IEEE Workshop on Stereo and Multi-Baseline Vision, December 9–10, 2001

Range-Space Approach for Generalized Multiple Baseline Stereo and Direct Virtual View Synthesis

Kim C. Ng[1], Mohan Trivedi[2], and Hiroshi Ishiguro[3]

[1]*AST – La Jolla Lab, STMicroelectronics, 4690 Executive Dr, San Diego CA 92121, USA*
[2]*Computer Vision & Robotics Research Lab, University of California–San Diego, CA 92093, USA*
[3]*Department of Computer and Communication Sciences, Wakayama University, Japan*

Abstract

In this paper a new "*range-space*" approach, for rendering visual models using a network of multiple omni-directional vision sensors (ODVS) is presented. This integrated approach allows for simultaneous extraction of 3-D range as well as visual models. The approach requires three distinct steps of analyzing multiple ODVS video input streams: 1) Search, 2) Match, and 3) Render. At the output, a user-specified view is rendered. This three-step process does not require 3D model of the scene to be provided.

Keywords: Range-space stereo; wide-baseline stereo; multiple baseline stereo; omni-directional video; view rendering; virtual walkthrough.

1. Introduction

Range-space approach is different from those where geometrically valid virtual views are derived using 3D models for the entire scene and then individual voxels are projected on to small windows of the virtual camera locations. Such approach requires depth recovery for every pixel and then rendering of individual voxels to the virtual view becomes enormously expensive. In the range-space approach, only a small set of voxels along the desired viewpoint is appropriately rendered.

The system is an extension of previous multi-camera systems. However, it differs them in five distinguishing aspects. (1), Multiple omni-directional vision sensors (ODVS) are exploited to provide wide scene coverage, data redundancy, and a way to synthesize arbitrary views by composing multiple cameras' pixel color. Omni-directional images (ODI) provide large overlapping region that is necessary for stereo matching, and their unique periodical signal makes smooth walkthrough possible. (2), Cameras are configured at arbitrary disparate location. A number of cameras are grouped into a video cluster for a given viewpoint. Cameras within the video cluster are selected based on robust statistics for matching. (3), Both 3D and views are generated simultaneously, in which virtual and real worlds are immersed into one, in contrast to the total virtual of Virtual Reality. No 3D model is needed. (4), Distributed computing is utilized to allow many views generated concurrently. (5), Efficient data structure and image caches are exploited to speed up view generation process.

Range-space approach needs to both estimate depth and recover color simultaneously without the intermediate sequential steps, so that virtual view synthesis with range estimation is performed exactly at the user-specified viewpoint. The challenges encountered in stereo vision have a straightforward and effective solution in the range-space approach. The research addresses five major challenges of wide-baseline omni-directional stereo: Scaling Effect, Foreshortening Effect, Window Cutoff, Specular Highlight, and Occlusion. When searching in the range space, it enables arbitrary disparate multi-camera configurations and it helps overcome the first three challenges. Cameras are selected through robust statistics to lessen the effects of Specular Highlight and Occlusion.

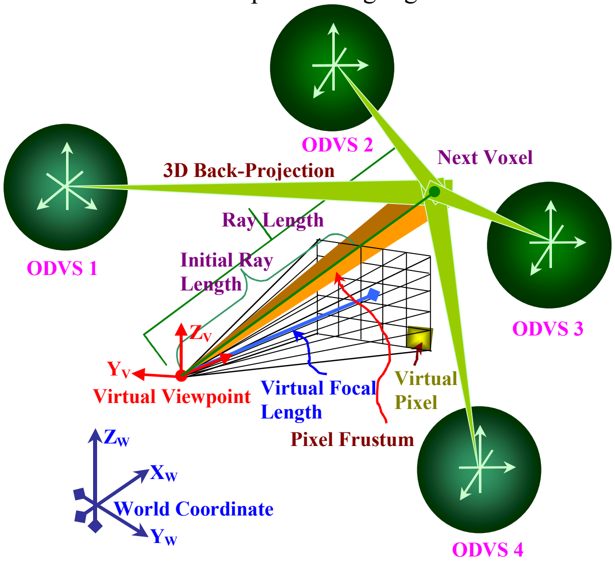

Figure 1 Illustration of the Range-Space Approach.

Figure 1 shows a view of four ODVS in a multiple baseline stereo configuration. The starting point for the range-space search is at the virtual viewpoint. For each pixel on that virtual image plane, we project a *pixel frustum* whose left, right, top, and bottom boundaries are aligned with the edges of the pixel and whose frustum tip is at the virtual viewpoint. The frustum extends outwards, away from the virtual camera and into the real-world environment. One such frustum is illustrated in the figure.

After [1], the only work, which is the closest to the similar efforts of both generating views and extracting depth using multi-baseline stereo, were done by [2][3]. We have shown the technical details of Range-Space Approach to efficiently generate views in [4]. The tracking and view synthesis can be integrated for surveillance and monitoring application [5][6]. Walking person's views and tracking views of a walking person were demonstrated in [5]. In [3], three panoramic images were used to extract 3D data from the scene. Using these 3D data, new views were generated. The work in [2] arranged their 51 standard cameras to form a studio dome. The authors searched and matched in the image space. Recently, they generated view by interpolating between two selected views instead of using texture map [7]. McMillan and Bishop [8], who first introduced the term of "image-based rendering [9][10]," have also generated panoramic images by rotating cameras (although the work is not involved in multiple baseline stereo). They devised an efficient mean of transferring known image disparity values between cylindrical panoramic images to a new virtual view. The work of [11] used simple edgel primitive to add realism and details using recovered view-dependent texture maps and depth displacements. Multiple Perspective Interactive (MPI) Video [12] utilized a sea of cameras, which were arranged facing inwards at the central region of activities, to perform model-based motion analysis to a set of image sequences. Three-dimensional models of these moving objects were computed and integrated with a priori environmental models.

In the following section, we present the details of Range-Space Search, Match, and Render. In Section 5, we show the results of virtual view synthesis and smooth virtual walkthroughs.

2. Range-Space Search

2.1. Search Length Determination

Searching in the range space has different types of challenges than searching directly in the disparity/image space. In the image space, the amount of search is bounded by the available number of pixels or simply by the image resolution. We are trying to utilize this resolution aspect to limit the search in the range space as well, but with more complexity involved. The search criterion is such that the interval between two voxels contains maximum depth resolution and accuracy. Also, mean time, the overlapping 3D region in the search has to be minimized to reduce repeated computational effort. Put in other words, we would like to move exactly one pixel away from the current pixel location to the next.

Searching in the range-space has four important benefits:

1) It allows having the maximum freedom to control the span of range to traverse.
2) It allows choosing the depth resolution to estimate.
3) The image-rectifying process is embedded in the range-space search process.
4) The motion of each camera's pixel on the epipolar lines are directly controlled by the motion of the voxel in the range space with respect to the placement and resolution of each individual camera.

