BAYESIAN BASED 3D SHAPE RECONSTRUCTION FROM VIDEO

Nirmalya Ghosh and Bir Bhanu
Center for Research in Intelligent Systems (CRIS), University of California, Riverside, CA 92521, USA
{nirmalya, bhanu}@ee.ucr.edu

ABSTRACT

In a video sequence with a 3D rigid object moving, changing shapes of the 2D projections provide interrelated spatio-temporal cues for incremental 3D shape reconstruction. This paper describes a probabilistic approach for intelligent view-integration to build 3D model of vehicles from traffic videos collected from an uncalibrated static camera. The proposed Bayesian net framework allows the handling of uncertainties in a systematic manner. The performance is verified with several types of vehicles in different videos.

Index Terms – Learning, 3D shape from video

1. INTRODUCTION

Shapes may change due to activities of flexible objects (e.g., human face), or due to 3D-to-2D projections from different viewpoints (e.g., a moving object in video). For face, changing shapes give problem to face recognition algorithms. While in 3D reconstruction, changing shapes and their interrelations provide valuable cues for incremental view integration. Actually, this spatio-temporal information flow is the key to 3D perception of shapes in 2D videos. This paper presents a Bayesian net framework [15] to handle uncertainties in the evidences and conditional dependencies among the intermediate variables to build 3D model of the moving vehicles in typical traffic scenarios. Uncalibrated static video camera provides different 2D views that gradually change to reveal new 2D features and/or hide previously seen features. Incremental view integration sews the evidences to learn the entire 3D model.

Most of the 3D model building methods use either range data [1, 2] or calibrated camera setup [3, 4, 5]. Range data is used for pose registration [1] or shape correlation and integration [2]. With calibrated setup, multi-view stereo reconstructions uses min-graph cut and triangulation [3], or polarization [4], or dictionary of primitives [5]. Some use modeling of structures from reflections [6]. For vehicle 3D model building from traffic video, range sensors are too cumbersome and calibrated setups [7, 16] with turntable based toy problems are often impractical. In [8], extended Bayesian net (EBN) is used to reconstruct 3D model of buildings from multiple aerial views. But motion, dynamics and temporal order are not utilized.

Most of the vehicle-centric image processing and computer vision methods are in 2D. A recent work [9] has used similar views, pose templates and primary edge directions [9] where entire vehicle is seen, which is *not* a valid assumption for traffic video with vehicles in motion [10]. 3D models have been seldom used, for 3D-2D projection, in the context of extended Kalman filter (EKF) based tracking [13], or edge/region based tracking under calibrated setup [11, 12]. But calibrating traffic cameras is a difficult proposition and hence avoided in [10] where 3D generic model and directional templates are used to map 2D-to-3D, using incremental clustering for 3D vehicle model reconstruction, with a non-probabilistic reliability measure. Notably, this work bypassed the stochastic nature of frame-based features and hence the probabilities of the 3D estimations over time.

None of the previous works has explicitly used spatio-temporal evidences across the frames to build 3D model with systematic handling of uncertainty in video sequences. *The **key contributions** of the current work are: **(1)** 3D model building of several vehicles from uncalibrated videos and **(2)** stochastic scalable view integration in video.*

2. TECHNICAL APPROACH

We use a single 3D generic model of vehicles (Fig 1(a)) that can represent several types of vehicles (Fig 1(b)). With the vehicle moving in an uncontrolled traffic scene, views of the vehicle changes gradually with different 2D features having different lifecycles that are interdependent spatially and temporally. On the basis of the 2D features observed so far in the video sequence, parameters of the generic model are incrementally tuned to instantiate the 3D model of the target vehicle. The approach consists of the following steps:

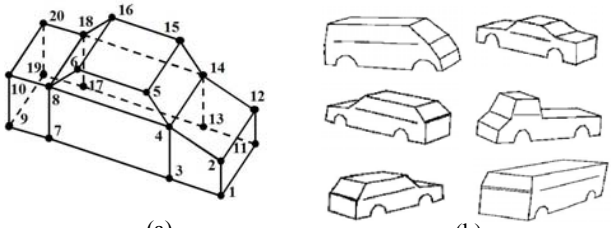

Fig 1: **(a)** 20-vertex-15-surface generic vehicle model with generic vertex numbers **(b)** Examples of represented vehicles.

