Range Segmentation Using Visibility Constraints

Leonid Taycher (lodrion@ai.mit.edu) and Trevor Darrell (trevor@ai.mit.edu)
Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 02139

Abstract.
Visibility constraints can aid the segmentation of foreground objects in a scene observed with multiple range imagers. Points may be labeled as foreground if they can be determined to occlude some space in the scene that we expect to be empty. Visibility constraints from a second range view can provide evidence of such occlusions. We present an efficient algorithm to estimate foreground points in each range view using explicit epipolar search. In cases where the background pattern is stationary, we show how visibility constraints from other views can generate virtual background values at points with no valid depth in the primary view. We demonstrate the performance of both algorithms for detecting people in indoor office environments.

1. Introduction

Object segmentation is an important preliminary step for many high-level vision tasks, including person detection and tracking. State-of-the-art systems (Wren et al., 1995, Brumitt et al., 2000, Beymer and Konolige, 1999, Grimson et al., 1998) use foreground/background classification followed by pixel clustering and analysis. A common approach to foreground detection is to maintain a background model and label all pixels that differ significantly from this model as foreground. Several segmentation methods have been proposed which use background models based on color/intensity (Wren et al., 1995, Toyama et al., 1999, Stauffer and Grimson, 1999), stereo range (Beymer and Konolige, 1999) or both (Krumm et al., 2000, Harville et al., 2001).

Ideally, these systems should be robust to rapid illumination variation, such as from outdoor weather or indoor video projection systems. Generally, non-adaptive color-based models suffer from varying illumination. Adaptive color models (Stauffer and Grimson, 1999) are more stable under lighting changes, but can erroneously incorporate objects that stop moving into the background model. Range-based background models can be illumination invariant, but are usually sparse when obtained from optical stereo. Sparse range data causes an inherent ambiguity in classifying valid range values in the locations where the model is invalid. If all such locations are labeled as foreground, then pixels where the range data becomes available due to illumination change (e.g. shadows or overhead projection on the otherwise uniform walls) will be detected as foreground (Figure 1(d)). The opposite approach, where only locations where both model and the input range values are valid are considered, may lead to underdetection of the foreground objects (Figure 1(g)). To avoid such ambiguity and the resulting illumination dependence, stereo range background models have been either used in conjunction with color models (Harville et al., 2001), or are built using observations from widely varying illumination and imaging conditions (Darrell et al., 2001).

We overcome the problems with sparse range backgrounds by using visibility constraints obtained from multiple range views of the scene. We define foreground as a set of points occluding the free space that is expected to be visible by the camera, in contrast to background-based approaches that describe it as deviation from the expected observation (the model). The occlusion relationships may be computed for every collection of concurrent scene views, leading us to an instantaneous foreground segmentation algorithm (Section 4); or, if the “background” scene is expected to remain static, they may be precomputed once, resulting in a virtual background generation algorithm (Section 5). We further formulate an approach to clustering detected foreground points based on visibility constraints(Section 4.1).

Our method relies on access to multiple widely spaced range views of the scene, that may be available in surveillance or smart environment applications. Access to range views simplifies detection of the free space, as all of the space between the observed 3-D location and the imager’s center of projection can be presumed to be empty. The occlusion relationships are computed by exploring the observed point’s optical ray in the other range views (e.g. by scanning a point’s

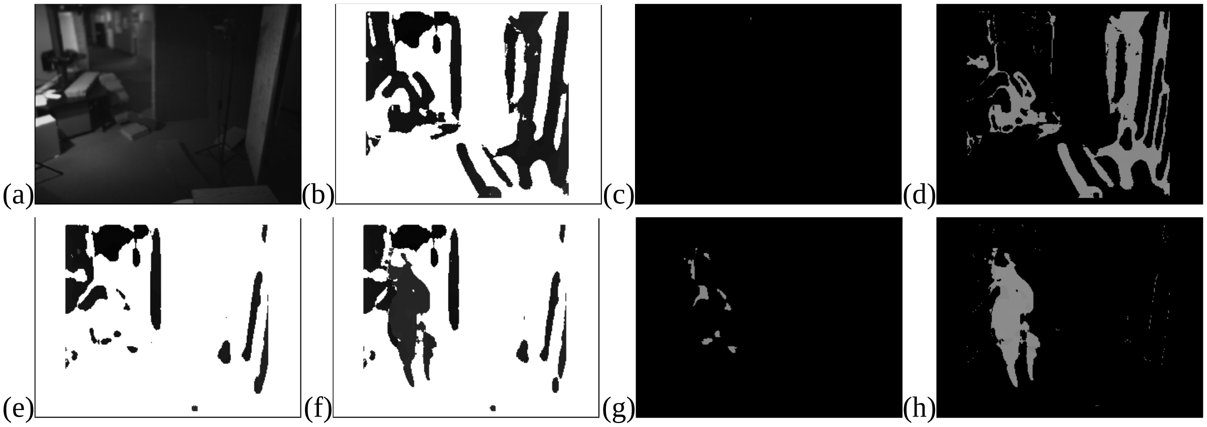


Figure 1. Background Subtraction ambiguities. Given a sparse (invalid disparity values are shown in white) background model (e) of a scene (a), a new range image with a foreground person (f), and a new range image with no foreground object but a changed illumination condition (b), we see that a conservative segmentation (c,g) misses many foreground points on the object. However the alternative approach (d,h), has many false positives when the illumination changes, and erroneously includes background points in the foreground. To achieve illumination invariance one must adopt a conservative approach and obtain very dense range background models.

epipolar lines). We believe ours is the first method for range image segmentation using image-based (non-voxel) free space computation.

