Plan-view trajectory estimation with dense stereo background models

T. Darrell, D. Demirdjian, N. Checka, P. Felzenszwalb
MIT Artificial Intelligence Lab
Cambridge MA
trevor,demirdji,checka,pff@ai.mit.edu

Abstract

In a known environment, objects may be tracked in multiple views using a set of background models. Stereo-based models can be illumination-invariant, but often have undefined values which inevitably lead to foreground classification errors. We derive dense stereo models for object tracking using long-term, extended dynamic-range imagery, and by detecting and interpolating uniform but unoccluded planar regions. Foreground points are detected quickly in new images using pruned disparity search. We adopt a "late-segmentation" strategy, using an integrated plan-view density representation. Foreground points are segmented into object regions only when a trajectory is finally estimated, using a dynamic programming-based method. Object entry and exit are optimally determined and are not restricted to special spatial zones.

1 Introduction

Tracking people in known environments has recently become an active area of research in computer vision. Several person tracking systems have been developed to detect the number of people present as well as their 3-D position over time. These systems generally use a combination of foreground/background classification, clustering of novel points, and trajectory estimation in one or more camera views [18, 16, 10, 13, 7, 17, 5]

Many color-based approaches to background modeling have considerable difficulty with fast illumination variation due to changing lighting and/or video projection. To overcome this, several authors have advocated the use of background shape models based on stereo range data [7, 5, 11]. Unfortunately, the background models built by these systems are often sparse, due to the many regions of uniform brightness where stereo estimation fails in a typical background training sequence. Additionally, most of these systems are based on exhaustive stereo disparity search.

In contrast, we show here how dense, fast range-based tracking can be performed with modest computational complexity. We recover dense depth data using multiple-gain imaging and long-term observation approaches. We match uniform but unoccluded planar regions in the scene and interpolate their interior range values. We apply ordered disparity search techniques to prune most of the disparity search computation during foreground detection and disparity estimation, yielding a fast, illumination-insensitive 3-D tracking system.

When objects are moving on a ground plane and are observed from multiple widely-separated viewpoints, rendering an orthographic vertical projection of foreground activity is useful [13, 3, 2, 17]. A "plan-view" image facilitates correspondence in time since only 2D search is required. Typically, previous systems would segment foreground data into regions *prior to* projecting into a plan-view, followed by region-level tracking and integration, potentially leading to sub-optimal segmentation and/or object fragmentation. (But see [12] for a way to smooth fragmented trajectories.)

Instead, we develop an approach which altogether avoids any early segmentation of the foreground data. We merge the plan-view images from each viewpoint and estimate over time a set of trajectories that best accounts for the integrated foreground density. Trajectory estimation is performed using a dynamic programming-based algorithm, which can optimally estimate the position over time as well as the entry and exit locations of an object. This contrasts previous approaches, which generally used instantaneous measures, and/or specific object creation zones to decide on the number of objects per frame [3, 13].

In the next section, we detail our new algorithm for computing dense range-based foreground estimates and for fast estimation of foreground disparities. Following that, we introduce a plan-view tracking representation and our algorithm for optimally estimating trajectories with limited temporal extent. We show how this method can accurately detect the entry and exit of objects without constraints on the spatial location of such events. We finish with a discussion of the overall system's performance and implications, as well as possible avenues for future work.

