Structure and Motion from Road-Driving Stereo Sequences

Hoang Trinh
Toyota Technological Institute
Chicago IL
ntrinh@ttic.edu

David McAllester
Toyota Technological Institute
Chicago IL
mcallester@ttic.edu

Abstract

We introduce a unified framework for scene structure and motion estimation on road-driving stereo sequences. This framework is based on the slanted-plane scene model that has become widely popular in the stereo vision community. Our algorithm iteratively and alternately solves for scene structure and motion. Surface estimation is done using our own slanted-plane stereo algorithm. Motion estimation is achieved by solving a MRF labeling problem using Loopy Belief Propagation. We show that with some specific assumptions about the motion of the camera and the scene, the motion estimation problem can be reduced to a 1D search problem along the epipolar lines. We also propose a novel evaluation metrics, based on the notion of view prediction error. This metrics can be used to evaluate the performance of structure and motion estimation algorithms on stereo sequences without ground truth data. Experimental results on road-driving stereo sequences demonstrate that our algorithm successfully improves the view prediction error although it was not designed to directly optimize this quantity.

1. Introduction

Multi-view video sequence is the richest form of image data for scene reconstruction. A subtype in this form that we consider very interesting are stereo video sequences. So far, to our knowledge, this useful data type has hardly been investigated and exploited for the purpose of recovering scene structure and motion. Some of the very few research work that has discussed this issue was introduced by D. Min et al. [14] and F. Huguet et al. [10], in which the authors presented variational methods for scene flow estimation from calibrated stereo image sequences. This data type is interesting for two reasons. First, as opposed to a multi-view video sequence, a stereo sequence can easily be captured using only one moving calibrated binocular camera. Second, this data type provides us with enough constraints to conveniently compute location and motion of scene points simultaneously, since coupling dense stereo matching with motion estimation helps decrease the number of unknowns per image point. More specifically, for stereo sequences, we can formulate structure and motion estimation as an energy minimization problem, in which the model is either an extension of a stereo vision model ([16]) to also handle scene point motion, or an extension of a Structure from Motion or optical flow model ([18]) under a stereo setup. One example of using this latter formulation is the work presented in [21].

In this paper, we follow the former approach. Based on our slanted-plane stereo model, we develop a new algorithm for structure and motion estimation on road-driving stereo sequences. This framework is based on the slanted-plane scene model that has become widely popular in the stereo vision community [17]. Our algorithm iteratively and alternately solves for scene structure and motion. Surface estimation is done using our own slanted-plane stereo algorithm. Motion estimation is achieved by solving a MRF labeling problem using Loopy Belief Propagation. We show that with some specific assumptions about the motion of the camera and the scene, the motion estimation problem can be reduced to a 1D search problem along the epipolar lines. We also propose a novel evaluation metrics, based on the notion of view prediction error. Again, it is also much easier and more affordable to collect unlabeled stereo video data than to collect stereo video with associated ground truth (which usually requires using multiple sensors followed by a data fusion step). We argue that view prediction error can be used to evaluate the performance of structure and motion estimation algorithms on stereo sequences without ground truth data. Experimental results on road-driving stereo sequences support our argument by demonstrating that our algorithm successfully improve the view prediction error although it was not designed to directly optimize this quantity. We believe view prediction can be used as a quantitative performance measure on unlabeled multi-view datasets in a variety of applications.

The paper is organized as follows. In section 2, we describe our slanted-plane stereo algorithm. Our proposed unified framework is presented in section 3. Section 4

demonstrates experimental results on several stereo road-driving sequences. Also in this section, we introduce a new evaluation metrics based on the concept of view prediction error. Conclusions are in section 5.

2. The Slanted Plane Stereo Model

2.1. The Model

The use of slanted-plane model for stereo vision was first proposed by BirchField and Tomasi in [4] as a special case of the affine model. Since then the model has been widely applied in many other stereo algorithms, including most of state-of-the-art algorithms in the current Middlebury benchmark. ([12, 20, 15], etc). The model assumes that the scene structure is locally planar - the scene is composed of a set of planar regions. This is a reasonable assumption, based on the fact that most surfaces found in a natural environment can be approximated by a plane or a set of planes.

