View Synthesis of Scenes with Man-made Objects Using Uncalibrated Cameras

Geetika Sharma
Department of Mathematics
IIT, Delhi
geetikas@gmail.com

Santanu Chaudhury
Department of Electrical Engineering
IIT, Delhi
santanuc@ee.iitd.ernet.in

J.B. Srivastava
Department of Mathematics
IIT, Delhi
jbsrivas@maths.iitd.ernet.in

Abstract

We propose a scheme for view synthesis of scenes containing man-made objects from images taken by arbitrary, uncalibrated cameras. Under the assumption of availability of the correspondence of three vanishing points,in general position, our scheme computes z-buffer values that can be used for handling occlusions in the synthesized view. This requires the computation of the infinite homography. We also present an alternate formulation of the technique which works with the same assumptions but does not require infinite homography computation. We present experimental results to establish the validity of both formulations.

1. Introduction

In this paper we have addressed the problem of synthesizing new views of a scene using multiple images taken by uncalibrated cameras. We make no assumptions on the motion or internal parameters of the cameras. Infact, our scheme can be used to synthesize novel views using frames extracted from a motion picture and we provide examples of the same. Our technique for view synthesis requires that the correspondence of three vanishing points, in general position, be available in the given views. The knowledge of the vanishing points is used to compute the infinite homography between two given views. We also give an alternate formulation that does not require computation of the infinite homography and is motivated by [3]. In both formulations we compute a z-buffer value for each corresponding point that can be used to resolve changes in visibility in the new view correctly. We have proposed a technique for view synthesis under the simpler assumption of translating cameras in [11].

In [3], three vanishing points in general position are used to setup a coordinate system in the world. Two points in a single image are said to be *corresponding points* if one is on one of the coordinate planes and the other is on the line through the first point parallel to the coordinate axis perpendicular to the plane. Reconstruction from a single image can be done but only for those points for which atleast one of the *corresponding points* is known. Our technique uses two or more views to reconstruct each point visible in atleast two of the given views. Also, [3] uses two views to make certain measurements, for example, the affine distance of the camera centres to a chosen world plane. However, they require the ratio of the distances of two reference points from the world plane to be known. We require only that the correspondence of three vanishing points be given.

Other work in view synthesis includes [1] which uses the trifocal tensor between three views taken by uncalibrated cameras. However, they cannot handle occlusions in the new view. A novel representation consisting of multiple depth and intensity levels for new view synthesis is proposed in [2], however, it requires calibrated cameras. In [4], the constraints imposed by weak-perspective and paraperspective cameras are used for view synthesis while we work with perspective cameras. In [6], [9] and [12], a set of views of a scene taken by uncalibrated cameras are used to reconstruct the scene in Projective Grid Space defined by the epipolar geometry between two chosen basis cameras. This reconstruction provides a volumetric projective model and new views are synthesized using interpolation. However, changes in visibility cannot be resolved. In [7], a method for obtaining a quasi-dense disparity map and a novel representation of a pair of images taken by uncalibrated cameras is proposed. They handle occlusions by rendering new views in a heuristically determined order using disparity values. Computing disparity from image information alone requires rectified images which correspond to a translation of the camera in the direction of the u axis of the image. Our technique can handle occlusions using z-buffer values and arbitrary orientation and location of the input cameras. View Morphing, [10], also works with uncalibrated cameras. They first rectify and then interpolate the given views to generate a new view. A projective transformation of the interpolated new view changes the direction of gaze. However, they can synthesize novel views only along the line joining the centres of projection of the given views and use disparity to handle occlusions.

In section 2 we describe our scheme based on the infinite homography and give the alternate formulation in section 3. We conclude in section 4.

2. Synthesis Scheme Using H_∞

We begin with a description of the technique using two input views.

