamma_C} + \frac{\Delta G_{xy}}{\gamma_G}\right)} \tag{16}$$

where ΔG_{xy} is the geometric distance from pixel x to y in the 2D image grid. γ_C and γ_G are constant parameters controlling the shape of the weighting function, which are determined empirically.

Pixel similarity is then an aggregation with the soft windows defined by the weights as

$$f_s(x_l, x_r) = \frac{\sum_{y_l, y_r \in W(x_l) \times W(x_r)} w_{x_l y_l} w_{x_r y_r} d(y_l, y_r)}{\sum_{y_l, y_r \in W(x_l) \times W(x_r)} w_{x_l y_l} w_{x_r y_r}} \tag{17}$$

where $W(x)$ is the support window around x; $d(y_l, y_r)$ represents the pixel dissimilarity using Birchfield and Tomasi's approach;[21] x_l and y_l are pixels in the left view; x_r and y_r are pixels in the right view.

Stereo Triangulation

Given a vector of matched pixels (matched pixels are calculated using methods introduced in Section Dense Depth Generation), we compute the pixel's depth by triangulating matched pixels into 3D space. The basic idea of stereo triangulation is first to compute the two 3D lines by back-projection the matched pixels given pre-calibrated projection matrix, and then compute the 3D intersection point of two lines. Algorithm 1 shows an example of calculating this 3D intersection by matrix computation.

Algorithm 1 Stereo Triangulation

P1,P2 are projective matrix. x1,x2 are matched pixels

for $i = 0$ to *numpairs* **do**
$A = zeros(4, 4)$
$A(1, :) = x1(1, i) * P1(3, :) - P1(1, :)$
$A(2, :) = x1(2, i) * P1(3, :) - P1(2, :)$
$A(3, :) = x2(1, i) * P1(3, :) - P2(1, :)$
$A(4, :) = x2(2, i) * P1(3, :) - P2(2, :)$
$M = A(1 : 4, 1 : 3)$
$b = -A(1 : 4, 4)$
$X(:, k) = inv(M'M)M' * b$
end for

Clustering by Fusing Depth and Color

By fusing depth and color information, we cluster objects into several categories by

$$p_{i,k} = w_d d_i + w_c c_i \tag{18}$$

where $p_{i,k}$ is the probability of current pixel which belongs to category k. w_d and w_c are coefficients for depth and color information. Combining depth and color can robustly cluster objects into correct categories, particularly when color is noise (object boundaries) or sparse (textureless regions).

One nice feature of our formulation is that is can be easily incorporated with other priors, such as shapes. This, in a sense, extends traditional template matching algorithms with rich information (depth). In our experiment, we use simple shapes like circle and cylinder to recognize interested objects (oranges and cup) in the scene. If the size of template patch T is a $j \times k$, the image (I) size is $m \times n$ and the center of p is (s, t), we compute the probability of template matching results by

$$p(s, t) = \frac{\sum_{(x,y) \in T_i} (I(x, y) - \bar{I})(T(x - s, y - t) - \bar{w})}{\sum_{(x,y) \in T_i} ((I(x, y) - \bar{I})^2 (T(x - s, y - t) - \bar{w})^2)^{1/2}} \tag{19}$$

We experimented two methods: hierarchical cluster and K-means to do clustering. The former method can be divided into Agglomerative and Divisive. In agglomerative clustering, each object is initially placed in its own group. The two *closest* groups are combined into a single group. In divisive clustering, all objects are initially placed into a single group. The two objects that are in the same group but are *farthest* away are used as seed points for two groups. All objects in this group are placed into the new group that has the closest seed. This procedure continues until a threshold distance is reached. K-means method uses the actual observations of objects or individuals in data, and not just their proximities. These differences often mean that K-means is more suitable for clustering large amounts of data.

