Parameterized Variety Based View Synthesis Scheme for Multi-view 3DTV

Mansi Sharma, Santanu Chaudhury, and Brejesh Lall

Department of Electrical Engineering, Indian Institute of Technology, Delhi
mansisharmaiitd@gmail.com, {santanuc,brejesh}@ee.iitd.ac.in

Abstract. This paper presents a novel parameterized variety based view synthesis scheme for 3DTV and multi-view systems. We have generalized the parameterized image variety approach to image based rendering proposed in [1] to handle full perspective cameras. An algebraic geometry framework is proposed for the parameterization of the variety associated with full perspective images, by image positions of three reference scene points. A complete parameterization of the 3D scene is constructed. This allows to generate realistic novel views from arbitrary viewpoints without explicit 3D reconstruction, taking few multi-view images as input from uncalibrated cameras.

Another contribution of this paper is to provide a generalised and flexible architecture based on this variety model for multi-view 3DTV. The novelty of the architecture lies in merging this variety based approach with standard depth image based view synthesis pipeline, without explicitly obtaining sparse or dense 3D points. This integrated framework subsequently overcomes the problems associated with existing depth based representations. The key aspects of this joint framework are: 1) Synthesis of artifacts free novel views from arbitrary camera positions for wide angle viewing. 2) Generation of signal representation compatible with standard multi-view systems. 3) Extraction of reliable view dependent depth maps from arbitrary virtual viewpoints without recovering exact 3D points. 4) Intuitive interface for virtual view specification based on scene content. Experimental results on standard multi-view sequences are presented to demonstrate the effectiveness of the proposed scheme.

1 Introduction

Over the intervening years, 3DTV technology has matured significantly to provide a realistic 3D impression of the scene. Multi-view systems (e.g. multi-view autostereoscopic displays) emerged as a core technology for 3DTV. The foremost requirement of these systems is the generation of high quality multi-view images. A variety of different 3D video representations exist to support these advanced 3D systems, with their own features and limitations. Multi-view video provides high quality 3D content and support wide angle viewing, but requires large amount of data to be processed. This needs sophisticated coding and bandwidth efficient transmission schemes. Video-plus-depth representation is quite popular for rendering of 3D views. It consists of monoscopic color video accompanied

K.M. Lee et al. (Eds.): ACCV 2012, Part IV, LNCS 7727, pp. 538–551, 2013.

with per-pixel depth data. As it explicitly contains 3D geometry information, virtual views can be rendered by depth image based rendering (DIBR) technique. This format is widely accepted as it is easily adapted to different 2D/3D display systems but does not support wide angle viewing. This is because DIBR falls into the category of point based rendering algorithms, and thus suffers from resampling problem, which possibly cause ghosting artifacts to appear in the rendered views. Moreover, the annoying visual artifacts (like holes, cracks) are present in the synthesized views due to inherent visibility and disocclusion problems. To support wide range multiview 3D displays, multi-view video-plus-depth is more appropriate. Rendered view quality is better as the representation uses more than one texture (color) and depth data. It avoids high complexity and maintain moderate size of the data. However, artifacts still occur in the synthesized views due to complex error prone processing steps and depth based rendering. Although DIBR based systems greatly reduce the bandwidth requirement as only two streams are needed to generate multi-view images, they are not suitable for high quality view generation from potentially arbitrary viewpoints.

For addressing these issues, a novel parameterized variety based representation and rendering scheme for multiview 3DTV systems is presented. The method construct a minimal parameterization of 3D space using a relatively small number of captured scene views. The scene is assumed to be captured by multiple uncalibrated cameras located at arbitrary positions. It has been shown earlier [1] that the set V of all views of n 3D points is a six dimensional variety of vector space R^{2n} for weak perspective, paraperspective and full perspective cameras. The parameterization of the variety in weak perspective and paraperspective cases were proposed earlier[1]. Our major contribution lies in the generalization of this approach to full perspective cameras. Euclidean constraints associated with the perspective cameras are explicitly taken into account. This yields a system of five quadratic multivariate polynomial equations, termed as parameterized image variety or PIV associated with the scene. This extension of variety based approach to full perspective cameras has a major advantage. It constructs a complete parameterization of 3D space (in terms of structure coefficients) which is not the case in weak and paraperspective cases as explained in [1]. The coefficients defining the PIV, allows to render novel views from arbitrary viewpoints without explicit 3D reconstruction. The technique produces photorealistic novel images without explicit depth recovery, therefore overcomes the most common problems associated with depth based methods. Moreover, using relatively less input views, large number of views can be synthesized from arbitrary viewpoints. These facts give the primary motivation to use this variety based approach for 3DTV view generation instead of depth based methods.