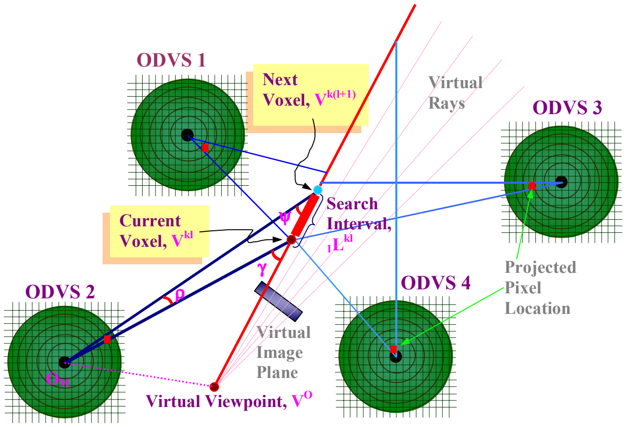

Figure 2 Range-Space Search. The criterion is to preserve the maximum depth resolution. In this example, ODVS 2 that contains the least visual information decides where the next voxel is.

Figure 2 explains the searching algorithm in the range space. Points V^O, V^{kl}, and O_M are known. V^{kl} is the l^{th} voxel on the k^{th} ray; O_{Mi} is the i^{th} mirror's focal point. Then the angle γ can be calculated with cosine's law. Based on the projected pixel location of V^{kl} and the assumption of a ray cone, the pixel resolution can be determined as $\rho \approx \tan^{-1}\frac{1}{r}$, where r is the radius from the ODI center to the projected pixel location. This angular resolution is purely a function of where the projected pixel location is. The line that connects points V^{kl} and $V^{k(l+1)}$ has to pass through the center of the circular cone, and both points are on the perimeter edge. Finally, the searching interval ${}_1L^{kl}$ (or the 1-step search length from the l^{th} voxel to the $(l+1)^{\text{th}}$ voxel on the k^{th} ray) is found using sine's law. These searching intervals of all cameras are compared; the shortest one decides the next voxel location. Given a voxel, V^{kl}, at the l^{th} point on the k^{th} ray, its corresponding pixel can be found in the ODI, $\mathbf{I} = \{\mathbf{I}_1, \cdots, \mathbf{I}_N\}$, as $\mathbf{P}^{\mathbf{kl}} = \{P_1^{kl}, \cdots, P_N^{kl}\}$, where

$P_1^{kl} \in \mathbf{I}_1, \cdots, P_N^{kl} \in \mathbf{I}_N$.[1] The mechanical searching cycle repeats until none of the pixels moves any further while the search length approaching infinity. In other words, all of the voxels from thereon are projected into the same set of pixels, $\mathbf{P^{kl}}$.

The trajectory of range-space search is sinusoidal when the virtual rays are projected to the ODI (Figure 3). The sinusoidal curves have discontinuities that are caused by the non-matching regions. It is now obvious that range-space search can accommodate the problem of Window Cutoff (corresponding windows that do not represent the same surface).

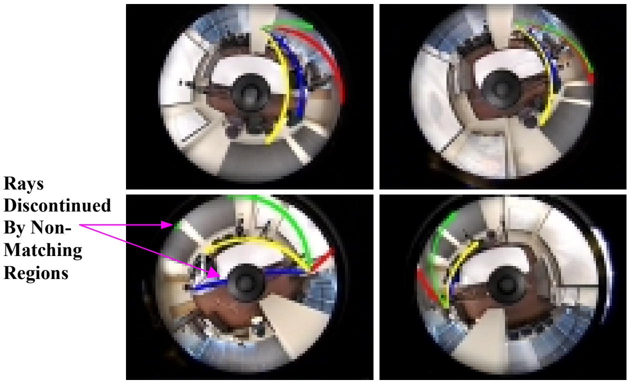

Figure 3 Trajectory of virtual rays in ODI.

2.2. Matching Template Derivation and Adjustments

In this section, we describe how to solve the Scaling and Foreshortening Effects. To solve the Scaling Effect (object appears larger in a closer camera than in a farther one) caused by the wide-baseline configuration, the size of the matching templates have to be adjusted in accordance with the camera arrangement. The derived matching templates will have the same visual information $\mathrm{f}\left(T_i^{kl}\right) = \mathrm{f}\left(T_j^{kl}\right)$, but with different template sizes $\Delta\left(T_i^{kl}\right) \neq \Delta\left(T_j^{kl}\right)$, where $i, j = 1, \cdots, N$ and $i \neq j$. The set of templates is denoted as $\mathbf{T^{kl}} = \left\{T_1^{kl}, \cdots, T_N^{kl}\right\}$, where $T_1^{kl} \in \mathbf{I}_1, \cdots, T_N^{kl} \in \mathbf{I}_N$. In general when $\mathrm{f}\left(T_i^{kl}\right) = \mathrm{f}\left(T_j^{kl}\right)$, $\Delta\left(T_i^{kl}\right)$ is not equal to $\Delta\left(T_j^{kl}\right)$, where $\Delta\left(T_i^{kl}\right) < \Delta\left(I_i\right)$. Again, $\mathrm{f}\left(T_i^{kl}\right) \approx \mathrm{f}\left(T_j^{kl}\right)$ and $\Delta\left(T_i^{kl}\right) \approx \Delta\left(T_j^{kl}\right)$ only if $\Delta(T_i^{kl}) \to 0$ and $\left\|V^{kl} - O_{Ci}\right\| \to \infty$, or $\left\|V^{kl} - O_{Ci}\right\| = \left\|V^{kl} - O_{Cj}\right\|$. Every voxel in V^{kl} corresponds to a group of pixels, i.e. a template. In a wide-baseline omni-directional stereoscopic system, characterized by $\left\|V^{kl} - O_{Ci}\right\| >> \left\|V^{kl} - O_{Cj}\right\|$, the closest sensor to the voxel has the resulting larger template size as oppose to the smallest one for the farthest sensor. Again, due to the non-uniformity in the omni sensing, the pixel resolution also need be taken into account besides the consideration of the distance between the voxel and the sensor itself.

Typical matching techniques in the image space utilizing fixed template size are unable to solve this scaling effect. *Their assumption of having fixed template size for every camera means that the physical objects in the range space change shapes and sizes with respect to the cameras*. Obviously, that is physically incorrect! Wang and Ohnishi [13] had similar idea of adjusting templates that they projected segmented patches to the range space, and then performing the deformable template matching by hypothesis-and-verification procedure. Their method requires starting out in the image domain to get those edge features segmented. Segmentation of objects in the image is a difficult task and error-prone due to noise. In range-space approach, we assume that every object, which occupies the range space, has a spherical shape. The range-space algorithms, however, fixate the volume of the spherical object (volumetric template), not the template size in the image space. The sensor that sees the shortest radius of the spherical object (spherical radius) determines the volume of the sphere for template derivation. (Or the variation that the longest radius is used to determine the volume of the spherical object. That will guarantee that the farthest camera with the lowest angular resolution will see the minimum number of pixels specified in the volumetric template size, such as 3x3, 5x5, or 7x7. All other cameras' template, which has shorter distance and higher angular resolution to the current voxel, V^{kl}.)

