J Sign Process Syst (2014) 76:33–46
DOI 10.1007/s11265-013-0805-8

Depth Estimation and Video Synthesis for 2D to 3D Video Conversion

Chien-Chih Han · Hsu-Feng Hsiao

Received: 29 March 2013 / Revised: 7 June 2013 / Accepted: 7 June 2013 / Published online: 5 July 2013

Abstract With the recent progress of multi-view devices and the corresponding signal processing techniques, stereoscopic viewing experience has been introduced to the public with growing interest. To create depth perception in human vision, two different video sequences in binocular vision are required for viewers. Those videos can be either captured by 3D-enabled cameras or synthesized as needed. The primary contribution of this paper is to establish two transformation models for stationary scenes and non-stationary objects in a given view, respectively. The models can be used for the production of corresponding stereoscopic videos as a viewer would have seen at the original event of the scene. The transformation model to estimate the depth information for stationary scenes is based on the information of the vanishing point and vanishing lines of the given video. The transformation model for non-stationary regions is the result of combining the motion analysis of the non-stationary regions and the transformation model for stationary scenes to estimate the depth information. The performance of the models is evaluated using subjective 3D video quality evaluation and objective quality evaluation on the synthesized views. Performance comparison with the ground truth and a famous multi-view video synthesis algorithm, VSRS, which requires six views to complete synthesis, is also presented. It is shown that the proposed method can provide better perceptual 3D video quality with natural depth perception.

Keywords View synthesis · 2D to 3D video conversion · Vanishing point · Motion analysis

C.-C. Han · H.-F. Hsiao (✉)
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
e-mail: hillhsiao@cs.nctu.edu.tw

1 Introduction

The depth perception of stereoscopic vision can be created by feeding each eye with two different video sequences or images in binocular vision with proper parallax. Unless the video scenes are captured with a pair of synchronized cameras that are separated and directed to the scenes properly, view synthesis is usually performed to generate two views for stereoscopic vision.

The approaches of view synthesis could be roughly classified in terms of the types of video sources, including multi-view videos and traditional monocular videos. Stereoscopic view synthesis from multi-view videos has attracted much attention during the standard development recently. Many of the view interpolation algorithms assume that video sequences are captured with aligned cameras. The depth map for video synthesis is created through estimating the disparity vector map [1–3]. In [1], the pixel-based disparity vectors are estimated using stereo matching and a graph cut algorithm. The cameras are assumed to be lined up at regular separation in horizontal direction as shown in Fig. 1 where *NL* and *NR* represent the original views used to synthesize the virtual views *OL* and *OR*. *D_NL* and *D_NR* are the corresponding depth maps generated with the assistance of their left and right views.

The general criterion of stereo matching is to search for the disparity value minimizing a cost function between one view and its neighboring view. With the depth map, the intermediate view can be synthesized accordingly.

Monocular videos such as most DVD titles are more popular, but the information of monocular videos is much less in comparison with the information of multi-view videos. Compared with the multi-view based depth map estimation, depth map estimation from monocular videos presents greater challenge. In [4], it is assumed that the camera motion contains translational motion, and the scene is stationary. With such strong assumption, the camera motion can be tracked and the virtual view can be warped using motion parallax without recovering the depth map. The camera motion is tracked first and an optimization algorithm is utilized to determine the base frame. The optimization algorithm is designed to have three properties: the realistic of stereoscopic effects after warping, the similarity between the warped views and original ones, and the temporal smoothness. Relative parallax is adopted in [4] instead of absolute parallax, and the warping error is decreased by minimizing the displacement of viewpoints.

In the case of immobile camera setting, the approach to estimate the map of disparity vectors between the given video and the synthesized one usually relies on the detection of moving objects. In [5], the moving objects are segmented using a motion/edge registration technique to avoid jitter of motion error which is a common problem in motion segmentation. The identified moving objects are classified as the nearest to the camera in order to create the rough depth map. For stationary objects in the scene, the work in [6] is based on a few heuristics to not only retrieve vanishing lines and vanishing points, but also generate the depth map according to the gradient of vanishing lines. The work in [7] is also utilized to assign the sky as the farthest location.

In this paper, two transformation models are proposed for stationary scenes and non-stationary objects, respectively. The design of the transformation model for stationary scenes was inspired by the concept in [6]. The developed model is derived analytically and it incorporates the information of vanishing point and vanishing lines in the estimation of depth for each pixel in a given monocular video. For non-stationary objects, motion analysis is performed first, and the model for stationary scenes is modified for the non-stationary objects such that the synthesized virtual view can be more realistic and natural.

The main objective of synthesizing views using the proposed methods is with intent to produce stereoscopic perception as a reviewer would have seen at the original event of the scene, instead of magnifying the depth perception. However, if the enhancement or magnification of the depth perception is desirable, it can be accomplished easily by warping the depth information resulted from the transformation models. Without loss of generality, the camera is assumed to be stationary in this paper. In practice, if the camera is not stationary, many global motion compensation algorithms, such as the one in [8], can be incorporated first so that the processed video can be regarded as captured by a stationary camera. In addition, the transformation model for stationary scenes is not affected even if the global motion is not compensated well.

The remainder of this paper is organized as follows. Section 2 presents the transformation models for stationary scenes and for non-stationary objects. The 2D to 3D view synthesis procedure which takes advantage of the developed models is also described in Section 2, and the 2D to 3D view synthesis system is developed to demonstrate the usefulness of the models in Section 3. The experimental results based on perceptual 3D video quality evaluation and objective quality evaluation are shown in Section 4. Finally, the conclusion remarks are shown in Section 5.

2 Transformation Models and View Synthesis

In this paper, two transformation models are proposed. The first model is to estimate pixel-based depth map for stationary scenes through coordinate transformation. The second model is to estimate depth information for non-stationary objects. With the obtained depth information, the desired view for 3D vision can then be synthesized after calculating corresponding disparity between two views.

2.1 Transformation Model for Stationary Scenes

The vanishing point and lines can assist the conversion of monocular videos to stereoscopic videos as also discussed in [9]. Furthermore, the main vanishing lines can be used to separate the vertical plane from the horizontal plane in a video scene.

The image coordinate system (x, y) and the world coordinate system (w, h, d) are illustrated in Fig. 2 where C_O: $(0, h_C, 0)$ and C_S: $(w_{cs}, h_{cs}, 0)$ are the positions of the original camera and the virtual camera, respectively. The depth value of a point in the world coordinate system is defined as its d component of the three-dimensional coordinates. The X axis of the image coordinate system is assumed to be parallel to the Waxis of the world coordinate system and those two axes lie on the same plane. The objective of the transformation model for stationary scenes is to find the d component, which is the depth information, for a pixel located at (x, y).

To find the corresponding point in the world coordinate system according to the projected location on the imaging plane, the projected points of the background on the imaging plane in a given video can be categorized as either one of the two groups: the points on the horizontal plane and the points on the vertical plane.

The points on the horizontal plane are considered first. For two points in the world coordinate system, O_1: (w_{O1}, h_{O1}, d_{O1}) and O_2: (w_{O2}, h_{O2}, d_{O2}) where h_{O1} equals h_{O2}, the projected points of the points O_1 and O_2 on the imaging plane are P_1: (w_{P1}, h_{P1}, d_{P1}) and P_2: (w_{P2}, h_{P2}, d_{P2}), respectively. An imaging plane is also referred as a captured video frame.

Since O_1 is on the line $\overleftrightarrow{C_OP_1}$ and O_2 is on the line $\overleftrightarrow{C_OP_2}$, O_1 and O_2 can also be represented as $(w_{P1}t, (h_{P1}\text{-}h_C)t, d_{P1}t)$ and $(w_{P2}s, (h_{P2}\text{-} h_C)s, d_{P2}s)$, respectively, where t and s are constants. It is obvious that t will be equal to s if h_{P1} is equal to h_{P2} but not equal to h_C. Consequently, since d_{P1} equals d_{P2}, d_{O1} shall equal d_{O2}. It means that the depth values of the two points which are on the same horizontal plane (i.e., the H components of all points on the plane are the same) will be the same if their projected points on the imaging plane have the same H components in the world coordinate system.