This variety model is used to build a new flexible multi-view 3DTV system that allows to render high quality virtual views of a 3D scene from arbitrary camera positions. Typical application of the methodology is in 3D viewing of wide range of indoor and outdoor urban scenes. The proposed system integrates two different view synthesis pipelines (transfer-based and depth-based) into one common framework. For merging the two different approaches without explicitly

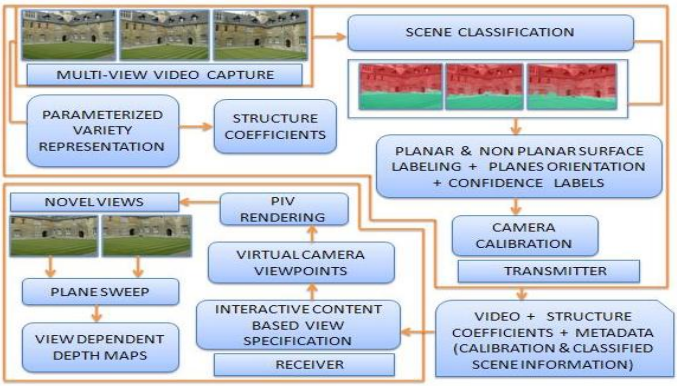


Fig. 1. A flexible variety based multi-view 3DTV system

obtaining sparse or dense set of 3D points, a statistical learning method is used[2]. Multiple input views are classified to detect planar and non planar regions along with their orientations and associated confidence measures. These are used to calibrate the input cameras. The calibration and orientation information of the classified input views are used to automatically define novel viewpoints with respect to the existing viewpoints. Novel views are synthesized using computed structure coefficients and PIV rendering. The realistic virtual views are obtained without using any calibration and depth information. Recovered orientation information of detected planes in classified input views are used to obtain dense depth maps (for all input and novel synthesized views) by using a plane sweep algorithm[3]. The orientations give the directions for sweeping planes, and thus avoid the need to reconstruct sparse or dense 3D points for depth estimation. This integrated framework supports standard video-plus-depth and multi-view video-plus-depth workflows, by generating reliable depth maps for all input and arbitrary virtual camera views. Extracted high quality view dependent depth maps can be used for content creation and 3D post production applications for multi-view displays. The architecture also provides important system features, such as an intuitive way of specifying virtual viewpoints based on content of the scene without complicated user intervention, compatibility with existing multi-view coding standards and adaptability to different 2D/3D displays.

2 Parameterized Variety Based Multi-view 3DTV System

Fig. 1 gives an overview of the proposed system. The proposed methodology is applicable for interactive 3D viewing of wide range of multiplanar environments like indoor and outdoor urban scenes. The input to the system are a collection of multi-view images taken from arbitrary, uncalibrated cameras. There are two stages of processing for signal generation. The first stage involves the construction of parameterized variety representation of the 3D scene (in terms

of structure coefficients). The system automatically establishes the sparse point correspondences across the multiple input views using scale invariant feature transform (SIFT) detector. Using the establish correspondences, parameterized variety is constructed. The coefficients defining the variety are computed and stored. These structure coefficients are the representative of the geometry information of the scene. The second stage is to use the classifier of [2] to classify all the input views, and identify all planar (vertical and horizontal) and nonplanar regions along with their orientations and associated confidence labels. The classification does not rely on any calibration or 3D scene information. Input cameras are self calibrated using the inter image homographies obtained from the located set of coplanar points across the views and applying the method presented in [4].

The outcome of these two stages are computed structure parameters, calibration and scene classification information (orientations, confidence labels etc.) of the input views. Thus, the structure coefficients along with the video forms the signal representation. Calibration and scene classification information are embedded as metadata part of the signal. The signal generation is an offline process. The generated signal is encoded and transmitted. At the receiver end, the user interactively selects certain part (e.g. a wall) of the scene in one of the input image. The system automatically specifies the virtual viewpoints using the plane orientation information of that part of the image. The virtual viewpoints are defined as such that the selected part (i.e. wall) is best viewed. A series of high quality virtual views are generated using the transmitted structure coefficients and PIV rendering, without using any calibration and explicit depth information. The calibration information of input views is used only in automatic viewpoint specification. Although PIV requires no explicit depth data to render virtual views, it is possible to extract the dense depth maps of novel synthesized images without obtaining dense 3D points, using the decoded classification information of the input views. A plane sweep approach is basically followed [3] for extracting view dependent depth maps. Instead of identifying the surface normals by analyses of dense 3D points through structure from motion, orientation information of the classified planes is used to identify the directions for sweeping. This gives additional flexibility to the architecture to support existing multi-view systems that relies on depth based representations. The other advantages of this signal representation are:

1. The representation is bandwidth efficient as one needs to transmit relatively small number of multiple views. The structure parameters and metadata can be efficiently encoded and transmitted with a less overhead. It is even compatible with existing multi-view coding and compression schemes.
2. In DIBR based systems, coding/transmission artifacts generally occur in the depth maps (blocking effects, ringing artifacts around the edges etc.). In our approach, depth maps of input and virtual views are obtained at the receiver end using the signal representation only, which ensures its good quality.

The details of each component are presented in the following sections.

2.1 Signal Generation

To generate the required signal, two stage of processing is involved 1) Given multi-view images, construct the parameterized representation of the 3D scene, and estimate the corresponding structure coefficients. 2) Obtain the scene classification and calibration information of the input views.