Volumetric template can also easily accommodate the Foreshortening Effect. The volumetric template of the primary camera is rotated in many directions about the fixation point at the current voxel, V^{kl}. The matching template of the rest of the cameras is the back-projection of the primary camera's volumetric template to their respective image coordinate system. The volumetric template is simplified as a plane, which occupies the range space with its center aligned at the current voxel. When the template size is small or the true object's surface is large, the plane approximates the surface. The derived matching templates will have the same visual information $\mathrm{f}\left(T_i^{kl}\right) = \mathrm{f}\left(T_j^{kl}\right)$, but with different template shapes $\mathrm{s}\left(T_i^{kl}\right) \neq \mathrm{s}\left(T_j^{kl}\right)$. When compared to the work of Maimone and Shafer using Local Spatial Frequency representation [14], the rotation of 3D template is more intuitive; nevertheless, it is also more time-consuming.

Figure 4 shows four templates of four virtual voxels created using the methods discussed. Observe that the size

[1] Omni-directional Images (ODI) represents the world in the spherical coordinate system. Therefore, every voxel will have a corresponding pixel, provided that there is no occlusion.

of the templates appears different in different cameras. These templates' size is varying in accordance with the camera's distance to the voxels and with the voxels' corresponding pixel resolutions.

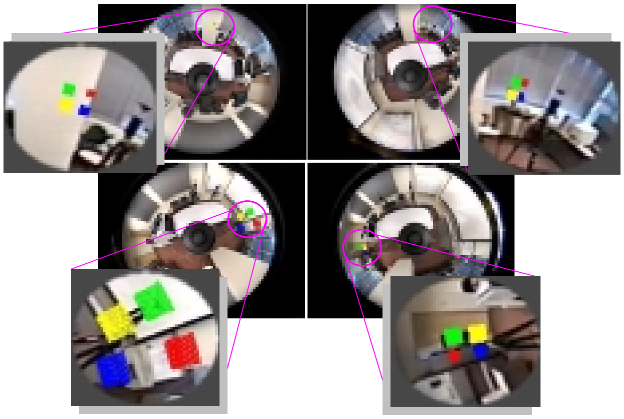

Figure 4 Templates of four virtual voxels in four different cameras.

3. Range-Space Match

We are processing both color images and color-edge images in parallel using identical mechanisms to search, derive matching templates, and match. Robust statistics is used to select camera set out of the set of images, because some of the cameras may be occluded or having significant color deviation from the others caused by the specular reflection or the sensor noise. After the preliminary (first pass) search and match are finished for the entire virtual image, the stored attributes derived from the area-based color match and the area-based color-edge match are combined and analyzed for each virtual pixel to locate the confident seeds for influencing their less confident neighboring pixels. Confident seeds are the virtual pixels, which have the global minimum for both color match and color-edge match occurred at the same estimated 3D location along the range search. In the second pass of the matching algorithms, these seeds are growing simultaneously and competing to correct their 8-neighbors' 3D and color. There are various levels of *Volume Growing*. When one level of seeds stops growing based on the characteristics of the matching curves, the next level of seeds are examined, uncovered, and grown. The final level is to fill the occluded regions or the regions where have high matching errors (high deviation in colors) using geometrical interpolation.

The following highlights and discusses the special features of range-space matching algorithms:

1) Volume growing in the range space is used to reduce false matches based on the characteristics derived from the matching curves using edge conformity. The matching criterions for volume growing are all based on the continuity assumption. Edges yield the information of possible geometrical discontinuity.
2) Robust statistics has to be applied to accommodate the problems of having non-uniform number of cameras being used for matching along the virtual ray.
3) A range-space match must recover 3D and color simultaneously. There is no reference camera/color that can be assumed unless the viewpoint is lying exactly at one of the cameras' center.
4) Every virtual pixel can be processed in parallel and with identical mechanisms independently from each other. Computation time is independent of the complexity of the scene composition. It is completely a function of the number of cameras used and the viewpoint location, as opposed to the feature-based matching that depends on the scene structure.

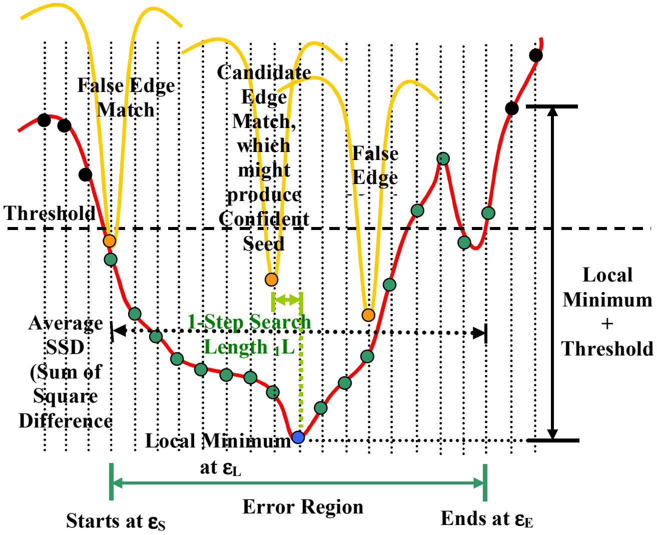

Figure 5 Definition of error region and from where the matching attributes are derived.

In regard with the first uniqueness, the concept of error region is used for deriving attributes in the volume growing process (Figure 5). A region is considered instead of investigating each individual raw matching error. An error region begins when the matching error is the first time lower than a set threshold and ends when the matching error is higher than the threshold plus the local minimum error. We do not pick the lowest error along the virtual ray as the best match. Both area-based color match and area-based color-edge match are used to support the correctness of each other's global minimum. They are processed in parallel using the same searching and matching mechanisms. The area-based color-edge match is relatively insensitive to the set threshold of the edge map, due to the verification support from the color match, and thus it is not as sensitive to the thickness and incompleteness of the detected edges. The method here is treating color-edge match as just another area-based match. The area-based color match is also rather insensitive to the choice of template size, because the matching results is not utilized directly to finalize a good match. It also has to be verified by a good edge match. Thus, no large (or coarse-to-fine) template is necessary. The matching does not assume a one-to-one correspondence. When the matching errors on a particular virtual ray are higher than the acceptable threshold (for example, a normalized value of 32x32), the pixel is marked low confident (either occlusion or specular highlight occurs).