Our stereo vision model is a slanted plane model involving shape from texture cues, almost the same as that introduced in [19]. Our stereo algorithm infers a disparity plane for each superpixel by minimizing a high-dimensional global energy function. The energy function involves three terms - a correspondence energy measuring the degree to which the left and right images agree under the induced disparity map, a smoothness energy measuring the smoothness of the induced depth map, and an texture energy.

$$E(Z) = E_M(Z) + E_S(Z) + E_T(Z) \tag{1}$$

where Z is the assignment of a disparity plane to each superpixel in the left image. We denote X to be the left image, Y to be a right image. For each superpixel i of X we have that Z specifies three plane parameters A_i B_i, and C_i. Given an assignment Z of plane parameters to superpixels we define the disparity $d(p)$ for any pixel p as follows where $i(p)$ is the superpixel containing p and x_p and y_p are the image coordinates of p.

$$d(p) = A_{i(p)}x_p + B_{i(p)}y_p + C_{i(p)} \tag{2}$$

So by equation (2) we have that Z assigns a real valued disparity to each pixel.

To define the smoothness energy we write $(p, q) \in B_{i,j}$ if p is a pixel in superpixel i, q is a pixel in superpixel j, and p and q are adjacent pixels (p is directly above, below, left or right of q). The smoothness energy is defined as follows where τ_S and λ_S are parameters of the energy, i ranges over all superpixels, $N(i)$ is the set of superpixels adjacent to i, and z is the overall assignment of three dimensional vectors of plane parameters (A_i, B_i, C_i) for all superpixels i:

$$E_S(Z) = \sum_{i,j \in N(i)} \min \left(\tau_S, \sum_{(p,q) \in B_{i,j}} \lambda_S |d(p) - d(q)| \right) \tag{3}$$

Intuitively the minimization with τ_S corresponds to interpreting the entire boundary between i and j as either an occlusion boundary or as a joining of two planes on the same object.

Next we consider the match energy. We write $p - d(p)$ for the pixel in Y that corresponds to the pixel p in image X under the disparity $d(p)$. For color images we construct a nine dimensional feature vector $\Phi^X(p)$ and $\Phi^Y(p)$ for the pixel p in the images X and Y respectively. The vector $\Phi^X(p)$ consists of three (bias gain corrected) color values plus a six dimensional color gradient vector and similarly for Φ^Y_p. We write $\Phi^X_k(p)$ for the kth component of the vector $\Phi^X(p)$. The match energy is defined as follows where λ_k are nine scalar parameters of the match energy.

$$E_M(Z) = \sum_{p \in X} \sum_k \lambda_k (\Phi^x_k(p) - \Phi^y_k(p - d(p)) \,)^2 \tag{4}$$

$E_T(Z)$ is the texture energy term which we call the HOG (Histogram of Oriented Gradients) component. The idea of using HOG ([6])as a surface orientation cue has for the first time been carefully investigated in [19]. Since the surface orientation of a plane is a component of the plane parameters, this surface orientation cue can be embedded into our existing slanted-plane stereo model as an additional energy term. Specifically this texture energy measures the degree to which the surface orientation at each point agrees with the monocular texture based surface orientation cue (more details can be found in [19]).

For the model parameters (τ_S, λ_S, λ_k, etc), we use the same values as those trained by unsupervised learning in ([19]). The energy terms in (1) can also be interpreted in a probabilistic way as follows. The combination of the smoothness energy term and the texture energy term can be interpreted as an energy $E_Z(X, Z, \beta_Z)$, which determines the probability $P(Z|X, \beta_Z)$. The match energy term can be interpreted as an energy $E_Y(X, Y, Z, \beta_Y)$, which determines $P(Y|X, Z, \beta_Y)$.