Step 1: Feature Extraction

We use 2D corners and linear edge-segments (referred as "*lines*" subsequently) as the 2D features. Due to video-smoothness, lane-based-motion and structural-rigidity

constraints, instead of independent frame feature extractions in [10], we employ global prediction-verification method to reduce computational complexity significantly. For every new features appearing for the first time, we initialize 2D location and track the global frame-structure of the vehicle using 11x11 window search with intensity based matching over 5x5 patches around the 2D corners.

Prior probability $p(e_i)$ of individual edge-point (e_i) is defined by its normalized edge-magnitude. Least square line fit [14] of close edge-points defines a line (E). The probability p(E) is defined by normal distances (d_i) of the edge-points (e_i) from the line E and the $p(e_i)$'s (eqn (1)).

$$p(E) = \frac{\sum_{i:\text{all edge points for this line}} [p(e_i)/(d_i + 1)]}{\sum_i [1/(d_i + 1)]} \quad (1)$$

The intersections of lines (E) define the 2D vertices (V). When several lines meet at close points $\{V_p\}$, we take the weighted mean of them as V. The conditional probability of a vertex, given the lines, (i.e., $p(V| E_i, E_j)$), is defined (as in eqn (2)) by the distances $d_{V_p,V}$ between the centroid V and the corresponding individual points V_p's.

$$p(V|E_i, E_j) = \left[1/(d_{V_p V} + 1)\right] \Big/ \sum_{p:\text{all possible pairs}} \left[1/(d_{V_p V} + 1)\right]$$
$$p\,(V|E_i, i \in \text{all connected edges}) = 1 - \prod_{p:\text{all closely intesecting pairs}} \left(1 - p\,(V|E_{p,1}, E_{p,2})\right) \quad (2)$$

Step 2: Motion estimation

Local displacements are algebraic difference vectors (D_i) of the locations of corresponding corners (V_i) seen in *both* the consecutive video frames. Hence conditional probability of each D given V's is always 1. The global 2D motion for current frame (M^t) is defined by the mean of the Gaussian fit of the D's. Conditional probability of the M^t given the D's is defined by the similarity between their directions, as in eqn (3). M^t also depends on M^{t-1}, the motion in the last frame. In a high data rate (frames per second (fps)) video, smooth flow of the traffic implies that the global motion can change only slowly across the frames. Thus, the current frame motion (M^t) should be close to the last frame motion (M^{t-1}) to satisfy this smoothness. The conditional dependence $p(M^t | M^{t-1})$ is defined by the probability mass for the Gaussian pdf of M^{t-1} at current value M^t. Present work considers only smooth motion direction.

$$p(M^t | D_i^t\ \forall i, M^{t-1}) = p(M^t | D_i^t\ \forall i).\, p(M^t | M^{t-1})$$
$$p(M^t | D_i^t\ \forall i) = \prod_{k:\text{all D's}} p(M^t | D_i^t) = \prod_{k:\text{all D's}} \left(\frac{\left[1/(0.2 * \|D_i^t - M^t\| + 1)\right]}{\sum_{k:\text{all D's}} \left[1/(0.2 * \|D_k^t - M^t\| + 1)\right]} \right) \quad (3)$$