2.1.1 Parameterized Variety Representation of 3D Scene

Suppose we observe three scene points Q_0, Q_1, Q_2 whose images $q_0 = (u_0, v_0)^T$, $q_1 = (u_1, v_1)^T$, $q_2 = (u_2, v_2)^T$ are not collinear. Define the coordinate vectors of these points in a Euclidean coordinate system as $Q_0 = (0, 0, 0)^T$, $Q_1 = (1, 0, 0)^T$ and $Q_2 = (p', q', 0)^T$. The values of p' and q' are nonzero but (a priori) unknown. Point PIV is parameterize using these three scene points. Consider a point $Q = (x', y', z')^T$ and its projection $\boldsymbol{q} = (u, v)^T$ in the image plane. The values of (x', y', z') are unknown. The image (x_i, y_i) of any scene point X_i=$[X, Y, Z]^T$ under perspective camera model[12] can be written as

$$\lambda_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} m_1^T & T_x \\ m_2^T & T_y \\ m_3^T & T_z \end{bmatrix} \begin{bmatrix} X_i \\ 1 \end{bmatrix}, \tag{1}$$

where, λ_i is the projective depth of point X_i. In normalized image coordinates m_1, m_2, m_3 represent the rows of the rotation matrix. The Euclidean constraints associated with the full perspective cameras are:

$$\begin{cases} \mid m_1 \mid^2 = \mid m_2 \mid^2, \mid m_2 \mid^2 = \mid m_3 \mid^2, \\ m_1 \cdot m_2 = 0, m_1 \cdot m_3 = 0, m_2 \cdot m_3 = 0. \end{cases} \tag{2}$$

Projecting Q_0, Q_1, Q_2 and Q under perspective camera model Eq. (1), yields

$$m_1 = BU, m_2 = BV, m_3 = BL, \tag{3}$$

where, $B = \begin{pmatrix} 1 & 0 & 0 \\ \vartheta & \mu & 0 \\ \varsigma_4/z' & \varsigma_5/z' & 1/z' \end{pmatrix}$, $U \stackrel{def}{=} \begin{pmatrix} \lambda_1 u_1 - \lambda_0 u_0 \\ \lambda_2 u_2 - \lambda_0 u_0 \\ \lambda u - \lambda_0 u_0 \end{pmatrix}$, $V \stackrel{def}{=} \begin{pmatrix} \lambda_1 v_1 - \lambda_0 v_0 \\ \lambda_2 v_2 - \lambda_0 v_0 \\ \lambda v - \lambda_0 v_0 \end{pmatrix}$, $L \stackrel{def}{=} \begin{pmatrix} \lambda_1 - \lambda_0 \\ \lambda_2 - \lambda_0 \\ \lambda - \lambda_0 \end{pmatrix}$ and $\vartheta = -p'/q', \mu = 1/q', \varsigma_4 = -(x' + \vartheta y'), \varsigma_5 = -\mu y'$. The λ_0, λ_1, λ_2, λ are the projective depth associated with points Q_0, Q_1, Q_2 and Q. Using Eq. (3) and letting $C_s \stackrel{def}{=} z'^2 B^T B$, full perspective constraints Eq. (2) can be written as

$$\begin{cases} U^T C_s U - V^T C_s V = 0, V^T C_s V - L^T C_s L = 0, \\ U^T C_s V = 0, U^T C_s L = 0, V^T C_s L = 0, \end{cases} \tag{4}$$

with

$$C_s = \begin{pmatrix} \varsigma_1 & \varsigma_2 & \varsigma_4 \\ \varsigma_2 & \varsigma_3 & \varsigma_5 \\ \varsigma_4 & \varsigma_5 & 1 \end{pmatrix}, and \begin{cases} \varsigma_1 = (1 + \vartheta^2) z'^2 + \varsigma_4^2 \\ \varsigma_2 = \vartheta \mu z'^2 + \varsigma_4 \varsigma_5 \\ \varsigma_3 = \mu^2 z'^2 + \varsigma_5^2. \end{cases} \tag{5}$$

Substituting U, V, L, C_s in Eq. (4) and defining the variables $g_1 = \frac{\lambda_1}{\lambda_0}$, $g_2 = \frac{\lambda_2}{\lambda_0}$, $g_3 = \frac{\lambda}{\lambda_0}$, we get a system of five quadratic equations $\{f_1, f_2, f_3, f_4, f_5\}$ in eight unknown variables $\varsigma_1, \varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5, g_1, g_2, g_3$. Five structure parameters $\varsigma_1, \varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5$ remain fixed, when four scene points Q_0, Q_1, Q_2 and Q are rigidly attached to each other. Choosing three points Q_0, Q_1, Q_2 as a reference triangle from n points and writing Eq. (4) for the remaining ones yield a set of $5n - 15$ quadratic equations in $8n - 24$ unknowns. This is the PIV. The structure parameters construct the parameterization of the set of all perspective images of the scene. The parameters are calculated by eliminating three variables g_1, g_2 and g_3 from five quadratic equations $\{f_1, f_2, f_3, f_4, f_5\}$. To eliminate the three variables g_1, g_2 and g_3, we follow closely the approach adopted in [5] to solve recognition problem for six lines. Elimination is performed in stages by applying Cayley-Dixon-Kapur-Saxena-Yang method (KSY method) [13].