The transformation model for pixels on the horizontal plane is described below, followed by the derivation. The transformation algorithm for pixels on the vertical plane is derived thereafter.

2.1.1 Transformation Model for Stationary Scene on the Horizontal Plane

Given a point (x, y) on the horizontal plane in the image coordinate system projected from a point (w, h, d) in the world coordinate, the W component w and the D component d can be determined as follows.

$$d = \frac{h_C\left(f^2 - \left(y_{vanish} - \frac{H_{frame}}{2}\right)\left(\frac{H_{frame}}{2} - y\right)\right)}{f\left(y_{vanish} - y\right)}, \tag{1}$$

$$w = \left(x - \frac{W_{frame}}{2}\right)\sqrt{\frac{d^2 + h_C{}^2}{f^2 + \left(y - \frac{H_{frame}}{2}\right)^2}}, \tag{2}$$

where h_C is the height from the horizontal plane to the camera C_O in world coordinate as shown in Eq. (3):

$$h_C = \frac{L{\cdot}f{\cdot}y_{vanish}}{f^2 - \frac{H_{frame}}{2}\left(y_{vanish} - \frac{H_{frame}}{2}\right)}. \tag{3}$$

y_{vanish} in the equations above is the Y component of the vanishing point on the imaging plane. The focal length of the camera C_O is f; L is the distance between the camera and the bottom of the captured video frames in the world coordinate; the video resolution is W_{frame} by H_{frame}.

The derivations of Eqs. (1), (2), and (3) are as follows:

Because the vanishing point can be regarded as the farthest point projected on the imaging plane, the line from the camera to the vanishing point is parallel to the horizontal plane in the world coordinate. If the camera is aimed at the farthest point, the vanishing point will be located at the middle of the imaging plane, as shown in the case of Fig. 3a. Otherwise, the vanishing point will be either below or above the middle of the imaging plane as shown in Fig. 3b and c. In those figures, V is the vanishing point of the given monocular video, M and B are the middle and the bottom points of the monocular video, respectively. The dotted segment between the green lines represents the transection of the monocular video.

Derivation of Eq. (1):

Without loss of generality, we first take the case where the camera is aimed at the lower horizontal plane as example. Suppose that the point P is on the vertical line in the central of the monocular video as shown in Fig. 4a. For any point P' which has the same Y component as P, those two points will appear as a single point marked as P in the figure on the transection of the monocular video. Alternatively, the relation of points is shown in Fig. 5. The corresponding points of P and P' on the horizontal plane are J and J', respectively. As long as P' has the same Y component as P, its corresponding point J' on the horizontal plane will have the same depth $\overline{SJ}$ as the point J.

As shown in Fig. 4a, $\angle OJS$ is equal to $\angle JOV$, where $\angle JOV$ can be divided into $\angle VOM$ and $\angle POM$. In the following derivation, the length of $\overline{SO}$, the height from the horizontal plane to the camera, is denoted as h_c. $\angle VMO$ and $\angle PMO$ are both right angles, and $\overline{OM}$ is the focal length f. The length d of $\overline{SJ}$ can be determined as shown in (4).

$$\begin{aligned}
&\tan(\angle OJS) = \tan(\angle JOV) = \tan(\angle VOM + \angle POM)\\
&\Rightarrow \frac{h_C}{d} = \frac{\sin(\angle VOM + \angle POM)}{\cos(\angle VOM + \angle POM)}\\
&\Rightarrow \frac{h_C}{d} = \frac{\sin(\angle VOM)\cos(\angle POM) + \cos(\angle VOM)\sin(\angle POM)}{\cos(\angle VOM)\cos(\angle POM) - \sin(\angle VOM)\sin(\angle POM)}\\
&\Rightarrow \frac{h_C}{d} = \frac{\overline{MV}*\overline{OM} + \overline{OM}*\overline{PM}}{\overline{OM}^2 - \overline{MV}*\overline{PM}} = \frac{\overline{MV}*f + f*\overline{PM}}{f^2 - \overline{MV}*\overline{PM}}\\
&\Rightarrow d = \frac{h_C\left(f^2 - \overline{MV}*\overline{PM}\right)}{f\left(\overline{PM} + \overline{MV}\right)} = \frac{h_C\left(f^2 - \overline{MV}*\overline{PM}\right)}{f * \overline{PV}}\\
&\Rightarrow d = \frac{h_C\left(f^2 - \left(y_{vanish} - \frac{H_{frame}}{2}\right)\left(\frac{H_{frame}}{2} - y\right)\right)}{f\left(y_{vanish} - y\right)}.
\end{aligned} \tag{4}$$

Since the corresponding point J' on the horizontal plane for any point P' with the same Y component as P on the monocular video will have the same depth d as calculated in Eq. (4), the derivation of Eq. (1) is completed for the case shown in Fig. 4a.

On the other hand, if point P is located between V and M as shown in Fig. 4b, the derivation of Eq. (1) is similar to the case in Fig. 4a except that $\overline{PM}$ is positive in Fig. 4a but it is regarded as negative in Fig. 4b. Also, $\angle OJS$ is equal to $\angle VOM$ minus $\angle POM$.

If the camera is aimed at the upper horizontal plane, the derivation is also similar to the procedure above. In either Fig. 4a or Fig. 4b, the length of $\overline{MV}$ is positive but it is regarded as negative in Fig. 4c. Also, $\angle OJS$ is equal to $\angle POM$ minus $\angle VOM$.

Derivation of Eq. (2):

The width w of J' in the world coordinate can be simply estimated using the theorem about similar

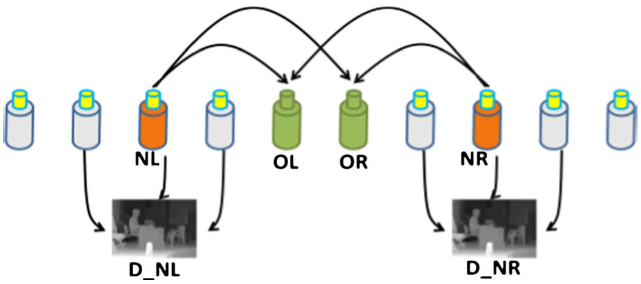


Figure 1 Camera configuration for depth map generation and view synthesis with multi-view videos.

triangles. According to the similar triangles in Fig. 5, $\overline{JJ'}$ can be determined if the values of $\overline{OP}$, $\overline{OJ}$, and $\overline{PP'}$ are available.

From Figs. 4(a) to 5, the lengths of $\overline{OP}$ and $\overline{OJ}$ can be calculated using Eqs. (5) and (6), respectively. The length of $\overline{PP'}$ is shown in Eq. (7).

$$\overline{OP} = \sqrt{\overline{OM}^2 + \overline{MP}^2} = \sqrt{f^2 + \left(y - \frac{H_{frame}}{2}\right)^2}. \tag{5}$$

$$\overline{OJ} = \sqrt{\overline{OS}^2 + \overline{SJ}^2} = \sqrt{h_C{}^2 + d^2}. \tag{6}$$

$$\overline{PP'} = x - \frac{W_{frame}}{2}. \tag{7}$$

Then, $\overline{JJ'}$ can be derived as shown in Eq. (8).

$$w = \overline{JJ'} = \overline{PP'}\frac{\overline{OJ}}{\overline{OP}} = \left(x - \frac{W_{frame}}{2}\right)\frac{\sqrt{d^2 + h_C{}^2}}{\sqrt{f^2 + \left(y - \frac{H_{frame}}{2}\right)^2}}. \tag{8}$$

This completes the derivation of Eq. (2).

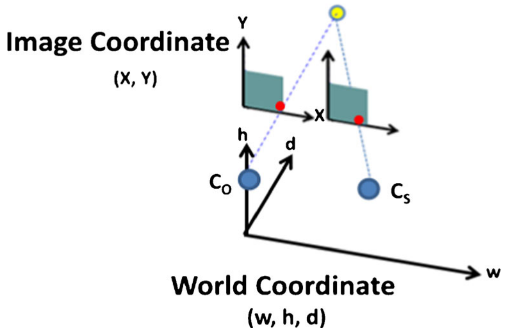


Figure 2 The image coordinate system and the world coordinate system.