We begin with a review of the previous work on 2-D and 3-D foreground segmentation and object modeling. We then give an overview of the notation used in the rest of the paper. Section 4 describes our instantaneous foreground segmentation algorithm, and proposes a visibility constraint-based clustering algorithm that relies on both proximity of pixels in the image plane and the estimated extent of the objects along the corresponding optical rays. In Section 5 we propose a method for creating dense virtual backgrounds for stationary scenes. Finally, we demonstrate the results using our algorithm for detecting people in an indoor office environment.

2. Related Work

A common approach to foreground object segmentation is to model the expected distribution of colors in each background pixel as either a single (Wren et al., 1995, Matusik et al., 2000) or a mixture of (Stauffer and Grimson, 1999) of Gaussians, or using a Kalman filter (Ridder et al., 1995). The pixel is then labeled as foreground if its value has small probability given the model. The single Gaussian approach performs well if lighting conditions are unchanged or vary slowly. While it allows for fast segmentation, it is not robust to significant illumination changes. The mixture of Gaussians and Kalman filter approaches can handle some illumination variations, but recover slowly after fast local changes (e.g. from overhead projection). The color-based segmentation approaches can also undersegment the scene when the foreground objects have color similar to learned background.

Range-based segmentation methods e.g. (Beymer and Konolige, 1999) are much more robust to lighting changes, but are unreliable in the absence of dense background models (Figure 1). Most stereo systems unfortunately produce sparse range values in the low texture regions. To overcome this, (Krumm et al., 2000, Harville et al., 2001) use combined color and range mixture model (allowing invalid range). Several approaches for clustering foreground pixels using range have been proposed, such as grouping neighboring points that have similar disparities (Krumm et al., 2000).

Our epipolar line search is a similar computation to algorithms proposed for the rendering of image-based visual hulls (Matusik et al., 2000). The key difference is that our method takes as input unsegmented noisy range data and evaluates 3-D visibility per ray, while the visual hull method presumes segmented color images as input and simply identifies non-empty pixels along the epipolar lines in other views. Also related are space carving and coloring methods (Kutulakos and Seitz, 2000, Slabaugh et al., 2001), which split the space into *voxels* and use color consistency across multiple cameras to locate opaque voxels and to detect free space. These methods are quite general, and work with an arbitrary set of monocular views. They also require the construction of a volumetric representation of the scene for reconstruction or segmentation. We are interested in algorithms that perform segmentation solely in the image domain, without computing a volumetric reconstruction.

3. Notations

While in the algorithms described below we assume stereo disparity input, they may be adapted to use any appropriate range inputs from any depth sensor.

An ideal (rectified) stereo rig may be completely described by the baseline B, focal length f and the image coordinates of the principal point (c_x, c_y). The following equations describe a relation between point (x, y) in the disparity image I_D and the corresponding 3-D location (X, Y, Z).

$$\begin{cases} \overline{x} = x - c_x = f\dfrac{X}{Z} \\ \overline{y} = y - c_y = f\dfrac{Y}{Z} \\ d = I_D(x, y) = f\dfrac{B}{Z} \end{cases} \qquad (1)$$

As has been shown in (Demirdjian and Darrell, 2001) , we can express the transformation between camera associated disparity coordinates (x, y, d) and camera-centered Euclidean coordinates (X, Y, Z) as a linear projective space transformation,

$$\mathbf{P}_{/E} \simeq \mathbf{T}^{\mathbf{E}}_{\mathbf{D}} \mathbf{p}_{/D} \qquad (2)$$

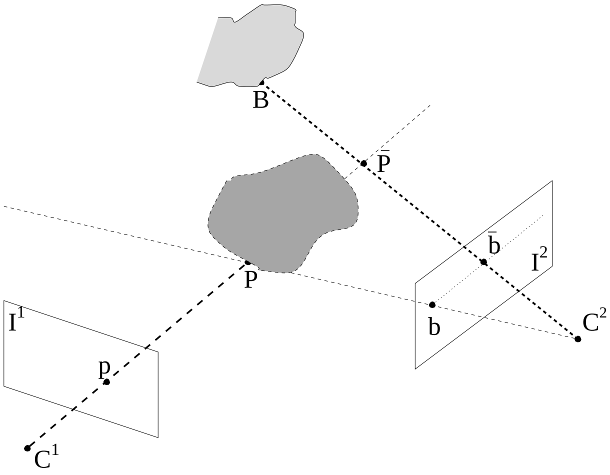


Figure 2. Visibility-based segmentation. Observation of $\mathbf{B}$ in C^2 allows us to infer that $\mathbf{P}$ in C^1 is foreground. Point $\mathbf{B}$ visible in I^2 (projecting to C^2 disparity point $\overline{\mathbf{b}}$) lies behind point $\overline{\mathbf{P}}$ relative to C^2, and thus provides evidence for existence of *free space* behind $\mathbf{P}$ (projecting to $\mathbf{b}$) by demonstrating that $\overline{\mathbf{P}}$ is transparent.

where

$$\mathbf{T}^{\mathbf{E}}_{\mathbf{D}} = \mathbf{T}_{\mathbf{C}}(f, B, c_x, c_y) = \begin{pmatrix} B & 0 & 0 & -c_x B \\ 0 & B & 0 & -c_y B \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{P}_{/E} = \begin{pmatrix} P_X \\ P_Y \\ P_Z \\ 1 \end{pmatrix}, \mathbf{p}_{/D} = \begin{pmatrix} p_x \\ p_y \\ p_d = I_D(p_x, p_y) \\ 1 \end{pmatrix} \qquad (3)$$

The disparity point $\mathbf{p}$ is considered valid if $I_D(p_x, p_y)$ contains a valid value.