0-7695-1143-0/01 $10.00 (C) 2001 IEEE

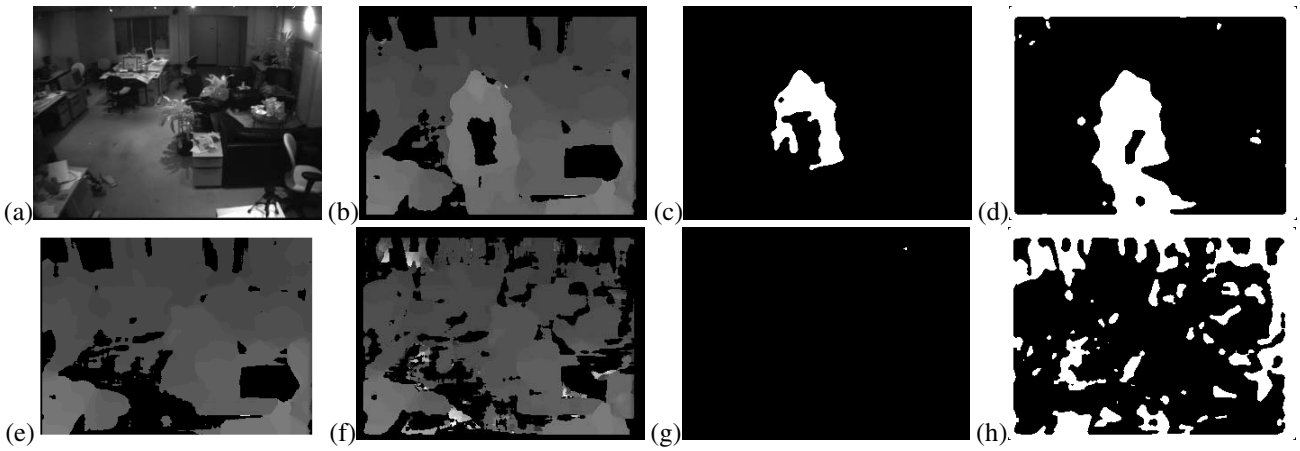

Figure 1: The problem with sparse range backgrounds. Given a sparse background model (e) of a scene (a), a new range image with a foreground person (b), and a new range image with no foreground object but a changed illumination condition (f), we see that a conservative segmentation (c,g) misses many 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.

2 Range-based foreground detection

Segmentation of foreground regions using range measurements is inherently robust to the illumination variation that disrupts most color-based approaches. However, when range data is used directly to build background models, experience shows that the models are often sparse and are well-defined at fewer points than in a color model [7, 5]. These background models will have pixels which are set to an "unknown" depth value.

With a sparse depth map, one has to decide whether to detect foreground pixels when the background is invalid and a new range value is observed. The conservative option–not declaring the pixel foreground–will forever prevent detection of any valid foreground points in the empty regions of the background. This will lead to Type I errors (Figure 1(c).) The alternative approach, declaring a pixel to be foreground even when the background is undefined, leads to Type II errors. If imaging condition changes such that a previously uniform background region suddenly has contrast and a defined range value, then a background point will incorrectly be declared a foreground point. (Figure 1(h)). This can commonly happen when the illumination level changes and pixels de-saturate, or when shadows or other projections are cast on a uniform surface.

To overcome this problem we construct dense background models with long-term and extended dynamic-range data. We can resolve depth values within unoccluded, uniform, and planar regions using a constraint on the appearance of these regions in two views. We also use predictive disparity search to prune unnecessary computation and quickly estimate foreground regions of new images.

2.1 Extended dynamic range stereo

Variable gain (or equivalently, variable aperture or extended dynamic range) imaging has been developed for intensity model acquisition with wide dynamic range [6, 15], but has not to our knowledge been applied to stereo range estimation. This particularly benefits disparity estimation, since different regions of a scene likely will have maximal contrast at different camera gain settings. With any single gain setting, a large portion of the image may be (de-)saturated, yielding unknown depth values.

Correspondence matching with extended range intensity images is straightforward–one can simply apply traditional matching algorithms to floating point pixel intensity data. However, acquiring extended range intensity images from a conventional camera requires a relatively precise photometric calibration of each camera. If photometric calibration is not available, we can approximate the result by separately computing disparity at each gain setting, and integrating the results over multiple observations.

Our basic stereo engine is based on classic window-based robust correspondence. If $I^L_{g,t}(\vec{x})$ and $I^R_{g,t}(\vec{x})$ are the rectified left and right intensity images acquired at time t and gain level g, indexed by pixel location $\vec{x} = [x\ y]^T$, we denote the quality of a disparity match using a match function

$$D(I^L_{g,t}, I^R_{g,t}, \vec{x}, d) =$$

$$-\sum_{\vec{n}\in\mathcal{N}} ||T_{\vec{x},\vec{n}}(I^L_{g,t}(\vec{x}+\vec{n}) - T_{\vec{x},\vec{n}}(I^R_{g,t}(\vec{x}+\vec{n}+[d\ 0]^T))||_r$$