Algorithm A
Input:
1) A collection of N input images and n point correspondences.
2) Three points $q_0 = (u_0, v_0)$, $q_1 = (u_1, v_1)$, $q_2 = (u_2, v_2)$ out of n points are chosen as reference points.

For $i = 1..N$ and $s = 1...n - 3$ {
Step 1: Substitute the known values of the eight parameters u_{0i}, v_{0i}, u_{1i}, v_{1i}, u_{2i}, v_{2i}, u_{is}, v_{is} (rational or integral) to quadratic polynomials. This reduces the size and complexity of the polynomials.
Step 2: Choose to work over a finite field like $Z_p\,[g_1, g_2, g_3]\,/(g_1^2 - 3, g_2^2 - 5, g_3^2 - 7)$, where p is a large prime and Z_p is a finite field of order p. This eliminated higher degree terms in g_1, g_2, g_3 occurring at intermediate steps and greatly speed up the computation.
Step 3: Apply KSY to eliminate two variables g_1 and g_2 from three equations $\{f_{1i}, f_{2i}, f_{3i}\}$ obtaining a polynomial q_1 in variables $\{\varsigma_1, \varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5, g_3\}$.
Step 4: Apply KSY to eliminate two variables g_1 and g_2 from three equations $\{f_{3i}, f_{4i}, f_{5i}\}$ obtaining a polynomial q_2 in variables $\{\varsigma_1, \varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5, g_3\}$.
Step 5: Apply KSY to eliminate g_3 from $\{q_1, q_2\}$ to get the final resultant Res.
Step 6: Subsequent higher orders (greater than one) in any of the variables ς_1, ς_2, ς_3, ς_4, ς_5 occuring in Res can be mod out. Choose to mod out by quadratic irreducible polynomial (like $\varsigma_j^2 - 11$ $(j = 1...5)$).
Step 7: Apply numerical techniques (like Jenkins-Traub method [6]) to solve Res. Stored estimated parameters in D_{is}. }
Step 8: Perform singular value decomposition of matrix D_{is} to refine the parameters and store them as a vector $\varsigma = (\varsigma_1, \varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5, 1)^T$.

2.1.2 Classification and Surface Labeling

Scene classification and surface labeling of input views is performed by using the methodology of [2]. The authors perform the labeling of the different regions of an image into different geometric classes based upon their 3D orientation with respect to the capturing camera. This machine learning approach model the

appearance of geometric classes from a set of training images. No calibration and 3D geometry information is required. Using that the likelihood of each of the possible classes for each pixel is estimated. Regions are mainly categorized as ground (horizontal), sky and vertical (left, right or center) and non planar surfaces either porous or solid. The signal components are the orientations, labels and associated confidence measures of each classified region.

2.1.3 Camera Self Calibration Using Scene Planes

Cameras are self calibrated using one or several planar regions obtained from the classified scenes. Point correspondences are establish and detected features are matched across the views. Outliers are removed by robustly fitting fundamental matrix between pairs of views. From the remaining inliers, points belonging to different planes are separated out. For this purpose, any one image can be used as the camera placement is arbitrary and corresponding points may have different labels across the views. From the located coplanar points across the views, inter image homographies have been estimated using the normalized DLT algorithm [7]. Image of absolute conic ω is determined using plane homographies by applying the method similar to [4].

The generated signal (i.e. structure parameters, classified scene and calibration data) is encoded and transmitted. At the receiver end, virtual viewpoints are specified and novel views are synthesized using decoded signal information.

2.2 Viewpoint Specification and Synthesis

We propose an intuitive and practical way for virtual view specification based on the content of the scene. In general, a viewpoint can be specified by performing a translation and rotation with respect to any input view to determine its position and direction. But it is impractical to ask a TV viewer to do this. A more practical way is to start with a given input view and let the user to choose the viewpoint based on scene content. This allows to see the novel 3D views of the chosen part as well as arbitrary virtual views of the entire scene. This content based relative viewpoint moving, in an interactive manner, is much more convenient. Typical application of it is in 3D viewing of indoor and outdoor scenes like building, shopping malls etc. Typical characteristics of such scenes (i.e. extracted planar and nonplanar patches) facilitate in automatic detection of positions and orientations. Our system is designed to synthesize continuum of virtual views from one viewpoint to some other (arbitrary) viewpoint.

2.2.1 View Specification Based on Scene Content

The user interactively selects a part of the scene in any given input view, through an external interface. For instance, if a wall of a monument in input scene is selected, a new viewpoint is defined automatically such that the wall is fronto parallel. We make use of the fact that the best view of a plane is fronto parallel. The orientation information of this part (i.e. wall), obtained from the decoded signal is used to determine the direction in which reference camera

($P^{'}_{ref} = K^{'}_{ref} R^{'}_{ref}[I| - C^{'}_{ref}]$) has to be rotated. Once the direction is specified, the virtual camera matrix ($P^{'}_{final} = K^{'}_{final} R^{'}_{final}[I| - C^{'}_{final}]$) is chosen as:

1. For plane corresponding to right part of the scene, a rotation matrix R_y is defined for rotation about the positive Y axis by an angle φ confined within the angle formed by plane normal and principal axis of the camera. The final virtual camera matrix is chosen as:

$$K^{'}_{final} = K^{'}_{ref},\ R^{'}_{final} = R_Y * R^{'}_{ref},\ C^{'}_{final} = C^{'}_{ref} + [0; 0; t]$$

   A small translation step t is required to keep the intermediate virtual camera view within the image bound. The factor t also provides a zoom in effect since effectively the camera is moving into the image.
2. Similarly for left and ground plane, rotation matrix is calculated for rotation about the negative "Y" and positive "X" axis respectively. For the center plane no rotation matrix is calculated. A gradual interpolation of camera matrices is performed from $P^{'}_{ref}$ to $P^{'}_{final}$ using varying interpolation factor $\alpha \in [0..1]$. Sperical linear interpolation *"slerp"* is applied to each row of the camera matrix.

$$K_\alpha = K^{'}_{ref},\ R_\alpha = slerp(R^{'}_{ref}, R^{'}_{final}, \alpha),\ C_\alpha = C^{'}_{ref} * \alpha + C^{'}_{final} * (1 - \alpha)$$

   These intermediate camera matrices are used to synthesize a continuum of virtual views, and thus we get a feeling of the wall turning towards us. Novel occlusion free views are synthesized using decoded structure parameters and PIV rendering.

2.2.2 Novel View Synthesis Using PIV Rendering

Novel views can be rendered by specifying image positions q_0, q_1 and q_2 for three reference points Q_0, Q_1 and Q_2 at the virtual viewpoint and computing the corresponding image positions of all other points. The algorithm for synthesis of a novel view I_{nv} is summarized as:

1. Define a new view by specifying image positions $q_0 = (u_0, v_0)$, $q_1 = (u_1, v_1)$, $q_2 = (u_2, v_2)$ of three reference points at virtual viewpoint. Let it be $q'_0 = (u'_0, v'_0)$, $q'_1 = (u'_1, v'_1)$, $q'_2 = (u'_2, v'_2)$.
2. Substitute q'_0, q'_1, q'_2 in Eq. (4) in place of q_0, q_1, q_2. Using computed structure coefficients, render the image positions $(u, v)'s$ of all other corresponding points in the new view by solving quadratic equations Eq. (4). Any visibility issue can be resolved using obtained $g'_3 s$ (scaled depth value as $g_3 = \frac{\lambda}{\lambda_0}$) for each corresponding point as z-coordinate values.
3. Triangulate the new view I_{nv} using the rendered points as vertices [11]. Assign a depth to each triangle by taking the mean depth of its three vertices. Sort the triangles in descending order of depth. Texture map the triangles from the given input views in decreasing order of depth. For each pixel p_{nv}, in the current triangle t_c of the novel view, compute the barycentric coordinates

of the pixels in I_{nv}. Find the pixels corresponding to p_{nv} in given input views $I_1, ..., I_N$ by computing the affine combination of the barycentric coordinates and the vertices of the same triangle t_c in $I_i(i = 1..N)$. Find the front-most triangles, the corresponding pixel lies in $I_i(i = 1..N)$. If any of the front-most triangle is the same as the triangle t_c , use the intensity from that triangle. If not, color the pixel black.

2.3 Depth Map Estimation

Depth maps for each of the input images and novel synthesized images are obtained by performing plane sweeping. In our approach, we follow closely to [3]. The basic steps involved are:

1. *Sweeping directions estimation*: Scene classification outputs a labeled image, where each pixel is assigned the label of the geometric class which most likely represents it and also the confidence measures associated with each geometric label. Pixels grouped after the classification are collected and planes are robustly fitted. Let Λ_{kl} denote M family of parallel depth planes, denoted as $\Lambda_{kl} = [n_k^T \;\; d_{kl}]$, $\{k = 1, ..., M\}$. The subscript l indices over number of planes corresponding to k^{th} family and n_k denotes unit length normal of the k^{th} family planes. The depth range $[d_{near}^k \;\; d_{far}^k]$ for each family is obtained empirically.
2. *Obtaining the sweeping planes*: Once the sweeping directions n_k are determined, the actual planes used in sweeping are obtained by varying d_{kl} obtained from the previous step.
3. *Warping*: Homography H_{Λ_{kl},P_i} induced by each of the planes Λ_{kl} is determined between two images (obtained at different camera positions). Let $P_{ref} = K_{ref}R_{ref}[I| - C_{ref}]$ and $P_i = K_iR_i[I| - C_i]$ be the camera projection matrices for the reference view I_{ref} and the other camera view I_i. The homography H_{Λ_{kl},P_i} is used to warp image I_i to obtain I_i^*. Missing pixels are interpolated using bilinear interpolation. For the warped image I_i^*, cost metric is defined as a function of pixel (x, y) in the reference view I_{ref} and for each of the plane Λ_{kl} as:

$$C(x, y, \Lambda_{kl}) = \sum_{(\delta_x,\delta_y)\in W} |I_{ref}(x-\delta_x, y-\delta_y) - I_i^*(x-\delta_x, y-\delta_y)| - \sigma * log(F_{\Lambda_{kl}}(x, y))$$

   where, σ is weight factor which is learned by experiments, W is 3×3 neighbourhood of the pixel and $F_{\Lambda_{kl}}(x, y)$ is the probability that the pixel (x, y) belongs to the plane Λ_{kl}. It is obtained in terms of confidence measures.
4. *Best plane selection*: The simplest possible technique is to choose the plane of minimum cost as $\widehat{\Lambda_{kl}}(x, y) = \arg\min_{\Lambda_{kl}} C(x, y, \Lambda_{kl})$. However, noise is still observed due to incorrectly assigned planes. The solution is formulated in an energy minimization framework similar to [3] and minimize it using techniques like graph cuts [8]. Once the plane label is correctly identified for each pixel (x, y), depth map of image I_{ref} is estimated.