In the matching process, due to the disparate views of multiple camera stereo, multiple true surfaces are possibly observed. The Uniqueness Constraint that traditionally applied to the disparity-space match becomes invalid in the case here. For a desired virtual image size of S, there exist S number of rays, $\mathbf{R}=\{\mathbf{R}^{1},\cdots,\mathbf{R}^{S}\}$ from the viewpoint, V^{O}. Each ray in $\mathbf{R}$ is discretized into collinear voxels, $\mathbf{R}^{\mathbf{k}}=\{V^{k1},\cdots,V^{k(q-1)},V^{kq}\}$, where $k=1,\cdots,S$ and $1\le q<\infty$. Considering the set of virtual rays, $\mathbf{R}$, each ray will eventually hit a surface. Therefore, there exists at least one voxel, V^{ka}, on each ray intersecting a physical surface, where $V^{ka}\in\mathbf{R}^{\mathbf{k}}$. The set of true voxels, which intersects physical surfaces, is denoted as ${}_{\mathbf{T}}\mathbf{R}^{\mathbf{k}}=\{{}_{T}V^{k1},\cdots,\ {}_{T}V^{k(n-1)},\ {}_{T}V^{kn}\}$ where ${}_{\mathbf{T}}\mathbf{R}^{\mathbf{k}}\subset\mathbf{R}^{\mathbf{k}}$, and ${}_{\mathbf{T}}\mathbf{R}^{\mathbf{k}}\neq\phi$. Since matching process includes noise, false matches exist. Then, the set of possible voxels can be ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}=\{{}_{P}V^{k1},\cdots,\ {}_{P}V^{k(q-1)},\ {}_{P}V^{kq}\}$, where ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}\subseteq\mathbf{R}^{\mathbf{k}}$, $k=1,\cdots,S$ and $1\le q<\infty$. Both multiple true matches and multiple false matches can exist in ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}$.

We consider a local minimum as a region of 3D points (“error region” in Figure 5). Eight attributes derived from every candidate $SSD^{k\varepsilon_{L}}$ (the error region with local minimum at ε_{L} that passes below the threshold line) are stored for this delayed matching process. Each local minimum forms its own region. The region starts when the error after ε_{S} is the first time lower than the threshold line and ends when the error at ε_{E} is higher than the threshold plus the local minimum error. It is important to note that the edge's ${}_{E}SSD^{k\varepsilon_{L}^{l}\prime}$ is recorded as the SSD of the color match at the local minimum of the color-edge match.

Recall that there are multiple of these candidates, $SSD^{k\varepsilon_{L}^{l}}$ and ${}_{E}SSD^{k\varepsilon_{L}^{l}}$, associated with any given virtual ray when multiple local minimums exist. Now that the error region is considered, the set of possible voxels is modified as ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}=\{{}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{0}}},\cdots,\ {}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{(q-1)}}},\ {}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{q}}}\}$, where ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}\subseteq\mathbf{R}^{\mathbf{k}}$, $k=1,\cdots,S$ and $0\le q<\infty$. When $q=0$, it means that there is no local minimum exists, that is when all SSD^{kl} are higher than the set threshold. Usually, this happens when occlusion or specular highlight occurs. Each element in ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}$ covers a number of voxels that spans the error region. The number of voxels includes ${}_{P}V^{k\varepsilon_{L}^{i}}$, ${}_{P}V^{k\varepsilon_{S}^{i}}$, and ${}_{P}V^{k\varepsilon_{E}^{i}}\in{}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{i}}}$, where $i=0,\cdots,q$.

Both multiple true matches and multiple false matches can exist in ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}$. It is necessary that ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}\cap\{{}_{T}V^{kl}\mid{}_{T}V^{kl}\in{}_{\mathbf{T}}\mathbf{R}^{\mathbf{k}}\}\neq\phi$, in order for the first, true voxel to be recovered for a ray at a particular viewpoint. To extract the first, true voxel out of ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}$, we make the following assumptions:

1) At ${}_{T}V^{kl}$, the voxel is seen by the majority number of cameras. That is ${}_{\mathbf{T}}\mathbf{T}^{\mathbf{kl}}>{}_{\mathbf{F}}\mathbf{T}^{\mathbf{kl}}$, when $T_{i}^{kl}\in{}_{\mathbf{T}}\mathbf{T}^{\mathbf{kl}}$ and we know that $\mathrm{f}(T_{i}^{kl})$ is true. In short, ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}\cap\{{}_{T}V^{kl}\mid{}_{T}V^{kl}\in{}_{\mathbf{T}}\mathbf{R}^{\mathbf{k}}\}\neq\phi$.
2) Every object's surface in view will be accompanied by sharp edges. All of the edges, which bound the surface of an object, can be detected.
3) Edge match is supported by the color match.
4) The rays that pass through the object borders have only one local minimum.
5) The object surfaces are continuous. Therefore, the distance of objects varies smoothly with viewing direction, except at object borders (defined by edges)— Continuity Constraint.
6) The images and the matching process have negligible noise.

The information at an edge is a more reliable one, which serves to locate the confident seed to uncover other true voxels. Assumption 2 indicates that there is at least one seed available. When both color match and color-edge match are used, the set of possible voxels becomes ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}=\{{}_{P}V^{k1},\cdots,\ {}_{P}V^{k(q-1)},\ {}_{P}V^{kq};\ {}_{P}E^{k1},\cdots,\ {}_{P}E^{k(p-1)},\ {}_{P}E^{kp}\}$, where $p\le q$. Due to Assumption 3, $\forall\ {}_{P}E^{kb}\in\{{}_{P}E^{k1},\cdots,\ {}_{P}E^{kp}\}$ from the color-edge match, there is a corresponding ${}_{P}V^{ka}\in\{{}_{P}V^{k1},\cdots,\ {}_{P}V^{kq}\}$ from the color match, and $\left\|{}_{P}V^{ka}-V^{O}\right\|=\left\|{}_{P}E^{kb}-V^{O}\right\|$. When both color match and color-edge match are used with the concept of volume growing, under Assumption 3, the set of possible voxels is modified again as ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}=\{{}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{0}}},\cdots,\ {}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{(q-1)}}},\ {}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{q}}};\ {}_{\mathbf{P}}\mathbf{E}^{\mathbf{k\varepsilon^{0}}},\cdots,\ {}_{\mathbf{P}}\mathbf{E}^{\mathbf{k\varepsilon^{(p-1)}}},\ {}_{\mathbf{P}}\mathbf{E}^{\mathbf{k\varepsilon}^{p}}\}$ where $p\le q$. The elements in each ${}_{\mathbf{P}}\mathbf{E}^{\mathbf{k\varepsilon^{b}}}$ must satisfy the three conditions: (1), $\left\|{}_{P}V^{k\varepsilon_{L}^{a}}-{}_{P}E^{k\varepsilon_{L}^{b}}\right\|\le{}_{1}L^{k\varepsilon_{L}^{a}}$; (2), ${}_{E}SSD^{k\varepsilon_{L}^{b}\prime}\le\overline{SSD^{k\varepsilon^{a}}}$; (3), $L^{k\varepsilon_{S}^{a}}\le{}_{E}L^{k\varepsilon_{L}^{b}}\le L^{k\varepsilon_{E}^{a}}$, i.e. ${}_{\mathbf{P}}\mathbf{V}^{\mathbf{k\varepsilon^{a}}}$ and ${}_{\mathbf{P}}\mathbf{E}^{\mathbf{k\varepsilon^{b}}}$ are in the same error region. Assumption 4 says there is at most one local minimum. When multiple edges exist in a ${}_{\mathbf{P}}\mathbf{E}^{\mathbf{k}\varepsilon^1}$ within the global minimum, we pick the first candidate edge match; the set at the object borders becomes ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}'} = \left\{{}_{P}V^{k\varepsilon_L^1};\ {}_{P}E^{k\varepsilon_L^1}\right\}$. With the set ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}'} = \left\{{}_{P}V^{k\varepsilon_L^1};\ {}_{P}E^{k\varepsilon_L^1}\right\}$ exists, a confident seed is found. Based on the Continuity Constraint, given a true voxel ${}_{P}V^{k\varepsilon^a}$ in ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}}$, there exist at least another true voxel ${}_{P}V^{k_8\varepsilon^b}$ in the 8-neighboring of $\mathbf{R}^{\mathbf{k}}$, such that ${}_{P}V^{k_8\varepsilon^b}$ is within the object borders and $\left\|{}_{P}V^{k\varepsilon^a} - {}_{P}V^{k_8\varepsilon^b}\right\| \leq {}_{1}L^{k\varepsilon^a}$. The new point is turned into another seed pixel at the next iteration. Eventually, a surface is grown within the bound of the edges. The minimum number of surface is 1, while the maximum number of surface equals the number of rays having the set of ${}_{\mathbf{P}}\mathbf{R}^{\mathbf{k}'} = \left\{{}_{P}V^{k\varepsilon_L^1};\ {}_{P}E^{k\varepsilon_L^1}\right\}$. The grown surface will have a maximum deviation of $\sum_{k=1}^{S} {}_{1}L^{k\varepsilon^a}$ from the first seed.