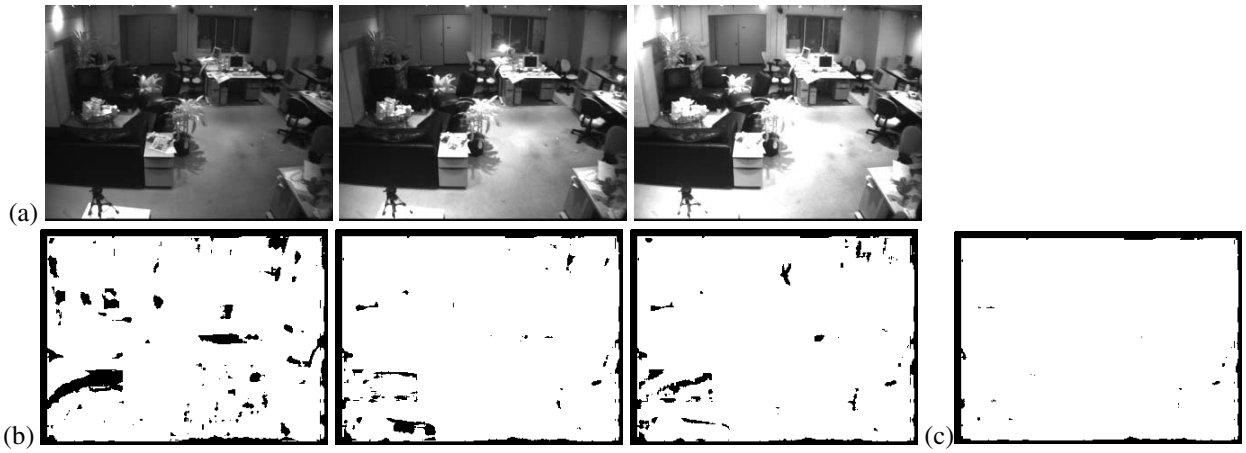

Figure 2: Stages in building a dense background model. (a) Examples of variable gain and illumination conditions for a scene. (b) Map of valid disparity values for each condition. (c) Map of valid disparity values for integrated disparity map.

for some suitable neighborhood $\mathcal{N}$ and robust transform T (e.g., [19]) and/or robust norm $||\cdot||_r$ (e.g., [4]).

At each pixel, we can apply the standard technique of evaluating D across all disparities, testing whether these values can be approximated well by a single mode. If so we set the match value $\hat{d}(\vec{x}, g, t)$ to be the disparity with the highest match, and $\hat{D}(\vec{x}, g, t)$ to be the match value. When the distribution is not approximated by a single mode of sufficient height, we declare the pixel invalid $\hat{d}(\vec{x}, g, t) = null$. We compute a global estimate of σ_D, the standard deviation of $\hat{D}$, based on $\hat{D}$ values when the same $\hat{d}$ is chosen in two consecutive frames.

2.2 Long term observation

To acquire background models online without an training sequence of foreground-free observations, we use a variant of the algorithms presented in [7] and [9]. During background training, we incrementally compute a histogram of disparity values across a range of time and gain conditions. Given a new range observation $\hat{d}(\vec{x}, g, t)$, we increment a histogram $H_B(\vec{x}, \hat{d}(\vec{x}, g, t))$. After a period of background training, we determine the background disparity values by inspecting H_B to find modes which are large and far. We note that choosing the farthest modes helps to reduce problems of disparity ambiguity in periodic textures.

We choose the background values to be those which are less than the median valid disparity over time, $d_B(\vec{x})$.[1] We then set a background weight map W_B to have the same values as H_B, except that any values greater than d_B are set to zero. We require that the ratio of valid to invalid range observations at a pixel be sufficiently large to keep a background value. If this ratio is not greater than a threshold (typically $\beta = 0.1$), then we set the background to be undefined, $d_B(\vec{x}) = null$, and W_B, H_B to be uniformly zero.

Gathering background observations over long-term sequences has the advantage that lighting variation can be included in the background training set. Extreme lighting variation is useful, since it can cause previously uniform regions to have apparent contrast. Either the overall (ambient) or relative (shadow or projected texture) illumination level can cause contrast to appear where previously the image region was uniform. As with variable gain, each illumination condition will likely yield a different pattern of valid/invalid range.

We generally choose to observe the natural variation of illumination, e.g., from natural outdoor light entering through windows, and from users' regular actuation of indoor lighting appliances. However, it is also possible to specifically train the background using active illumination if desired. During a background training period we also set the camera parameters to cycle through the range of possible gain settings. Figure 2(a) shows a set of images taken under different gain and illumination conditions, and Figure 2(b) shows the map of valid range pixels computed from stereo pairs in each of these conditions. Figure 2(c) shows the map of valid disparity values for an integrated background recovered from the set of variable gain and illumination training data.

2.3 Detection of unoccluded uniform planar regions

In indoor environments there are many planar uniform regions on which disparity is difficult to obtain, even with long-term, variable-gain observations. In general, determining whether to interpolate range values within such surfaces without inferring global structure in the scene is difficult.

[1] For scenes where the background is covered more than half the time, we could use a rank filter with lower threshold.

However, there is one special case that is locally computable and proves to be particularly useful for our purposes. When a planar uniform region is unoccluded in two views, the extent of the homogeneous patch in each view will be equivalent, according to the homography determined by the stereo viewing geometry and the orientation of the patch in the world. This condition often happens in practice–for example in saturated pixels in dark or light clothing worn by a person in the scene (e.g., Figure 3).