Derivation of Eq. (3):

If the point P is placed at the location of point B, d is equal to L. In this case, h_C can be obtained using Eq. (4).

2.1.2 Transformation Model for Stationary Scene on the Vertical Plane

For a projected point P_v of the vertical plane, the first step to find the depth d and width w of the corresponding point in the real scene is searching for the corresponding *foothold* on the horizontal plane. The corresponding foothold is defined as the point on the horizontal plane in the world coordinate with the same depth and the width as the point P_v. The depth and width of the corresponding foothold on the horizontal plane can then be derived using Eqs. (1) and (2).

In the world coordinate system, the imaging plane can be expressed as:

$$a\boldsymbol{h} + b\boldsymbol{d} = c, \tag{9}$$

where a, b, and c are constants. In Fig. 6, it is assumed that there are two points p_1: (W_{p1}, H_{p1}, D_{p1}) and f_1: (W_{f1}, H_{f1}, D_{f1}) in the world coordinate, where H_{p1} is not equal to H_{f1} but W_{p1} equals W_{f1} and D_{p1} equals D_{f1}. p_1 is the corresponding point in the world coordinate that projects to the point (x, y) in the image coordinate. The projected points on the screen plane are p_2: $(W_{p1}t$, $(H_{p1}$-$h_C)t$, $D_{p1}t)$ and f_2: $(W_{f1}s$, $(H_{f1}$-$h_C)s$, $D_{f1}s)$, respectively, where s and t are constants. Since p_2 and f_2 are on the imaging plane, Eq. (10) can be obtained.

$$\left[aH_{p1} - ah_C + bD_{p1}\right]t = \left[aH_{f1} - ah_C + bD_{f1}\right]s = c. \tag{10}$$

From Eq. (10) and the relation between points p_1 and f_1, the widths ($W_{p1}t$ and $W_{f1}s$) on the imaging plane of these points will be the same only if either a is zero or W_{p1} (=W_{f1}) is zero.

In other words, it means that the projected points of p_1 and f_1 on the imaging plane will have the same width in the world coordinate (or the same value of x-axis in the image coordinate) when either the projected point p_2 is located on the central line expressed in (11):

$$a\boldsymbol{h} + b\boldsymbol{d} = c, \boldsymbol{w} = 0 \tag{11}$$

, or the camera is aimed at the vanishing point as shown in Fig. 3a. At these cases, the corresponding foothold of the point p_2 projected on the imaging plane is the intersection of the following two lines. The first line ($a\boldsymbol{h}$+$b\boldsymbol{d}$=c, $\boldsymbol{w}$=$W_{p1}t$) has the same width with the corresponding point. The second line is the vanishing line which separates vertical and horizontal planes.

For the case where both W_{p1} and a are not zeros, the projected foothold of the point p_2 on the imaging plane is the intersection of two lines: the line $\overleftrightarrow{p_2 f_2}$ on the imaging plane and the vanishing line which separates vertical and horizontal planes.

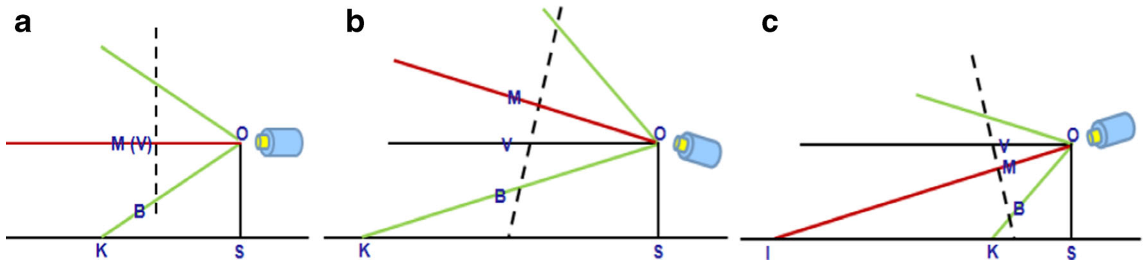


Figure 3 The related positions between vanishing point and the middle of the monocular video when the camera is aimed at different locations.

In order to find the location of point f_2, the length $\overline{f_3f_2}$ is determined first. It is assumed that there are four points, p_3: $(0, (H_{p1}\text{-}h_C)\, t, D_{p1}t)$, f_3: $(0, (H_{f1}\text{-}h_C)\, s, D_{f1}s)$, p_4: $(0, (H_{p1}\text{-}h_C)\, t, 0)$, and f_4: $(0, (H_{f1}\text{-}h_C)\, s, 0)$ where p_3 and f_3 are on the central line expressed in (11), as shown in Fig. 6.

Since W_{p1} equals W_{f1} and D_{p1} equals D_{f1}, $\cos(\angle p_3p_4p_2)$ and $\cos(\angle f_3f_4f_2)$ shown in (12) and (13) are equal as well. In addition to the fact that $\angle p_2p_3p_4$ and $\angle f_2f_3f_4$ are both right angles, $\Delta p_2p_3p_4$ and $\Delta f_2f_3f_4$ are similar triangles. Therefore, $\overline{f_3f_2}$ can be derived when $\overline{p_3p_2}$, $\overline{p_4p_3}$, and $\overline{f_4f_3}$ are known.

$$\cos(\angle p_3p_4p_2) = \frac{\overrightarrow{p_4p_3}\bullet\overrightarrow{p_4p_2}}{\left\|\overrightarrow{p_4p_3}\right\| \times \left\|\overrightarrow{p_4p_2}\right\|} = \frac{(0,0,D_{p1}t)\bullet(W_{p1}t,0,D_{p1}t)}{(D_{p1}t)\sqrt{(W_{p1}t)^2+(D_{p1}t)^2}} = \frac{(D_{p1}t)^2}{D_{p1}t^2\sqrt{W_{p1}{}^2+D_{p1}{}^2}} = \frac{D_{p1}}{\sqrt{W_{p1}{}^2+D_{p1}{}^2}}. \tag{12}$$

$$\cos(\angle f_3f_4f_2) = \frac{\overrightarrow{f_4f_3}\bullet\overrightarrow{f_4f_2}}{\left\|\overrightarrow{f_4f_3}\right\| \times \left\|\overrightarrow{f_4f_2}\right\|} = \frac{(0,0,D_{f1}s)\bullet(W_{f1}s,0,D_{f1}s)}{(D_{f1}s)\sqrt{(W_{f1}s)^2+(D_{f1}s)^2}} = \frac{(D_{f1}s)^2}{D_{f1}s^2\sqrt{W_{f1}{}^2+D_{f1}{}^2}} = \frac{D_{f1}}{\sqrt{W_{f1}{}^2+D_{f1}{}^2}}. \tag{13}$$

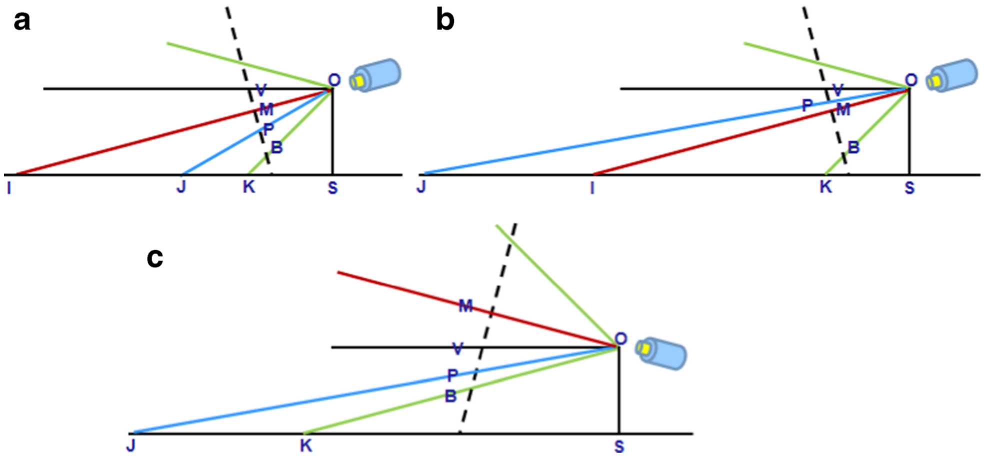
The • operators shown in (12) and (13) are inner products.

