DEPTH PROPAGATION WITH TENSOR VOTING FOR 2D-TO-3D VIDEO CONVERSION

Kuanyu Ju *Yong Li* *Hongkai Xiong*[*]

Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ABSTRACT

Recently, 2D-to-3D video conversion has raised great interest because of the increasing demand of 3D video contents. A critical component in semi-automatic methods is to estimate the depth information of non-key frames from key frames. To overcome the sensitiveness to occlusion in existing schemes, a depth propagation method is proposed in this paper where tensor voting is leveraged to eliminate occlusion points, thereby derive accurate sparse depth. Furthermore, high dimensional tensor is constructed with coordinate, motion and color features to represent points in the sparse depth. A dense depth can be generated via tensor-voting based interpolation. It is robust against object occlusion and inaccurate motion estimation because tensor voting can efficiently make denoising and capture the structures. Experiment results demonstrate that the proposed method outperforms state-of-the-art semi-automatic techniques.

Index Terms— 2D-to-3D video conversion, depth estimation, tensor voting, motion tracking

1. INTRODUCTION

Three dimensional (3D) video can provide an enhanced visual experience with depth perception beyond conventional 2D contents. With the growth of 3D display devices, the increasing demand for 3D contents has aroused a significant challenge to the 3D industry. A promising way is to produce new 3D videos from massive existing monocular 2D videos [1]. A typical 2D-to-3D conversion process consists of two steps: depth estimation and depth-based rendering. Depth estimation is a critical problem because synthesized stereo views cannot be well generated by depth-based rendering without accurate depth (e.g. DIBR [2]).

The existing 2D-to-3D techniques can be divided into two categories: fully automatic methods and semi-automatic methods [3, 4, 5], depending on whether man-machine interactions are involved in depth estimation. Fully-automatic methods are limited to restricted scenarios, thus they do not work well for arbitrary scenes. In contrast, semi-automatic methods can balance 3D content quality with production cost, which makes them more effective and flexible. Aiming at desirable 3D quality, semi-automatic methods exploit a skilled operator who assigns depth to selected key frames in 2D videos. Later, the depth information can be propagated automatically from the key frames to non-key frames over the entire video sequence. Depth propagation is a major part of depth estimation, thereby playing a critical role in semi-automatic methods. In [4], depth is attained by bilateral filtering and refined through a block-based motion compensation from previous frames. In 2011, it was extended in [6], where the depth map is propagated by shifted bilateral filtering with motion information. Li et al. [7] propagated depth for non-key frames via bi-directional motion estimation, where bi-directional motion vectors are estimated to determine the depth propagation strategy. In [9], motion vectors are estimated by the Horn-Schunk optical flow estimation. To alleviate error propagation, post-filtering is performed before estimating depth to the next frame. However, the methods are still sensitive to occlusion and inaccurate motion estimation. The wrong matches can be easily introduced near the occlusion boundary, leading to inaccurate depth map. Such problems motivate us to develop a robust method to propagate accurate depth for 2D to 3D video conversion.

In this paper, a novel depth propagation method is proposed where tensor voting is utilized to eliminate occlusion points, thereby deriving accurate sparse depth. Then, we incorporate the coordinate, motion and color features to represent points in sparse depth as high dimensional tensor. A dense depth can be derived via tensor-voting based interpolation. The proposed method is robust against object occlusion and inaccurate motion estimation because tensor voting can efficiently make denoising and capture the structure. Experiment results demonstrate that the proposed method outperforms state of the art semi-automatic techniques.

The rest of the paper is organized as follows: Section 2 describes the proposed method to propagate depth information accurately for 2D to 3D video conversion. The experimental results in Section 3 are validated to reflect the effectiveness. Finally, Section 4 concludes this paper.

2. THE PROPOSED APPROACH

The proposed method consists of two stages: sparse depth estimation and tensor-voting based depth interpolation. As shown in Fig.1, given two successive frames F_t and F_{t+1},

[*]The work has been partially supported by the NSFC grants 61425011, U1201255, 61271218, 61529101, and Shu Guang project (13SG13).