Step 3: Orientation estimation

An object-centered coordinate system (OCC) avoids the requirement of calibrated setup for 3D model building. The right hand lower corner of the frontal face of the vehicle (corner 1 in Fig 1(a)) is taken as the OCC origin. Three lines (E_{01}, E_{02}, E_{03}) intersecting at the origin define the directions of the 2D projections (Cr) of the 3D OCC axes. Line (closely) parallel to M^t is the 3D Y-axis, i.e., Cr(2). Line (closely) parallel to the image-Y axis, is the 3D Z-axis, i.e. Cr(3) (we constrain Cr(3) = 90°). And the line furthest from the direction of M^t, is the 3D X-axis, i.e. Cr(1). The conditional probability of Cr, given (E_{01}, E_{02}, E_{03}) and M^t, is defined by the angular relations between them, as in eqn (4).

$$p(Cr \mid E_{01}, E_{02}, E_{03}, M^t) = \frac{\left|Cr(1) - M^t\right| - \left|Cr(2) - M^t\right|}{\sum_{i:1,2,3} \left|Cr(i) - M^t\right|} \quad (4)$$

For street constraints, orientation can change only in azimuth. Hence we use *directional template* method proposed in [10]. Each directional template vector (T_i) in the library contains (i) 3D azimuth angle ($T_{i,1}$) (ii) 2D angles of the projections of 3D coordinate axes for this azimuth ($T_{i,2\text{-}4}$), and (iii) the corresponding scale factors ($T_{i,5\text{-}7}$) in OCC axes directions to map 2D observed distances to 3D estimated distance. As these templates (T_i, i=1, 2, .. 180) are known a priori, they are root nodes with equal probabilities.

The closest match of *Cr* vector in the template library, (precisely, between $Cr_{1\text{-}3}$ and $\{T_{i,2\text{-}4}\}_{i=1}^{180}$) determines the 3D orientation (azimuth) of the vehicle and this closest template is called 2D-to-3D mapping vector for the present frame (MV^t). Its conditional probability depends on the Euclidian distances between Cr and the directional templates, actually similarity between them, as shown in eqn (5). MV^t also depends on MV^{t-1}, the mapping vector for the previous frame. Street lane restricts wide variation in orientation (azimuth) of a moving vehicle, specifically in a high fps video. So the mapping vector in the current frame (MV^t) causally depends on its counterpart in the last frame (MV^{t-1}). The conditional dependence $p(MV^t | MV^{t-1})$ is defined by the probability mass of Gaussian pdf of MV^{t-1} at MV^t.

$$p(MV^t \mid Cr, T_{i^*}, MV^{t-1}) = p(MV^t \mid Cr, T_{i^*}).\, p(MV^t \mid MV^{t-1})$$
$$p(MV^t \mid Cr, T_{i^*}) = \left[1/\min_{k:2,3,4} \left|T_{i^*}(k) - Cr\right|\right] \Big/ \left[\sum_{k:2,3,4} 1/\left|T_{i^*}(k) - Cr\right|\right] \quad (5)$$

Step 4: Mapping 2D features to 3D - Estimation Propagation

We start mapping from the OCC origin taken as [0 0 0] in 3D. A path, consisting of connected lines (E) from the OCC origin to any particular vertex (V), is used to propagate the 2D-to-3D mapping. Among several possible paths, we select the best one with the highest unified score (F). F is defined for the collection of lines in the path, and depends on three factors (in the descending order of importance): **(i)** the number of lines *not* parallel to the OCC axes directions (from MV), **(ii)** the number of lines, and **(iii)** the average probability of existence of the lines (E). Conditional probability of the best path is defined by the mean of the prior probabilities of the lines in the path. Due to newly encountered features and different noise statistics for different frames, the best path for the same corner can evolve over the frame sequence. Thus, proposed method incrementally considers available evidences in the video.