2.1. Synthesis from Two Views

Let $P = (X, Y, Z)^T$ be a point in the scene and $p^i = (u^i, v^i, 1)^T$ its image in the i^{th} view, $i = 1, 2$. The projection equations for the two views I_1 and I_2 are given by ([5])

$$\lambda^1 p^1 = K_1(R_1 P + t_1) \quad (1)$$
$$\lambda^2 p^2 = K_2(R_2 P + t_2) \quad (2)$$

where,

$$K_i = \begin{pmatrix} f^i s_u^i & 0 & x_0^i \\ 0 & f^i s_v^i & y_0^i \\ 0 & 0 & 1 \end{pmatrix}$$

$i = 1, 2$, are the matrices of internal parameters, R_i are the rotation matrices and $t_i = (x_i, y_i, z_i)^T$, $i = 1, 2$, are the translation vectors. Note that λ^i is the depth of the point P in the coordinate system of the ith camera. We assume that $R_i \neq I$ and $t_i \neq (0, 0, 0)^T$, $i = 1, 2$, so that neither camera is aligned with the world coordinate system. This allows us to further assume that the origin of the world coordinate system is visible in the given views. So, if we have n point correspondences given between I_1 and I_2, without any loss of generality, we may choose one of them as the origin of the world coordinate system. Let p_0^i denote its image in the ith view. Then, $\lambda_0^i p_0^i = K_i t_i$, $i = 1, 2$.

Eliminating P from (1) and (2), we get

$$\begin{aligned} \lambda^2 p^2 &= K_2 R_2 R_1^{-1} K_1^{-1}(\lambda^1 p^1 - \lambda_0^1 p_0^1) + \lambda_0^2 p_0^2 \\ &= H_\infty^{12}(\lambda^1 p^1 - \lambda_0^1 p_0^1) + \lambda_0^2 p_0^2 \quad (3) \end{aligned}$$

where H_∞^{12} is infinite homography from I_1 to I_2. If H_∞^{12} is known, the only unknowns in (3) are λ_0^1, λ_0^2 which are fixed and λ^1, λ^2 for each point correspondence. Since the equations are linear, we can solve for all the unknowns, given $n \geq 2$ point correspondences.

Note that in (3) the camera parameters cannot be eliminated and are replaced by the infinite homography H_∞^{12}, which models them upto an unknown affine transformation. Computing the infinite homography amounts to an affine reconstruction of the scene and the λ's that we compute are the depths for this reconstruction which is an affine transformation of the true Euclidean scene. Our technique for synthesizing new frames then produces correct perspective views of the affinely transformed scene. Thus, when compared with the given views, the synthesized views suffer from affine distortion which may be removed using metric rectification techniques [8]. Note, however, that we do not do an explicit reconstruction of the scene. We only compute λ's which act as z-buffer values required for occlusion handling.

2.1.1 Synthesis of Virtual Views

Suppose I_1 is translated to obtain the new view I_s. Then, I_s and I_1 have the same orientation and internal parameters but different positions, i.e., the matrix of camera internals for I_s is K_1 and the rotation matrix is R_1. Then, the projection equation for the novel view I_s is

$$\lambda^s p^s = K_1(R_1 P + t_1 + t_s)$$

Using (1), we get

$$\lambda^s p^s = \lambda^1 p^1 + K_1 t_s \quad (4)$$
$$\lambda_0^s p_0^s = \lambda_0^1 p_0^1 + K_1 t_s \quad (5)$$

Equating the last coordinate on both sides and rearranging, we get

$$z_s = \lambda_0^s - \lambda_0^1$$

Since, λ_0^1 is fixed by the first view, different choices of λ_0^s correspond to different z translations of the camera. So, the translation of the virtual camera in the z direction may be chosen by giving a value to λ_0^s. Now, equating the first coordinate on both sides and rearranging, we get

$$x_s = \frac{\lambda_0^s u_0^s - \lambda_0^1 u_0^1}{f^1 s_u^1}$$

Again, since λ_0^1, u_0^1, f^1 and s_u^1 are fixed by the first view and λ_0^s has been fixed in the previous step, different values of u_0^s correspond to different translations x_s in the x direction. Thus, the x translation of the virtual camera may be chosen, interactively, by giving a value to u_0^s. Similarly, the y translation of the virtual camera may be chosen by giving a value to v_0^s. It follows that a choice of p_0^s and λ_0^s fixes the translation of the virtual camera and specifies the new view. Substituting $K_1 t_s$ from (5) in (4), we get

$$\lambda^s p^s = \lambda^1 p^1 + \lambda_0^s p_0^s - \lambda_0^1 p_0^1 \quad (6)$$

The positions and z-buffer values λ^s of point correspondences in the new view can be obtained from the above equation. Since we do not assume dense correspondences, the new view and the given views are triangulated using corresponding points as vertices. Triangles in the new view are then texture mapped by combining textures from corresponding triangles in the given views. The computed λ^s's act as z-buffer values for corresponding points and they can be used to handle occlusions while rendering the new view. Thus, given n point correspondences the algorithm for synthesis of new views is as follows:

