논문97-2-2-05

Video Content Manipulation Using 3D Analysis for MPEG-4

Sanghoon Sull*

Abstract

This paper is concerned with realistic manipulation of content in video sequences. Manipulation of content in video sequences is one of the content-based functionalities for MPEG-4 Visual standard. We present an approach to synthesizing video sequences by using the intermediate outputs of three-dimensional (3D) motion and depth analysis. For concreteness, we focus on video showing 3D motion of an observer relative to a scene containing planar runways (or roads). We first present a simple runway (or road) model. Then, we describe a method of identifying the runway (or road) boundary in the image using *the Point of Heading Direction* (PHD) which is defined as the image of, the ray along which a camera moves. The 3D motion of the camera is obtained from one of the existing 3D analysis methods. Then, a video sequence containing a runway is manipulated by (i) coloring the scene part above a vanishing line, say blue, to show sky, (ii) filling in the occluded scene parts, and (iii) overlaying the identified runway edges and placing yellow disks in them, simulating lights. Experimental results for a real video sequence are presented.

I. Introduction

Manipulation of content in video sequences is one of the content-based functionalities for MPEG-4 Visual standard. The MPEG-4 Visual standard will allow the hybrid coding of natural images and video together with synthetic (computer generated) scenes. The Visual standard will comprise tools and algorithms supporting the coding of natural video sequences as well as tools to support the compression of synthetic two-dimensional (2D) and three-dimensional (3D) graphic parameters (i.e. compression of synthetic text, coloring, road boundary or simulating light). Generally, it is not easy to find such graphic parameters which gives a realistc synthesis to an input video without 3D information about the scene and the camera motion although the use of 2D mesh [1] sometimes works. Thus, it is important to estimate the relative 3D motion of the camera and the 3D depth of the scene. However, it is known that the 3D estimation of motion and depth for an arbitrary scene is an underconstrained problem and so it is difficult to obtain reliable estimates. Thus, we focus on the video sequences showing 3D motion of an observer relative to the scene containing planar runways (or roads) although our approach can be applied to manipulate the video containing piecewise planar objects which is taken from a camera mounted on a moving platform.

In this paper, we first present a simple runway (or road) model which consists of straight lines. The extracted line features in the input images are used to find the corresponding 3D model lines and compute the parameters of the runway model. Although the step of line detection is necessary compared with the other road-following methods [2, 3], we have two advantages :i) The computation becomes simple due to the smaller number of features. ii) The computed parameters are more robust since the line features, which are global, are less noise sensitive. We assume that the direction of a runway is approximately parallel to the projection of the heading direction of a moving camera onto ground, which is often the case. This domain specific knowledge drastically reduces the search space for the candidate lines for a runway model. To use the knowledge, we

* 고려대학교 전기전자전파공학부
School of Electrical Engineering, Korea University, Seoul

must find the heading direction of a moving camera. The focus of expansion (FOE) usually gives a good clue about the heading direction of the motion when the translation is dominant except for the *forward screw motion* which involves a large translation and rotation about a direction of translation. Therefore, we propose a method of finding the heading direction for a camera in general motion by using the screw motion description [4]. We define the *Point of Heading Direction* (PHD) as the image of the ray along which the camera moves where the displacement vector can be shown to approach zero for the forward screw motion. The FOE is a special case of PHD. Burger and Bhanu [5] introduced a fuzzy focus of expansion that provides the approximate heading direction of a moving camera. Even though FOE gives a good clue about the heading direction for the land vehicles, it could provide a wrong heading direction for a flying aircraft which is rotating about the heading direction as shown in Fig. 2. We use the 3D motion and depth estimates(the relative velocity, position and orientation of the ground with respect to a camera coordinates system) obtained from one [6] of the existing 3D analysis methods. Note that we do not propose a new 3D motion estimation method but we demonstrate an application of the 3D analysis methods for manipulating video contents. After identifying a runway by using the analysis results, the input video is manipulated by (i) coloring the scene part above a vanishing line, say blue, to show sky, (ii) filling in the occluded scene parts, and (iii) overlaying the identified runway edges and placing yellow disks in them, simulating lights. (A vanishing line is defined as the intersection of the image plane with a plane which includes the camera center and is parallel to the object plane.) All of these enhancements are examples of synthesis guided by 3D analysis.