Regardless of the location of point p_3 shown in Fig. 7, the search of the point f_2 is the same. $\overline{PC}$ is vertical to $\overline{OC}$; $\overline{OM}$ is vertical to the imaging plane which is represented by the dash line in Fig. 7, and $\overline{OM}$ is equal to f as shown in Fig. 7. Also, $\overline{p_3p_2}$ and $\overline{p_4p_3}$ can be determined as the following.

$$\overline{p_3p_2} = x - \frac{W_{frame}}{2} \tag{14}$$

$$\begin{aligned}\overline{p_4p_3} &= D_{p1}t = \overline{OC} = \overline{OV} + \overline{VC}\\ &= \overline{OV} + \overline{Vp_3}\sin(\angle Cp_3V)\end{aligned} \tag{15}$$

Figure 4 Illustration for the derivation of *d* and *h* with the point *P* located at different locations.



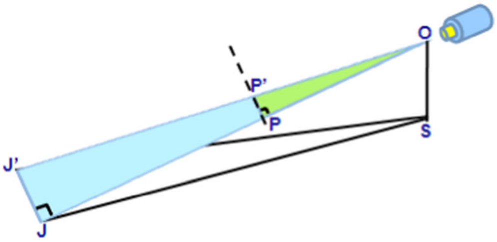


Figure 5 Another viewpoint of Fig. 4a for illustration.

Since ΔCp_3V is similar to ΔMOV, $\angle Cp_3V$ is equal to $\angle MOV$. Therefore, $\overline{p_4p_3}$ can be calculated in (16).

$$\overline{p_4p_3} = \overline{OV} + \overline{Vp_3}\,\sin(\angle MOV) = \overline{OV} + \overline{Vp_3}\,\frac{\overline{MV}}{\overline{OV}} = \sqrt{f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2} + (y-y_{vanish})\frac{\left(\frac{H_{frame}}{2}-y_{vanish}\right)}{\sqrt{f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2}} \tag{16}$$

Either the point V or M can be regarded as point f_3 when calculating $\overline{f_4f_3}$. If the point V doubles as f_3, $\overline{f_3f_2}$ can be derived as shown in Eq. (17).

$$\begin{aligned}
\overline{f_4f_3}|_{V=f_3} &= \overline{OV} = \sqrt{f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2}, \\
\overline{f_3f_2}|_{V=f_3} &= \overline{f_4f_3}\,\frac{\overline{p_3p_2}}{\overline{p_4p_3}} = \frac{\left[f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2\right]\left(x-\frac{W_{frame}}{2}\right)}{\left[f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2\right]+(y-y_{vanish})\left(\frac{H_{frame}}{2}-y_{vanish}\right)}.
\end{aligned} \tag{17}$$

On the other hand, $\overline{f_3f_2}$ is shown in Eq. (18) when $M=f_3$.

$$\begin{aligned}
\overline{f_4f_3}|_{M=f_3} &= \overline{OM}\cos(\angle MOV) = \frac{f^2}{\sqrt{f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2}}, \\
\overline{f_3f_2}|_{M=f_3} &= \overline{f_4f_3}\,\frac{\overline{p_3p_2}}{\overline{p_4p_3}} = \frac{f^2\left(x-\frac{W_{frame}}{2}\right)}{\left[f^2+\left(\frac{H_{frame}}{2}-y_{vanish}\right)^2\right]+(y-y_{vanish})\left(\frac{H_{frame}}{2}-y_{vanish}\right)}.
\end{aligned} \tag{18}$$

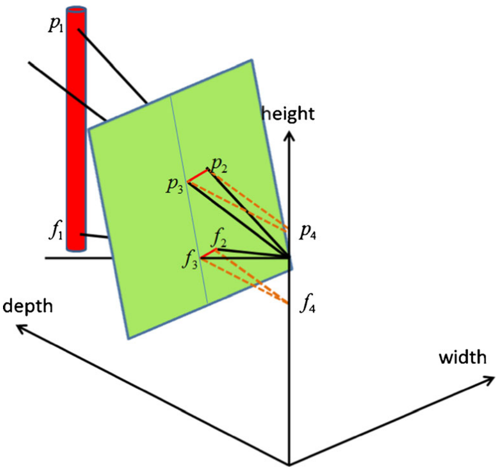


Figure 6 Illustration for the relation between $\overline{p_3p_2}$ and $\overline{f_3f_2}$.

And then, the required line p_2f_2 can be found when we link p_2 and f_2 on the imaging plane. The projection of the corresponding foothold of the point p_2 on the imaging plane is the intersection of two lines, the line p_2f_2 and the vanishing line which separates vertical and horizontal planes. The two-dimensional image coordinate of the foothold projection on the imaging plane can then be used to find the depth and width of the corresponding foothold in the world coordinate using Eqs. (1) and (2). Therefore, the width w and the depth d of the point p_1 in the world coordinate which is projected as the point (x, y) on the vertical plane in the image coordinate are obtained.

2.2 Transformation Model for Non-stationary Objects

Compared to the view synthesis in multi-view videos, there is much less information for the non-stationary

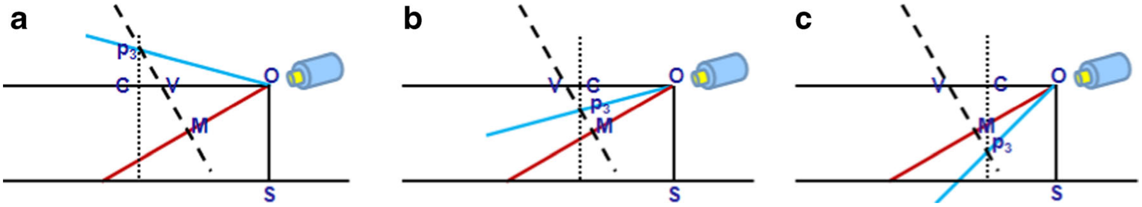


Figure 7 The cross-sectional view on plane $\boldsymbol{w}$=0 from Fig. 6, when **a** the point p_3 is above the vanishing point, **b** p_3 is below the vanishing point but above the middle of image, **c** p_3 is below the middle of image.

objects in a monocular video and the inaccuracy of the synthesized results is usually expectable. In [5], the moving objects are regarded as the nearest part to the camera and their depth values are assumed to be a constant, which is a highly simplified assumption. In this section, the focus is to obtain more reasonable depth values for the non-stationary objects.

In the proposed model for non-stationary objects, the initial depth value for each object is determined first and the value is further adjusted with the assistance of the transformation model described in Section 2.1. For the initial depth value of moving objects, the following observation is utilized. If an object is closer to the camera, the magnitude of its projected motion will be larger. On the other hand, the magnitude of its projected motion will be smaller if the object is farther. In [10], it is also mentioned that the retinal images of objects closer to the eye are displaced more quickly than the retinal images of more distant objects. In order to improve the accuracy of motion estimation, only the boundaries of moving objects are considered. The initial depth value shown in (19) is calculated using linear interpolation and non-linear warping based on the magnitude $\|mv\|$ of the motion vector mv.

$$d_{init} = \min\left\{f_{d-mv} = \left(f_{NL}(mv)\cdot f_{L-Depth}(mv)\right), d_{bottom}\right\}, \tag{19}$$

where function $f_{L\text{-}Depth}$ determines the depth value linearly according to the motion vector, and function f_{NL} warps the estimated depth value nonlinearly. The functions $f_{L\text{-}Depth}$ and f_{NL} are shown in (20) and (21).

$$f_{L-Depth}(mv) = \left(\frac{\|mv_{\max}\|-\|mv\|}{\|mv_{\max}\|}\right)\cdot d_{farthest}, \tag{20}$$

$$f_{NL}(mv) = \left(\frac{\|mv\|}{\|mv_{\max}\|}\right)^m, \tag{21}$$

where $d_{farthest}$ is the depth value of the farthest point on the horizontal plane of the given video. The exponent m in (21) is the nonlinear factor, which is 2 in this paper.