1. Setup a system of equations using (3) and compute λ^i's, $i = 1, 2$.

2. Specify a new view by giving values to u_0^s, v_0^s and λ_0^s.

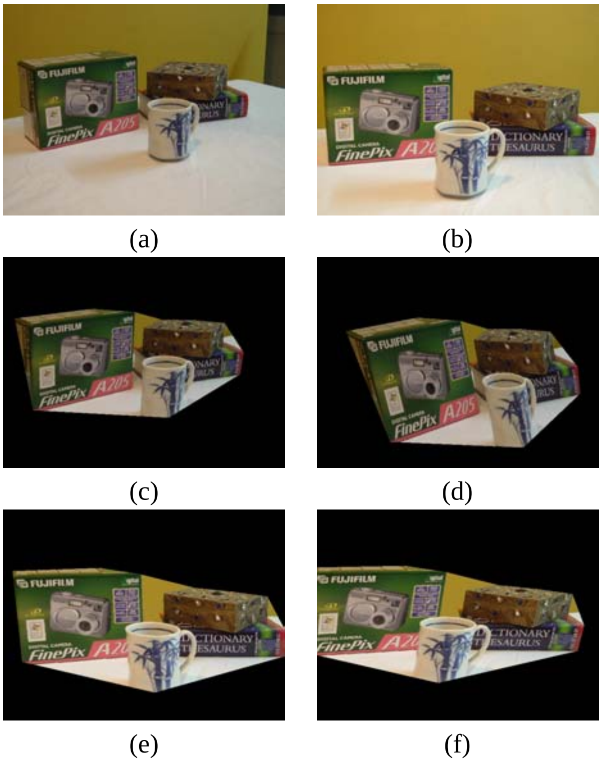


Figure 1: (a) and (b) are the input views (c) and (d) are the synthesized views parallel to (a) and (e) and (f) are synthesized views parallel to (b).

3. Render the position of corresponding points in the new view using (6). Resolve any visibility issues using λ^s's as z-buffer values.

4. Triangulate the new view using the rendered points as vertices and texture map the triangles from the given views.

Figure 1 shows the results for this section. (a) and (b) are the input views. (c)-(f) are the synthesized views with (c) and (d) parallel to (a) and (e) and (f) parallel to (b). Observe the affine distortion in the synthesized views.

2.1.2 Characterisation of View points

We now give a charaterisation of those viewpoints for which it is guaranteed that all points within the fields of view of the given cameras are also in the field of view of the virtual camera. This is required because it is possible that the chosen translation of the virtual camera is so large that none of the corresponding points are within its field of view. We assume that the motion of the camera is of translation in the $x-z$ plane and rotation about the y axis.

Without loss of generality, we may assume that the synthesized view is parallel to the first view. The characterisation of the viewpoints can be done in the coordinate system of the first camera since the new view is specified with respect to the position of the first camera. The first camera is then given by $K_1[I\,|\,0]$ and the second camera by $K_2[R_1\,|\,t_1]$. Also, we assume that, R_1, the rotation between the second and first cameras is

$$R_1 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

and that $t_1 = (x_1, 0, z_1)^T$ is the translation of C_2.

The view-volume of a camera is defined by four planes which intersect at the centre of projection and pass through the boundaries of the image plane. In order to ensure that all points in the fields of view of the given views are also in the FOV of the novel view, we will intersect the view-volumes of the given cameras and require that the view-volume of the new camera contain the intersection. For the first view, let planes defining the view volume be π_1, π_2, π_3 and π_4. Let $a^1 \le u^1 \le b^1$ and $c^1 \le v^1 \le d^1$, be the bounds for the first image plane and $a^2 \le u^2 \le b^2$ and $c^2 \le v^2 \le d^2$, be the bounds for the second image plane. Then, the equations of these planes are

$$\pi_1 : f^1X - a_1Z = 0\ ,\ \pi_2 : f^1Y - d_1Z = 0$$
$$\pi_3 : f^1X - b_1Z = 0\ ,\ \pi_4 : f^1Y - c_1Z = 0$$

Let N_i denote the normal of π_i, $i = 1,2,3,4$, that points outside the view-volume. Let $P = (X, Y, Z)^T$ be any point inside the view-volume. Then, the dot product of N_i, $i = 1,2,3,4$, with the vector joining any point on the plane π_i and P is less than zero. Choosing the centre of projection as the point on the plane, P will lie inside the view volume if,

$$\frac{a^1}{f^1}Z < X < \frac{b^1}{f^1}Z \tag{7}$$
$$\frac{c^1}{f^1}Z < Y < \frac{d^1}{f^1}Z \tag{8}$$