Section II presents a simple 3D runway model. Section III describes how to find the heading direction of motion for a moving camera. Section IV describes our algorithm consisting of five major steps. The first step extracts line features from the input images. The second step computes the heading direction from the estimated values of $\boldsymbol{R}$ and $\vec{T}$. The third step finds the correspondences between the 3D model lines and the two-dimensional (2D) lines in each frame. The search space for finding correspondences becomes small by using the heading direction as stated earlier. The fourth step noniteratively computes the model parameters from the candidate lines whose correspondences were found in the third step. Then, the fifth step describes how to manipulate the parts of the original images corresponding to the identified runway. Finally, Section V presents the details of the implementation and the results obtained in the experiment with a video sequence of 33 images, digitized from a commercially available laserdisc of a film taken from a flying aircraft.

II. Three-Dimensional Runway Model

In this section, we describe a simple 3D model for a runway (or road) shown in Fig. 1.

Let (X_c, Y_c, Z_c) be a 3D coordinate system attached to a moving camera. The 2D coordinates (X, Y) represent the reference coordinates on the ground. The projection of heading direction $\vec{H}_{k,F}$ of the moving camera onto the ground is denoted by $\vec{V}_{k,F}$ at t_k. We define $\vec{N}_{k,GND}$ as the unit surface normal of the ground with respect to the camera coordinates at t_k. Given $\vec{H}_{k,F}$ and $\vec{N}_{k,GND}$, $\vec{V}_{k,F}$ is easily computed. Note that all of the three 3D vectors are expressed in the camera coordinates. The reference ground coordinate system (X, Y) is chosen such that i) both the X-axis and Y-axis are perpendicular t $\vec{N}_{1,GND}$ and ii) the X-axis is parallel to the known direction of the vector $\vec{V}_{1,F}$

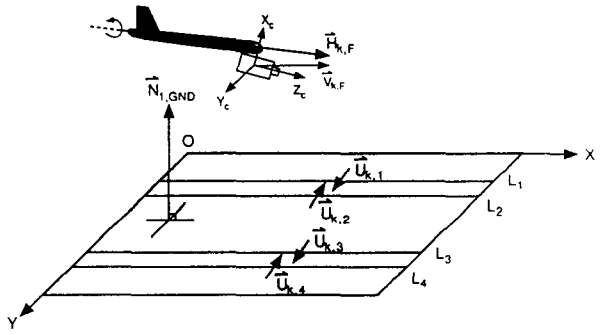


Fig. 1. Scenario of a moving camera over a runway

The four parallel lines $(L_1, L_2, L_3$ and $L_4)$ are used to model boundaries of a runway where L_4 is defined as the rightmost boundary of a runway with respect to the direction of the vector $\vec{V}_{1,F}$. The equations for the four lines are given by

$$\begin{aligned} L_1 &: A_1X+B_1Y+C_1=0 \\ L_2 &: A_2X+B_2Y+C_2=0 \\ L_3 &: A_3X+B_3Y+C_3=0 \\ L_4 &: A_4X+B_4Y+C_4=0 \end{aligned} \tag{1}$$

Although there are 12 unknowns for the four line, we can reduce the number of unknowns to 5 by using the following two geometrical constraints:

i) The four lines are parallel.

ii) The distance between L_1 and L_2 is equal to the distance between L_3 and L_4.

Since the first constraint implies the same slope for each line, we have

$$\begin{aligned} A_1 &= A_2 = A_3 = A_4 \\ B_1 &= B_2 = B_3 = B_4 \end{aligned} \tag{2}$$

Using the above relationships and the second constraint, we can derive

$$C_1 - C_2 = C_3 - C_4 \tag{3}$$

Therefore, the four lines are expressed as follows:

$$\begin{aligned} L_1 &: A_1X+B_1Y+C_1=0 \\ L_2 &: A_1X+B_1Y+C_2=0 \\ L_3 &: A_1X+B_1Y+C_3=0 \\ L_4 &: A_1X+B_1Y-C_1+C_2+C_3=0 \end{aligned} \tag{4}$$

We estimate the set of five unknowns $\boldsymbol{X}=[A_1, B_1, C_1, C_2, C_3]'$up to a scale factor from the candidate lines in a batch of frames.

Ⅲ. Computation of the Heading Direction of a Moving Camera

In this section, we describe how to find the heading direction of a moving camera.

We are concerned with the motion of a camera i) when the aircraft is approaching a runway to land (or when it is taking off a runway), and ii) when it is following the runway on the ground before takeoff and after landing. The second case is similar to the road following when a road boundary can be piecewisely approximated by straight lines.