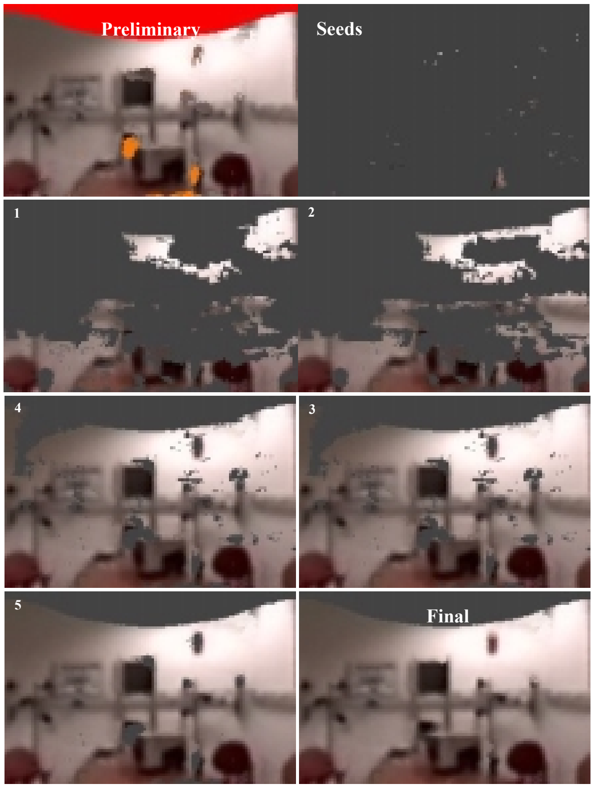

Figure 6 Snap shots of the Volume Growing process. The final image (bottom-right) shows improvement over the initial image (top-left).

When more than one seed are competing for influencing the same neighboring pixel during the growing process and if all of them meet the continuity constraint, all of the depths along the viewing direction are valid. Then, because of the back-to-front projection concept, the farther depths are occluded by the closer ones. The closest depth dominates the filling of virtual pixel's color. That means that the voxel chosen for the view synthesis is the first element in ${}_{\mathbf{T}}\mathbf{R}^{\mathbf{k}}$, which has the closest distance to the viewpoint.

Figure 6 shows a few snap shots of the growing process. The synthesized images are each 212x140. The top left image is synthesized by picking the lowest error along each virtual ray. The red-coded region is due to the cropping of the original ODI, while the orange-coded regions are the low confident regions. Visual errors can be easily spotted. The final image is the result from Volume Growing after a total of 5 steps. The steps are indicated with the numbers.

4. Range-Space Render

After a good match is found along the virtual ray, we recover both the voxel, $V^{k\varepsilon^a}$, and its corresponding pixel, $P_i^{k\varepsilon^a}$, for every valid camera within the camera set, $i = 1, \cdots, N_{CS}^{k\varepsilon^a}$. The valid cameras in a camera set are the cameras, which do not violate the matching conditions and are not the outliers. Their pixel's color, $P_i^{k\varepsilon^a}$, are mixed. The composite color, $P_V^{k\varepsilon^a}$, is used to fill the virtual pixel. This process needs to be carried out carefully; otherwise, incorrect pixel color will cause visually unpleasant virtual view, even if the 3D is precisely determined (correct range but incorrect color type of error occurs).

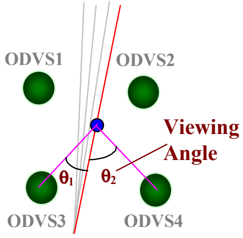

Figure 7 Viewing angle is the angle between the virtual ray and the ray connecting the valid camera's center to the estimated voxel.

The virtual pixel color is the weighted composite color of multiple cameras' pixel. The weights are a function of the valid cameras' viewing angle (see Figure 7) to the estimated voxel and their respective matching error at that voxel. They are independent of the cameras' distance to the subject voxel. When using this weighted scheme, not a sole camera's pixel can fully dominate the composite color, even when the virtual viewpoint coincides exactly at one of the camera's center. Therefore, in order for the synthesized view to be clear and sharp, so must the estimated range be accurate. In other words, *when we see a clear synthesized view, we are quite sure that the* *underlying 3D of the virtual pixels are accurate, unless the color in the scene is rather homogeneous.*

When the viewing angle of a camera is zero, which means the virtual ray completely coincides with the camera's physical ray, the camera's pixel has stronger influence in the process of virtual pixel synthesis. The opposite case is when the angle is 180 degrees apart, then that camera pixel will have the least influence on the composite color. Also, the greater the matching error, the smaller the influence a camera has on the color synthesis. The composite color of a virtual pixel can be computed as

$$P_V^{k\varepsilon^a} = \frac{\sum_{j=1}^{N_{CS}^{k\varepsilon^a}} \left(\frac{\sum_{i=1}^{N_{CS}^{k\varepsilon^a}} \theta_i - \theta_j}{\sum_{i=1}^{N_{CS}} \theta_i} * \frac{\sum_{i=1}^{N_{CS}^{k\varepsilon^a}} SSD_i^{k\varepsilon^a} - SSD_j^{k\varepsilon^a}}{\sum_{i=1}^{N_{CS}} SSD_i^{k\varepsilon^a}} * P_j^{k\varepsilon^a} \right)}{\frac{\left(N_{CS}^{k\varepsilon^a} - 1\right)^2}{N_{CS}^{k\varepsilon^a}}}$$

,

where θ are the viewing angles, SSD are the matching errors, $P_j^{k\varepsilon^a}$ are the pixel color of physical cameras, and $N_{CS}^{k\varepsilon^a}$ is the number of valid cameras within a camera set. $P_V^{k\varepsilon^a}$ is the resulting composite color. Three channels, Red, Green, and Blue are performed separately.