DOI: 10.1002/cav

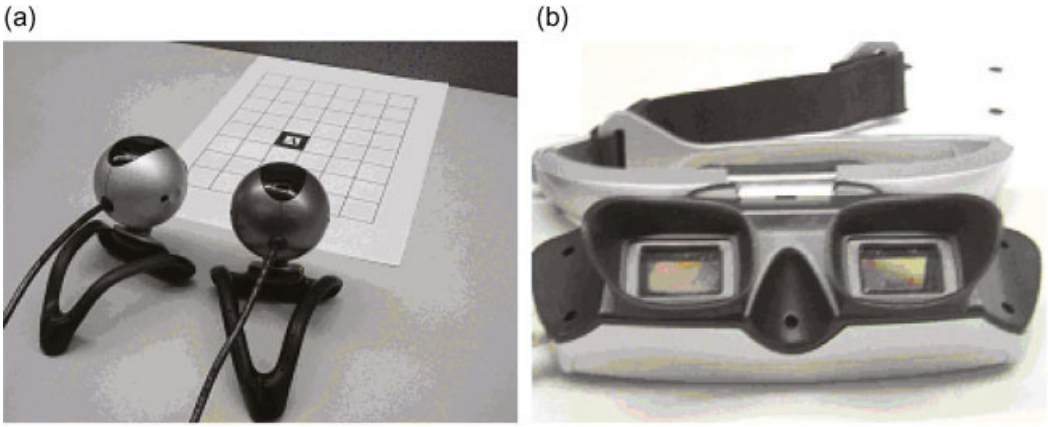


Figure 8. Two cameras and the HMD used in our VARs. (a) Two logitech web cameras. (b) Head mounted display.

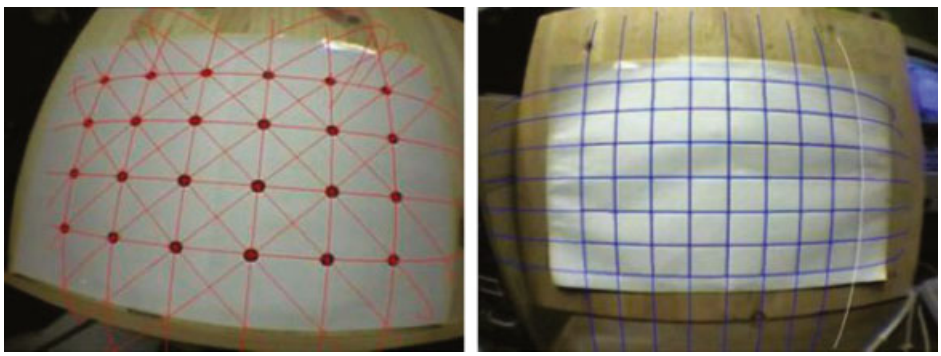


Figure 9. Example of radial distortion from ARToolkit Software Package [24].

Experiment Results

We simulate a VAR system by two video cameras. A head mounted display (HMD) displays merged scenes. In experiments, user wears the HMD and can move in a working volume around $2 \times 2 \times 2$ m. Figure 8 shows a screen capture of such devices.

Calibration

We calibrate two cameras using traditional camera calibration method.[23] The basic idea of this method is to define a world space for two cameras and estimate the geometric transformation matrix from one to the other. It captures several pattern box images where image corners can be easily detected, and then users manually select all matched pixels for each image pair. These matched pixels are triangulated to 3*D* space using Algorithm 1. One of the point is selected as the origin. The projective matrixes are computed using a least square approach. One noticeable artifact of web cameras are the radial distortion (see Figure 9). Using Zhang's method we can efficiently remove this artifact by adjusting the radial distortion parameters.

Epipolar Geometry

Above calibration method is designed for a calibrated volume, which means it only works when cameras are fixed. For unmounted cameras, we compute the fundamental matrix using the method introduced in Section Matching with Uncertainty. We test our approach from 4 test images and compare the results with Normalized Least Square method. We also compare the uncertainty matrix using identity matrix and the covariance matrix. Figure 10 shows an example of calculated epipolar lines, and Table I shows the numerical comparison of different methods. We can see the accuracy of pixel matching is improved around 7 times in average with uncertainty propagation.

DOI: 10.1002/cav

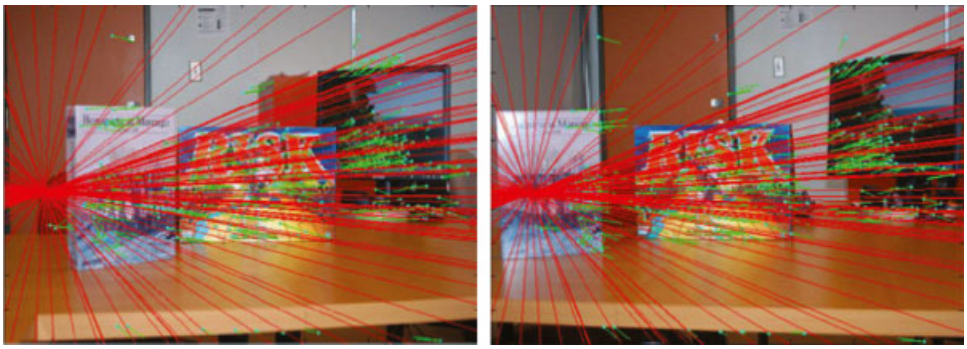


Figure 10. Example of epipolar lines.