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

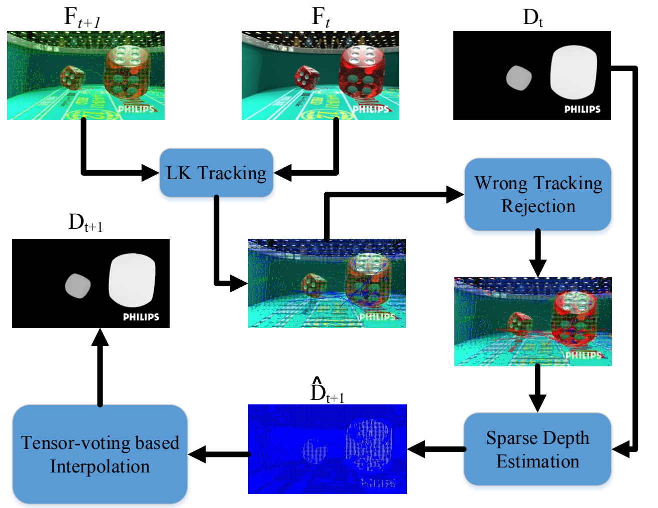

Fig. 1. The framework of the proposed method. Green dots in F_{t+1} denotes initial points. The blue arrows denotes motion vectors while the red arrows denote wrong motion vectors.

we aim to derive the unknown D_{t+1} from known depth D_t of F_t. In the first stage, we uniformly sample the initial points in F_{t+1}. Then, LK tracking is performed to obtain motion vectors (v_x, v_y) of these initial points (x, y). Subsequently, we utilize tensor voting [8] in the 4D space to capture reliable points (x, y, v_x, v_y), which belong to curves or surfaces. The corresponding sparse depth $\hat{D}_{t+1}$ can be obtained. In the second stage, we incorporate coordinate, motion and color features to represent reliable points in 8D space $(x, y, v_x, v_y, r, g, b, d)$. Corresponding 8D tensor is constructed at each point. The final depth D_{t+1} is estimated via tensor-voting based interpolation from the sparse depth $\hat{D}_{t+1}$.

2.1. Sparse Depth Estimation

Since motion-based depth propagation methods [7, 9] are sensitive to object occlusions, which typically occur along the boundaries of objects. Points in occluded areas always get wrong matches during backward tracking. Because tensor voting is noniterative and can tell whether a point is at certain structures or an isolated point, it is adapted in 4D space to eliminate isolated points and overcome this problem. Given a pixel (x, y) with motion vector (v_x, v_y), we represent the corresponding point as (x, y, v_x, v_y) in 4D space.

Initially, a set of initial points are uniformly sampled with certain pixel step in the current frame F_{t+1}. After initial points are selected, LK tracker [10] is applied to find corresponding matches. Considering that homogeneous areas may be problematic for the LK tracker, we remove points of homogeneous areas based on the small eigenvalue of the structure tensor.

Later, (v_x, v_y) is estimated for each initial point (x, y). Combining coordinate (x, y) and motion vector (v_x, v_y), the resulted candidates appear as a cloud of (x, y, v_x, v_y) points in the 4D space. We represent each 4D point as a second order, symmetric, non-negative definite tensor T, which is equivalent to a 4×4 matrix and can be decomposed according to Eq. 1 when $N = 4$:

$$\begin{aligned} T &= \sum_{i=1}^{N} \lambda_i e_i e_i^T \\ &= \sum_{i=1}^{N} [(\lambda_i - \lambda_{i+1}) \sum_{k=1}^{i} e_k e_k^T] + \lambda_N \sum_{i=1}^{N} e_i e_i^T \end{aligned} \tag{1}$$