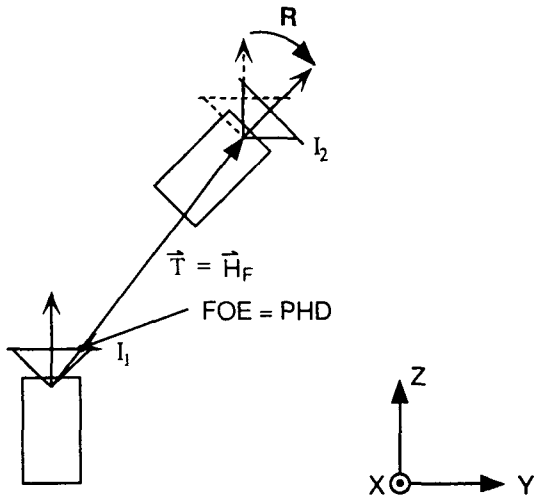
A general rigid motion is described by a translation and a rotation about an axis through the origin of a camera coordinate system. Consider a point P on an object in 3D. Let $\vec{X}=[X, Y, Z]'$ be a 3D coordinate vector of P at time t_1 and let $\vec{X'}=[X, Y, Z]'$ be the corresponding vector at time t_2. Let $\vec{T}$ and $\mathbf{R}$ denote the translation vector and the rotation about the unit axis $\vec{n_w}=[n_X, n_Y, n_Z]'$ through the origin by an angle w, respectively. Thenm we have

$$\vec{x'} = R(\vec{X}+\vec{T}) \tag{5}$$

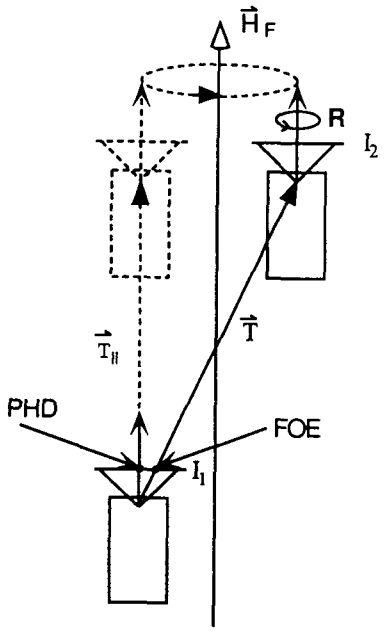
If we view the camera motion as the separate steps of translation ($\vec{T}$) and rotation $\mathbf{R}$ as in Eq. (5), the FOE, which is located at the intersection of the 3D vector $\vec{T}$ and the image plane I_1, gives a good clue about the heading direction ($\vec{H}_F$) for the land vehicle as shown in Fig. 2(a) [5]. However, for the aircraft which is capable of forward screw motion, the direction which is given by the 3D vector $\vec{T}$ is not the heading direction as shown in Fig. 2(b). Therefore, we have to develop a (qualitative) criterion which determines whether the aircraft is in the forward screw motion or not. Further, for the forward screw motion, we would like to identify the heading direction. The interpretation of $\mathbf{R}$ and $\vec{T}$ is important to describe correctly the camera motion.

When an aircraft is approaching a runway to land, its actual motion can be described as a large forward motion with a small up and down jiggling. To describe this motion naturally, we use a screw motion which was proposed by Chasles' theorem [4]. A general rigid body motion can be described by a translation along a unique axis and a rotation about the same axis. Such a description of rigid motion is called a *screw*, and the translation and the unique axis are called the *screw translation* and the *screw axis*, respectively. For a given pair of $\vec{n}_w$ and ω, if $\vec{n}_w$ is in the opposite direction of the screw translation, we reverse the sign of $\vec{n}_w$ and replace ω by $2\pi - \omega$. Note that the screw axis is not defined when the rotation angle is zero (a pure translational case). For a rotation axis $\vec{n}_w$, we denote the parallel and perpendicular translation components by $\vec{T}_{\parallel}$ and $\vec{T}_{\perp}$, respectively. Defining $\vec{T}'$ as $\mathbf{R}\vec{T}'$ for the notational convenience, we decompose $\vec{T}'$ as follows:

$$\vec{T} = \vec{T}_{\parallel} + \vec{T}_{\perp} \tag{6}$$



(a)



(b)

Fig. 2. Illustration of heading direction ($\vec{H}_F$) for two cases. (I_1:Image at t_1, I_2:Image at t_2) (a) $\vec{T}$ point the heading direction ($\vec{H}_F$) (the camera rotates about X axis). (b) $\vec{T}$ points the wrong direction (the carmera rotates about the Z-axis, i.e., forward screw motion).