2.2. The Inference Algorithm and Results

Given a pair of images (X, Y), and a given segmentation of X, and a given setting of the model parameters, the inference problem is to find an assignment Z of plane parameters to superpixels so as to minimize the total energy in (1). The energy defines a Markov random field. More specifically, the match energy and the texture energy defines the unary potential on each superpixel independently and the smoothness energy defines the binary potential on pairs of adjacent superpixels.

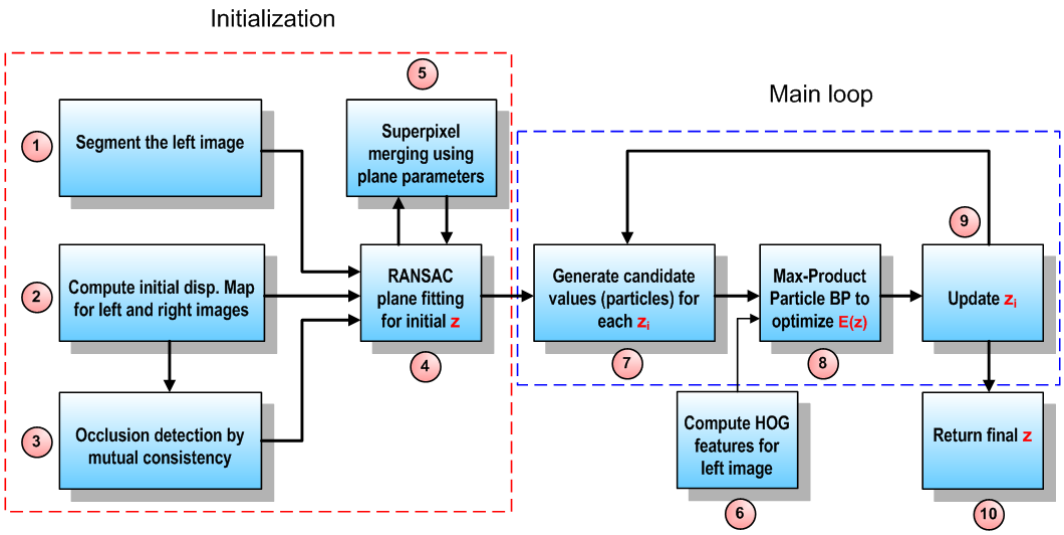

Figure 1. Our inference algorithm.

The complete inference algorithm is illustrated in Figure 1. The steps of the algorithm include:

1. We compute a segmentation using the Felzenszwalb-Huttenlocher segmentation algorithm [7]. Although this segmentation algorithm has a parameter governing the number of superpixels, we do not attempt to tune this parameter with our training algorithm. A more principled approach would include the segmentation itself in the energy functional and tune the segmentation parameter along with the other parameters of the model. However by holding the segmentation parameter fixed, we can treat the segmentation as part of the input data.

2. We run the Felzenszwalb-Huttenlocher's efficient loopy BP stereo algorithm ([8]) to compute the disparity maps for both images X and Y.

3. Mutual consistency check requires that for a particular pixel in the left image, the disparity values between the left and right disparity maps are consistent, i.e:

$$d_L(p) = d_R(p - d_L(p)) \tag{5}$$

   The pixel is considered occluded if (5) does not hold, and considered non-occluded otherwise.

4. The plane fitting is performed in the disparity space, and is applied per superpixel to obtain an initial disparity plane. This is done robustly using RANSAC [9] on the disparity values of the non-occluded pixels only. (similar to [15])

5. This step is optional. In this step we implement a superpixel grouping algorithm similar to the one used in step 1. However each time we consider merging two adjacent superpixels, we also look at the difference in their disparity planes computed from step 4. Intuitively we prefer merging adjacent superpixels having very similar disparity planes. Note that if used, this step may change the initial segmentation generated by step 1. Then the output is iteratively fed back into the plane fitting in step 4.

6. At each pixel p in the left image X, we compute a HOG vector $H(p)$ which is a 24 dimensional feature vector consisting of three 8 dimensional normalized edge orientation histograms an 8 dimensional orientation histogram is computed at three different scales.