In the rest of the paper we will refer to the arrangement illustrated in the Figure 2. There are N ($N = 2$ in the figure) fully calibrated stereo rigs $C^i, i = 1..N$, with associated disparity-to-Euclidean camera coordinate transforms $\mathbf{T}^{\mathbf{E^i}}_{\mathbf{D^i}} = \mathbf{T}_{\mathbf{C}}(f^i, B^i, c^i_x, c^i_y)$, image planes I^i, and disparity images I^i_D. The Euclidean coordinate transformations between each pair of views i, j ($\mathbf{T}^{\mathbf{E^i}}_{\mathbf{E^j}}$) are also assumed to be known.

4. Instantaneous Foreground Segmentation

If the background scene is dynamic or no previous observations are available, then the only information about the scene is a set of range images simultaneously obtained from different viewpoints. In this case we can assume that all space visible by the cameras is empty, and compute the visibility constraints directly from the instantaneous data. We compute the occlusion relationships for each range view separately, using other views as *complementary information*.

Let us consider a range image I_D^i. Let a valid point in D^i, $\mathbf{p}$,

$$\mathbf{p}_{/D^i} \simeq \begin{pmatrix} p_x \\ p_y \\ p_d \\ 1 \end{pmatrix}$$

be the image of the point $\mathbf{P}$, $\mathbf{P}_{/E^i} \simeq T_{D^i}^{E^i}\mathbf{p}_{/D^i}$. Points $\mathbf{C^i}$ and $\mathbf{P}$ define an optical ray in the camera C^i. As the point $\mathbf{P}$ was imaged, we can conclude that all points on the line segment $(\mathbf{C^i}, \mathbf{P})$ are transparent, and that $\mathbf{P}$ is not. I_D^i does not provide us with any information about points $\overline{\mathbf{P}}$ that lie on the optical ray $[\mathbf{C^i}, \mathbf{P})$ beyond $\mathbf{P}$, and are occluded by it. In order to determine whether $\mathbf{p}$ belongs to foreground, i.e. some of $\overline{\mathbf{P}}$s are transparent, we will use the rest of available views.

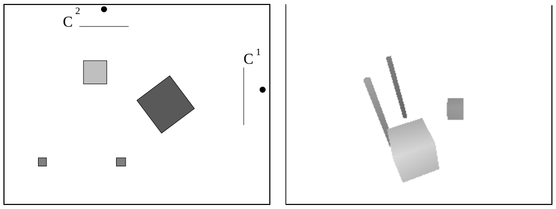
Let I_D^j be a range image taken with a camera C^j, such that $\mathbf{C^j}$, $\mathbf{C^i}$, and $\mathbf{P}$ are not collinear. Let point $\overline{\mathbf{P}}$ project to D^j point $\overline{\mathbf{b}}$:

$$\overline{\mathbf{b}}_{/D^j} = \begin{pmatrix} \overline{b}_x \\ \overline{b}_y \\ \overline{b}_d \\ 1 \end{pmatrix} \simeq \mathbf{T_{E^j}^{D^j}} \mathbf{T_{E^i}^{E^j}} \overline{\mathbf{P}}_{/E^i}.$$

Points $(\overline{b}_x, \overline{b}_y)$ corresponding to all $\overline{\mathbf{P}}$s make up a ray of the line epipolar to point (p_x, p_y) in I^i. If the observed value $I_D^j(\overline{b}_x, \overline{b}_y)$ is valid, and $I_D^j(\overline{b}_x, \overline{b}_y) < \overline{b}_d$, then the point $\mathbf{B}$, observed by C^j,

$$\mathbf{B}_{/E^j} \simeq \mathbf{T_{D^j}^{E^j}} \begin{pmatrix} \overline{b}_x \\ \overline{b}_y \\ I_D^j(\overline{b}_x, \overline{b}_y) \\ 1 \end{pmatrix}$$

lies beyond $\overline{\mathbf{P}}$ on the projective ray $[\mathbf{C^j}, \mathbf{B})$. From this observation we can conclude that $\overline{\mathbf{P}}$ is transparent. Each such observation $I_D^j(\overline{b}_x, \overline{b}_y)$ can be considered to be the *evidence* for point $\mathbf{p}$ belonging to foreground in view I_D^i under our definition. We present fast algorithm for generating points $\overline{\mathbf{b}}_{/D^j}$ corresponding to a particular $\mathbf{p}_{/D^i}$ in Section 6.