If the correspondences at the border of the region are well approximated by a homography and are consistent with the epipolar stereo constraint, we can test to see whether the shape of the homogeneous patch in each view is related by the appropriate warping function. We can compute a boolean image indicating the extent of the uniform region in each image, warp one according to the given homography, and compare the overlap of the resulting images. If they agree, then we mark this to be a uniform solid plane hypothesis.

There are two degenerate cases: a region from a window looking out to a completely featureless background, and a solid but non-planar region with no apparent texture. By maintaining the hypothesis throughout an extended time period, we can alleviate the impact of the former and only keep the hypothesis if no object is ever seen further in depth at any point within the region. The latter degeneracy seems unavoidable, but has not been a problem in practice.

Our method is related to the classic idea of region stereo, however we do not attempt to perform a dense color segmentation of the image as a pre-processing step, and only opportunistically apply the technique when there are large uniform connected components in the scene. We also make explicit use of planar homographies and the epipolar constraint to constrain possible matches.

2.4 Fast foreground disparity estimation

Fast foreground detection is necessary to detect rapidly moving objects and dynamic activity patterns. Traditional approaches to tracking with stereo-based backgrounds usually have relied on general-purpose stereo computation, which exhaustively searches for the best disparity matches at each frame. However most of this computation is unnecessary for scenes with predictable backgrounds, as pointed out by [11]. They demonstrated how disparity testing could find foreground silhouettes, given a previously computed static background model. We have extended their method to the case of dynamic backgrounds with multiple range modes, and to predict the entire range image, including disparities in foreground regions.

We use our background disparity weights, W_B together with similar weights corresponding to short-term foreground predictions, W_F, to guide the disparity search in a new frame. For each new range match we increment W_F as we increment W_B above, but only after we reduce the previous values of W_F by a constant factor (typically $\gamma = 0.9$):

$$W_F(\vec{x}, d) = \gamma W_F(\vec{x}, d) + \delta(d - \hat{d}(\vec{x}, g, t))$$

where $\delta(x)$ is the impulse function at $x = 0$.

We maintain a separate W_B and W_F for each pixel in the image. Given a new frame at gain g and time t we find those (d_i^*, w_i^*) with sufficiently large value:

$$||D(I_{g,t}^L, I_{g,t}^R, \vec{x}, d_i^*) - \hat{D}(\vec{x}, g, t)||^2 \ <?\ \rho\sigma_D^2$$

and whose background or foreground weight is also above a threshold, typically $\alpha = 0.25$. (Usually $\rho = 2$.) If no such candidate is confirmed, we compute all disparity values and estimate $\hat{d}$ using the conventional approach described above. If the selected disparity is less than $d_B(\vec{x})$, we label it foreground. Points for which $d_B = null$ or $\hat{d} = null$ are by definition not foreground. During a foreground detection phase we have the option of using the automatic gain control setting, or searching through the range of gain levels. If run-time speed is the primary concern, we choose the former approach.

This search pruning optimization can dramatically reduce run-time costs when the foreground regions of a scene are relatively small or are moving slowly. The final result of range-based foreground detection is a map of foreground pixels $\vec{p}_j = (c_j, x_j, y_j, t_j, d_j)$, each from a particular location x, y in camera c at time t with disparity d.

3 Plan-view trajectory estimation

We combine information from multiple stereo views to estimate the trajectory of objects over time. The true extent of an individual object in a given image is generally difficult to identify. An optimal trajectory segmentation ought to consider the assignment of an individual pixel to all possible trajectories estimated over time. Systems which perform an early segmentation and grouping of foreground data before trajectory estimation preclude this possibility.

We adopt a late-segmentation strategy, finding the best trajectory in an integrated spatio-temporal representation that combines foreground pixels from each view. By assuming that objects move on a ground plane and do not overlap in the vertical dimension in our environment, a "plan-view assumption" allows us to completely model instantaneous foreground information as a 2-D orthographic density projection. Over time, we compute a 3-D spatio-temporal plan-view volume.