Scene	Number of matches	NTLS	$C = I$	$C =$ Cov
Box	34	0.3936	0.0379	0.0183
	45	0.6819	0.0382	0.0115
	67	0.4533	0.0218	0.0068
	133	0.6507	0.0136	0.0015
	256	0.5670	0.0022	4.6326e-004
Orange	12	2.8308	2.0855	2.0849
	16	1.0694	0.6008	0.5885
	23	1.2897	0.0103	0.0093
	46	1.5675	0.0074	0.0074
	90	3.4614	0.3863	0.1308
Fruits	108	133.8546	19.1246	18.3812
	144	146.1588	10.2616	8.6902
	216	142.5185	8.6392	2.8756
	432	162.3101	2.5625	0.8746
	864	177.9800	1.8746	0.3863
Radio	16	1.2564	0.6736	0.5936
	21	0.4348	0.3981	0.1908
	32	0.7628	0.9310	0.1312
	63	0.7419	0.0558	0.0364
	126	0.7162	0.0221	0.0018

Table 1. Comparison of estimating epipolar geometry

Framework of Occlusion Handling

We present the occlusion handling process in Figure 11 according to previous introduction. Altogether, four modules are divided and each is indexed by dash line in different color. The input module reads 2 streams of captured images and they are used in the initialization module to calculate the Epipolar constrain or matching pixels. The depth estimation module computes the depth for each interested object, and finally the output module concludes the occlusion relationship and renders the synthesized image into HMD.

In Figure 12, we show an example of estimated depth. Part (a) shows the depth map of real objects. Part (b) shows the depth map of observer's hand. They are rendered in different colors. Part (c) shows the result of occlusion handling. We can see parts of the cup are occluded by user's hand, and parts of user's hand are occluded by the second orange. Part (d) shows the result of correct occlusion registration.

DOI: 10.1002/cav

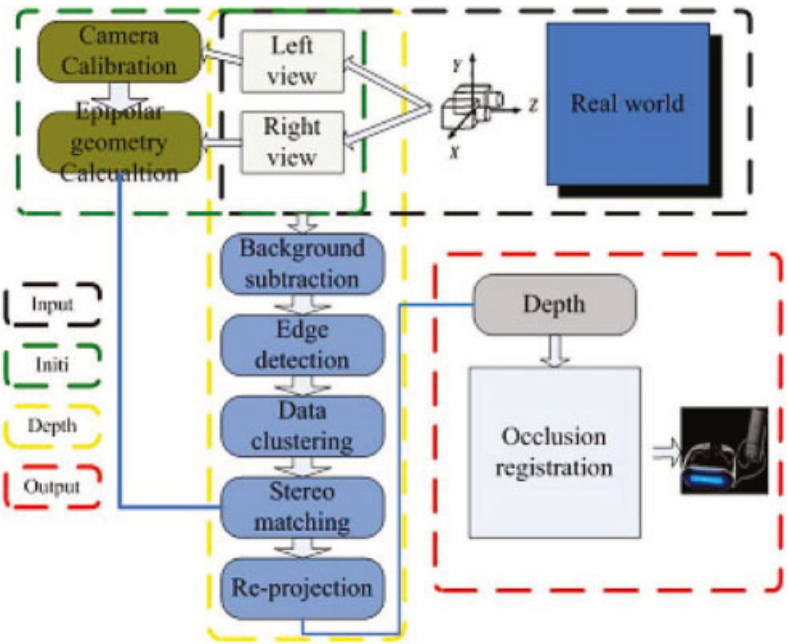


Figure 11. Algorithm structure of occlusion registration.