An alternate form of weighting function that does not include the matching error can be expressed as

$$P_V^{k\varepsilon^a} = \frac{\sum_{j=1}^{N_{CS}^{k\varepsilon^a}} \left(\frac{\sum_{i=1}^{N_{CS}^{k\varepsilon^a}} \theta_i - \theta_j}{\sum_{i=1}^{N_{CS}^{k\varepsilon^a}} \theta_i} * P_j^{k\varepsilon^a} \right)}{N_{CS}^{k\varepsilon^a} - 1}.$$

In this case, when a physical ray coincides with the virtual ray, i.e. $\theta = 0$, the camera will dominate the composite color, especially if only two cameras are valid within the camera set ($N_{CS}^{k\varepsilon^a} = 2$).

5. Experimental Results

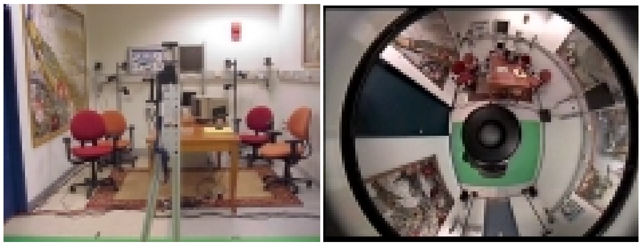

Figure 8 A scene for visual modeling.

Virtual view synthesis experiments are performed inside a room (shown in Figure 8). The room size is 258 x 128 sq. in. It is a box-like structure. The omni-directional images are captured using a single hyperboloidal mirror with vertical field of view of about 270 degrees. The images were taken at regular intervals defined by the grids on the green cardboard (Figure 9). Camera 1 is the reference camera coordinate, from where all the estimated 3D in the experiments are measured. The highlighted cameras are the ones formed a video cluster for those viewpoints. There are a total of 36 cameras/images taken inside the scene. These cameras can form various combinations of video clusters with many possible choices of baselines and number of cameras.

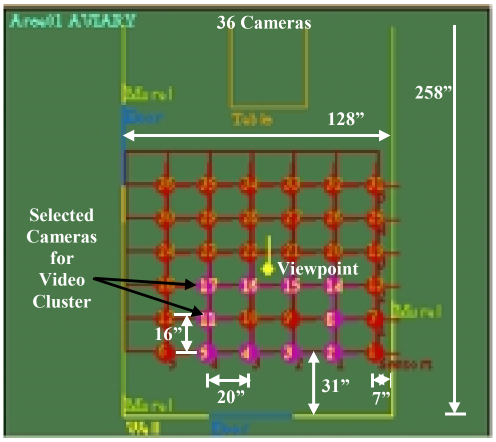

Figure 9 Sensor layout.

5.1. Virtual View Synthesis

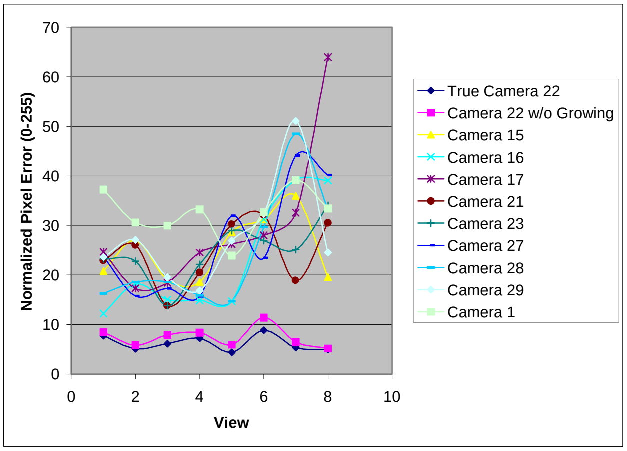

Figure 10 Error comparison with the cameras in the video cluster and with the true camera.

For the experiment, we use the Cameras 15, 16, 17, 21, 23, 27, 28, and 29 for view synthesis. The virtual viewpoint is at the Camera 22's projection center. The synthesized panoramic view is compared with the real panoramic view at the same viewpoint, with the panoramic view of the eight cameras that were used for matching, and with the panoramic view of a camera that is the farthest from the cluster and is uncorrelated. The result from the farthest camera gives us a reference for comparison. From

Figure 10, the plot clearly shows that the synthesized views have the lowest error when compared to the real views of Camera 22. None of the cameras that were used for matching comes close to the resemblance between the synthesized views and Camera 22's views. Camera 1, which is the farthest camera away from the video cluster, shows no relation to the synthesized views, and its error remains consistently high. Without volume growing (average error is 7.43), the average view synthesis error is 1.22 per pixel value higher than the one with volume growing (average error is 6.22). The improvement is 16.35%.

Figure 11 shows real and synthesized views. The numbers on the figures give names to Panoramas 1 to 6. Panoramas 1 to 3 are real. They are from Camera 29, 15, and 22, respectively. Cameras 15 and 29 have the widest baseline of 51 inches within the video cluster. Synthesized Panoramas 4 and 5 are supposed to resemble Panorama 3. Panorama 5 is the one before volume growing. Errors can be readily observed. Again, the red-coded regions in Panorama 5 are due to the cropping of the original ODI. The orange-coded regions are the occluded regions or the regions where high color deviation occurs due to specular lighting effect. The color-coded regions with red, orange, and gray are not compared. Panorama 6 shows the difference between Panorama 3 (real) and Panorama 4 (synthesized). The errors are mainly at the high frequency regions. Figure 12 shows the close-up views before and after volume growing. The dramatic improvement in view synthesis after volume growing can be easily observed.