Similarly, if π'_1, π'_2, π'_3 and π'_4 are the planes defining the second view volume, then
$\pi'_1 : (f^2\cos\theta + a^2\sin\theta)X + (f^2\sin\theta - a^2\cos\theta)Z + f^2x_1 - a^2z_1 = 0$
$\pi'_2 : d^2\sin\theta X + f^2Y - d^2\cos\theta Z - d^2z_1 = 0$
$\pi'_3 : (f^2\cos\theta + b^2\sin\theta)X + (f^2\sin\theta - b^2\cos\theta)Z + f^2x_1 - b^2z_1 = 0$
$\pi'_4 : c^2\sin\theta X + f^2Y - c^2\cos\theta Z - c^2z_1 = 0$

Now, for a point P to lie in the view-volume of the second camera the following conditions must be satisfied.

$$\begin{gathered} \frac{-f^2\sin\theta + a^2\cos\theta}{f^2\cos\theta + a^2\sin\theta}(Z + z_1) - x_1 < X \\ X < \frac{-f^2\sin\theta + b^2\cos\theta}{f^2\cos\theta + b^2\sin\theta}(Z + z_1) - x_1 \end{gathered} \tag{9}$$

$$\frac{c^2\cos\theta}{f^2}(Z+z_1) - \frac{c^2\sin\theta}{f^2}(X+x_1) < Y$$
$$Y < \frac{-d^2\cos\theta}{f^2}(Z+z_1) + \frac{d^2\sin\theta}{f^2}(X+x_1) \tag{10}$$

A point P which lies in both the view-volumes will be visible in both the images. Thus, for corresponding points inequalities (7), (8), (9) and (10) hold simultaneously.

A new view is parallel to the first view and has camera matrix $K_1[I \mid t_s]$ where, $t_s = (x_s, y_s, z_s)^T$. For a point P visible in both the view-volumes to be visible in the view-volume of the new view, we must have

$$\frac{a^1}{f^1}(Z+z_s) - x_s < X < \frac{b^1}{f^1}(Z+z_s) - x_s \tag{11}$$
$$\frac{c^1}{f^1}(Z+z_s) - y_s < Y < \frac{d^1}{f^1}(Z+z_s) - y_s \tag{12}$$

For the new view to be such that every point visible in the given views is also visible in the new view, (7), (8), (9) and (10) must imply (11) and (12). It follows that the translation x_s should be such that

$$x_s > \frac{a^1}{f^1}z_s - \max(0, \frac{-f^2\sin\theta + a^2\cos\theta}{f^2\cos\theta + a^2\sin\theta}(Z+z_1) - \frac{a^1}{f^1}Z - x_1)$$
$$x_s < \frac{b^1}{f^1}z_s - \min(0, \frac{-f^2\sin\theta + b^2\cos\theta}{f^2\cos\theta + b^2\sin\theta}(Z+z_1) - \frac{b^1}{f^1}Z - x_1)$$

Note the presence of Z in the above inequalities. This Z is the depth of the point P in the coordinate system of the first camera. The λ^1 computed for each corresponding point in the synthesis algorithm is precisely this depth. Thus, we may replace Z by $\max(Z)$ and $\min(Z)$ by comparing the λ^1's. If the cameras internal parameters and the motion parameters of the second camera are known we can obtain a relation for x_s and y_s.

2.2. Synthesis from $\mathbf{N} \geq 3$ Views

Suppose now that we are given three or more views of a scene. We would like to extract the maximum number of point correspondences from these views and use them to render the new view. This will give us more vertices for triangulation and a better coverage of the scene. We choose a point visible in all the three views as the origin of the world coordinate system. Computing the infinite homography between different pairs of views, we can obtain λ's for all point correspondences. A new view is specified parallel to one of the given views and the positions of corresponding points in the new view can be obtained using (5).

3. Alternate Formulation

In this section we present an alternate formulation of our technique for view synthesis from multiple uncalibrated cameras.