We project (x_j, y_j, d_j) from each foreground point $\vec{p}_j$ into world coordinates (U_j, V_j, W_j) using the calibration given by camera c_j. (See Figure 4.) U, V are chosen to be orthogonal axes on the ground plane, and W normal to the

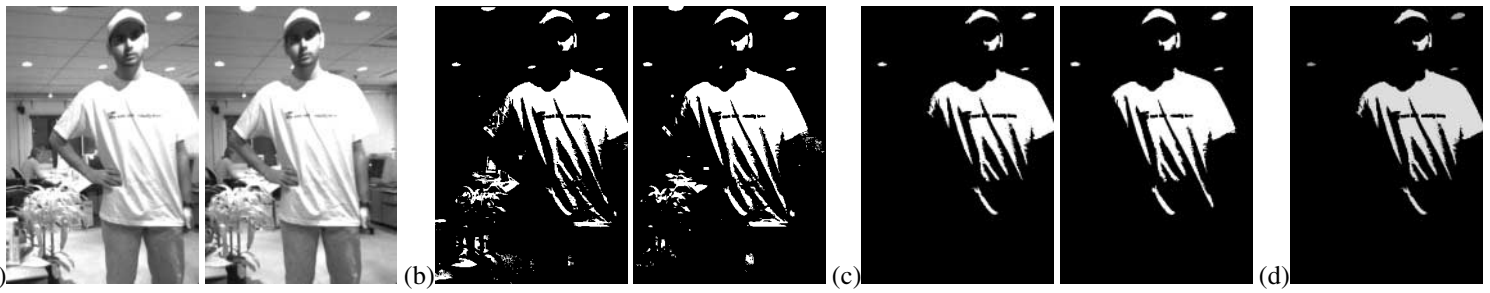

Figure 3: Disparity estimation with uniform planar unoccluded regions. We match a candidate homography between connected components of uniform regions in two views. Unoccluded planes will yield connected component shapes that exactly match according to the given homography. Occluded planes, such as the computer monitor under the user's arm in the far background, or non-planar objects, such as the plant in the foreground, will not have equivalent shapes and will not be matched. In this example uniform regions were determined by saturated pixels, and the plane was restricted to be fronto-parallel so that image plane translation would determine equivalent matches. (a) shows a stereo pair, (b) saturated pixels in each view, (c) connected components that exactly match in both views for some translation, (d) computed disparity values for each region, shown in grayscale.

ground plane. We then compute the spatio-temporal plan view volume, with $P(u,v,t) = \sum_{\{\vec{p}_j | U_j=u, V_j=v, t_j=t\}} 1$.

3.1 Distance transform-based dynamic programming

We characterize the quality of a trajectory by its smoothness over time, and the support of the object track in each time frame. Given $P(u,v,t)$ and a set of possible poses and positions of the object at each time step, we characterize a single optimal trajectory over all time as,

$$L^* = \arg\max_{L} \sum_{0<t\leq T} M(\vec{l}_t) - \sum_{0<t<T} d(\vec{l}_t, \vec{l}_{t+1}) \qquad (1)$$

where $\vec{l}_t$ is one of the possible discrete 2-D location, size and pose parameters of the object in frame t, $L = \{\vec{l}_t \mid 0 < t < T\}$ is the object trajectory, $M(\vec{l}_t)$ is the support of the object track at location $\vec{l}_t$, and $d(\vec{l}_t, \vec{l}_{t+1})$ is cost of matching $\vec{l}_t$ with $\vec{l}_{t+1}$. We compute support to be the integral of the plan-view density within the shape given by $\mathcal{S}(\vec{l}_t)$:

$$M(\vec{l}_t) = \sum_{(x,y)\in\mathcal{S}(\vec{l}_t)} P(x,y,t)$$

Classically, it is possible to solve equation (1) with complexity $O(m^2T)$ using dynamic programming techniques [1]. Unfortunately, m is the number of discrete configurations, so m^2 can grow prohibitively large. Using the distance transform formulation introduced in [8], we can reduce this complexity to $O(mT)$, by restricting the form of d to be the norm or norm squared of transformed location values. We simply set $d(\vec{l}_t, \vec{l}_{t+1}) = (\vec{l}_t - \vec{l}_{t+1})^T \mathbf{V}^{-1} (\vec{l}_t - \vec{l}_{t+1})$, where $\mathbf{V}$ is a diagonal matrix of variances for each pose parameter. This simply says that each pose parameter should change slowly over time.

To solve equation (1), we first compute the best value of the final $\vec{l}_T$, as a function of the location at the previous time using dynamic programming:

$$B_T(\vec{l}_{T-1}) = \max_{\vec{l}_T}(M(\vec{l}_T) - d(\vec{l}_{T-1}, \vec{l}_T))$$

Recursively, we then compute the best value of $\vec{l}_t$ as a function of $\vec{l}_{t-1}$

$$B_t(\vec{l}_{t-1}) = \max_{\vec{l}_t}(M(\vec{l}_T) - d(\vec{l}_{t-1}, \vec{l}_t) + B_{t+1}(\vec{l}_t))$$

and finally $B_0 = \max_{\vec{l}_0}(M(\vec{l}_0) + B_1(\vec{l}_0))$. The optimal trajectory is then given by replacing $\max$ with $\arg\max$ in the above equations and reversing the recursion to compute the optimal location at each time, $\vec{l}_t^*$. This method finds a single optimal trajectory from the initial time 0 to the final time T.