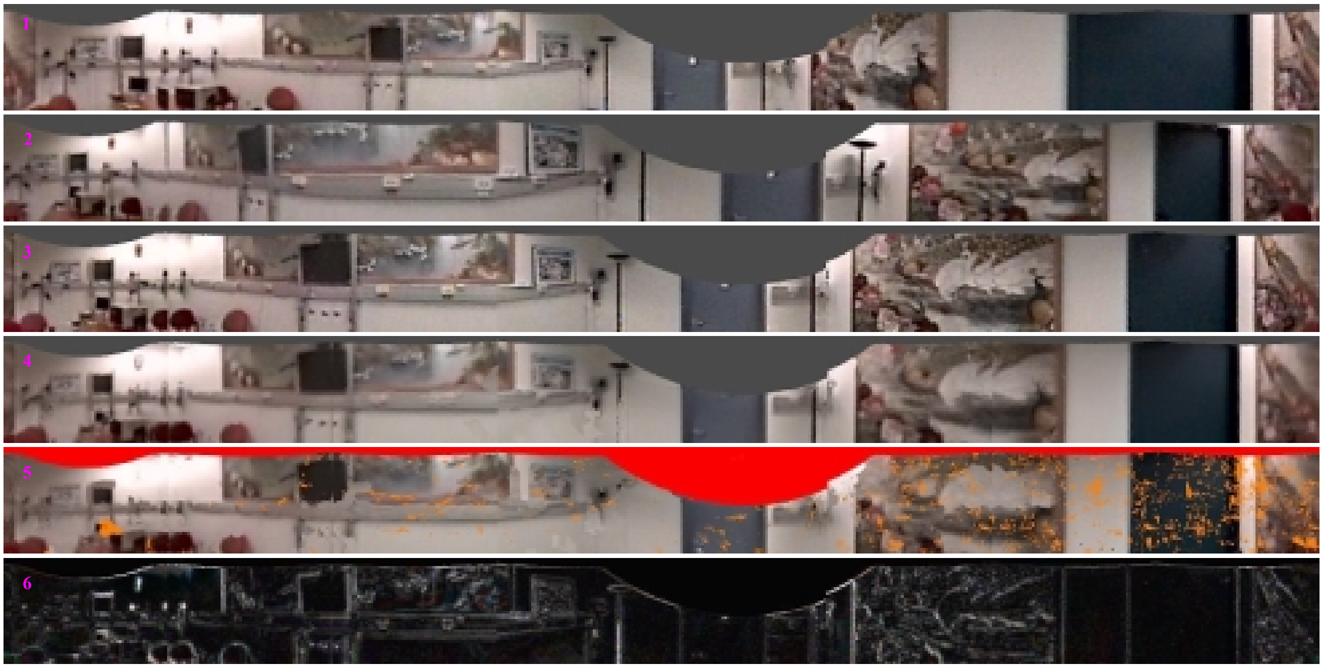

Figure 11 Real and synthesized panoramic views.

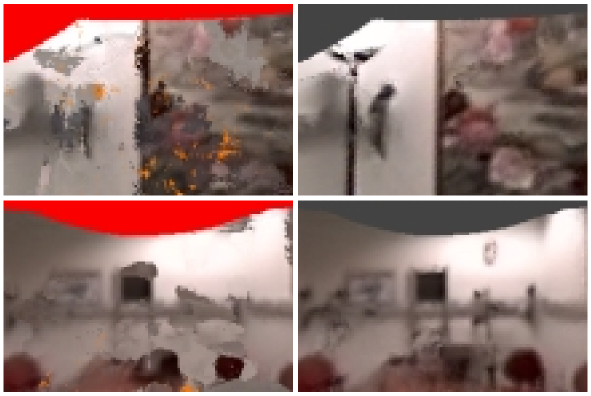

Figure 12 Synthesized views before (left) and after volume growing (right).

5.2. Demonstrations of Smooth 3D Virtual Walkthrough

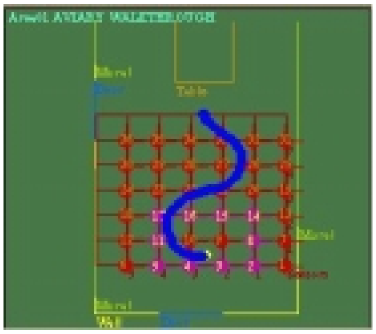

Figure 13 Smooth walk path and video clusters.

The smooth walk path is shown in Figure 13. The walk path has a total of 168 inches in length. It is walking from the bottom yellow point to the top point. Three video clusters are available. Each cluster includes 10 cameras. Some cameras are shared by two clusters. A total of 22 cameras were used for the entire path. When the viewpoints are within and a little beyond the cluster (based on the viewing direction and viewpoint), that cluster is selected to generate views. Views 1 to 39 used the first cluster, while views 40 to 77 used the second one. The rest was using the third cluster. Ninety views were synthesized along this path. In this work, we show the views at discrete sampling intervals of every 8th view in Figure 14. The view sequence is from left to right and top to bottom. Figure 15 shows the 12 consecutive smooth views, from the 75th view to the 86th view, toward the end of the walkthrough after passing ahead the third video cluster. Smoothness in the views can be easily observed. The walkthrough includes both simultaneous translation and rotation. The peacock on the mural becomes larger in view and the chair is seen less at the end of the walk.

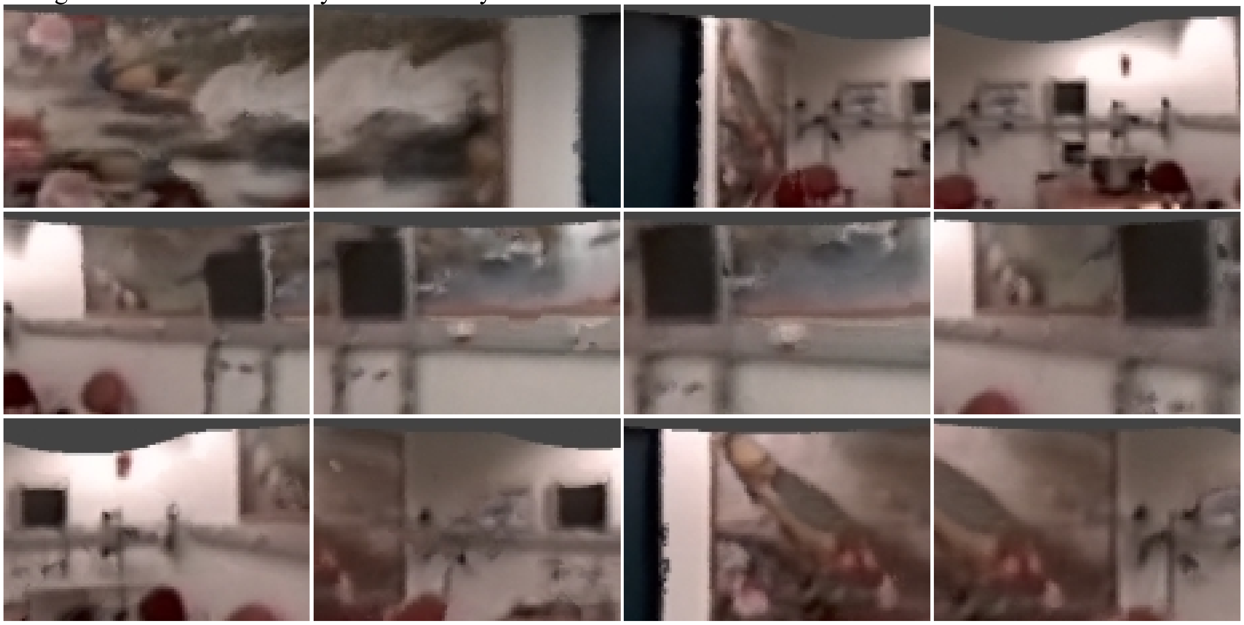

Figure 14 Twelve discrete synthesized views extracted from smooth virtual walkthrough.