7. Let C_i be a set of candidate values for z_i derived by repeatedly adding random noise to the current value of z_i.

8. Run discrete max-product BP with the finite value set C_i for each node i.

9. Set z_i to be the best value for i found in step 8 and repeat.

10. The iteration can be stopped after a fixed number of iterations or when the energy is no longer reduced.

Note that the main loop of the algorithm, including steps 7, 8, 9, is very similar to the Particle-based Belief Propagation algorithm described in [11], which is closely related to previous work in [13]. The main differences of our stereo algorithm in this paper with the one in [19] are in steps 3, 4

and 5. Steps 3 and 5 were not included in [19], while with step 4, we replace the plane fitting using linear regression by the robust RANSAC plane fitting, only taking into account the non-occluded pixels computed from step 3. The results in table 1 demonstrate that these modifications improve the performance in the Middlebury benchmark.

We have also run experiments on a set of rectified stereo pairs taken from the Stanford color stereo dataset [1] which has been used to train monocular depth estimation [1, 3, 2]. The images cover different types of outdoor scenes (buildings, grass, forests, trees, bushes, etc.) and some indoor scenes. Our results are directly comparable to [2], i.e we use the same training set and test set. Table 2 shows that we outperform their results.

3. Proposed approach for Structure and Motion

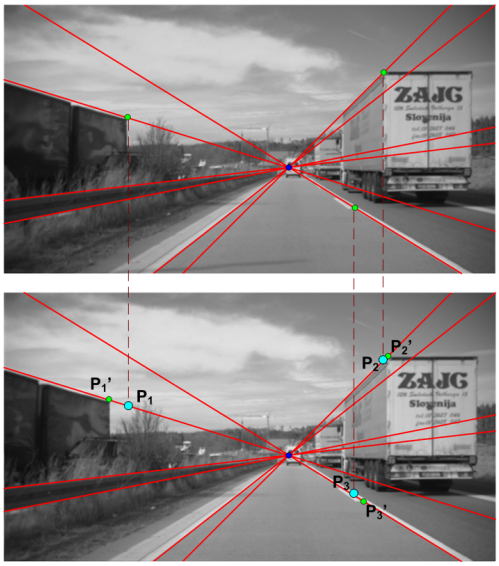

Figure 2. As the camera moves forward, each pixel moves along its corresponding epipolar line. The distance it moves depends on both their depth and velocity. Since the truck on the left has highest velocity w.r.t the camera, P_1 moves the longest distance. On the other hand, the truck on the right has almost zero velocity w.r.t the camera, hence P_2 almost stands still. P_3 reflects the motion of the camera w.r.t the road (the scene background).

In this section, we present a detailed description of our unified formulation for scene structure and motion. At each time step t in the video sequence we consider the stereo pair at time t and at time $t+1$ (Figure 4). We only observe the three frames: I_L^t, I_R^t and I_L^{t+1} and the frame I_R^{t+1} is unobserved. We want to estimate the 3-D location and motion for all pixels in I_L^t. To simplify the derivation of our framework, we use the following assumptions: the camera's viewing direction is the same direction of the Z axis, and all motion vectors in the scene are parallel to the Z axis. Later in section 4 we show that our approach still delivers good estimates even in sequences where these assumptions are violated, e.g: when there is small camera rotation, or rotation of moving objects.

[1] http://ai.stanford.edu/ asaxena/learningdepth/data

3.1. Connection between Disparity and Motion

Our assumptions about the motion of the camera and the scene guarantee the following constraints:

- The image location of the epipoles in all frames of the sequence is constant: As the epipole is simply the projection of the next camera center on the previous frame, this is obvious since the camera always moves along its viewing direction.
- From one frame to the next, a pixel translates along the epipolar line connecting that pixel and the epipole: since all 3D points in the scene move in the same direction with the Z axis, a 3D point always stay in the same 3D line which is parallel to the Z axis. Therefore the epipolar line, which is the projection of this 3D line on the image plane, stays constant, given that the epipole also stays constant. (Figure 2)