where

$$\vec{T}' \overset{\text{def}}{=} \vec{T}_{\|} + \vec{T}_{\perp} \tag{7}$$

$$\vec{T}_{\perp} \overset{\text{def}}{=} \vec{T}' - \vec{T}_{\|} \tag{8}$$

Then, the motion of an object is described by

$$\begin{aligned} \vec{X}' &= \boldsymbol{R}\vec{X} + \vec{T}' \\ &= \boldsymbol{R}\vec{X} + \vec{T}_{\|} + \vec{T}_{\perp} \end{aligned} \tag{9}$$

Since the screw axis need not pass through the origin, we denote the displacement vector for the screw axis by $\vec{C}=[C_X, C_Y, C_Z]'$. To make $\vec{C}$ a unique vector, it is constrained to be $\vec{C} \cdot \vec{n}_\omega = 0$ as shown in Fig. 3. Then, a screw motion is represented by

$$\vec{X}' - \vec{C} = \boldsymbol{R}(\vec{X}' - \vec{C}) + \vec{T}_{\|} \tag{10}$$

Rewriting the above equation, we have

$$\vec{X}' = \boldsymbol{R}\vec{X} + (\boldsymbol{I} - \boldsymbol{R})\vec{C} + \vec{T}_{\|} \tag{11}$$

From Eqs. (9) and (11), we have

$$\vec{T}' = (\boldsymbol{I} - \boldsymbol{R})\vec{C} \tag{12}$$

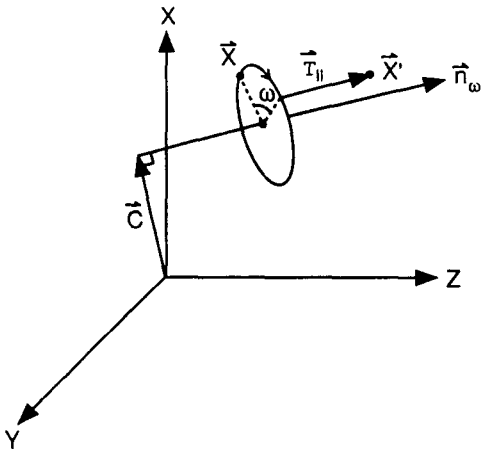


Fig. 3. Illustration of screw motion

We note that the term $\vec{T}_{\|}$ defined in Eq. (7) represents a translation along the rotation axis. In Eq.

(12), the determinant of, (I-R) is zero since $\vec{C}$ is not uniquely defined by the equation alone. If we select one solution for $\vec{C}$ that has the minimum length, the condition $\vec{C} \cdot \vec{n}_w = 0$ is automatically satisfied. Therefore, we find the unique $\vec{C}$ by computing the pseudoinverse [7]. We can also compute $\vec{C}$ from a geometrical relationship which was presented by H. Chen in [4]:

$$\vec{C} = \frac{\left(\vec{T}_\perp + \vec{n}_\omega \times \vec{T}_\perp \cot(\frac{\omega}{2}) \right)}{2}, \quad \omega \neq 0 \tag{13}$$

For the pure translational case, we define $\vec{C}$ as an infinite vector. Now, we qualitatively define the *forward screw* motion if the following criterion is satisfied:

$$\frac{|\vec{C}|}{d_{GND}} < \delta_{sc} \tag{14}$$

where d_{GND} represents the perpendicular distance from the camera origin to the runway plane which can be estimated from the motion analysis stage and δ_{sc} is the predetermined threshold. The above criterion serves as a good indicator to determine whether the motion is the forward screw or not. For example, for the translational motion with a very small rotation angle, the value of $\frac{|\vec{C}|}{d_{GND}}$ is very large. When the camera moves with the translation followed by the rotation about the axis perpendicular to the heading direction as shown in Fig. 2(a), the value of $\frac{|\vec{C}|}{d_{GND}}$ is also very large. However, when the camera moves with the translation followed by the rotation about the axis parallel to the heading direction as shown in Fig. 2(b), the value of $\frac{|\vec{C}|}{d_{GND}}$ is small. For the forward screw motion, we define the heading direction as the direction of $\vec{T}_\parallel$. In summary, if Eq. (14) is satisfied, the heading direction $\vec{H}_F$ is the same as that of $\vec{T}_\parallel$. Otherwise, the heading direction $\vec{H}_F$ is defined as that of $\vec{T}$.