where λ_i denote the eigenvalues in descending order and e_i are the corresponding eigenvectors. Because the initial matches do not provide any orientation, we encode each tensor T in the 4D space as ball tensor, which is an identity matrix. Through the voting step, each tensor collects votes from its neighbors using Eq. 2,

$$\begin{aligned} R(A) &= \sum_{B_i \in N(A)} B_{vote}(A, B_i) \\ &= \sum_{B_i \in N(A)} e^{-(\frac{s^2}{\sigma^2})} \left(I - \frac{\vec{v}\vec{v}^T}{\|\vec{v}^T\vec{v}\|} \right) \end{aligned} \tag{2}$$

where A is a vote receiver point, B_i is a voter point in the neighborhood of A, $\vec{v} = \overrightarrow{B_i A}$, $s = |\vec{v}|$, σ is a parameter. $R(A)$ is also a second order, symmetric, non-negative definite tensor which can be decomposed using Eq. 1. We can get the 2D variety saliency $\lambda_2 - \lambda_3$ of A. If a point results from a wrong match, it is more like an isolated in the 4D space. Because the isolated point collects weak votes from its neighbors, it has small 2D variety saliency value. We eliminate isolated points with small saliency value. Thus, reliable points set $P = \{P_i, i = 1, \cdots, N\}$ is obtained with N points. Finally, the sparse depth $\hat{D}_{t+1}$ is derived at location of reliable points by shifted bilateral filtering [6].

2.2. Tensor-voting based Depth Interpolation

In this subsection, unknown depth value in D_{t+1} can be estimated from obtained reliable points set $P = \{P_i, i = 1, \cdots, N\}$ and the corresponding sparse depth $\hat{D}_{t+1}$. We encapsulate position (x, y), motion vector (v_x, v_y) and color information (r, g, b) to represent all points of D_{t+1} as $(x, y, v_x, v_y, r, g, b, d)$ in the 8D space. d is the depth value in pixel (x, y), and only depth value d_{P_i} in P are observed. We denote unknown points set as $Q = \{Q_i, i = 1, \cdots, M\}$. In order to estimate depth d_{Q_i}, we take the 8D space as input-output space, where input space is $(x, y, v_x, v_y, r, g, b)$ and output variable is d. Unobserved depth values d_Q can be inferred from d_P in sparse depth $\hat{D}_{t+1}$ by tensor-voting based interpolation.

Table 1. Tensor Interpretation in the 8D Space

Dim	Saliency	Normals	Tangents
0	λ_8	e_1,e_2,$\cdots$,e_7,e_8	none
1	$\lambda_7 - \lambda_8$	e_1,e_2,$\cdots$,e_7	e_8
$\vdots$	$\vdots$	$\vdots$	$\vdots$
5	$\lambda_3 - \lambda_4$	e_1,e_2,e_3	e_4,e_5,e_6,e_7,e_8
6	$\lambda_2 - \lambda_3$	e_1,e_2	e_3,e_4,$\cdots$,e_8
7	$\lambda_1 - \lambda_2$	e_1	e_2,e_3,$\cdots$,e_8

We assume that each point $P_i \in P$ is lying on a manifold. Tensor voting is used to extract local structures in this manifold. The local structure is characterized by normal and tangent vectors. The structure in the 8D space can be represented as parametric equations: $x = x, y = y, r = r, g = g, b = b, v_x = v_x(x, y), v_y = v_y(x, y), d = d(x, y, r, g, b)$. Because these equations are controlled by the five parameters (x, y, r, g, b). The local structure can be characterized by three normal vectors and five tangent vectors, as shown in Table 1. The five tangent vectors span a tangent space V_i of P_i, as Eq. 3.

$$V = span\{e_4, e_5, e_6, e_7, e_8\} \tag{3}$$

The local smoothness around P_i is maintained in the derived tangent space. Thus, we can interpolate a new point in the neighborhood of P_i.