3.2 Trajectory start/end determination

A difficult challenge in person tracking systems has been the estimation of the number of people a given environment under general entry and exit conditions. Previous systems [13] have relied on specific spatial zones to delimit the start and end of person trajectories, while other systems generally use instantaneous criteria to initiate or terminate a new track. We take advantage of our spatio-temporal plan-view representation to optimally estimate trajectory extent as well as shape in a single dynamic programming optimization.

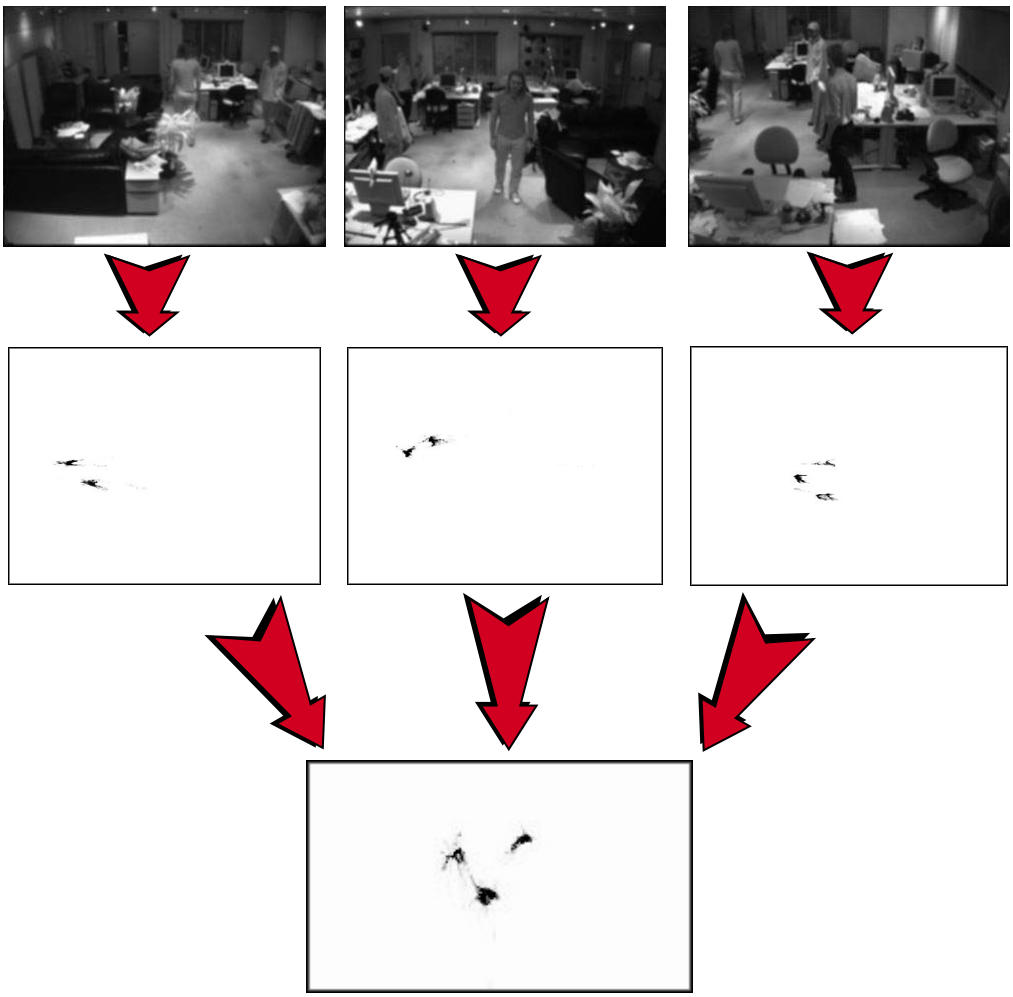

Figure 4: Foreground points are projected from each view individually to a plan view representation, then are integrated into a single spatio-temporal sequence before segmentation into individual trajectories is performed.

We extend the above method to find trajectories with explicit start and end points. Let

$$(L^*, s^*, e^*) = \arg\max_{L,s,e} \sum_{s \le t \le e} (M(\vec{l}_t) - \psi) + \sum_{s \le t < e} d(\vec{l}_t, \vec{l}_{t+1}) \tag{2}$$

where ψ is the cost of extending (s^*, e^*) per time unit.