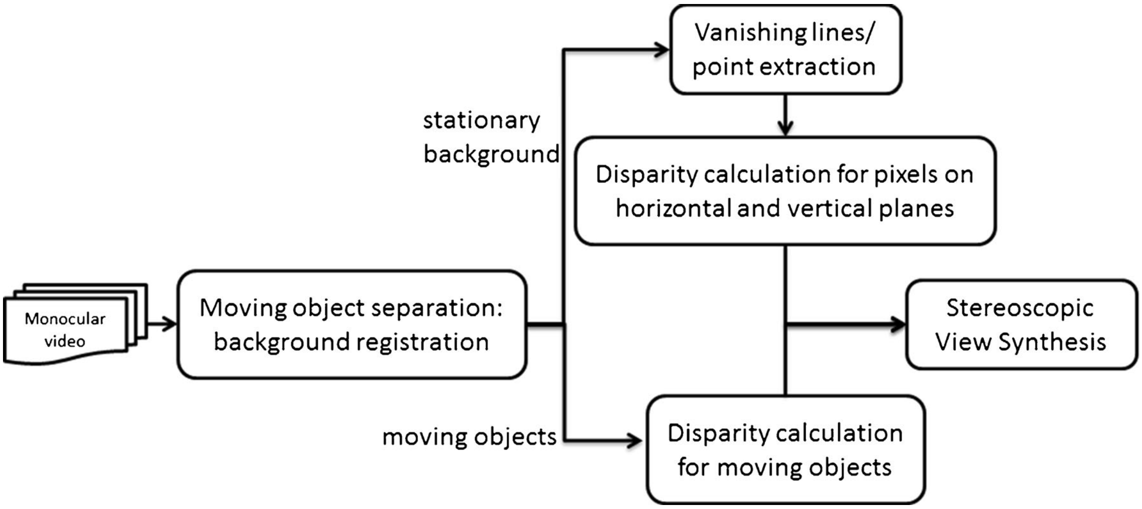

Vanishing lines/ point extraction
stationary background
Disparity calculation for pixels on horizontal and vertical planes
Monocular video
Moving object separation: background registration
Stereoscopic View Synthesis
moving objects
Disparity calculation for moving objects

Figure 8 Block diagram of the 2D to 3D view synthesis system.

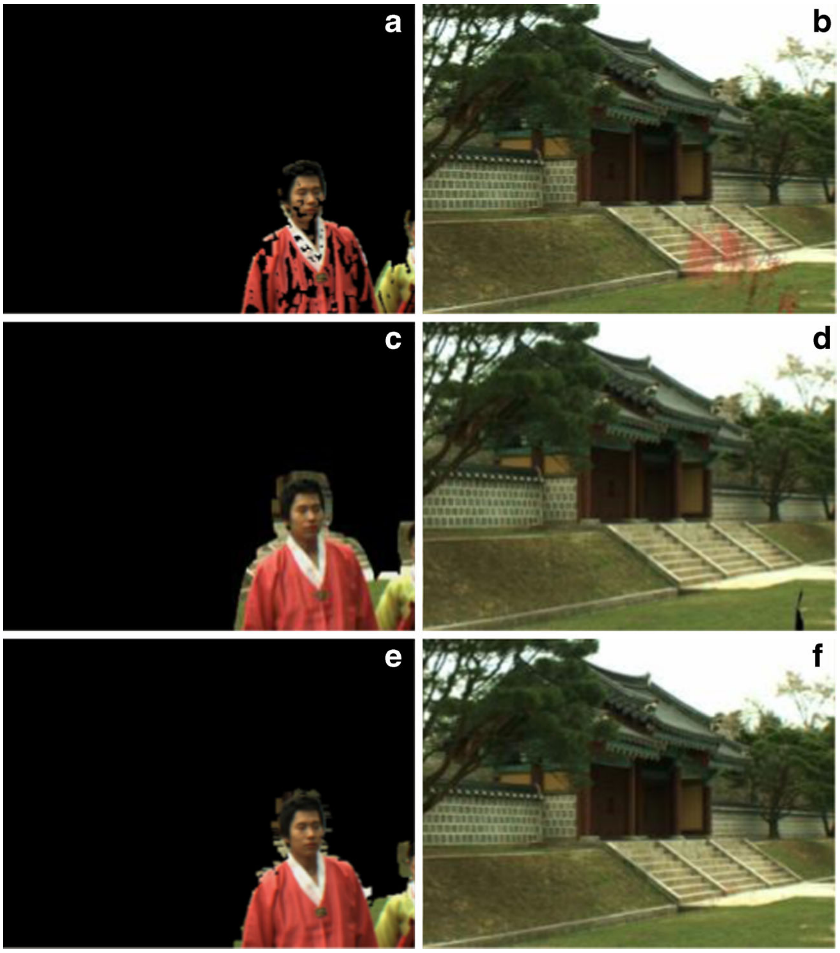


Figure 9 Frame difference of video *lovebird1* (camera 1) with different operations: **a** frame difference only, **c** frame difference with unrestricted dilation, **e** frame difference with restricted dilation. **b**, **d**, and **f** are the results of background registration, respectively.

However, the initial depth value $f_{d\text{-}mv}$ shown in (19) can be estimated in error. To alleviate the error, the location of the connected point of the background with the bottom of a moving object is determined in order to find out the corresponding depth d_{bottom} using the derivation in Section 2.1. If the depth value is less than d_{bottom}, there is a higher probability for the depth estimation to be incorrect. In this case, d_{bottom} is considered as the depth of the corresponding moving object instead.

After the initial depth value is determined for an object which appears for the first time, the depth of this object in the later frames is updated in sequence as follows.

After the initial depth value is obtained, the corresponding foothold $P_{fh}(x, y)$ of the object on the horizontal plane can be calculated. Note that the foothold of a point P_v is defined as the point on the horizontal plane in the world coordinate with the same depth and the width as the point P_v. Therefore, the y value of the point P_{fh} can be derived using Eq. (1). The foothold $P'_{fh}(x', y')$ of the same object in the next frame can then be determined with the motion vector of this moving object. The depth of the relocated foothold, which is also the depth of the moving object, can then be obtained using the same model in (1).

2.3 Disparity Calculation for View Synthesis

With the depth information calculated using the models introduced in Section 2.1 and 2.2, the desired view V_R can then be synthesized without much effort.

The locations of the original camera C_O and the virtual camera C_S are illustrated in Fig. 2. Since the disparity between the two views only involves horizontal difference, the Y components in the image coordinate system for a pair of corresponding pixels on the two views should be the same. There are two scenarios when calculating the disparity: (1) for pixels on the horizontal plane of the background part, and (2) for pixels on the vertical plane or for pixels of the non-stationary objects.

For each pixel (x_h, y_h) on the horizontal plane of the background part in view V_L, the w and d components in the world coordinate are first calculated using (1) and (2). With the values of w and d, the corresponding X component x_r on the synthesized view V_R in the image