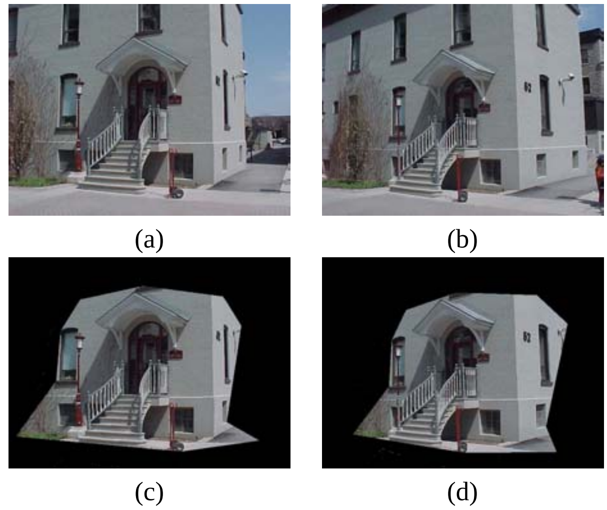


Figure 2: (a) and (b) are two images of an outdoor scene. (c) is a new view parallel to (a) and & (d) is a new view parallel to (b).

Suppose we are given two views, I_1 and I_2, of a scene such that the correspondence of three vanishing points, in general position, can be established between them. We may setup a world coordinate system with axes in the directions of these vanishing points and any world point, whose correspondence is available in the given views, as origin. Note that the coordinate system need not be orthogonal since the vanishing points need not be in mutually orthogonal directions. Let v_1, v_2, v_3 be the vanishing points in I_1 and v'_1, v'_2, v'_3 those in I_2. Then, the camera matrices of the two views may be parametrised as $M = [\alpha v_1\ \beta v_2\ \gamma v_3\ p_4]$ and $M' = [\alpha' v'_1\ \beta' v'_2\ \gamma' v'_3\ \delta' p'_4]$, where p_4 and p'_4 are the images of the origin of the world coordinate system in the two views.

We know from [3] that if $P = (X, Y, 0, 1)^T$ is a point on the world XY plane and $P' = (X, Y, Z, 1)^T$ is a point such that the line joining P and P' is parallel to the world Z axis, then,

$$\gamma Z = \frac{||p \times p'||}{(\bar{l}.p)||v_3 \times p'||}$$

where, p and p' are the images of P and P' in the first view, respectively, and $\bar{l} = \frac{l}{||l||} = p_4$, where l is the vanishing line in the first view of the world XY plane. Here, the origin, p_4, is chosen at the image point $\bar{l}$ so that the origin does not lie on the vanishing line l. Note that here only one view is required to compute γZ. Also, similar formulae can be derived for computing αX and βY for appropriately chosen points P and P'. In the following, we extend this idea to two views and give an algorithm to compute the coordinates $(\alpha X, \beta Y, \gamma Z)$ for all points whose correspondences are available in the two views.

3.1. Relating the Two Cameras

We first derive a novel relation between the parameters α, α' and δ', β, β' and δ' and γ, γ' and δ'.

Let $Q_1 = (X, 0, 0, 1)^T$ be a point on the world X axis. Let q_1 and q_1' be the images of Q_1 in the two images respectively. Then,

$$\begin{aligned} \lambda_1 q_1 &= \alpha v_1 X + p_4 \\ \Rightarrow (\lambda_1 q_1) \times q_1 &= \alpha X (v_1 \times q_1) + (p_4 \times q_1) = 0 \\ \Rightarrow \alpha X &= -\frac{||p_4 \times q_1||}{||v_1 \times q_1||} \end{aligned} \quad (13)$$

Similarly, from the second view,

$$\frac{\alpha'}{\delta'} X = -\frac{||p_4' \times q_1'||}{||v_1' \times q_1'||} \quad (14)$$

Dividing (14) by (13), we get,

$$\frac{\alpha'}{\alpha\delta'} = \frac{||p_4' \times q_1'|| \, ||v_1 \times q_1||}{||p_4 \times q_1|| \, ||v_1' \times q_1'||} \quad \Rightarrow \alpha' = C_1 \alpha \delta'$$

where $C_1 = \frac{||p_4' \times q_1'|| \, ||v_1 \times q_1||}{||p_4 \times q_1|| \, ||v_1' \times q_1'||}$. Similarly, choosing $Q_2 = (0, Y, 0, 1)^T$ on the world Y axis and $Q_3 = (0, 0, Z, 1)$ on the world Z axis, we get

$$\beta' = C_2 \beta \delta', \quad \gamma' = C_3 \gamma \delta'$$

where $C_2 = \frac{||p_4' \times q_2'|| \, ||v_2 \times q_1||}{||p_4 \times q_2|| \, ||v_2' \times q_2'||}$ and $C_3 = \frac{||p_4' \times q_3'|| \, ||v_3 \times q_3||}{||p_4 \times q_3|| \, ||v_3' \times q_3'||}$.