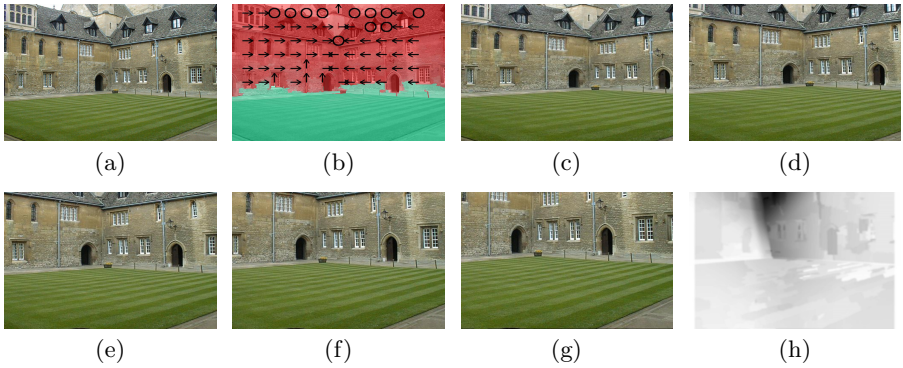


Fig. 2. (a) One of the input image of Merton. (b) Its scene classification and planes orientation. (c,g) Novel synthesized sequence when viewer selected the lower right part of the building. (h) Estimated dense depth map of novel view (g).

3 Implementation Details and Results

The scheme is implemented using MATLAB (R2009a) and MATHEMATICA. Mathematica implementation of KSY Dixon resultant algorithm is used [9] to perform the elimination and finding structure coefficients. Reduction over finite field is performed by interfacing with Sage version 5.0.1. The performance is extensively tested on various standard multi-view dataset and video sequences of indoor and outdoor scenes: 3D video[1], Visual geometry group[2], Kitchen[3] dataset. Test conditions consider both cases of simple and complex camera motion and also taken into account the scenes containing high detail and complex depth structures. Results with only static scenes are presented as it is difficult to perform comparative analysis with dynamic scenes in respect they are unrepeatable. The proposed scheme is workable for dynamic scenes also, by constructing parameterization of each temporal aspect independently.

3.1 View Synthesis Results Using PIV Rendering

The various steps of the proposed scheme are illustrated with Merton[2] dataset. All three images of Merton are used for the estimation of structure coefficients. Fig. (2(a),2(b)) shows the input view and its classified scene planes (green (horizontal), red (vertical)) and orientations (arrows). Fig. (2(c),2(g)) shows the novel synthesized views, when the user is intended to view the lower right wall of the scene closely. No rendering artifacts occur even if the camera is taking a steep turn towards the right part of the scene. Fig. 2(h) shows the estimated depth of novel view obtained from the procedure described in section 2.3.

[1] `http://www.cad.zju.edu.cn/home/gfzhang/projects/videodepth/data/`
[2] `http://www.robots.ox.ac.uk/~vgg/data/data-mview.html`

3.2 Comparative Analysis

The proposed rendering scheme has been compared with state of art DIBR technique [10]. An experiment with Kitchen[3] dataset is performed. Out of eight given camera views, five views (C_4 to C_8) are used for the estimation of structure coefficients (*Algorithm A*). A novel view is resynthesized from viewpoint corresponding to C_2 using PIV rendering. Another experiment is conducted using avaliable ground truth depth maps. Nearest camera views C_1 and C_3 are chosen as reference, and virtual view at camera C_2 are resynthesized using standard DIBR based view synthesis pipeline [10]. Resynthesized views from both methods are compared with the original one, to assess the quality. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) is measured for objective quality assessment. Rendering artifacts are clearly visible in resynthesized view obtained using DIBR, even after contour correction and hole filling Fig. 3(f) (see marked areas). Artifacts are visible where depth values are erroneous Fig. 3(e). The quality of the synthesized view using PIV rendering is comparatively much better Fig. 3(g). This experiment have revealed another important fact about full perspective PIV representation. The camera views C_1, C_2 and C_3 are not used in estimating the PIV coefficients, yet the rendered view quality at viewpoint C_2 is quite good. The occlusion is correctly handled. This means PIV can be used to extrapolate the views outside the camera basis from arbitrary viewpoints and even using small number of input views. Fig. (3(h),3(j)) shows the novel synthesized PIV views of Kitchen from arbitrary viewpoints.