(a) (b)

Figure 3. Scene used in the synthetic tests contains 4 objects – two foreground cubes and two "poles" that provide background information. (a) is the plan view of the scene, and (b) is the scene rendered in 3D

In the current implementation of the algorithm, we use the number of found *evidence* pixels as a measure of certainty that point $\mathbf{p}$ belongs to foreground. If more than one complementary camera is available, then the results from all of them may be combined to provide more robust output. We compute a map of the number of observed occluded free-space points:

$$\theta(\overline{\mathbf{b}}) = \begin{cases} 1 & I_D^j(\overline{b}_x, \overline{b}_y) \text{is valid and } \lambda I_D^j(\overline{b}_x, \overline{b}_y) < \overline{b}_d \\ 0 & \text{otherwise} \end{cases}$$
$$OFS(\mathbf{p}) = \sum_{\substack{\overline{\mathbf{b}}, \\ \text{for all } \overline{\mathbf{P}}}} \theta(\overline{\mathbf{b}}) \tag{4}$$

The factor $\lambda > 1$ is introduced to deal with noise inherent in disparity computation. Since we expect the stereo-based range to be less robust for locations that are far from the camera, we can classify $\mathbf{p}$ as foreground if $OFS(\mathbf{p}) > T_{OFS}(p_d)$, where $T_{OFS}(d) \sim 1/d$.

In order to segment the primary view, we search epipolar lines of every valid pixel in all complementary views. So if v is the number of valid pixels in the range image, and n is the total number of pixels in each range image, then the complexity of instantaneous foreground segmentation algorithm is $O(v\sqrt{n}N)$, where N is the number of available views.

Figure 4 demonstrates the results of applying our algorithm to synthetic non-noisy data. The algorithm was presented with two range views (a)

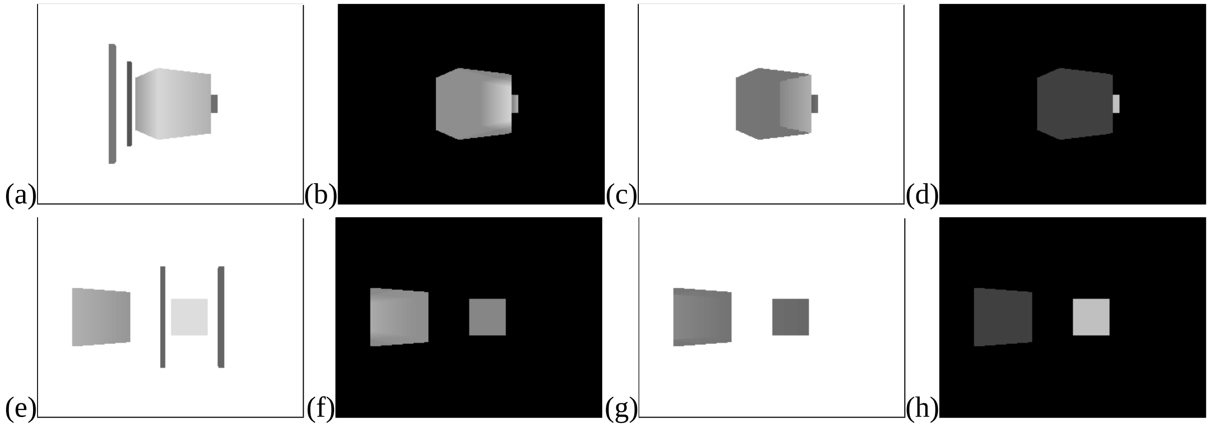


Figure 4. Example of segmentation with visibility constraints on scene in Figure 3. Given disparity views (a) and (e), the algorithm produces OFS maps: (b) and (f). The algorithm also outputs the disparity maps (I_{DB}) of the first visible free-space points behind each of the foreground point: (c) and (g). And the range clustering algorithm (Section 4.1) uses this information to produce the connected components: (d) and (h).

and (e) of the scene in Figure 3, and produced the OFS maps (b) and (f) correspondingly.

4.1. 3-D Connected Components

The method described in the previous section provides us with a measure of how much free space is occluded by each pixel in a given view. It can also be used to approximate the extent of the observed object(s) along each optical ray. We can use this information to cluster the foreground points using both proximity in image plane and the depth extent.

A naive approach would be to cluster the points in a single view based on proximity in either disparity or Euclidean space, and assume that each such cluster corresponds to a separate object. Such assumptions are correct in cases such as one in Figure 5(a), but lead to oversegmentation in the example in Figure 5(b), where parts of the same object have different depths.

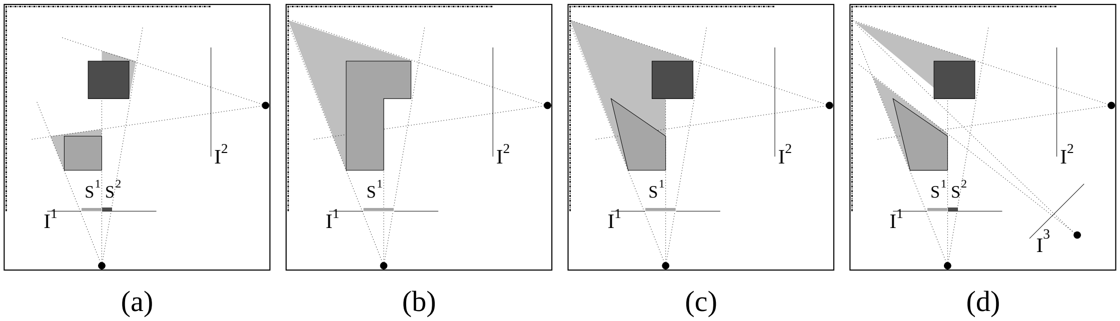
We use the regular connected components analysis supplemented with depth extent information as a clustering technique. For each optical ray (each pixel in the image plane) we can define a *optical line segment* that contains an object, if any. One end of this line segment is the observed 3D point ($\mathbf{P}$) corresponding to the disparity point $(x \;\, y \;\, I_D(x,y))^T$, and the other is $\hat{\mathbf{P}}$, the first free space point behind $\mathbf{P}$ that can be detected from the complementary views. Depending on the camera configuration and the texture availability, optical line segment $[\mathbf{P}, \hat{\mathbf{P}}]$ may be a more or less tight bound on the true extend of the object along the optical ray containing it.