In the case of pure translation, the displacement field consists of vectors of varying length that all pass through a single point in the image plane when extended. This point where the displacement vector is zero is called the FOE: it is the image of the ray along which the camera moves. For the forward screw motion, we can similarly define the PHD in the image plane where the displacement vector is zero. The PHD also gives the information on the direction along which the camera moves. Consider a 3D point X which is lying on the line through the screw axis at t_1

$$\vec{X} = \vec{C} + k\vec{n}_\omega \tag{15}$$

where k is a real number. Then, by using Eqs. (7) and (11), the corresponding point x' at t_2 is described as

$$\vec{X}' = k\vec{n}_\omega + \vec{C} + (\vec{T}' \cdot \vec{n}_\omega)\vec{n}_\omega \tag{16}$$

Hencem the image coordingates are given by

$$\begin{aligned} x &= \frac{kn_{X+}C_X}{kn_Z + C_Z} \\ y &= \frac{kn_{Y+}C_Y}{kn_Z + C_Z} \\ x' &= \frac{[k + (\vec{T} \cdot \vec{n}_\omega)]n_{X+}C_X}{[k + (\vec{T} \cdot \vec{n}_\omega)]n_{Z+}C_Z} \\ y' &= \frac{[k + \vec{T} \cdot \vec{n}_\omega)]n_{Y+}C_Y}{[k + (\vec{T} \cdot \vec{n}_\omega)]n_{Y+}C_Z} \end{aligned} \tag{17}$$

Then we can express the image coordinates (PHD_x, PHD_y) of the PHD as follows:

$$PHD_X = \lim_{k\to\infty} x = \lim_{k\to\infty} x' = \frac{n_X}{n_Z} \tag{18}$$

$$PHD_Y = \lim_{k\to\infty} y = \lim_{k\to\infty} y' = \frac{n_Y}{n_Z} \tag{19}$$

Therefore, if the point X is far away from the camera (which is valid in most cases), the displacement vector approaches to zero at PHD. In some sense, this is trivial since the points at a far distance from the camera are less affected by the translation. For the forward screw motion, the PHD gives the direction along which the camera moves. For the nonforward motion, we have another image point where the displacement vector is zero if $\vec{T}' \cdot \vec{n}_w$ is zero in Eg. (17).

If we have a good flow field, we can directly find the forward direction by locating the zero flow point in the image plane without estimating R and $\vec{T}$.

Our goal is to identify a runway from the detected lines by using the motion estimates obtained during a

3D analysis stage. Therefore, to reduce the search space, we use the knowledge that the direction of the runway is approximately parallel to the projection of the heading direction $\vec{H}_F$ of a moving camera into the ground.

Ⅳ. Algorithm

In this section, we describe the five major steps of the algorithm which were overviewed in Section 1.

1. Feature extraction

This step extracts the straight lines from the input images based on the method in [8]. For each line $l_{k,i}$ at t_k, we compute $length_{k,i}$(length), $contrast_{k,i}$ (difference in intensity of one side from the opposite along $l_{k,i}$) and $\vec{u}_{k,i}$ (the average image gradient direction at those image locations belonging to $l_{k,i}$). The starting point of $\vec{u}_{k,i}$ is defined at the midpoint of the line $l_{k,i}$ in the image at t_k. For example, the vector $\vec{u}_{k,i}$ is illustrated in Fig. 4 for the four detected lines in the image at t_k where the parts of the image between $l_{k,1}$ and $l_{k,2}$, and $l_{k,3}$ and $l_{k,4}$ are assumed to be brighter than the rest.

2. Computation of the heading direction

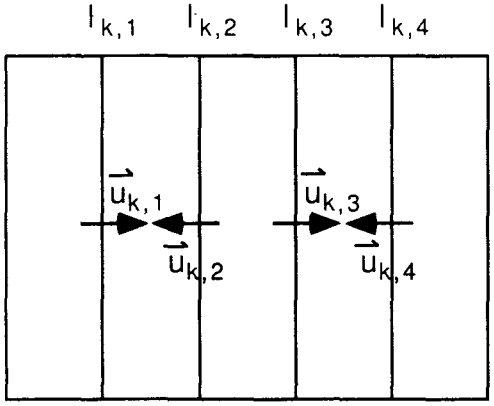


Fig. 4. Illustration of $\vec{u}_{k,i}$ (the average image gradient direction of the line $l_{k,i}$).

This step computes the heading direction $\vec{H}_{k,F}$ from the the estimated values of R and $\vec{T}$ (which are with respect to the camera coordinates system) between t_k and t_{k+1}. If Eq. (14) is satisfied, the heading direction $\vec{H}_{k,F}$ is defined as that of $\vec{n}_\omega$ at t_k. Otherwise, the heading direction $\vec{H}_{k,F}$ is defined as that of $\vec{T}$ at t_k.