Step 5: Single-frame 3D Model Estimation

2D to 3D mapping of a vertex (V) starts from the OCC origin (3D coordinates [0 0 0]) and propagates estimation along the best path (P) of the vertex. We find that 3D location of a vertex (V_i^{3D}) depending on the 2D vertex (V_i), the scale factors in the mapping vector (MV^t) and the best path of the vertex (P_i). The conditional probability in eqn (6) is related with average directional similarity of the lines (in the path) to the 3D OCC-axes projection directions in the mapping vector (MV^t).

$$p\left(V_i^{3D} \mid P_i, V_i, MV^t\right) = \sum_{k:\ \text{lines in } P_i} p(E_k) * \exp\left[-0.5 * \min_{m:2,3,4} \left|\angle E_k - MV^t(m)\right|\right] \Big/ \sum_k 1 \quad (6)$$

The generic model (in Fig 1(a)) provides two structural constraint matrices for vertices (VSC) and lines (LSC), as defined in the eqn (7) and (8) respectively. As these are known a priori, they have prior probabilities equal to 1.

$$VSC(i,j) = \begin{cases} 0 & \text{vertices i \& j not connected} \\ 1 & \text{connected by a line paralle to OCC X - axis} \\ 2 & \text{connected by a line parallel to OCC Y - axis} \\ 3 & \text{connected by a line parallel to OCC Z - axis} \\ 4 & \text{connected by a line NOT parallel to any OCC axis} \end{cases} \quad (7)$$

$$p\left(VSC(i,j)\right) = 1 \quad \forall i,j$$

$$LSC(i,j) = \begin{cases} 0 & \text{no relation for line i \& j} \\ 1 & \text{parallel} \\ 2 & \text{colinear/concurrent} \\ 3 & \text{colinear/seperate} \\ 4 & \text{perpendicular/concurrent} \\ 5 & \text{perpendicular/seperate} \\ 6 & \text{just concurrent} \end{cases} \quad (8)$$

$$p\left(LSC(i,j)\right) = 1 \quad \forall i,j$$

Step 6: Incremental 3D Model Estimation

We also compute structural relations for the estimates of the 3D vertices (VSR) and the 2D lines (LSR) by eqn (7) and (8). Comparison of (VSR and LSR) with their generic counterparts (VSC and LSC) provides structural similarity. The conditional probabilities of the 3D model parameters, given the structural constraints, $p(V_i^M \mid VSC)$ and $p(E_i^M \mid LSC)$ are defined by these similarity values, normalized by the total number of vertices and lines in the generic model.

The 3D location of a vertex (V_i^M) of the 3D incremental model of the current vehicle are computed by a linear combination of the estimate from only the current frame (V_i^{3D}) from eqn (6) and its incremental counterpart in the previous frame ($V_i^{M\_last}$), where the weight for the last one depends on the number of frames previously seen (i.e., reliability depend on the number of frames considered). The conditional probability of the vertices in the incremental 3D model $p\left(V_i^M \mid V_i^{3D}, V_i^{M\_last}, VSC\right)$ is also defined as a weighted normalized sum of the individual conditional factors $p(V_i^{3D})$ (from eqn (6)), $p(V_i^{M\_last})$, and $p(V_i^M \mid VSC)$. Similar is the case for $p\left(E_i^M \mid E_i^{3D}, E_i^{M\_last}, LSC\right)$. In all of the last three weighted sums, weights depend on the number of frames considered for the computation (i.e., parameters of the previous frame are weighted more than the one from only the present frame). Note that the 3D locations of the vertices along with their connectivity matrix from the 2D line features specify the entire 3D model. Hence we define a unified model probability by the average of the probabilities of the *visible* vertices (eqn (9)).

$$p\left(Model \mid \{V_i^M \ \forall i\}\right) = \frac{\sum_{i:\text{seen vertices}} p\left(V_i^M \mid V_i^{3D}, V_i^{M\_last}, VSC\right)}{\sum_{i:\text{seen vertices}} 1} \quad (9)$$

3. EXPERIMENTAL RESULTS

We have used traffic video data collected in two different locations with a static uncalibrated video camera and several types of vehicles. Sample frames of the videos in Fig 2 show the changing shapes of the 2D projections of the vehicles in the image plane.