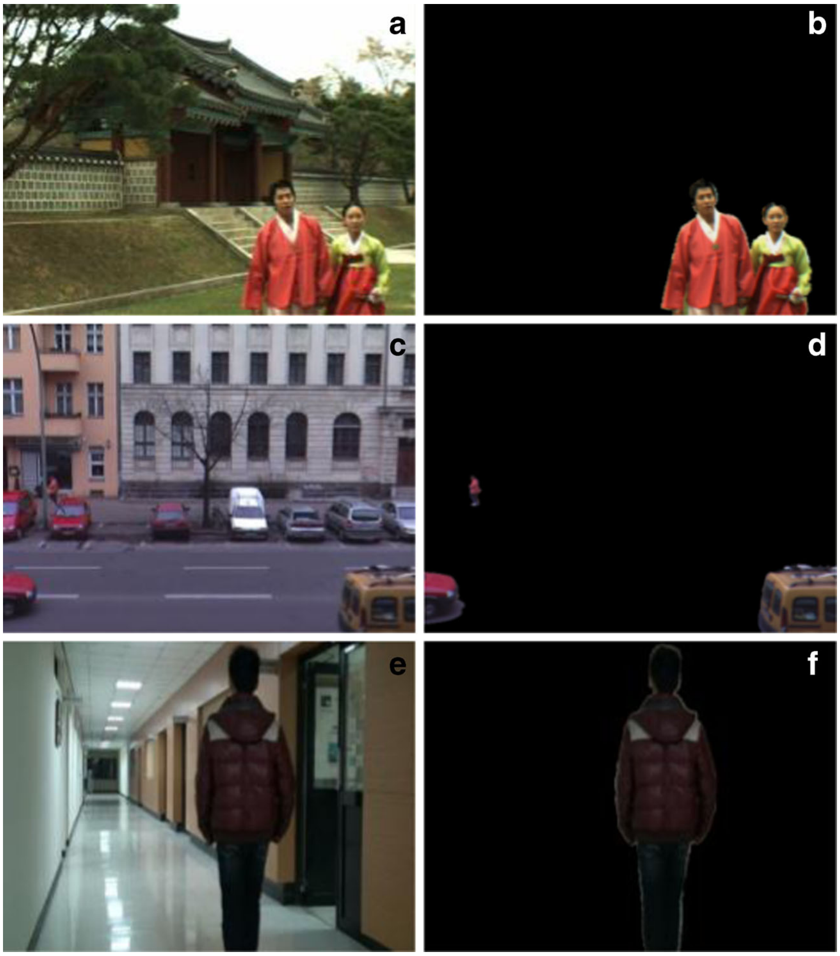


Figure 10 The results of moving object separation. **a** and **b** are from the *lovebird1* (camera 1); **c** and **d** are from *Alt Moabit* (camera 7); **e** and **f** are from the homemade video, *hallway*. After the moving object separation, the final object masks are shown in **b**, **d**, and **f**.

coordinate can be determined using (22), which is derived from (2).

$$x_r = \frac{w}{\sqrt{\frac{d^2 + {h_C}^2}{f^2 + \left(y_h - \frac{H_{frame}}{2}\right)^2}}} + \frac{W_{frame}}{2}. \tag{22}$$

For the pixels (x_v,y_v) in view V_L on the vertical plane, the w and d components of the corresponding point $p_1(w, h,d)$ in the world coordinate can also be calculated using (1) and (2) through their footholds. For the pixels of the non-stationary objects, the calculation of w and d components follows the same way.

As shown in Fig. 6, let p_2 be the projected point on the view V_R. In addition to p_1, p_2, p_3, and p_4 in Fig. 6, two more points $p_5(0, h,d)$ and $p_6(0, h, 0)$ are used for the derivation of x_r. Note that p_3 is the projected point on the view V_R from p_5. Since $\angle p_3p_4p_2$ equals $\angle p_5p_6p_1$, and $\angle p_2p_3p_4 = \angle p_1p_5p_6 = 90°$, $\Delta p_2p_3p_4$ and $\Delta p_1p_5p_6$ are similar triangles. Therefore,

$$\overline{p_3\,p_2} = x_r - \frac{W_{frame}}{2} = \frac{w}{d} \cdot \overline{p_4\,p_3}. \tag{23}$$

Using Eq. (23) and the length of $\overline{p_4\,p_3}$ in (16), the X component x_r on the synthesized view V_R in the image coordinate system can be calculated.

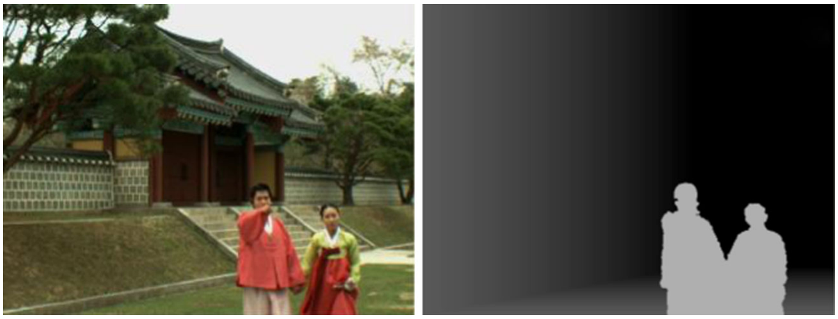


Figure 11 A frame of *lovebird1*: synthesized view (*left*); depth map (*right*), using the proposed system.

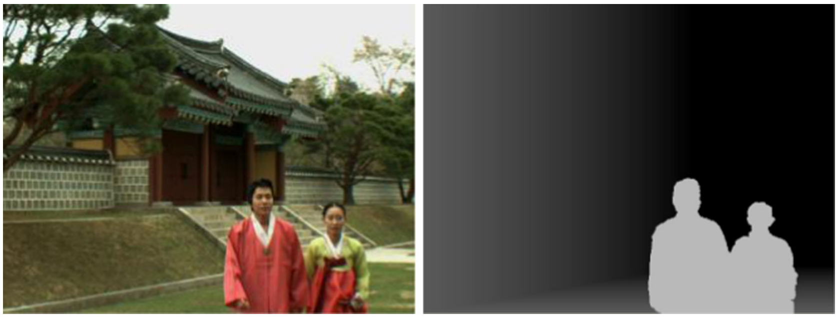


Figure 12 The next frame of *lovebird1*: synthesized view (*left*); depth map (*right*), using the proposed system.

3 The 2D to 3D View Synthesis System

To show the usefulness of the proposed models described in Section 2, a complete 2D to 3D view synthesis system was implemented. The block diagram of the developed 2D to 3D view synthesis system is shown in Fig. 8. The non-stationary objects are first separated from the background in a given video. The vanishing lines and point are extracted from the processed video that only contains the background part. There are many existent methods available to perform background separation and to extract vanishing lines/point. In this paper, we adopted the background registration technique for background separation, due to its simplicity and satisfactory results. In addition, edge information is calculated for the extraction of the vanishing lines and point. Details are given in the following subsections. After those two parts, the view synthesis can be completed using the transformation models described in Section 2.1 and 2.2, as well as the disparity calculation described in Section 2.3.

3.1 Background Registration for Object Separation

The moving object separation in the proposed framework of 2D to 3D conversion is mainly based on the moving object segmentation in [11] with modification. Unlike other object segmentation algorithms which are based on motion [12, 13], the algorithm in [11] is based on background registration to distinguish objects from background in a video with the assumption that the global motion due to camera movement has been properly compensated and the background region can be considered stationary. The steps of the modified moving object separation include frame difference estimation, background registration, and object detection.

The estimation of frame difference is the fundamental operator to find the changed regions that is called frame difference mask. The idea is to calculate the difference between the consecutive frames against a threshold. However, the frame difference mask distinguished using the threshold method only represents the rough shape of the moving objects as shown in Fig. 9a. Since the background consists of stationary pixels in this algorithm, the regions mistakenly considered stationary will result in erroneous judgment as shown in Fig. 9b.

To counter the drawback, we modified the algorithm by spreading the rough shape of the frame difference mask to the neighborhood using pixel-based dilation. However, if the dilation is performed over the entire region of the detected pixels, more stationary pixels could be considered as moving ones, instead of only the neighborhood that shows similar texture as the detected pixels. Specifically, if a pixel in the current frame is similar to its neighbors but the pixel of the same coordinates in the previous frame is not, it usually means that the object moves out of the captured scene. On the other hand, if a pixel in the previous frame is similar to its neighbors but that pixel of the same coordinates in the current frame is not, most likely it suggests that the object moves into the video frame. Therefore, the dilation operator is only performed when a pixel is similar to the neighboring pixels as well as the pixels of the same coordinates over several consecutive frames.

For each video frame, the frame difference mask roughly represents whether a pixel is stationary or not. If a pixel has

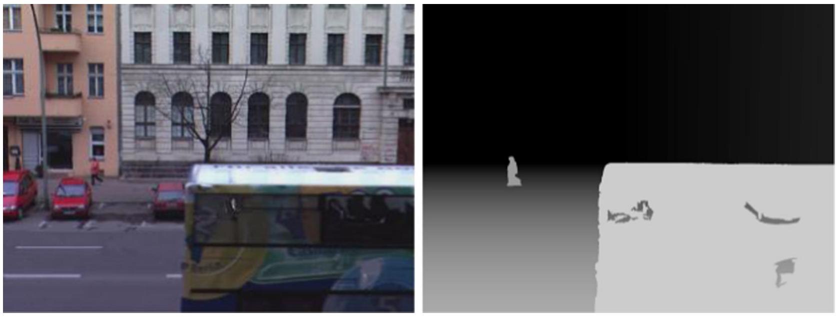


Figure 13 *Alt Moabit*: synthesized view (*left*); depth map (*right*), using the proposed system.