Consider a 3D point P in the scene. Let R be the distance from the point to the Z axis. Let r_t be the 2D distance from the projection of P to the epipole in the video frame at time t. We can derive the following mathematical relation between r_t, r_{t+1}, v_t and d_t as follows (This relationship is also illustrated in Figure 3):

$$\begin{aligned} r_t &= \frac{Rf}{z_t} = \frac{Rf}{fh/d_t} = \frac{Rd_t}{h} \\ r_{t+1} &= \frac{Rf}{z_{t+1}} = \frac{Rf}{z_t - \Delta z_t} \\ &= \frac{Rf}{fh/d_t - \Delta z_t} = \frac{Rd_t}{h(1 - d_t \Delta z_t / fh)} \end{aligned}$$

$$\frac{r_{t+1}}{r_t} = \frac{1}{1 - d_t \Delta z_t / fh} = \frac{1}{1 - d_t v_t} \tag{6}$$

$$\frac{d_{t+1}}{d_t} = \frac{r_{t+1}}{r_t} = \frac{1}{1 - d_t v_t} \tag{7}$$

where: f is the focal length, z_t is the depth of P at time t (the actual distance of P w.r.t the camera), h is the stereo baseline, d_t is the disparity of the projected pixel of P, v_t is the velocity value of P. Note that in the derivation above we make use of the following relationship between d_t and z_t: $z_t = fh/d_t$.

	Tsukuba			Venus			Teddy			Cones			Avg.
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	bad
Hoang et al. [19]	3.12	5.22	13.9	1.03	1.17	11.5	7.08	7.30	16.1	6.90	10.7	16.0	8.33%
Our algorithm	2.58	4.66	11.8	0.47	0.64	6.1	6.72	6.98	16.1	6.93	9.33	16.6	7.41%

Table 1. Performance on the Middleburry stereo evaluation. The numbers shown are for unsupervised training with texture features.

	RMS Disparity Error (pixels)	Average Error $\lvert \log_{10} Z - \log_{10} \hat{Z} \rvert$
Saxena et al. [2]		.074
Our algorithm	1.1608	.0723

Table 2. RMS disparity error (in pixels) and average error (average base 10 logarithm of the multiplicative error) on the Stanford stereo pairs for our algorithm.

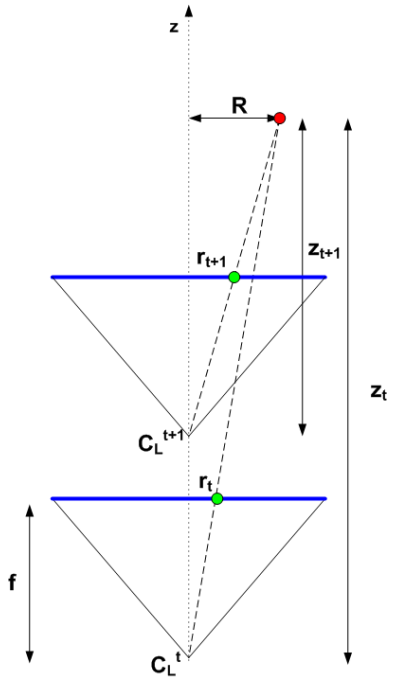

Figure 3. The geometric interpretation of equation (6).

3.2. 3-frame Model for Depth and Motion estimation

At each time step t in the video sequence we consider the stereo pair at time t and the left frame at time $t+1$. The frames are labeled as in Figure 4. By equation (6), we can see that if we know d_t and v_t, then we can compute r_{t+1}, which means solving for the correspondence between I_L^t and I_L^{t+1}. In other words, we converted the 2D optical flow field problem to the stereo disparity estimation problem and a 1D velocity assignment problem. We define the following energy functions:

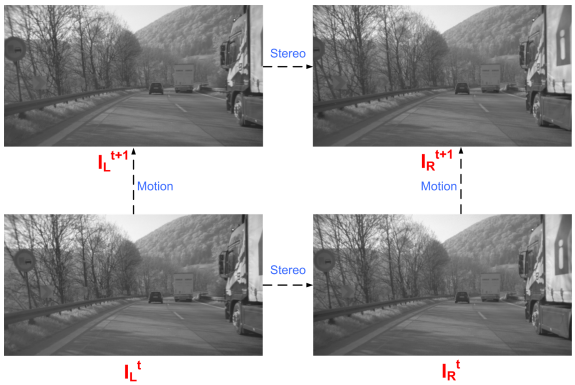

Figure 4.

$$E = E_{Stereo} + E_S^v + E_M^v \quad (8)$$

$$E_S^v = \sum_{P,Q} \min\left(\tau_v, \lambda_v \left|v_P - v_Q\right|\right) \quad (9)$$

$$E_M^v = \sum_p \sum_k \lambda_k \left(\phi_k^t(p) - \phi_k^{t+1}(p + Q(p, d_p, v_p)) \right)^2 \quad (10)$$

where E_{Stereo} is the function in (1), $Q(p, d_p, v_p)$ is the function mapping a pixel in I_L^t to a pixel in I_L^{t+1}, and $\phi^t(p)$ is the k-dimensional feature vector corresponding to pixel p in frame at time t. The objective is to find the optimal co-assignment for depth and velocity that minimize the global energy function in (8).

$$(d^*, v^*) = \underset{(d,v)}{\operatorname{argmin}}(E)$$

The optimization of E_{Stereo} is introduced in section 2.

Minimizing the sum of the other two energy terms: $E_{Motion} = E_S^v + E_M^v$ can be considered another MRF labeling problem: we want to assign a velocity value to each superpixel such that E_{Motion} is minimized. The first term E_S^v penalizes the difference in velocity between two adjacent superpixels, the second term E_M^v is the data cost of assigning a velocity v_p to pixel p, taking into account the image data agreement between frame I_L^t and frame I_L^{t+1}. We solve this MRF labeling problem by Loopy Belief Propagation. We used the max-product BP algorithm with conceptually parallel updates. In our actual implementation however, they were performed sequentially.

Finally, once we have obtained an initial estimate of v_t, we can fix v_t and optimize the function $E^{3frame}_{Stereo} = E_{Stereo} + E^v_M$. Note that this function has 1 more data term compared to the previous stereo function as we can now incorporate data from I^{t+1}_L. We construct an iterative algorithm alternately optimizing the disparity d_t and the velocity v_t for frame I^t_L. Here is the outline of our algorithm:

1. Compute the epipole at I^t_L using a version of the 8-point algorithm.
2. Estimate the disparity map d_t of I^t_L using our stereo algorithm.
3. Use SIFT feature matching to obtain a set of initial velocity values: Each pair of matched SIFT features give us 1 initial velocity.
4. Estimate v_t given d_t: Run Loopy BP to optimize E_{Motion} and assign a velocity value to each superpixel in I^t_L.
5. Fixing v_t, reestimate d_t by optimizing E^{3frame}_{stereo}.
6. Repeat from Step 4.

4. Experimental Results

4.1. Evaluation Metrics

View prediction is a general notion that has been motivated by the two-view learning approach in machine learning [5]. In two-view learning, the goal is to predict the second view given the first. In our context the goal of view prediction is to predict the unobserved data y given the observed data x and the latent variables w.We can express a probabilistic interpretation of view prediction as follows:

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{m} \ln 1/P_w(y|x)$$

where $P_w(y|x)$ is the conditional probability of the unobserved data given the obserbed data parametrized by w. As our algorithm never observed I^{t+1}_R, in our problem setting we might define $P_w(y|x)$ as the $P(I^{t+1}_R|I^t_L, I^t_R, I^{t+1}_L, d_t, v_t)$. The idea is that the more accurate we estimate (d_t, v_t), the more accurate we can predict I^{t+1}_R.