We can thus describe the segment as the image plane coordinates x, y, the "front" disparity $I_D(x,y)$ and the "back" disparity $I_{DB}(x,y) < I_D(x,y)$. The connected components are then computed such that two points $\mathbf{p_1}$ and $\mathbf{p_2}$ belong to the same component if and only if there exists a 4-connected path in the image plane such that all points in the path belong to the foreground (as described in the previous section), *and* their depth ranges ($[I_{DB}(x,y), I_D(x,y)]$) intersect. Thus if free space can be detected between the objects, they will never be undersegmented.

When the algorithm is applied to the synthetic data of Figure 4(a, b), we obtain "back" disparity images (c) and (g), and the foreground pixels in (d) and (h) are correctly segmented into two clusters, corresponding to each of the foreground cubes.

5. Virtual Background Generation

The foreground classification method described in the previous section requires a rather large computational expense at run-time to correctly segment all objects in complicated backgrounds, and will label free-standing static objects (e.g. support columns) as foreground, necessitating extra postprocessing steps to discard them.



(a) (b) (c) (d)

Figure 5. Range-based Connected components. (a) the components S^1, and S^2 belong to separate objects, as their range clusters (lightly shaded regions) do not intersect. (b) A single range connected component found. (c) The algorithm undersegments the scene, since all the optical line segments of visible points make up a single connected component, but the same scene may be correctly segmented (d) if an extra view (I^3) is available.

In cases where the background scene is expected to be static, we can use observations in the absence of the foreground objects to precompute the *expected* visible space for each of the views. We describe the expected visible free space as an upper limit on depth (lower limit on disparity), and classify points with depth less than this limit as foreground. While this foreground definition is analogous to one used with regular range background modeling techniques, it allows us to use both directly observed and computed limits.

An approximation to this limit may be computed as a minimum of the range values at a pixel observed over time and with varying gain (Darrell et al., 2001), resulting in the statistically trained model I_B. This technique is not guaranteed to obtain valid values for all location in the image (e.g. uniformly colored walls would not normally produce valid range for any lighting). If no range data is available at the pixel, we can estimate this limit from visibility constraints obtained from "complementary" cameras, using a technique similar to that shown in previous section. For each point in I^i with invalid range we search the corresponding optical ray to detect all free space points along it that are visible by other cameras C^js, and select the one with the greatest depth (lowest disparity) as the *virtual background*.

We can inverse the order of computation in order to simplify the algorithm. Instead of searching along the optical rays of C^i, we use algorithm described in Section 6 to compute all free space points visible by C^j, $i \neq j$.

For each valid range point $\mathbf{b}$,

$$\mathbf{b}_{/D^j} \simeq \begin{pmatrix} b_x \\ b_y \\ I_D^j(b_x, b_y) \\ 1 \end{pmatrix}$$

all points on the optical ray between $\mathbf{B}$,

$$\mathbf{B}_{/E^j} \simeq \mathbf{T}^{\mathbf{E^j}}_{\mathbf{D^j}} \mathbf{b}_{/D^j}$$

and $\mathbf{C^j}$ are transparent and may be assumed to belong to free space. Thus any point $\overline{\mathbf{B}} \in (\mathbf{B}, \mathbf{C^j})$ is a candidate virtual background for the corresponding point in I^i, (p_x, p_y), such that

$$\mathbf{p}_{/D^i} = \begin{pmatrix} p_x \\ p_y \\ p_d \\ 1 \end{pmatrix} \simeq \mathbf{T}^{\mathbf{D^i}}_{\mathbf{E^i}} \mathbf{T}^{\mathbf{E^i}}_{\mathbf{E^j}} \overline{\mathbf{B}}_{/E^j} \qquad (5)$$

After a set of candidates for a single (discrete) image location ($\{\tilde{\mathbf{p}}_k = (\tilde{p}_x \; \tilde{p}_y \; \tilde{p}_{d_k})^T\}$) is computed, we select the virtual background value at the location $(\tilde{p}_x, \tilde{p}_y)$ as

$$I_V^i(\tilde{p}_x, \tilde{p}_y) = \min_k \tilde{p}_{d_k} \qquad (6)$$

We combine the directly observed background I_B and computed virtual background I_V into our background model I_{VB} by using directly observed values where available, and virtual values in the rest of locations,

$$I_{VB}(x, y) = \begin{cases} I_B(x, y) & I_B(x, y) \text{ is valid} \\ I_V(x, y) & \text{otherwise} \end{cases} \qquad (7)$$

6. Fast Computation of Visibility Constraints

In this section we detail our method for discretizing disparity-space projections of optical rays. Since this is the most time consuming part of the algorithms described in the previous sections, it is imperative that the point generation be extremely fast. We use Bresenham's algorithm (Bresenham, 1965) to discretize the C^i's optical ray of $\mathbf{P}$ relative to D^j.