Quality of the depth map obtained using classified scene data of input views is also accessed. Depth map of PIV resynthesized view Fig. 3(g) is determined using scene classification information of input views (C_4 to C_8). From dense correspondences between resynthesized PIV view and input views, points belonging to different planes are separated out. Fig. 3(k) shows its classification into different planar regions. The regions are divided into left (pink), right (red) or center (cream), ground (green) and ceiling (blue). Labels and associated confidence measures are shown in Fig. (3(l),3(q)). Fig. 3(r) shows the plane family labels obtained after sweeping and graph cut minimisation. Final determined depth map Fig. 3(s) is compared with ground truth. The PSNR value obtained is much better as compared to final depth map obtained using DIBR Fig. 3(e).

Fig. 4 shows the results on scenes containing complex planar and non-planar geometries. Annoying artifacts predominate the DIBR rendered view quality Fig. 4(a) as compared to the proposed method Fig. 4(b), when virtual viewpoint is far away from the original camera position. Fig. 4(c) highlight the shortcoming of DIBR with respect to zoom-in effects. The image quality degrades (holes, cracks) as one move more into the image because of the inherent sampling problem. Comparatively, rendered PIV vitual view Fig. 4(d) are quite realistic and superior in quality, even when the camera is zoomed more into the image. Subjective quality assesment (Tab. 1) has been carried out on a group of 17 human subjects, expressed by a 10 point continuous scale ranging from 1 (severe annoying artifacts) to 10 (imperceptible artifacts).

[3] `http://lttm.dei.unipd.it/downloads/kitchen/`

Table 1. Average mean opinion scores (MOS) and standard deviations (SD)

DIBR		PIV	
MOS (5.364)	SD (1.152)	MOS (8.975)	SD (1.143)

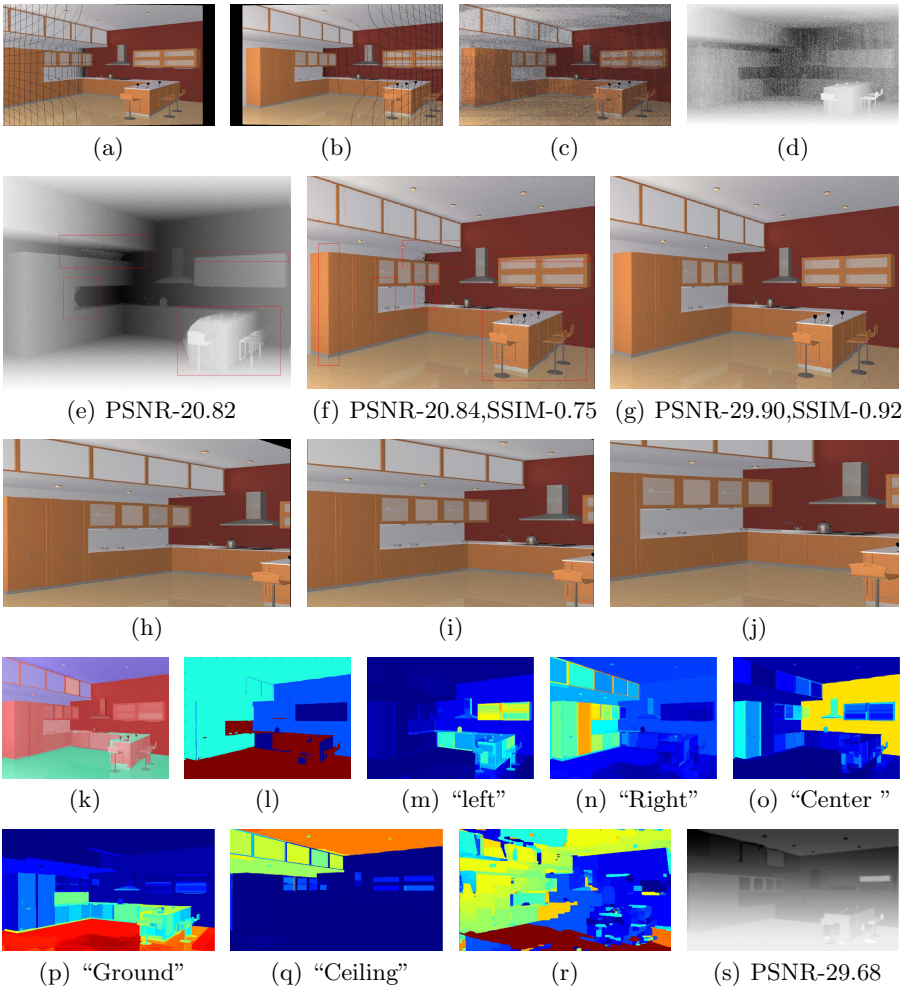
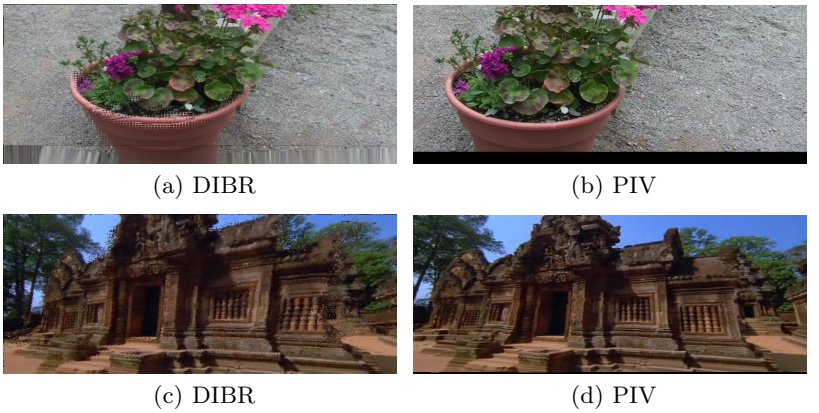