Given the estimated depth and velocity d_t, v_t, we can compute the prediction $\hat{I}^{t+1}_R$ as follows:

$$I^t_L \overset{d_t,v_t}{\rightarrow} I^{t+1}_L \overset{d_{t+1}}{\rightarrow} \hat{I}^{t+1}_R$$

where d_{t+1} is computed using (7). An example of a predicted fourth view is shown in Figure 5.

Figure 5. Top: original fourth frame I^{t+1}_R. Bottom: predicted fourth frame $\hat{I}^{t+1}_R$. The black pixels are missing values caused by bilinear interpolation. These pixels are excluded when computing the view prediction error.

The view prediction error is defined as the pixel RMS between I^{t+1}_R and $\hat{I}^{t+1}_R$:

$$err(I^{t+1}_R, \hat{I}^{t+1}_R) = \sqrt{\frac{\sum_{p=1}^{N}(I^{t+1}_R(p) - \hat{I}^{t+1}_R(p))^2}{N}} \quad (11)$$

In the absence of ground truth data, it is natural to use view prediction error as a useful tool for quantitative analysis, i.e. to measure how good the estimation of latent variables is. In the specific setting of our problem, the latent variables $w^* = (d^*, v^*)$.

4.2. Results on road-driving stereo sequences

We tested our algorithm on 7 gray scale road-driving stereo sequences provided by Daimler AG. Each sequence contains from 250 to 300 rectified, bias gain corrected stereo pairs, taken under different light condition and road setting. Ground truth depth and motion is not available for these datasets. The algorithm estimates the dense map of disparity and velocity value for each left frame in the sequence. Figure 6 demonstrates our results on several frames in a sequence.

Table 3 shows the average view prediction error (11) over seven video sequences after 3 iterations of the algorithm. The results shows that the algorithm successfully improved the estimation over time. The improvement stops after three iterations. Note that the pixel intensity values are scaled to be in the range of $[0 - 1]$.