For each valid disparity point $\mathbf{p}$ in D^i,

$$\mathbf{p}_{/D^i} = \begin{pmatrix} p_x \\ p_y \\ I^i_D(p_x, p_y) \\ 1 \end{pmatrix},$$

the corresponding optical ray in the camera C^i is $[\mathbf{C^i}, \mathbf{P})$, where

$$\mathbf{P}_{/E^i} \simeq \mathbf{T}^{\mathbf{E^i}}_{\mathbf{D^i}} \mathbf{p}_{/D^i}.$$

If we define the transformation from E^i to D^j,

$$\mathbf{\Gamma} = \mathbf{T}^{\mathbf{D^j}}_{\mathbf{E^j}} \mathbf{T}^{\mathbf{E^j}}_{\mathbf{E^i}} = \left(\begin{array}{c|c} & \gamma_x \\ \mathbf{\Gamma}_{4\times 3} & \gamma_y \\ & \gamma_d \\ & \gamma_w \end{array} \right). \tag{8}$$

The image of the ray $[\mathbf{C^i}, \mathbf{P})$ in D^j is $[\mathbf{\Gamma}(\mathbf{C^i}_{/E^i}), \mathbf{\Gamma}(\mathbf{P}_{/E^i}))$

Since $\mathbf{C^i}$ is the origin of E^i, i.e

$$\mathbf{C^i}_{/E^i} \simeq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

the ray in D^j is $[\boldsymbol{\gamma}, \mathbf{b})$, where

$$\mathbf{b}_{/D^j} = \begin{pmatrix} b_x \\ b_y \\ b_d \\ 1 \end{pmatrix} \simeq \mathbf{\Gamma} \mathbf{T}^{\mathbf{E^i}}_{\mathbf{D^i}} \mathbf{p}_{/D^i}.$$

After the two points on the ray are computed, we can represent any point on this ray,

$$\overline{\mathbf{b}} = \begin{pmatrix} \overline{b}_x \\ \overline{b}_y \\ \overline{b}_d \end{pmatrix}$$

in parametric form as $\overline{\mathbf{b}} = (b_x \ b_y \ b_d)^T + t\mathbf{u}$. Where

$$\mathbf{u}' = \begin{pmatrix} b_x\gamma_w - \gamma_x \\ b_y\gamma_w - \gamma_y \\ b_d\gamma_w - \gamma_d \end{pmatrix}$$
$$\mathbf{u} = \pm \frac{\mathbf{u}'}{\max\{|u'_x|, |u'_y|\}} \tag{9}$$

with sign selected so that $\mathbf{u}$ points "away" from $\boldsymbol{\gamma}$ when applied at $\mathbf{b}$. In this case the positive values of t correspond to the direction away from the center of projection, and should be used for foreground segmentation (Section 4), while the negative ts should be used in generating virtual backgrounds (Section 5).

We can then use Liang-Barsky line clipping algorithm (Foley et al., 1993) to select the region of the ray that should be explored. It is determined by the extend of available portion of I^j_D (the range image), and the requirements that the point $\mathbf{P}$ should be in front of both image planes I^i and I^j. As the result of clipping we obtain values t_{MIN} and t_{MAX}, such that if $t_{MIN} \le t \le t_{MAX}$, then $\overline{\mathbf{b}} = (b_x \ b_y \ b_d)^T + t\mathbf{u}$ is valid.

After $\mathbf{b}$, $\mathbf{u}$, t_{MIN} and t_{MAX} are computed, the straight forward application of Bresenham's algorithm generates all valid points $\overline{\mathbf{b}} = \mathbf{b} + t\mathbf{u}, t \in \Im, t_{MIN} \le t \le t_{MAX}$. These point can then processed as described in Sections 4 and 5.

Note that while ray processing is the most time consuming operation of the presented algorithms, each ray is independent of all other rays, and the algorithm can take advantage of parallel processing capabilities if they are available.

7. Experimental Results

We have tested our algorithms on both synthetic and live data. The live images were obtained using SRI Small Vision System hardware/software combination (Konolige, 1997). The cameras with $7.5mm$ lenses were placed to cover approximately $3m \times 3m$ area, with optical axes approximately perpendicular.

The results of applying our algorithms to sample images are presented in Figures 7, 8, 9. The images of the scene without foreground objects (including computed virtual background image) for the live image tests are shown in Figure 6

For each example with intensity image (a), stereo disparity image (b) and complementary disparity image (e), we compute the OFS map (c) using algorithm described in Section 4, with $\lambda = 1.1$. The algorithm achieved processing speed of 15fps on 1.5GHz Pentium 4, on half-resolution input images from our two camera installation (not including time used for stereo processing). The detected foreground pixels are then clustered using technique from Section 4.1, producing the "back" disparity map (f) and connected components (g). We finally apply the method presented in Section 5, using the background model in Figure 6(d), obtained from observations 6(b) and 6(c), to produce the virtual background segmentation (d). The segmentation obtained using classing foreground detection with the observed background model (6(b)), is provided for comparison in (h).

As can be seen in e.g. 7(c), the Instantaneous Foreground algorithm detected free space behind the lamp-post and labeled it as foreground, even though it was present in the empty scene (Figure 6(b)). The image was segmented correctly by the Virtual Background Segmentation (7(d)).

8. Conclusions

We have presented two novel range-based segmentation algorithms, that take advantage of availability of multiple, widely spaced stereo views. The semi real-time foreground segmentation algorithm relies on the visibility information obtained from other views to locate points that occlude *free space*. Since the algorithm does not maintain an explicit background model and uses only immediately available reliable range information, it is able to handle variable lighting conditions, but can incorrectly label parts of the background scene as foreground. We further extended the algorithm to use visibility constraints to improve clustering of the object points. The virtual backgrounds algorithm uses the visibility information to create dense range background images which can then be used with common real-time conservative background subtraction methods.