3. Matching

This step finds the best set of correspondences between the 3D model lines and the 2D line segments in each frame. An iterative approach is used to find the best correspondences in [9, 10]. However, if we use a brute force search to find the line correspondences, the search space is large, and therefore it is often difficult to reliably identify the 2D line segments for a runway.

Given the estimates of motion and orientation of a moving camera with respect to the ground, we use domain specific knowledge that the direction of the runway is approximately parallel to the projection ($\vec{V}_{k,F}$) of the heading direction $\vec{H}_{k,F}$ of a moving camera onto the ground at t_k. This drastically reduces the search space for the candidate 2D lines. We select the candidate 2D lines by comparing the direction of the 2D line $l_{k,i}$ in the image plane with $\vec{V}_{k,F}$ after it is backprojected to the 3D line $D_{k,i}$ in the reference ground coordinate system by using motion and structure estimates. Before backprojection, we first remove from further consideration the 2D lines whose directions are far from the one which is the projection of the forward motion onto the image plane at the point corresponding to the 2D center of the ground in each frame. This step is useful for removing the spurious lines which are present near the vanishing line in the image.

We also assume that the boundaries of the runway surface are painted with strips of brighter colors than is the runway surface as shown in Fig. 1. We define $\vec{U}_{k,i}$ as the 3D unit vector which is the backprojection of the 2D vector $\vec{u}_{k,i}$ into the 3D scene by using the motion and structure estimates. Since the reference ground coordinate system (X,Y) is chosen in such a way that the X-axis is parallel to $\vec{V}_{1,F}$ and L_4 is the rightmost boundary of a runway with respect to $\vec{V}_{k,F}$, the candidate lines of L_1 and L_3 should then satisfy $\vec{V}_{k,F} \times \vec{U}_{k,GND} \cdot \vec{N}_{K,i} < 0$. Further, the Y-coordinate of the candidate for L_1 should be smaller than that of L_3. The candidates for L_2 and L_4 are similarly constrained.

To find the candidate 2D line segments, we consider

only those 2D lines whose lengths are longer than *thre_length* and whose contrasts are larger than *thre-cont*. The matching algorithm is summarized below.

Matching Algorithm

$\vec{V}_{k,F}$ and $\vec{N}_{k,GND}$ are initially given. Let $L_i^{(k)}$ be a set of candidates for L_i at t_k.

$K_i^{(k)} \leftarrow \phi$
for each frame k
{
for each line $l_{k,j}$
if ($length_{k,i} > thre\_length$ and $contrast_{k,i} > thre\_cont$)
Backproject 2D line $l_{k,i}$ to 3D line $D_{k,i}$.
if (angle between $D_{k,i}$ and $\vec{V}_{k,F}$ < thre_angle)
Backproject 2D vector $\vec{u}_{k,i}$ to 3D vector $\vec{U}_{k,i}$.
$L_{1,3} \leftarrow \phi$
$L_{2,4} \leftarrow \phi$
if(($\vec{V}_{k,F} \times \vec{U}_{k,i}$) · $\vec{N}_{k,GND} < 0$)
$L_{1,3} \leftarrow l_{k,i}$.
else
$l_{2,4} \leftarrow l_{k,i}$.
}
if(cardinality of $L_{1,3} = 2$)
Include the line whose Y-coordinate is smaller into $L_1^{(k)}$ and the other line into $L_3^{(k)}$.
if(cardinality of $L_{2,4} = 2$)
Include the line whose Y-coordinate is smaller into $L_2^{(k)}$ and the other line into $L_4^{(k)}$.
}

We note here that some of the line segments are not visible due to occlusion in some frames. Further, even if line features are present in the scene, the detection process may often miss some of them. Since the unknown coefficients are computed from a sequence of the images, it is not necessary to find the set of correspondences in every frame.

In principle, we can use Hough Transform of 5D space for $(A_1, B_1, C_1, C_2, C_3)$ to find the correspondences in a batch of frames. Since the 5D space is too large to be searched, we would like to reduce the search space. In fact, (A_1, A_2) and (C_1, C_2, C_3) are decomposable since the former represents the slope of a line and the latter describes the distances between the lines. Therefore, the effective search space becomes 2D and 3D which can be done sequentially. Once the runway is identified in the earlier frames, this facilitates the tracking of the runway.