Fortunately, we can solve equation 2 by modifying the recursive algorithm above. First we replace $M(l)$ with $(M(l) - \psi)$ in the equations above. After computing each B_t, we inspect each location to check if $B_t(\vec{l}_{t-1})$ is negative. If so, we mark that location, indicating that if the object goes through $\vec{l}_{t-1}$, that should be the end of the trajectory. The start of the trajectory is given by the location maximizing $B_s(\vec{l}_s)$, over all locations and time frames. The trajectory can be found by tracing back from $\vec{l}^*_{s^*}$ until reaching a location marked as a trajectory end. The estimation of L^*, s^*, e^* for a single trajectory is optimal in that the dynamic programming method computes a global maxima of equation 2.

3.3 Implementation and Examples

Figure 5 shows the configuration of cameras in our test environment. To align multiple views, we expect to use an automatic calibration process where objects moving on the plane are used to determine the orientation of each camera view [14, 13]. However, this section's results were obtained with an approximate manual calibration based on hand selected correspondences in each camera view.

We collect variable gain and illumination images in our environment during a background acquisition phase. When there are multiple objects in the scene, we solve for a set of trajectories by first finding the trajectory with highest quality given by Equation 2, removing the points that contributed to its support in the plan-view sequence, and re-

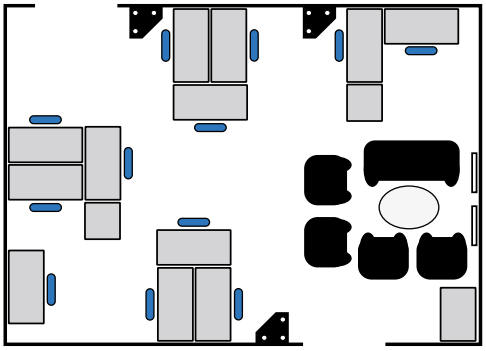

Figure 5: Plan view diagram of office environment used for example in previous figure. Workstations are shown as rectangles, and a sitting area is shown as chairs and couch around an oval table. Three trinocular stereo cameras, shown as black triangles with three circles, are drawn at the approximate location where they are mounted in the ceiling.

peating until no further trajectory can be found with positive quality.

For the examples shown in this section, we used a simplified shape model of rectangular regions with fixed size and pose, given by average human torso dimensions. We expect to extend our implementation shortly to include varying size and pose, which will allowing extended arm position tracking, etc. We discretized ground plane position to a $320x240$ grid, and set ψ to be the median value of $\hat{d}(l)$ evaluated at random locations. (Again, assuming that our scene is on average more than half background.) We truncate the time history at 50 frames. Figure 6 shows the result on a sequence of 3 people moving within our test environment.

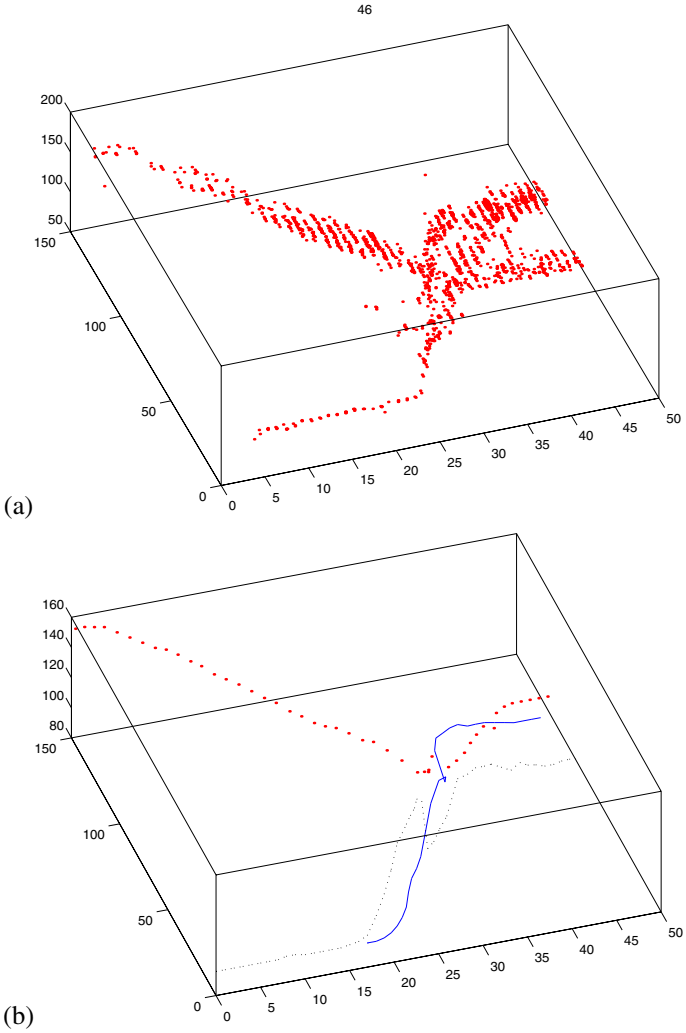

Figure 6: (a) (U,V,T) plot of plan-view foreground density over time for a sequence with three moving people. The third person enters around frame 20. (b) Segmentation using dynamic programming method described in text.