Acknowledgements

We are very grateful to Gregory Shakhnarovich and David Demirdjian for useful discussions. This work was funded in part by MIT Project Oxygen, DARPA HumanID program, and NTT.

References

Beymer, D. and K. Konolige: 1999, 'Real-time tracking of multiple people using continous detection'. In: *Proc. International Conference on Computer Vision (ICCV'99)*.

Bresenham, J. E.: 1965, 'Algorithm for Computer Control of Digital Plotter'. *IBM Syst. J.* **4**(1), 25–30.

Brumitt, B., B. Meyers, J. Krumm, A. Kern, and S. Shafer: 2000, 'EasyLiving: Technologies for intelligent environments'. In: *Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000*. pp. 12–29.

Darrell, T., D. Demirdjian, N. Checka, and P. Felzenszwalb: 2001, 'Plan-view trajectory estimation with dense stereo background models'. In: *Proc. International Conference on Computer Vision (ICCV'01)*.

Demirdjian, D. and T. Darrell: 2001, 'Motion estimation from disparity images'. In: *Proc. International Conference on Computer Vision (ICCV'01)*.

Foley, J. D., A. van Dam, S. K. Feiner, J. F. Hughes, and Phillips: 1993, *Introduction to Computer Graphics*. Reading, MA: Addison-Wesley.

Grimson, W., C. Stauffer, R. Romano, and L. Lee: 1998, 'Using adaptive tracking to classify and monitor activities in a site'. In: *Proceedings of CVPR'98*.

Harville, M., G. Gordon, and J. Woodfill: 2001, 'Foreground Segmentation Using Adaptive Mixture Models in Color and Depth'. In: *Workshop on Detection and Recognition of Events in Video.*

Konolige, K.: 1997, 'Small Vision System: Hardware and Implementation'. In: *Eighth International Symposium on Robotics Research, Japan*.

Krumm, J., S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer: 2000, 'Multi-Camera Multi-Person Tracking for EasyLiving'. In: *IEEE Workshop on Visual Surveillance*.

Kutulakos, K. N. and S. M. Seitz: 2000, 'A Theory of Shape by Space Carving'. *Int. Journal of Computer Vision* **38**(3), 199–218.

Matusik, W., C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan: 2000, 'Image-Based Visual Hulls'. In: K. Akeley (ed.): *Siggraph 2000, Computer Graphics Proceedings*. pp. 369–374, ACM Press / ACM SIGGRAPH / Addison Wesley Longman.

Ridder, C., O. Munkelt, and H. Kirchner: 1995, 'Adaptive background estimation and foreground detection using Kalman filtering'. In: *Proc. ICAM*. pp. 193–199, 1995.

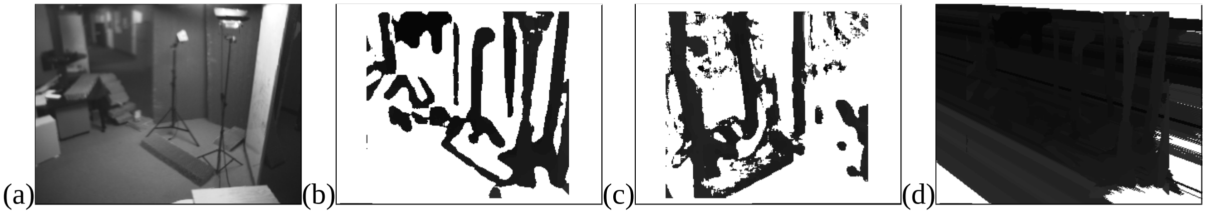


Figure 6. The images of the empty scene (invalid disparities are shown in white): (a) intensity, (b) disparity, (c) disparity view from the complementary camera, (d) generated virtual background image.

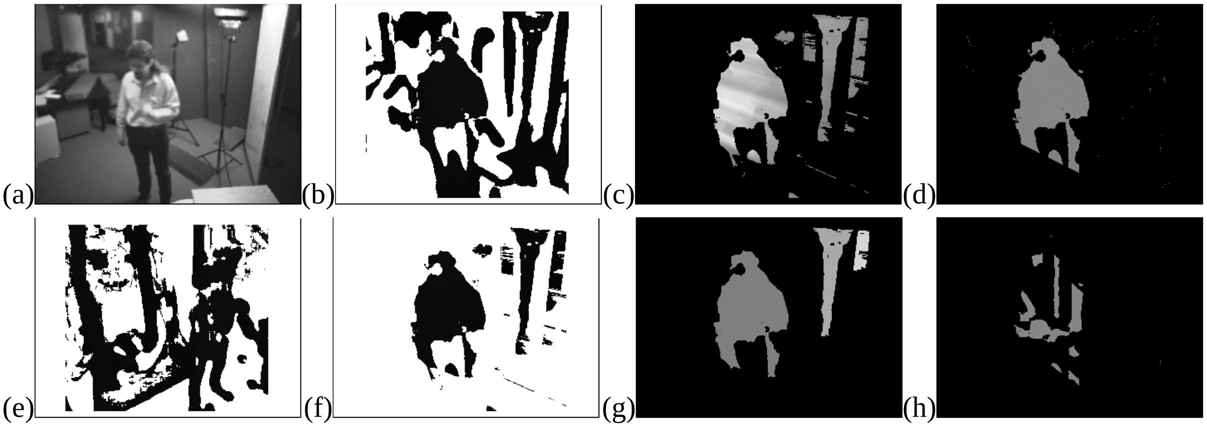


Figure 7. Foreground segmentation with a single foreground object: (a) intensity image, (b) disparity image, (c) OFS map, (d) virtual background segmentation (e) complementary disparity image, (f) estimated "back" disparity for detected foreground points, (g) Color-coded depth-extent connected components, (h) real background segmentation.