Fig 2: Sample frames of the traffic videos of several types of moving vehicles in two different traffic locations. **Rows: (1)** Sedan type 1, **(2)** Sedan type 2, **(3)** SUV, **(4)** Jeep, **(5)** Pickup, **(6)** Van.

We show the incrementally estimated 3D model of the Van learned from the projected-shape changes in the video in Fig 3. Initially, when we see only few frames, too little information over-estimates model correctness (Fig 4(f)). The drop in the probability is actually due to nullification of this effect. Then, as we see more number of frames, robustness increases and hence, in general, the probability of correctness of the 3D model features increases.

For brevity, we summarize the performance of the proposed work in terms of the learning curves of the Bayesian framework for the model probability in eqn (9).

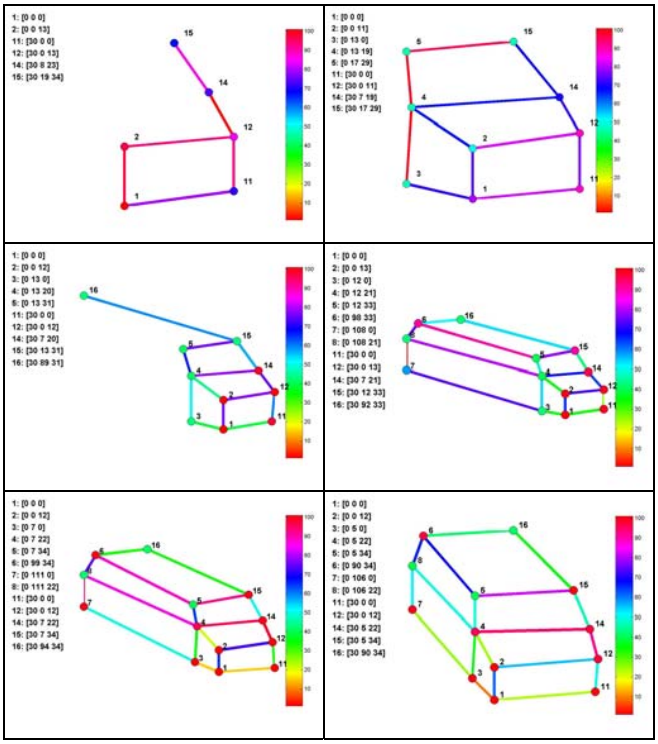

Fig 3: Incrementally learned 3D models from changing projections of the van, in row-column order, for frames 17, 22, 47, 77, 230, and 260 respectively. Probabilities of correctness of the 3D model parameters are color-coded (scale superimposed). Estimated 3D locations of the model corners are shown for each frame as well.