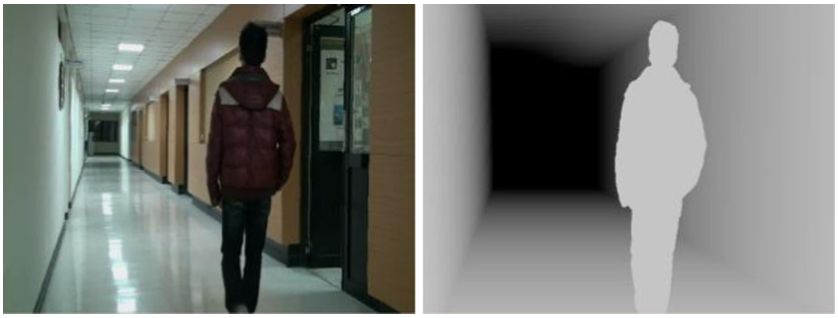


Figure 14 *hallway*: synthesized view (*left*); depth map (*right*), using the proposed system.

been stationary over several consecutive frames, it is considered as one of the background pixels in the background registration algorithm. To decrease possible artifacts of the determined background frame, the background pixels of the same location are averaged along the temporal axis.

Examples of the results of the frame difference masks and background registration are shown in Fig. 9 where Fig. 9c and e are the frame difference masks resulted from the pixel-based dilation operation without and with restriction mentioned above, respectively. There are less background pixels classified as part of the frame difference masks incorrectly with the restricted dilation. The background registration can be improved if less erroneous judgments are made. In Fig. 9d, there are some missing regions shown in black near the bottom-right of the image.

An object in each video frame that is either moving or temporarily stationary in a short time can then be classified after taking the difference of the registered background and the current frame.

To eliminate possible holes and cracks of segmented objects, dilation/erosion and small-region filtering are applied as also shown in [11]. As for the dilation operator, it is utilized to enlarge a region by filling the mask for each moving pixel. After dilation, it is necessary to perform erosion operator to keep the size similar. For the small-region filtering, it is considered noise if there are less connected pixels in an object. However, noise can appear not only in the foreground region but also in the background region. Small-region filtering should be performed in both regions to eliminate possible holes and cracks. A few examples of the results of moving object separation are shown in Fig. 10.

3.2 Extraction of Vanishing Lines and Point

Usually, the disparity information can be extracted after finding the corresponding points between multi-views. However, it cannot be done for monocular videos. If the camera moves slowly, the motion parallax could be used to estimate the disparity information. In the developed system in this paper, the camera is assumed to be stationary, and the disparity information can be estimated with only one view.

The extraction of vanishing lines can be misled by the moving objects. In the proposed method, they are removed first using the object mask mentioned in the previous subsection. Then, edge detection using the *Sobel* operator is performed to locate the edge information. Along each edge in a video frame, the number of edge points with similar gradient is counted. The edges are then sorted by the numbers of edge points as the candidates of vanishing lines. Most of the vanishing lines would intersect at the neighborhood of the vanishing point. To determine the most suitable vanishing point, all the intersections from the candidates of vanishing lines are determined first. The intersections are enclosed in the circles with various radiuses. The circle with more intersections and smaller radius is selected as the vanishing point.

According to the location of vanishing point, there are five important vanishing situations as defined in [6], including left case, right case, up case, down case, and inside case. After determining the vanishing situation according to the position of vanishing point, the main vanishing lines which divide the background into horizontal planes and vertical planes can be obtained for the proposed transformation models introduced in Section 2. In the situation of either left

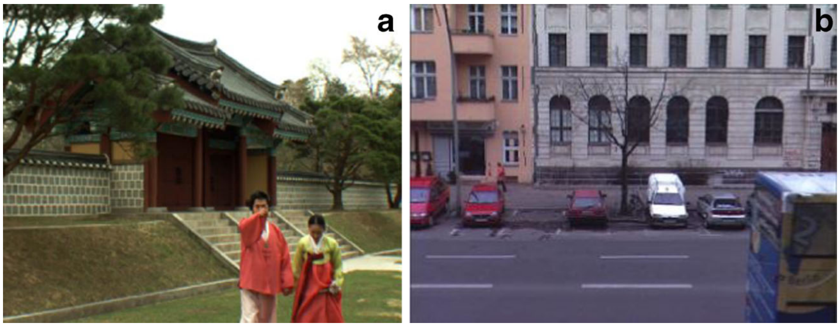


Figure 15 The synthesized results using VSRS **a** *lovebird1* **b** *Alt Moabit*.

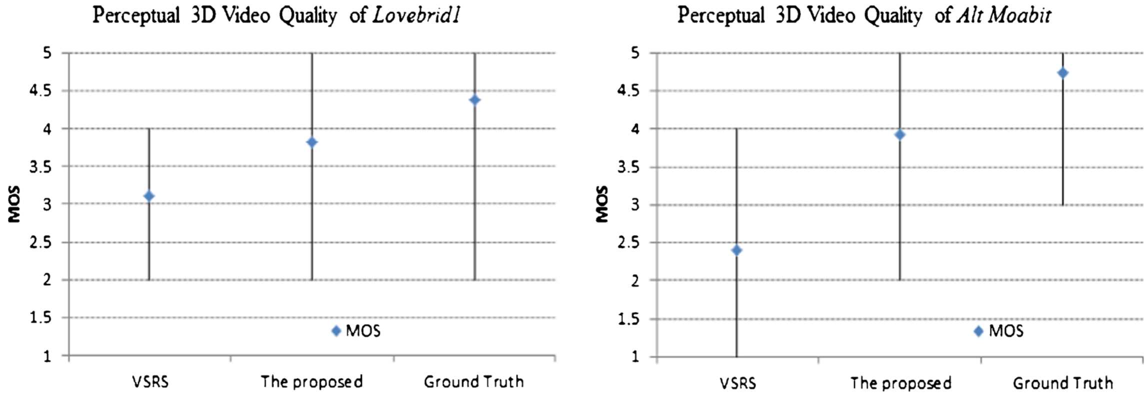


Figure 16 The MOS comparison of the synthesized results of sequences *lovebird1* (*left*) and *Alt Moabit* (*right*).

or right case, there is only one main vanishing line. The vanishing line with the most edge points in the region below the vanishing point is selected as the main vanishing line. In the situation of either up or down case, there are two main vanishing lines. The vanishing lines with the most edge points in either the left part or the right part of the frame are chosen as the main vanishing lines. The main vanishing lines of the inside case are selected similarly to the up or down case, except that there are four regions in the inside case.

Inaccurate positions of vanishing lines and vanishing point can cause erroneous estimation of disparity vectors. To reduce possible detection error of vanishing point, the region of vanishing point is obtained by intersecting vanishing points found in each frame. The mean point of the region is then considered as the final position of vanishing point.

4 Experimental Results

As to many 2D to 3D video/image conversion algorithms in the literature, such as [4~6], the performance of synthesized results is often quite limited due to lack of sufficient video/image information, compared to the video/image synthesis from multi-view.

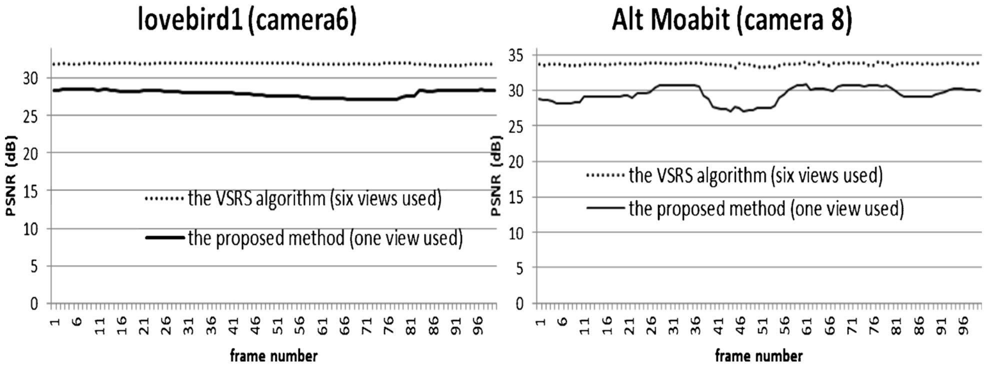
For the proposed models, the objective is to synthesize what the observer could have seen at the scene, instead of exaggerating the depth sensation. The magnification of perceived depth perception can be done quite easily, if it is desired. In this section, three video sequences are used to synthesize their corresponding desired view V_R for stereoscopic vision using the proposed method. Those video sequences are *lovebird1* [14], *Alt Moabit* [15], and selfmade video *hallway.* The developed system illustrated in Fig. 8 is used to synthesize camera 6 of the *lovebird1* sequences from camera 5, and to synthesize camera 8 of the *Alt Moabit* sequences from camera 7.