4 Discussion and Future Work

While the results are appealing, a problem remains: when the trajectory of two objects or people overlap, it is not possible from a foreground density representation to disambiguate trajectories if they subsequently separate. Appearance information can resolve this, as shown by [10]. Unfortunately, including this in the dynamic programming optimization would greatly increase the size of the state space of locations at each time frame, making the solution for the optimal trajectory impractical. Resolving this is a topic of ongoing and future work. We plan to use a trajectory-level correspondence process that uses a graph based on the overall trajectory data, computes aggregate appearance information along each edge (e.g., using color histograms), and then matches these features to resolve identity along each edge.

Our system currently uses an iterative approach to estimating multiple trajectories, and for each person solves a batch trajectory estimation problem. This is clearly impractical for real-time, on-line use. To overcome this limitation we are developing an incremental version of the algorithm that maintains an more compact representation of prior trajectory state.

Finally, we have left open the issue of what schedule of gain settings to use when building a background model and when detecting foreground points. A topic of future work is to determine the minimum number of samples needed to obtain a maximally dense integrated range image for a given scene or scene class.

5 Conclusions

This paper presents new algorithms which make tracking objects in widely varying illumination conditions possible. There are two main contributions presented.

First, we formulate stereo range estimation using extended dynamic-range imagery and show how a dense background model can be built with long-term observations. Without this, stereo range data is too sparse to construct a useful background model for tracking. We derive a constraint on the projection of planar uniform surfaces in stereo views and use this to interpolate range within such regions. We implement our scheme with a predictive, ordered disparity search technique, that prunes most of the computation typically required to process new images.

Second, we developed an optimal method for estimating trajectories without performing an initial segmentation of foreground points. Foreground points from multiple views are projected into a plan-view density representation, and are segmented into object regions as part of a globally optimal dynamic programming optimization. Estimation of trajectory extent (object entry/exit) is included in the global optimization, and does not require any spatial constraints. We demonstrated our prototype system estimating the trajectory of several people moving in an environment.

References

[1] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *PAMI*, 12(9):855–867, September 1990.

[2] D.J Beymer. Person counting using stereo. In *Workshop on Human Motion, Austin, Texas*, 2000.

[3] D.J. Beymer and K. Konolige. Real-time tracking of multiple people using stereo. In *Workshop on Frame-Rate Applications, ICCV'99*, 1999.

[4] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV93*, pages 231–236, 1996.

[5] T. Darrell, G.G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *IJCV*, 37(2):175–185, June 2000.

[6] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. *Proceedings of SIGGRAPH 97*, pages 369–378, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California.

[7] C. Eveland, K. Konolige, and R.C. Bolles. Background modeling for segmentation of video-rate stereo sequences. In *CVPR98*, pages 266–272, 1998.

[8] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient matching of pictorial structures. In *CVPR00*, pages II:66–73, 2000.

[9] G.G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *CVPR99*, pages II:459–464, 1999.

[10] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *PAMI*, 22(8):809–830, August 2000.

[11] Y.A. Ivanov, A.F. Bobick, and J. Liu. Fast lighting independent background subtraction. *IJCV*, 37(2):199–207, June 2000.

[12] P. Kornprobst and G. Medioni. Tracking segmented objects using tensor voting. In *CVPR00*, pages II:118–125, 2000.

[13] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *3rd IEEE Workshop on Visual Surveillance (Dublin,Ireland)*, 2000.

[14] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *PAMI*, 22(8):758–767, August 2000.

[15] S.K. Nayar and T. Mitsunaga. High dynamic range imaging: Spatially varying pixel exposures. In *CVPR00*, pages I:472–479, 2000.

[16] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR99*, pages II:246–252, 1999.

[17] M. Trivedi, S. Mikic, and S. Bhonsle. Active camera networks and semantic event databases for intelligent environments. In *IEEE Workshop on Human Modeling, Analysis and Synthesis*, 2000.

[18] C.R. Wren, A. Azarbayejani, T.J. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *PAMI*, 19(7):780–785, July 1997.

[19] R. Zahib and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proc. ECCV*, 1994.