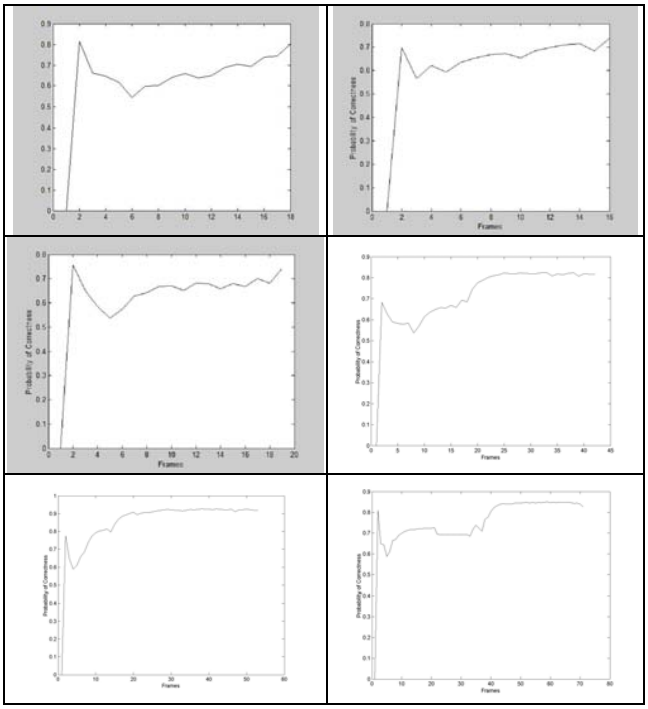

Fig 4: Probability of learned models over the length of videos (in row-column fashion): Sedan type 1, Sedan type 2, SUV, Jeep, Pickup, and Van. X-axis: Frame numbers, Y-axis: Model probability from eqn (9).

We acquired Car1 video data from the publication in [10]. For this video, very little top-view of the vehicle is visible. As a result the estimates for vertices (numbers > 10) on the far-side OCC YZ plane are not highly accurate. We get 0.8 probability of correctness by our proposed approach, compared to 0.65 reliability values in [10].

4. CONCLUSION

This paper shows how spatio-temporal interdependently changing projected 2D shapes of a rigid 3D object can be used to integrate views incrementally for 3D shape reconstruction. A Bayesian incremental learning method is proposed that tunes the parameters of a single 3D generic model to estimate the 3D model of the target vehicle. Performance for several types of vehicles shown in this work validates the potential of the proposed approach for 3D model building from uncalibrated videos of moving rigid objects.

Acknowledgements: This work was supported in part by NSF grants 0551741 and 0622176. The contents of the information do not reflect the position or policy of the US Government.

5. REFERENCES

[1] S-Y Park, & M. Subbarao. "Pose estimation and integration for complete 3D model reconstruction", WACV'02, 143-147.

[2] P. Claes, et al. "Partial surface integration based on variational implicit functions and surfaces for 3D model building". Intl Conf. 3-D Dig. Image & Mod, 2005, 31-38.

[3] A. Hornung, & L. Kobbelt, "Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding", IEEE CVPR'06 (1) 503-510.

[4] G. Atkinson, & E. Hancock, "Multi-view surface reconstruction using polarization", IEEE ICCV'05 (1) 309-316.

[5] A. Barbu, & S.-C. Zhu, "Incorporating visual knowledge representation in stereo reconstruction", ICCV'05(1) 572-579

[6] P.-H. Huang, & S.-H. Lai, "Contour-Based Structure from Reflection", IEEE CVPR'06 (1) 379-386.

[7] S. Seitz, et al. "A comparison and evaluation of multi-view stereo reconstruction algorithms", IEEE CVPR'06(1) 519-528

[8] Z.W. Kim, & R Nevatia. "Expandable Bayesian networks for 3D object description from multiple views and multiple mode inputs", IEEE Trans. PAMI 25(6): 769-774, 2003.

[9] Y. Shan, et al. "Unsupervised learning of discriminative edge measures for vehicle matching between non-overlapping cameras", IEEE CVPR'05 (1) 894-901.

[10] N. Ghosh, & B. Bhanu. "Incremental vehicle 3D modeling from video", ICPR'06, (3) 272-275.

[11] S. Gupte, et al. "Detection and classification of vehicles", IEEE Trans. Intelligent Transport. Sys. 3(1): 37-47, 2002.

[12] J. Lou, et al. "3-D model-based vehicle tracking". IEEE Trans. IP, 14(10): 1561-1569, 2005.

[13] D. Han et al. "Vehicle class recognition from video based on 3D curve probes". IEEE ICCV 2005, VS-PTS Workshop.

[14] P. D. Kovesi. MATLAB Functions. U. of Western Australia. <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>.

[15] K. B. Korb, & A. E. Nicholson. Bayesian artificial intelligence. Chapman & Hall CRC Press. 2004.

[16] R. Hartley, & A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge Univ. Press, March, 2004.