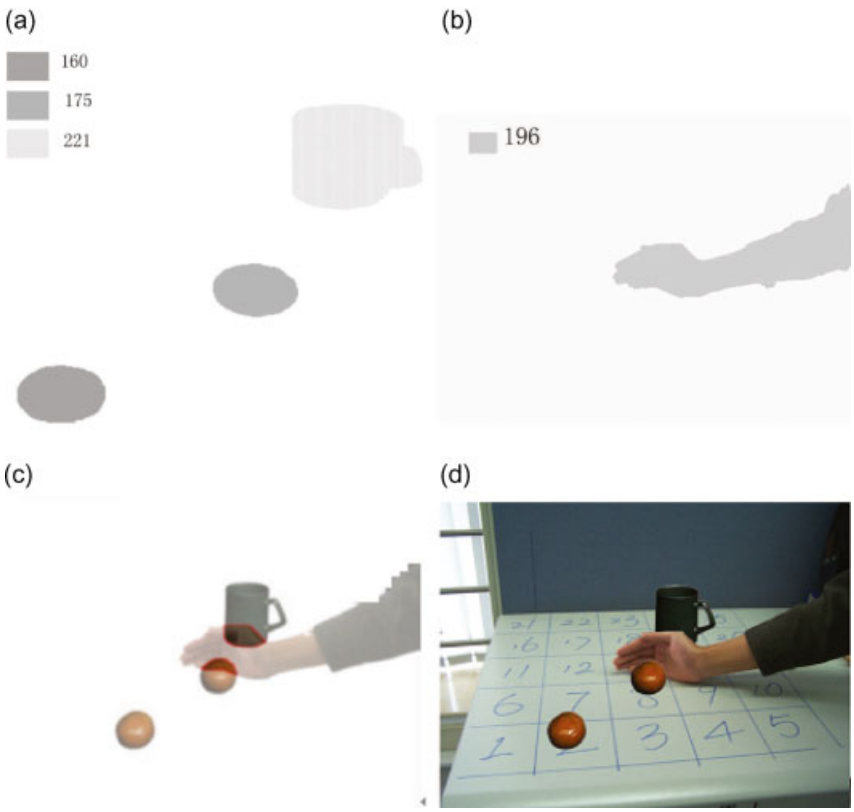


Figure 12. Occlusion registration. (a) Objects depth. (b) Hand depth. (c) Occlusion relationship. (d) Corrected synthesized occlusion.

Comp. Anim. Virtual Worlds 2010; **21**: 509–521
DOI: 10.1002/cav

Conclusion

In this paper, we provide an approach to resolve occlusion handling in VAR by depth information. The depth is estimated from matched pixels from stereo cameras. We fuse color and depth to extract interested objects from scenes. By only triangulating pixels on boundaries of interested objects, we can calculate the correct occlusion relationship fast. Results show that our approach is efficient and robust. In future, we envision extending current work on deformable objects where the objects' boundaries are more complex and occlusion handling is more challenging.

ACKNOWLEDGEMENTS

This research work is co-supported by NSFC project on Virtual Olympic Museum (grant no:60533080), 863 project on Digital Media Authoring Platform (grant no:2006AA01Z335), and Open Project of State Key Lab of VR, Beihang University.

References

1. Bajura M, Henry F, Ohbuchi R. Merging virtual reality with the real world: seenig ultrasound imagery within the patient. *Computers and Graphics* 1992; **26**(2): 203–210.
2. Berger M-O. Resolving occlusion in augmented reality: a contour based approach without 3d reconstruction. In *IEEE Computer Vision and Pattern Recognition*, 1997; 91–96.
3. Fuhrmann A, Hesina G, Faure F, Gervautz M. Occlusion in collaborative augmented environments. *Computers and Graphics* 1998; **23**(6): 255–261.
4. Pang Y, Yuan M-L, Nee A-Y-C, Ong S-K, Toumi K-Y. A markerless registration method for augmented reality based on affine properties. In *Proceedings of the 7th Australasian User Interface*, 2006; 25–32.
5. Gordon G, Billinghurst M, Bell M, *et al*. The use of dense stereo range data in augmented reality. In *Proceedings of the Mixed and Augmented Reality*, 2002, 14–23.
6. Vallerand S, Kanbara M, Yoloya N. Binocular vision-based augmented reality system with an increased registration depth using dynamic correction of feature positions. In *IEEE Virtual Reality Conference*, 2003; 22–26.
7. Howard IP, Rogers BJ. *Depth Perception in Seeing in Depth*. Oxford University: USA, 1995.
8. Bimber O, Raskar R. *Spatial Augmented Reality Merging Real and Virtual Worlds*. A. K. Peters: Wellesley, MA, 2005.
9. Yokoya N, Takemura H, Okuma T, Kanbara M. Stereo vision based video see-through mixed reality. In *Proceedings of the Mixed and Augmented Reality*, 1999; 85–94.
10. Lepetit V, Berger MO. A semi-automatic method for resolving occlusion in augmented reality. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000; 2225–2230.
11. Ohta Y, Sugaya Y, Igarashi H, Ohtsuki T, Taguchi K. Share-z: client/server depth sensing for see-through head mounted displays. In *Proceedings of the International Symposia on Mixed Reality*, 2001; 64–72.
12. Uratani K, Machida T, Kiyokawa K, Takemura H. A study of depth visualization techniques for virtual annotations in augmented reality. In *Proceedings of IEEE Virtual Reality Conference*, 2005; 295–296.
13. Wither J, Hollerer T. Pictorial depth cues for outdoor augmented reality. In *Proceedings of 9th IEEE International Symposium on Wearable Computers*, 2005; 92–99.
14. Fortin PA, Hebert P. Handling occlusions in real-time augmented reality: dealing with movable real and virtual objects. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, 2006; Q7 54.
15. Swan JE II, Jones A, Kolstad E, Livingston MA, Smallman HS. Egocentric depth judgments in optical, see-through augmented reality. *IEEE Transactions on Visualization and Computer Graphics* 2007; **13**: 429–442.
16. Hayashi K, Kato HK, Nishida S. Occlusion detection of real objects using contour based stereo matching. In *Proceedings of the International Conference on Augmented Tele-existence*, 2005; 180–186.
17. Hartley R, Zisserman A. *Multiple View Geometry in Computer Vision*. Cambridge University: Cambridge, UK, 2003.
18. Bogdan M, Peter M. Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2006; **28**(10): 1537–1552.
19. Harris C, Stephens MJ. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, 1988, 147–152.
20. Noriega P, Bascle B, Bernier O. Local kernel color histograms for background subtration. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2006; 67–74.
21. Birchfield S, Tomasi C. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1998; 401–406.
22. Yoon KJ, Kweon IS. Locally adaptive support-weight approach for visual correspondence search. In *Proceedings of IEEE Computer Society Computer Vision and Pattern Recognition*, 2005; 924–931.
23. Zhang Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2000; **22**: 1330 –1334.
24. Camera Calibration in ARtoolKit. Available online:http//www.hitt.washington.edu/artoolkit/documentation/usercalibration.htm.Accessed on 2009.