Slabaugh, G., B. Culbertson, and T. Malzhender: 2001, 'A Survey of Methods for Volumetric Schene Reconstruction from Photographs'. In: *VG'01*.

Stauffer, C. and W. Grimson: 1999, 'Adaptive background mixture models for real-time tracking'. In: *Proceedings of CVPR'99*.

Toyama, K., J. Krumm, B. Brumitt, and B. Meyes: 1999, 'Wallflower: Principles and practice of background maintenance'. In: *In Proc. International Conference on Computer Vision (ICCV'99)*.

Wren, C., A. Azarbayejani, T. Darrell, and A. Pentland: 1995, 'Pfinder: Real-time tracking of the human body'. In: *Photonics East, SPIE, volume 2615*.

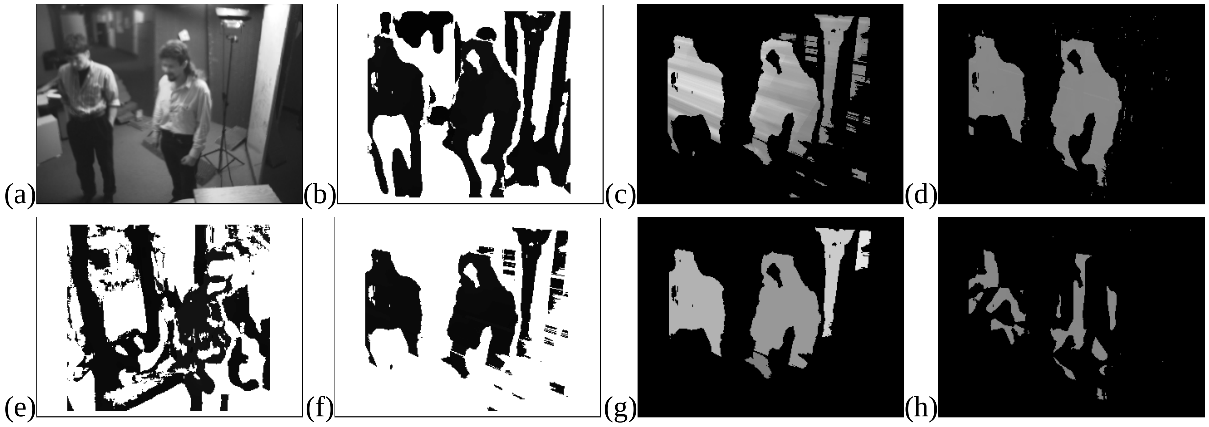


Figure 8. Foreground segmentation with two foreground objects: (a) intensity image, (b) disparity image, (c) OFS map, (d) virtual background segmentation (e) complementary disparity image, (f) estimated "back" disparity for detected foreground points, (g) Color-coded depth-extent connected components, (h) real background segmentation.

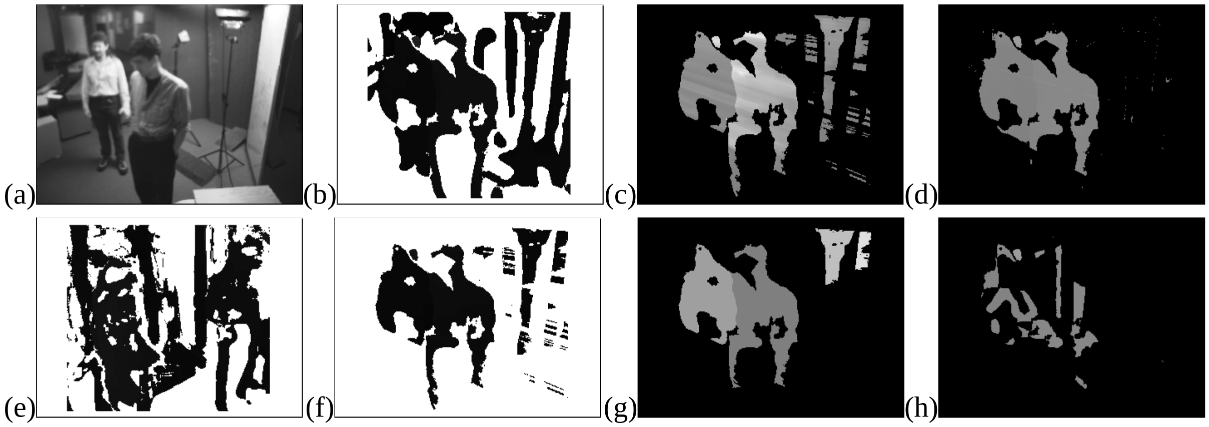


Figure 9. Foreground segmentation with two foreground objects with overlapping projections: (a) intensity image, (b) disparity image, (c) OFS map, (d) virtual background segmentation (e) complementary disparity image, (f) estimated "back" disparity for detected foreground points, (g) Color-coded depth-extent connected components, (h) real background segmentation.