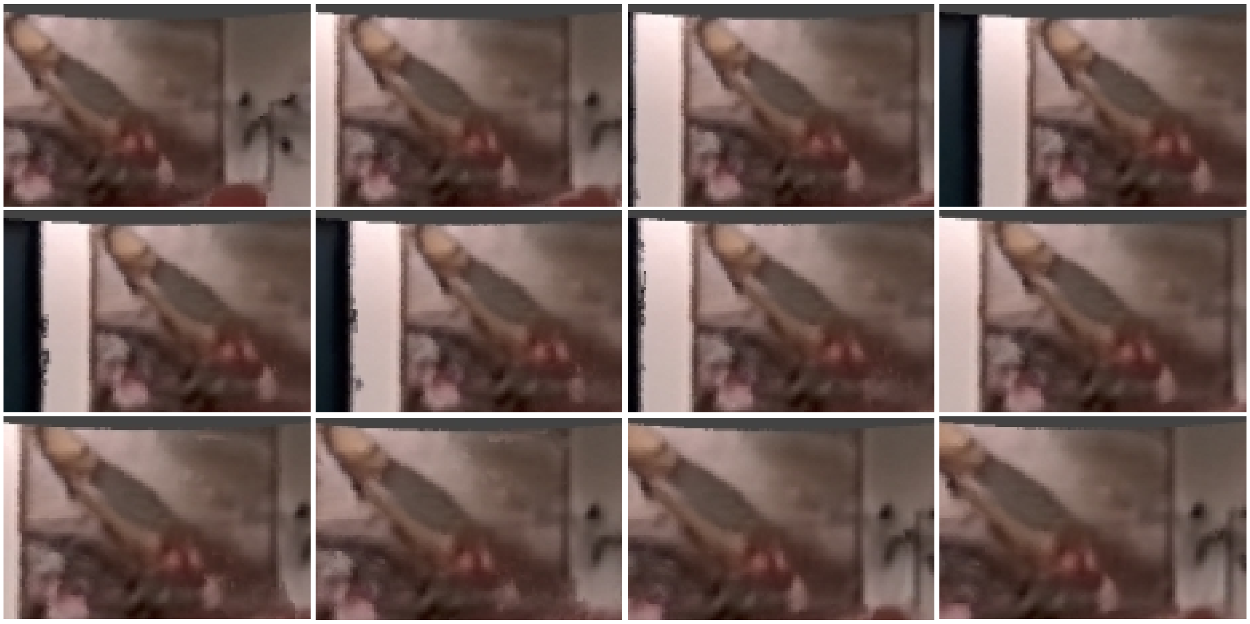

Figure 15 Twelve consecutive synthesized views extracted from the 75th view to the 86th view.

6. Concluding Remarks

In this paper, we have introduced a range-space search, match, and render technique. This system allows viewers freely walk through a remote dynamic environment concurrently with their individually desired viewpoint. When these views are synthesized from several wide-view ODI, only the necessary 3D are derived. The arrangement of the cameras can be arbitrary and the searching mechanism is general. Range-space search overcomes the problems of scaling effect, foreshortening effect, and window cutoff—three of the five major research challenges of wide-baseline omni-directional stereo. The multiple baseline stereo is also modified to handle occlusion and specular highlight. Cameras within a video cluster are chosen based on robust statistics. Volume growing has the major effects on reducing most of the false matches. Error region, which is the basis for volume growing, is defined and attributes are derived from the matching curves. Both area-based color match and area-based color-edge match are processed in parallel with identical mechanisms. The derived attributes are combined and analyzed in the range space to locate confident seeds. These confident seeds are used to correct their 8-neighboring pixels' 3D and color based on continuity constraint. Low confident regions are filled with geometrical interpolation. Virtual pixel color is synthesized using parameters of viewing angles and matching errors. The results show clear virtual view synthesis. View synthesis has an overall average error of 6.22 normalized pixel error and an overall average improvement of 16.35% over the results before volume growing. Viewers actively explore the scene. Smooth walkthrough is put together by pieces of views through time.

7. Acknowledgements

Our research was supported by the California Digital Media Innovation Program (DiMI) in partnership with Sony Electronics and Compaq Computers. We are pleased to acknowledge the assistance of our colleagues in the CVRR Laboratory, especially Mr. Rick Capella and Nils Lassiter who helped in the design of the AVIARY testbed.

REFERENCES

[1] M. Okutomi and T. Kanade, "A Multiple Baseline Stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.15, (no.4), p.353-63, April 1993.

[2] P. J. Narayanan, Peter W. Rander, and Takeo Kanade, "Constructing Virtual Worlds using Dense Stereo," *Proceedings of 6th IEEE International Conference on Computer Vision*, Bombay, India, p.3-10, January 1998.

[3] S. B. Kang and P. K. Desikan, "Virtual Navigation of Complex Scenes using Clusters of Cylindrical Panoramic Images," *Proceedings of Graphics Interface*, Vancouver, BC, Canada, p.223-32, June 1998.

[4] K. Ng, *3D Visual Modeling and Virtual View Synthesis: A Synergetic, Range-Space Stereo Approach using Omni-Directional Images*, Ph.D. Dissertation, University of California, San Diego, March 2000.

[5] K. Ng, H. Ishiguro, M. Trivedi, and T. Sogo, "Monitoring Dynamically Changing Environments by Ubiquitous Vision System," Proceedings of IEEE Workshop on Visual Surveillance, Fort Collins, Colorado, p.67-73, June 1999.

[6] K. Ng, H. Ishiguro, and M. Trivedi, "Multiple Omni-Directional Vision Sensors (ODVS) based Visual Modeling Approach," Conference and Video Proceedings of IEEE Visualization '99, San Francisco, California, October 1999.

[7] H. Saito, S. Baba, M. Kimura, S. Vedula, T. Kanade, "Appearance-based Virtual View Generation of Temporally-Varying Events from Multi-Camera Images in the 3D Room," *Proceedings of 2nd International Conference on 3-D Digital Imaging and Modeling*, Ottawa, Ont., Canada, p.516-25, October 1999.

[8] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-based Rendering System," *Computer Graphics (SIGGRAPH'95)*, p.39-46, August 1995.

[9] S. Kang, "A Survey of Image-based Rendering Techniques," *Proceedings of the SPIE*, vol.3641, San Jose, California, p.2-16, January 1999.

[10] Z. Zhang, "Image-based Geometrically Correct Photorealistic Scene/Object Modeling: A Review," *Proceedings 3rd Asian Conference on Computer Vision*, p.279-88, 1998.

[11] P. Debevec, C. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-based Approach," *SIGGRAPH'96*, New Orleans, Louisiana, p.11-20, August 1996.

[12] S. Moezzi, L. Tai, and P. Gerard, "Virtual View Generation for 3D Digital Video," *IEEE MultiMedia*, vol.4, (no.1), p.18-26, May 1997.

[13] ZengFu Wang and N. Ohnishi, "Deformable Template based Stereo," *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics*, vol.5, Vancouver, BC, Canada, p.3884-9, October 1995.

[14] M.W. Maimone and S.A. Shafer, "Modeling Foreshortening in Stereo Vision using Local Spatial Frequency," *Proceedings of 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol.1, Pittsburgh, PA, p.519-24, August 1995.