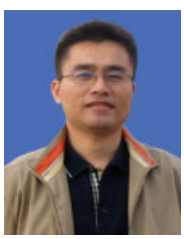
DOI: 10.1002/cav

Authors' biographies:

Jiejie Zhu was born in 1979 in Zhejiang Province, China, and he received his Bachelor Degree and Master Degree from the Computer Science Department in 2001 and 2004 from Zhejiang Normal University and Hangzhou Dianzi University, respectively. He received the Ph.D. degree in computer science from State Key Lab of CAD&CG, Zhejiang University, China, in 2007. He is currently a postdoctoral researcher in Computer Science Department, University of Central Florida. Before that, he was a postdoctoral researcher in Computer Science Department, University of Kentucky. His research interests included machine learning, computer vision, augmented reality, and human computer interaction.

Zhigeng Pan was born in 1965 in Jaingsu Province, and he received his Bachelor Degree and Master Degree from the Computer Science Department in 1987 and 1990 from Nanjing University respectively and Ph.D. Degree in 1993 from Zhejiang University. Since 1993, he has been working at the State Key Lab of CAD&CG, Zhejiang University. He has published more than 100 papers on international journals, national journals and international conferences. His research interests include distributed graphics, virtual reality, multimedia, digital entertainment. Professor Pan is a member of SIGGRAPH, Eurographics, IEEE, a senior member of the China Image and Graphics Association. He is on the director board of the International Society of VSMM (Virtual System and Multimedia), a member of IFIP Technical Committee on Entertainment Computing (acting as representative from China). Currently, he is the Editor-in-Chief of The International Journal of Virtual Reality, Co-EiC of LNCS Transactions on Edutainment. He is on the editorial board of International Journal of Image and Graphics, International Journal of CAD/CAM, Journal of Image and Graphics, Journal of CAD/CG, et al. He is the Director of the Virtual Reality Committee, China Society of Image and Graphics; He is on the Steering Committee member of VRCAI series conference. He is the organizing committee chair of VRAI'2002, ICIG'2004. He is the program co-chair of EGMM'2004 (Eurographics workshop on Multimedia), the program co-chair of Edutainment'2006, conference co-chair of ICAT'2006 and Edutainment'2008.

Sun Chao was born in April, 1980, and now he is a Ph.D. candidate in State Key Lab. of CAD&CG, Zhejiang University, P.R. China, specializing in computer science. His research interest is concentrated on computer vision, computer graphics, human computer interaction, and Augmented Reality, especially in multiple view geometry and applied mathematics associated in above fields.



Wenzhi Chen is an associate professor in the Computer Science Department of Zhejiang University. He received his Bachelor degree, Master's degree Ph.D. from the Computer Department, Zhejiang University in 1992, 1999, and 2005. His research interests include: virtual reality/virtual environment, visualization and image processing, computer architecture, operating System.

DOI: 10.1002/cav