To be able to compute the constants C_i, $i = 1, 2, 3$, we need the correspondences of the points Q_1, Q_2 and Q_3. We describe here how to compute q_1 and q_1'; q_i and q_i', $i = 2, 3$ can be computed in a similar manner. The image of the world X axis in the first view is the line through p_4, the image of the origin and v_1, the vanishing point in the X direction. So, q_1 may be chosen to be any point on this line. The corresponding point q_1' will lie on the epipolar line, l_1, of q_1. l_1 can be computed from the fundamental matrix, F_{12}, from the first to the second view. Also, q_1' must lie on the image of the X axis in the second view, which is the line through p_4' and v_1'. Thus, q_1' is the intersection of l_1 and the image of the X axis in the second view. So, each of the points q_i and q_i' can be computed and so the constants C_i, $i = 1, 2, 3$ can be determined. The camera matrix M' may then be parametrised as $M' = [C_1 \alpha \delta' v_1' \; C_2 \beta \delta' v_2' \; C_3 \gamma \delta' v_3' \; \delta' p_4']$.

3.2. Obtaining Z-Buffer Values and Coordinates

Now, let $P = (X, Y, Z, 1)$ be any point in the world whose correpondence, p^1 and p^2, in the two views is available. Then, we have

$$\lambda^1 p^1 = v_1(\alpha X) + v_2(\beta Y) + v_3(\gamma Z) + p_4 \quad (15)$$

$$\begin{aligned} \lambda^2 p^2 &= \delta' [C_1 v_1' (\alpha X) + C_2 v_2' (\beta Y) + C_3 v_3' (\gamma Z) + p_4'] \\ \frac{\lambda^2}{\delta'} p^2 &= C_1 v_1' (\alpha X) + C_2 v_2' (\beta Y) + C_3 v_3' (\gamma Z) + p_4' \end{aligned} \quad (16)$$

We get six linear equations in the five unknowns $\lambda^1, \frac{\lambda^2}{\delta'}, \alpha X, \beta Y, \gamma Z$. These can be solved using singular value decomposition. Thus, we can compute the z-buffer value λ^1 and the coordinates, $(\alpha X, \beta Y, \gamma Z)$, for each point whose correspondence is known. If the parameters α, β and γ are known then the exact value of (X, Y, Z) can be obtained.

3.3. Synthesis of Virtual Views

Once we have obtained λ^1, new views can be synthesized parallel to the first view using the algorithm of Section 2.1.1. The only difference is that we use equations (15) and (16) to compute the λ's. Also, new views parallel to the second view can be synthesized by interchanging the two views and computing λ's for the second view.

3.4. Synthesis from Three or More Views

Our method can be easily extended to handle three or more input views. We only require that the correspondence of the three vanishing points in different directions be available in all the given views and that the origin be visible in all the views. Once these conditions are satisfied, constants C_i relating the parameters α, β, γ of the first camera with those of other cameras may be found. Consequently, coordinates and z-buffer values for additional point correspondences may be obtained and new views parallel to any of the given views can be synthesized. Also, since we have z-buffer values for the corresponding points, we can handle changes in visibility in the new view.

3.5. Results and Discussion

Figure 3 shows the results using the alternate formulation. (a) and (b) are two input images from which virtual views (c) and (d) parallel to (a) are synthesized and virtual views (e) and (f) parallel to (b) are synthesized. Figure 1 (g) and (h) are two views synthesized using this formulation. Note that the affine distortion is much less in this case. Infact, if the vanishing points are chosen so that they correspond to three mutually orthogonal directions in the 3D world, there is only a scale distortion in the synthesized views. Figure 3 (g), (h), (i) and (j) are four views synthesized using vanishing points corresponding to mutually orthogonal directions.

Figure 2 (a) and (b) are two images of an outdoor scene. (c) is a novel view parallel to (a) while (d) is a novel view parallel to (b). Figure 5 (a) and (b) are two frames taken from the motion picture *Harry Potter and the Sorceror's Stone* and (c), (d) and (e) are virtual views parallel to (a) while (f), (g) and (h) are virtual views parallel to (b). Note that in all cases visibility changes have been correctly handled.