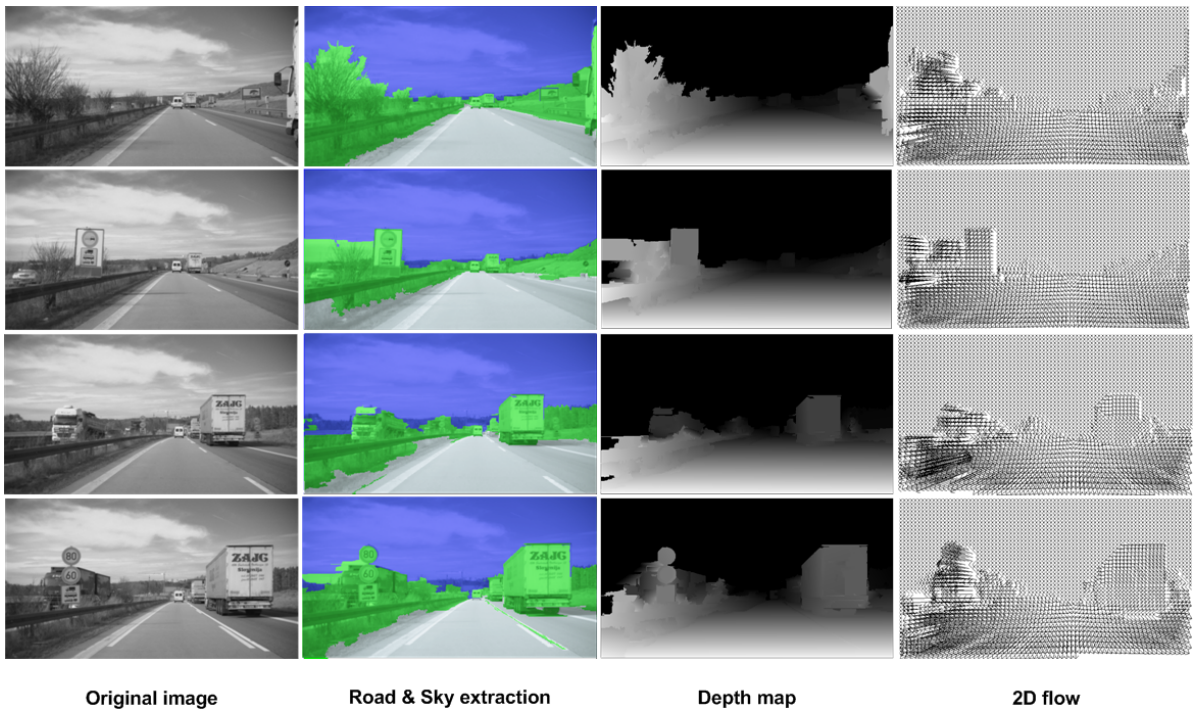

Figure 6. Results on a Daimler road driving sequence.

	Average View Prediction Error
Iter 1	0.0692
Iter 2	0.0622
Iter 3	0.0621

Table 3. Average view prediction error (in pixel intensity value) on 7 Daimler road-driving stereo sequences after each iteration of the algorithm.

5. Conclusion

In this paper we emphasized view prediction as a way of evaluating scene structure and motion estimation from stereo video data in the absence of any ground truth labels. We introduced a new algorithm for structure and motion estimation on road-driving stereo sequences. Based on specific assumptions about the motion of the camera and the scene, we can reduce the 2D optical flow problem to a 1D velocity value problem. Our algorithm iteratively and alternately solve for structure and motion. Scene structure estimation is done using our own plane-based stereo algorithm. Velocity estimation is completed by solving a MRF labeling problem using Loopy BP. Experiments on road-driving stereo sequences showed encouraging results, even with video sequences where the scene and camera motion do not fully comply with our assumptions.

Performance analysis was done using our novel evaluation metrics based on the notion of view prediction error. We argue that this evaluation metrics is quite appropriate for algorithms working with stereo sequences as well as multiple view image data, when ground truth data is not available. Our experimental results support this argument. We used hand-tuned parameters for our model in this paper. Ideally, these parameters should be estimated by an automated method, usually through learning using a labeled training data set. With view prediction error, we believe the problem we solve in this paper can be another setting where we can apply the unsupervised learning approach based on maximizing conditional likelihood. In other words, the model parameters can be learned using only unlabeled stereo video data. It is hoped that a general notion of view prediction will eventually facilitate unsupervised learning in a variety of cases with mutual information between views.

References

[1] A. Y. N. Ashutosh Saxena, Sung H. Chung. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, 2005.

[2] J. S. Ashutosh Saxena and A. Y. Ng. Depth estimation using monocular and stereo cues. In *International Joint Conference on Artificial Intelligence*, 2007.

[3] M. S. Ashutosh Saxena and A. Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 2007.

[4] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision*, 1999.

[5] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*, 1998.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficiently computing a good segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[9] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[10] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *International Conference on Computer Vision*, 2007.

[11] A. Ihler and D. McAllester. Particle belief propagation. In *International Conference on Artificial Intelligence and Statistics*, 2009.

[12] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *International Conference on Pattern Recognition*, 2006.

[13] D. Koller, U. Lerner, and D. Angelov. A general algorithm for approximate inference and its application to hybrid Bayes nets. In *Conference on Uncertainty in Artificial Intelligence 15*, pages 324–333, 1999.

[14] D. Min and K. Sohn. Edge-preserving simultaneous joint motion disparity estimation. In *International Conference on Pattern Recognition*, 2006.

[15] R. Y. H. S. Q. Yang, L. Wang and D. Nistr. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3), 2008.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 2002.

[17] D. Scharstein, R. Szeliski, and R. Zabih. http://vision.middlebury.edu/stereo/, 2001.

[18] J. L. S. R. M. J. B. Simon Baker, Daniel Scharstein and R. Szeliski. A database and evaluation methodology for optical flow. Technical Report MSR-TR-2009-179, December 2009.

[19] H. Trinh and D. McAllester. Unsupervised learning of stereo vision with monocular cues. In *British Machine Vision Conference*, 2009.

[20] Z. Wang and Z. Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, 2008.

[21] Y. Zhang and C. Kambhamettu. On 3d scene flow and structure estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.