Algorithm 1 is illustrated to infer d_Q. When d_Q has been inferred, depth of D_{t+1} is obtained. We denote P_t and Q_t as the projection of P and Q into the input space $(x, y, v_x, v_y, r, g, b)$, respectively. $P_{t,i}$ and $Q_{t,i}$ are the corresponding points of P_i and Q_i in input space. When estimating the depth of a point $Q_i \in Q$, we first find $Q_{t,i}$'s nearest neighbor point $P_{t,j} \in P_t$. Then, we can calculate the direction $\overrightarrow{P_{t,j}Q_{t,i}}$ in the input space and project it back into the 8D space. The selected $P_j \in P$ is taken as the starting point on the manifold. The desired direction $\vec{w}$ is the projection of the vector $\overrightarrow{P_{t,j}Q_{t,i}}$ on the tangent space V_j of P_j. Then, we take a small step along $\vec{w}$ towards Q_i to get $\hat{Q}$, according to $\hat{Q} = P_j + \tau\vec{w}$. The approximation stops when $\hat{Q}$ is within ε of Q_i. $\hat{Q}$ in the 8D space is the desired interpolated point for Q_i. Thus, the depth value of Q_i equals $d_{\hat{Q}}$. When the depth of all points in Q are interpolated, D_{t+1} is generated finally.

3. EXPERIMENTAL RESULTS

The proposed method is evaluated over ten video sequences, where eight test sequences are collected from the Philips WowVc© project, Sequence 9 "Interview" from Heinrich-Hertz-Institut, and "Inner-gate" from [7]. In order to verify the effectiveness of the proposed method, three popular depth estimation methods are compared, i.e., improved depth propagation (Varekamp et al.) [4], bi-direction motion estimation and compensation (Li et al.) [7], and motion compensation with trilateral filtering (Wang et al.) [9]. Several examples of the estimated depth maps are displayed in Fig. 2. It is clearly shown that the proposed method can obtain reliable depth estimation in occlusion boundary areas. For Sequence Philips-1, due to occlusion, existing methods [7, 9] misuse the depth of moving foreground object to estimate that of occluded background areas, while the proposed method has good performance.

Algorithm 1 Tensor Voting-based Depth Propagation

```
Task: Generate dense depth $D_{t+1}$.
Initialization: Input $P$, $Q$
for all $i \in [1, N]$ do
    Encode ball tensor $T_i$ as identity matrix $I$ for $P_i$
end for
Set $P_t$ as the projection of $P$ in input space
Set $Q_t$ as the projection of $Q$ in input space
Construct k-d trees of $P$ and $P_t$ for fast neighbor searching
2.Tangent Space Calculation by Voting
for all $i \in [1, N]$ do
    Compute ball voting $R(P_i)$ according to Eq.2
end for
for all $i \in [1, N]$ do
    Decompose $R_i$'s eigensystem according to Eq.1
    Calculate the tengent space $V_i$ of $P_i$
end for
3. Depth Estimation for $Q$
for all $i \in [1, M]$ do
    Find $P_{t,j}$ as the nearest neighbor of $Q_{t,i}$
    Project $\overrightarrow{Q_{t,i}P_{t,j}}$ into $V_j$ to get desirable direction $\vec{w}$
    $\hat{Q} \leftarrow P_j + \tau\vec{w}$
    while $\hat{Q}$ is not within $\epsilon$ of $Q_{t,i}$ do
        Set $\hat{Q}$ as a new start point
        Calculate $\hat{Q}$'s tengent space and get desirable direction $\vec{w}$
        $\hat{Q} \leftarrow \hat{Q} + \vec{w}$
    end while
    $d_{Q_i} \leftarrow d_{\hat{Q}}$
end for
4. Output:$D_{t+1}$
```

Furthermore, objective quality assessment is enabled to compare the proposed method with the aforementioned state-of-the-art methods. The original depth map of those sequences is taken as ground truth. The average mean squared error between original depth and propagated depth maps is calculated and listed in Table 2. It is easy to find that the proposed method outperforms other methods in all sequences. Besides, the relevant Structure Similarity (SSIM) index is utilized to evaluate the difference in structural information, which is demonstrated in Table 3. Obviously, the proposed method achieves the best performance because it preserves the depth structure with higher fidelity.