In the video of *lovebird1*, the moving objects move closer and closer to the camera. From the depth maps shown in Figs. 11 and 12, it is clear that the corresponding depth values for the moving objects decrease along the temporal axis. The results for video *Alt Moabit* and video *hallway* are shown in Figs. 13 and 14, respectively.

To further analyze the performance comparison of view synthesis, VSRS [3], a multi-view video synthesis platform, is also used to synthesize the same views for *lovebird1* and *Alt Moabit.*

However, in order for the VSRS to work, the depth maps for both camera 5 and camera 8 of the *lovebird1* sequences need to be estimated first using DERS [1] from their left and right views. Similarly, both camera 7 and camera 10 of the

Figure 17 The PSNR comparison of the synthesized results of sequences *lovebird1* (*left*) and *Alt Moabit* (*right*).



Alt Moabit sequences need to be estimated in advance. Therefore, six views from the *lovebird1* sequences and six views from the *Alt Moabit* sequence are required to complete the task of view synthesis in VSRS. The synthesized videos using VSRS have obvious artifacts. One example per video is shown in Fig. 15. Notice the artifact near the couple and the walking person, respectively. Since the depth estimation algorithm in DERS is based on pixel matching, occlusion regions usually lead to troublesome estimation of depth information.

In this section, subjective and objective quality evaluations are performed. The videos produced using the proposed models are compared with the videos produced using the VSRS. For subjective quality evaluation, the original 3D videos (ground truth) are also evaluated.

According to the subjective evaluation suggestion of 3D images and video sequences in [16], visual quality, depth quality, and comfort are usually taken into consideration during the 3D quality evaluation. In our experiments, 27 reviewers were invited to give mean opinion score (MOS) for each 3D video displayed on a 23″ LED 3D monitor with active-shutter technology. 13 of the reviewers are males, and 14 of them are females. The MOS ranges from 1 to 5; 1 is the lowest perceived quality, and 5 is the highest quality measurement. The results are shown in Fig. 16. The ground truth stands for the original captured 3D videos.

Even though the view synthesis using VSRS requires six views and the proposed system only needs one view, the perceptual quality of the proposed 2D to 3D view synthesis system is actually better than that of VSRS. The main reason is the artifacts introduced by VSRS as also described earlier. The synthesized views using the proposed method are quite smooth, instead.

For the objective quality evaluation, PSNR results are shown in Fig. 17. The comparison of the proposed method with the VSRS is not fair in nature, and it does not actually match the perceptual stereoscopic video quality. However, it can be regarded as a reference to see the performance of 2D to 3D conversion methods.

5 Conclusion

In this paper, two transformation models for stationary scenes and non-stationary objects are proposed. To synthesize the desired view, the depth maps of a given video are estimated using the proposed models for the background and for the moving objects separately. The transformation model for the stationary scene is constructed based on the geometric relationship and vanishing lines. The model for the non-stationary objects is to link the position of an object to its foothold on the stationary background, and the model for stationary scene is utilized for updating the depth of moving objects with the consideration of motion information. The 2D to 3D video synthesis system was developed to demonstrate the usefulness of the transformation models. According to the subjective quality evaluation, the proposed 2D to 3D conversion produces natural and smooth quality of stereoscopic vision, while the VSRS algorithm requires 5 more views and gives worse subjective 3D video quality.

References

1. Tanimoto, M., Fujii, T., & Suzuki, K. (2007). Multi-view depth map of Rena and Akko & Kayo. ISO/IEC JTC1/SC29/WG11, MPEG M14888.
2. Tanimoto, M., Fujii, T., & Suzuki, K., (2007). Experiment of view synthesis using multi-view depth. ISO/IEC JTC1/SC29/WG11, MPEG M14889.
3. Lee, C., & Ho, Y.-S., (2008). View synthesis tools for 3D video. ISO/IEC JTC1/SC29/WG11, MPEG M15851.
4. Zhang, G., Hua, W., Qin, X., Wong, T.-T., & Bao, H. (2007). Stereoscopic video synthesis from a monocular video. *IEEE Transactions on Visualization and Computer Graphics, 13*(4), 686–696.
5. Chang, Y.-L., Fang, C.-Y., Ding, L.-F., Chen, S.-Y., & Chen, L.-G. (2007). Depth map generation for 2D-to-3D conversion by short-term motion assisted color segmentation. *IEEE International Conference on Multimedia and Expo*, pp. 1958–1961.
6. Battiato, S., Capra, A., Curti, S., & La Cascia, M. (2004). 3D stereoscopic image pairs by depth-map generation. *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 124–131.
7. Comaniciu, D., & Meer, P. (1997). Robust analysis of feature spaces: color image segmentation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 750–755.
8. Chien, S.-Y., Huang, Y.-W., Hsieh, B.-Y., Ma, S.-Y., & Chen, L.-G. (2004). Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques. *IEEE Transactions on Multimedia, 6*(5), 732–748.
9. Horry, Y., Anjyo, K., & Arai, K. (1997). Tour into the picture: using a spidery mesh interface to make animation from a single image. *Proceedings of the 24th annual conference on computer graphics and interactive techniques*, pp. 225–232.
10. Kral, K. (1998). Side-to-Side head movements to obtain motion depth cues: a short review of research on the praying mantis. *Behavioural Processes, 43*(1), 71–77.
11. Chien, S.-Y., Ma, S.-Y., & Chen, L.-G. (2002). Efficient moving object segmentation algorithm using background registration technique. *IEEE Transactions on Circuits and Systems for Video Technology, 12*(7), 577–586.
12. Khan, S., & Shah, M. (2001). Object based segmentation of video using color, motion and spatial information. *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, 2*, II-746–II-751.
13. Krutz, A., Kunter, M., Mandal, M., & Frater, M. (2007). Motion-based object segmentation using sprites and anisotropic diffusion. *Eighth International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 35.
14. Um, G.-M., Bang, G., Hur, N., Kim, J., & Ho, Y.-S. (2008). 3D video test material of outdoor scene. ISO/IEC JTC1/SC29/WG11, MPEG M15371.
15. Feldmann, M., Mueller, F., Zilly, R., Tanger, K., Mueller, A., Smolic, P., et al. (2008). HHI test material for 3D video. ISO/IEC JTC1/SC29/WG11, MPEG M15413.

16. Hyunh-Thu, Q., Callet, P., & Barkowsky, M. (2010). Video quality assessment: from 2D to 3D challenges and future trends. *IEEE International Conference on Image Processing*, pp. 4025–4028.

Chien-Chih Han received the B.S. degree and the M.S. degree both in computer science from National Chiao Tung University, HsinChu, Taiwan, R.O.C. in 2007 and 2009, respectively. From 2009 to 2012, he was a software engineer at Cyberlink, Taipei, Taiwan. Since 2013, he has worked for Synopsys, Hsinchu, Taiwan.

Hsu-Feng Hsiao received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C. in 1995, the M.S. degree in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C. in 1997, and the Ph.D. degree in electrical engineering from University of Washington, Seattle, WA, USA in 2005.

He was an engineering officer at Communication Research Laboratory of the Ministry of National Defense, Taiwan from 1997 to 1999. From 2000 to 2001, he was a software engineer at HomeMeeting, Redmond, WA. He had been then a Research Assistant in the department of Electrical Engineering, University of Washington till 2005. Dr. Hsiao has been an Assistant Professor in the department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan since 2005. His research interests include multimedia signal processing and wired/wireless communications.