Fig. 3. (a,b) Warped virtual views from left (C_1) and right (C_3) reference camera at viewpoint C_2. (c) View obtained after contour correction and merging (a) and (b). (d) Depth map associated with (c). (e) Final depth map and virtual view (f) obtained after median filtering and hole filling [10] at C_2. (g) PIV resynthesized view at C_2 (h,j) PIV rendered virtual views from arbitrary viewpoints. (k) Classified PIV novel view (g). (l) Label associated with each geometric class. (m,q) Confidence with each label. (r) Graph cut minimized planes family labels. (s) Final depth map at C_2.



(a) DIBR (b) PIV

(c) DIBR (d) PIV

Fig. 4. (a,c) DIBR [10] rendered views of Flower and Temple sequence (holes, cracks). (b,d) Proposed PIV rendered views (realistic, fine texture details are preserved).

3.3 Performance Evaluation and Camera Calibration Results

Tab. 2 shows average CPU time for performing the elimination of variables using KSY method[9], finding structure parameters (*Algorithm A*) and rendering a novel view using estimated parameters. These timing are noted on Intel(R) Core(TM) i3 2.13 GHz PC with 3 GB of RAM with unoptimized matlab code.

Table 2. Computation time (sec) for estimation of structure coefficients (ESC) and rendering a novel view (RNV)

	Merton	Kitchen	Temple
No. of points	40	36	35
No. of input images	3	5	10
Time(sec)	10.32(ESC) 6.224(RNV)	15.48(ESC) 5.602(RNV)	30.01(ESC) 5.446(RNV)

3.4 Camera Calibration

To evaluate the performance of camera self calibration using classified scene data, results are compared with ground truth calibration data available with Temple[1] and Kitchen sequence. The number of cameras varies from 2 to 8 (Tab. 3).

4 Discussion and Conclusions

We present a flexible architecture for multi-view 3DTV build on a novel parameterized variety based representation and rendering scheme. The scheme allows to render a continuum of virtual views from arbitrary viewpoints using few sample

Table 3. Percentage error (%) in focal length estimation

No. of camera views	2	3	4	5	6	7	8
Temple	1.36	1.04	0.76	0.77	0.91	0.99	0.68
Kitchen	1.31	1.33	1.47	0.72	0.79	0.79	0.71

images. It provides a parameterization of all possible views and overcome the shortcomings of depth based methods. The signal representation is bandwidth efficient, compatible with standard multiview coding schemes and adaptable with 2D/3D displays. It duly supports the existing multi-view 3D systems based on depth based representations, by generating high quality views and per-view depth maps from arbitrary camera viewpoints. Looking at these advantages, rendering time is not a critical issue. It can be substantially reduced with GPU implementation of this scheme, which is our next target.

References

1. Genc, Y., Ponce, J.: Image Based rendering using parameterized image varieties. International Journal of Computer Vision 41, 143–170 (2001)
2. Hoeim, D., Efros, A.A., Hebert, M.: Geometrical context from a single image. In: International Conference on Computer Vision, vol. 1, pp. 654–661 (2005)
3. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-Time Plane Sweeping Stereo with Multiple Sweeping Directions. In: CVPR, pp. 1–8 (2007)
4. Sturm, P.F., Maybank, S.J.: On Plane Based Camera Calibration: A General Algorithm, Singularities, Applications. In: CVPR (1999)
5. Lewis, R.H., Stiller, P.F.: Solving the recognition problem for six lines using the Dixon resultant. IMACS 49 (1999)
6. Jenkins, M.A., Traub, J.F.: A three-stage variable-shift iteration for polynomial zeros and its relation to generalized rayleigh iteration. Number. Math. (1970)
7. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision (2002)
8. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. TPAMI 23, 1222–1239 (2001)
9. Nakos, G., Williams, R.: Elimination with the Dixon Resultant. Mathematica in education and research 6, 11–21 (1997)
10. Zinger, S., Doa, L., de With, P.H.N.: Free-viewpoint depth image based rendering. Visual Communication and Image Representation 21, 533–541 (2010)
11. Lhuillier, M., Quan, L.: Edge-constrained joint view triangulation for image interpolation. In: CVPR, vol. 2, pp. 218–224 (2000)
12. Horaud, R., Dornaika, F., Lamiroy, B., Christy, S.: Object pose: The link between weak perspective, paraperspective and full perspective. IJCV (1997)
13. Kapur, D., Saxena, T., Yang, L.: Algebraic and geometric reasoning using the Dixon resultants. In: ACM ISSAC, Oxford, England, pp. 99–107 (1994)