4. CONCLUSIONS

Depth propagation is a major part of depth estimation, which plays a critical role in semi-automatic 2D-to-3D video conversion. This paper presents a robust framework where tensor voting is utilized to eliminate occlusion points, thereby deriving accurate sparse depth. Then, we incorporate a variety

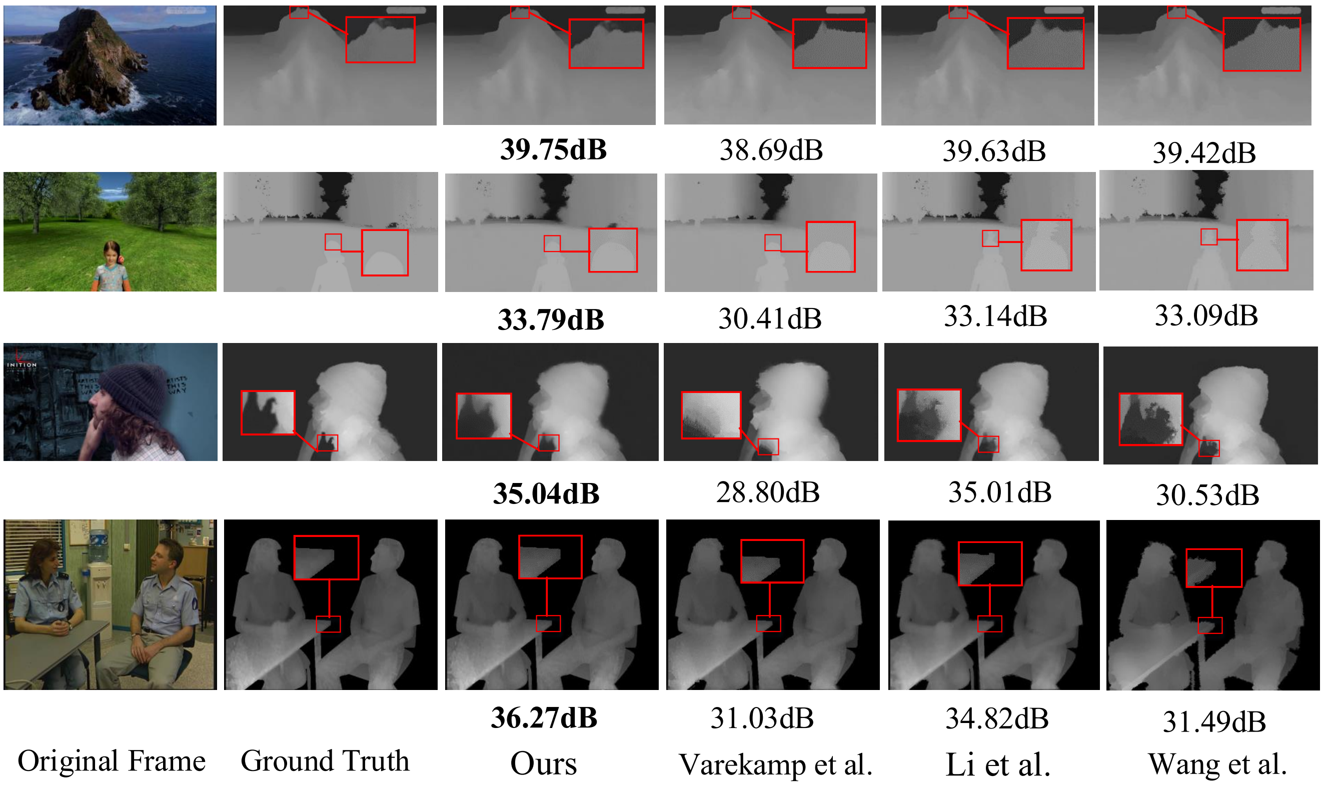

Fig. 2. Estimated depth maps of four testing sequences (from top to bottom): Inition-1, Philips-1, HeadRotate and Interview. The PSNR values of depth maps are shown, respectively.