Figure 4 shows the results using three images taken by arbitrary cameras. Using three views, we can get correspondences on the outlined box which would not be available had we used only views (a) and (b).

4. Conclusion

We have presented a technique for view synthesis from multiple uncalibrated cameras. Our scheme requires the knowledge of the infinite homography and computes z-buffer values for each corresponding point so that occlusions may be handled correctly in the new view. We also present an alternate formulation of our scheme that does not require the infinite homography but computes the z-buffer values for corresponding points.

References

[1] S. Avidan and A. Shashua. Novel View Synthesis by Cascading Trilinear Tensors. *IEEE Transactions on Visualization and Computer Graphics*, 4(4):293–306, 1998.

[2] N. L. Chang and A. Zakhor. Constructing a Multivalued Representation for View Synthesis. *International Journal of Computer Vision*, 45(2):157–190, 2001.

[3] A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD thesis, Department of Engineering Science, University of Oxford, 1999.

[4] Y. Genc and J. Ponce. Image-Based Rendering using Parametrised Image Varieties. *International Journal of Computer Vision*, 41(3):143–170, 2001.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[6] N. Inamoto and H. Saito. Intermediate View Generation of Soccer Scene from Multiple Views. In *Proc. ICPR*, 2002.

[7] M. Lhuillier and L. Quan. Image-Based Rendering by Joint View Triangulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(11):1051–1063, 2003.

[8] D. Liebowitz and A. Zisserman. Metric Rectification for Perspective Images of Planes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, 1998.

[9] H. Saito, M. Kimura, S. Yaguchi, and N. Inamoto. View Interpolation of Multiple Cameras Based on Projective Geometry. In *International Workshop on Pattern Recognition and Understanding for Visual Information Media.*, 2002.

[10] S. Seitz and C. Dyer. View Morphing. In *Proc. SIGGRAPH*, pages 21–30, 1996.

[11] G. Sharma, A. Kumar, S. Kamal, S. Chaudhury, and J. B. Srivastava. Novel View Synthesis Using a Translating Camera. To appear in *Pattern Recognition Letters*, 2004.

[12] S. Yaguchi and H. Saito. Arbitrary Viewpoint Video Synthesis from Uncalibrated Multiple Cameras. In *Proc. WSCG.*, 2002.