4. Fitting of runway model

To fit the candidate line segments to our model, we minimize the perpendicular distance between the model lines and the candidate line segments with respect to the set of five variables **X** defined earlier. By checking the resulting perpendicular distances, our hypothesis is verified. Referring to Fig. 5, we denote the candidate line for L_i in frame t_k converted to the reference ground coordinates (X,Y) by $D_{k,i}$. Let $P_{k,i}$ and $Q_{k,i}$ be the two end points of the candidate line $D_{k,i}$ whose respective coordinates $(X_{k,i}^P, Y_{k,i}^P)$ and $(X_{k,i}^Q, Y_{k,i}^Q)$ with respect to the reference ground coordinates are computed from the two end points of the candidate line segment in the image plane by using the motion and structure(plane equation for the ground) estimates obtained during the 3D analysis stage.

Our goal is to linearly estimate the coefficients **X** for the four model lines by using the candidate lines in each frame. We use the constraint that the candidate line $D_{k,i}$ and the model line L_i should be identical in the sense that the perpendicular distance from any point on $D_{k,i}$ to L_i should be zero. Therefore, the perpendicular distances from two end points on $D_{k,i}$ to L_i should be ideally zero. Let K be the total number of frames under consideration. Then, our objective function to be minimized with respect to the unknown coefficients of the model lines is

$$G(X) \stackrel{\text{def}}{=} \sum_{k=1}^{K} \sum_{i=1}^{4} \left(\frac{(A_i X_{k,i}^P + B_i Y_{k,i}^P + C_i)}{\sqrt{A_i^2 + B_i^2}} \right)^2 + \left(\frac{(A_i X_{k,i}^Q + B_i Y_{k,i}^Q + C_i)}{\sqrt{A_i^2 + B_i^2}} \right)^2 \quad (20)$$

We removed the absolute signs in the expression for the perpendicular distance since the sum of squared distances is minimized. Note here that the four candidate lines are not available at all frames. From Eq. (4), the unknowns to be estimated are the five unknowns $\boldsymbol{X} = [A_1, B_1, C_1, C_2, C_3]'$ where they are determined up to a scale factor. Further, by using Eq. (2), we can

factor out the constant denominator term $\sqrt{A_i^2+B_i^2}$ from G. Therefore, we need to minimize the following with respect to the five unknows A_1, B_1, C_1, D_2 and C_3:

$$\min_{X} (A_1^2+B_1^2)G(\boldsymbol{X}) \text{ subject to } \boldsymbol{X}'\boldsymbol{X}=1 \quad (21)$$

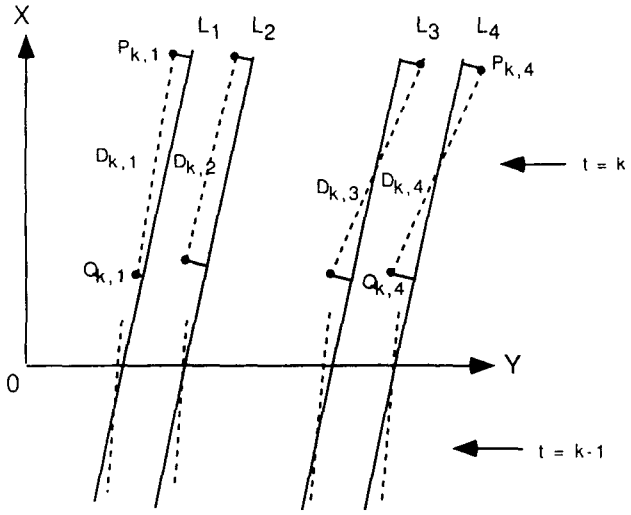


Fig. 5. Model of runway

Define a $8K\times5$ matrix H as follows:

$$H \overset{\text{def}}{=} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \end{bmatrix} \quad (22)$$

where

$$h_k = \begin{bmatrix} X_{k,1}^P & Y_{k,1}^P & 1 & 0 & 0 \\ X_{k,1}^Q & Y_{k,1}^Q & 1 & 0 & 0 \\ X_{k,2}^P & Y_{k,2}^P & 0 & 1 & 0 \\ X_{k,2}^Q & Y_{k,2}^Q & 0 & 1 & 0 \\ X_{k,3}^P & Y_{k,4}^P & 0 & 0 & 1 \\ X_{k,3}^Q & Y_{k,4}^Q & 0 & 0 & 1 \\ X_{k,4}^P & Y_{k,4}^P & -1 & 1 & 1 \\ X_{k,4}^Q & Y_{k,4}^Q & -1 & 1 & 1 \end{bmatrix} \quad (23)$$

Then we can rewrite Eq. (21) in a matrix form,

$$\min_{X} \boldsymbol{X}'\boldsymbol{H}'\boldsymbol{H}\boldsymbol{X} \quad \text{subject to } \boldsymbol{X}'\boldsymbol{X}=1 \quad (24)$$

Since this minimization problem is equivalent to finding the eigenvector corresponding to the minimum eigenvalue, we can noniteratively estimate the five parameters.