Table 2. Average Mean Squared Error

	Vare. [4]	Li [7]	Wang [9]	Ours
Inition-1	40.91	16.89	32.68	**11.28**
Inition-2	7.55	5.51	6.32	**3.97**
Philips-1	94.83	41.98	32.68	**25.12**
Philips-2	548.9	190.7	388.2	**163.2**
Dice-1	124.7	86.97	113.8	**52.79**
Dice-2	70.01	69.25	79.88	**37.92**
HeadRotate	79.78	19.27	57.99	**16.20**
Building	360.4	105.8	192.3	**69.28**
Interview	98.73	45.03	68.32	**30.13**
Inner-gate	529.9	156.4	195.3	**98.55**

Table 3. Structure Similarity Comparison Results

	Vare. [4]	Li [7]	Wang [9]	Ours
Inition-1	0.971	0.979	0.973	**0.980**
Inition-2	0.985	0.981	0.983	**0.991**
Philips-1	0.971	0.976	0.975	**0.980**
Philips-2	0.928	0.935	0.935	**0.967**
Dice-1	0.988	0.985	0.982	**0.989**
Dice-2	0.990	0.987	**0.990**	**0.990**
HeadRotate	0.976	0.987	0.979	**0.990**
Building	0.875	0.922	0.912	**0.932**
Interview	0.963	0.979	0.979	**0.984**
Inner-gate	0.900	0.930	0.917	**0.949**

of features to represent points in sparse depth as high dimensional tensor. A dense depth is formed via tensor-voting based interpolation. It is robust against object occlusion and inaccurate motion estimation, and could capture the structures.

5. REFERENCES

[1] J. Ko, M. Kim, and C. Kim, "2D-to-3D stereoscopic conversion: depth-map estimation in a 2D single-view image" in *Proc. SPIE Applications of Digital Image Processing*, San Diego, CA, USA, Sept. 2007, pp. 1-9.

[2] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, San Jose, CA, USA, May 2004, pp. 93-104.

[3] M. Kim, S. Park, H. Kim, and I. Artem, "Automatic conversion of two-dimensional video into stereoscopic

video," in *Proc. SPIE Three-Dimensional TV, Video, and Display*, Boston, MA, USA, Nov. 2005, pp. 601-610.

[4] C. Varekamp and B. Barenbrug, "Improved depth propagation for 2D-to-3D video conversion using keyframes," in *Proc. IET 4th European Conf. Visual Media Production (CVMP'07)*, London, UK, Nov. 2007, pp. 1-7.

[5] M. Guttmann, L. Wolf, and D. Cohen-Or, "Semi-automatic stereo extraction from video footage," in *Proc. Int'l Conf. Computer Vision (ICCV'09)*, Kyoto, Japan, Sept. 2009, pp. 136-142.

[6] X. Cao, Z. Li, and Q. Dai, "Semi-automatic 2D-to-3D conversion using disparity propagation," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 491-499, Jun. 2011.

[7] Z. Li, X. Cao, and Q. Dai, "A novel method for 2D-to-3D video conversion using bi-directional motion estimation," in *Proc. Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP'12)*, Kyoto, Japan, Mar. 2012, pp. 1429-1432.

[8] P. Mordohai, and G. Medioni, "Dimensionality Estimation, Manifold Learning and Function Approximation Using Tensor Voting," in *J. Mach. Learn. Res.*, vol.11, no.1, pp. 411-450, Mar. 2010.

[9] L. Wang, C. Jung, "Example-based video stereolization with foreground segmentation and depth propagation," *IEEE Trans. Multimedia*, vol.16, no.7, pp. 1905-1914, Nov. 2014.

[10] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," technical report, Intel Microprocessor Research Labs, 1999.