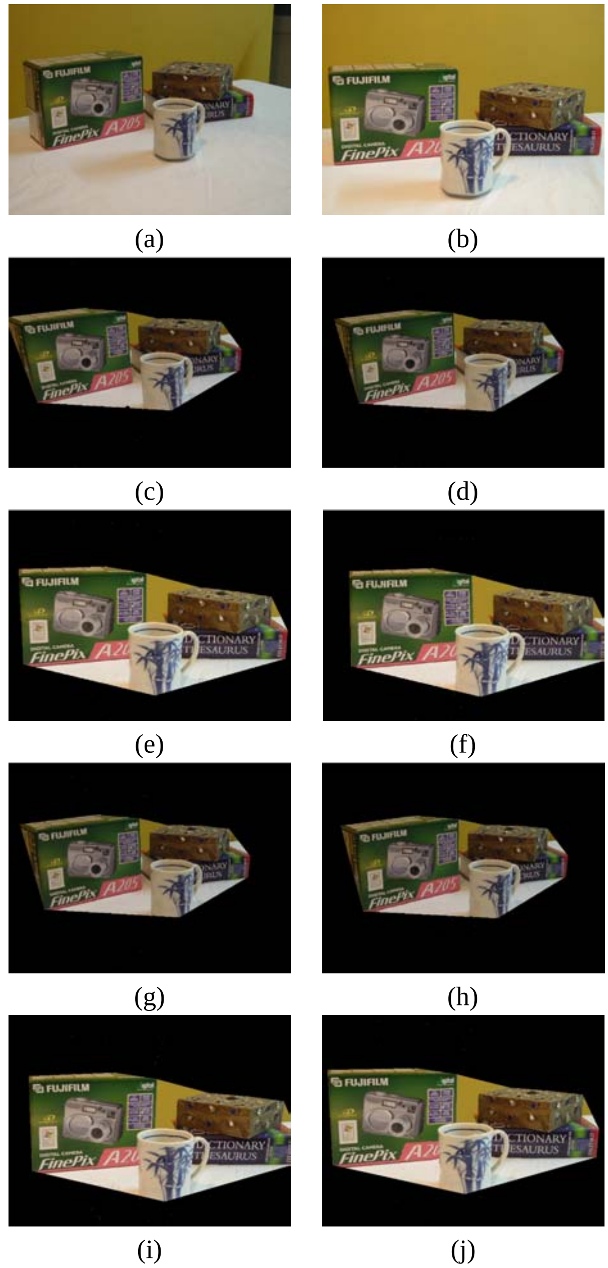
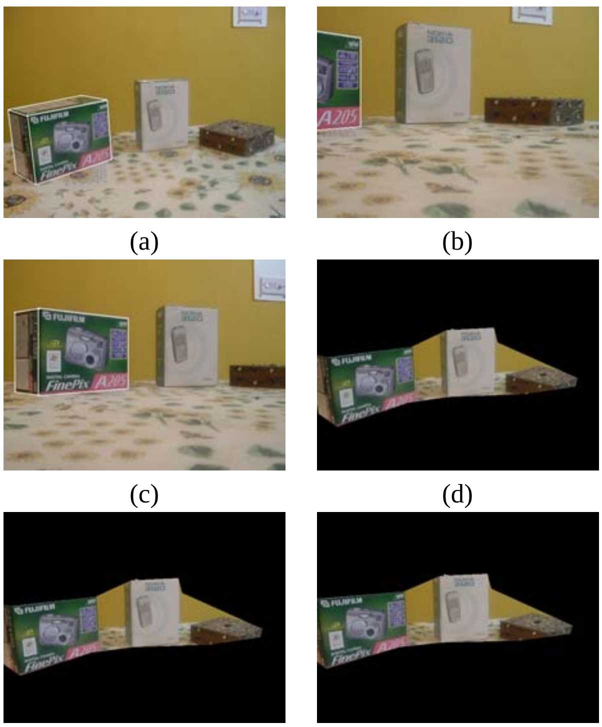
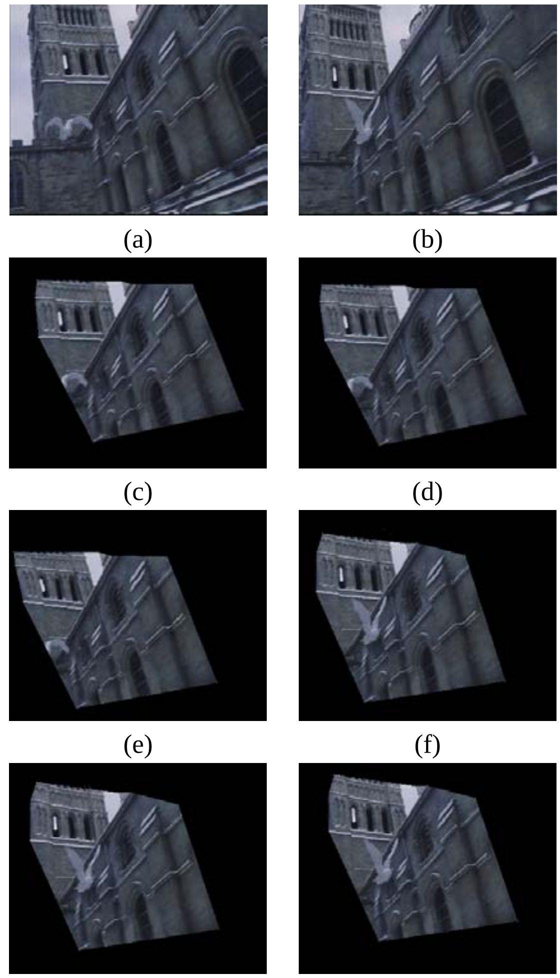


Figure 3: (a) and (b) are the input views (c) & (d) are synthesized views parallel to (a) and (e) & (f) are synthesized views parallel to (b).(g), (h), (i) & (j) are synthesized using vanishing points corresponding to mutually orthogonal directions in the 3D world.



(a) (b)
(c) (d)
(e) (f)

Figure 4: (a), (b) and (c) are the input views. (d),(e) and (f) are synthesized views parallel to (a).



(a) (b)
(c) (d)
(e) (f)
(g) (h)

Figure 5: (a) and (b) are two frames from the motion picture *Harry Potter and the Sorceror's Stone*. (c), (d) & (e) are synthesized views parallel to (a) and (f), (g) & (h) are synthesized views parallel to (b)