5. Manipulation of images

In this section, we first describe how to reconstruct the occluded parts of the objects such as the parts of the ground occluded by the bottom of an aircraft. Then, we present a method of manipulating the part of the images corresponding to the recognized runway.

5-1. Reconstruction of occluded parts

We reconstruct the parts of the ground occluded in some frames by an airplane or a moving vehicle if they appear in other frames. The reconstruction is done by using the flow field obtained from the analysis stage. The first step is to identify the parts of the image which are occluded. This can be done by registering the two images according to the given flow. Irani and Peleg [11] also showed an example of reconstruction of occlusions. The occluded parts are reconstructed by extrapolating the intensity values at the image locations corresponding to the ground which are visible in either directions in the time domain by using the given flow field. We first extrapolate in the forward direction through the image sequence and then in the backward direction. Note that one pixel usually does not correspond to one pixel in the next frame, for example, when the object is approaching. Therefore, the occluded parts in the image are reconstructed by considering the four subpixel points as follows (see Fig. 6):

For each pixel at(i,j) which belongs to the object in the image at t_k:

1. Consider the four rays starting at $(i-\frac{\Delta_V}{2}, j-\frac{\Delta_H}{2})$, $(i-\frac{\Delta_V}{2}, j+\frac{\Delta_H}{2})$, $(i+\frac{\Delta_V}{2}, j-\frac{\Delta_H}{2})$ and $(i+\frac{\Delta_V}{2}, j+\frac{\Delta_H}{2})$ *where* Δ_V *and* Δ_H *repersent the vertical and horizontal sizes of one pixel, respectively.*

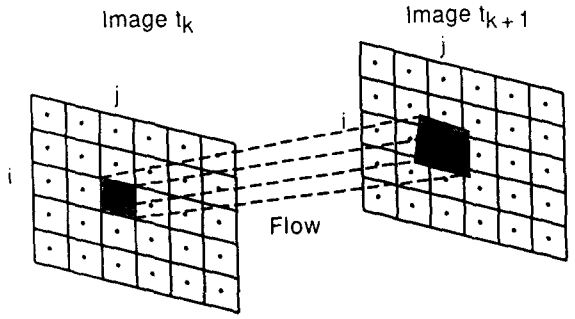


Fig. 6. Illustration of supersampled pixel mapping

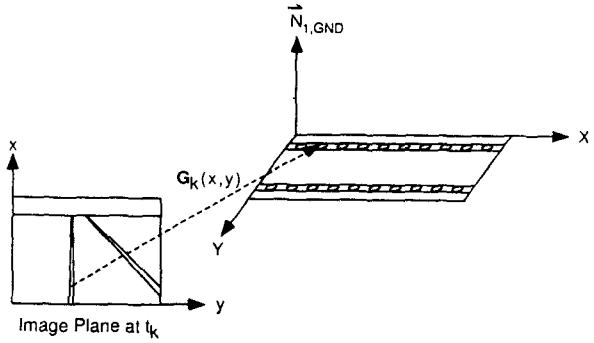


Fig. 7. Supersampling ray tracing for enhancement

2. Send the four rays into the next frame according to the given flow tfield.

For each pixel at t_{k+1}, compute the intensity value by averaginghe intensity values of all of the rays within that pixel coming from the previous frame.

3. Compute the color(intensity) value at the pixel P at t_k by averaging the values of the N_R rays within that pixel.

5-2. Synthesis of the recognized runway

After recognizing the runway, we synthesize the runway by adding an artificial circle pattern along the runway boundaries as shown in Fig. 7. The inside of each circle and the rest of the boundaries are painted with the yellow and black colors, respectively. The values of the radius (R_C) and the period (T_C) of the circle pattern are specified by the user.

Synthesis is performed by using the supersampling ray tracing technique[12]. Here, we send a predetermined number (N_R) of rays through the pixel and let each ray contribute one-N_Rth for the final color of the pixel. Since the coefficients for the runway model were determined in the previous step, we can easily find those pixels that belong to the runway boundaries. Now, the steps are described as follows:

For each pixel P which belongs to the side boundaries of the runway in the image at t_k:

1. Consider the N_R rays through it

2. Send the N_R rays to the ground coordinates where the image coordinate (x, y) is transformed to $(G_X(x,y), G_Y(x,y))$ according to the given flow field. If the ray hits the inside of the circles on the ground, the color value of the ray is set to yellow. Ohterwise, the value is set to to black.

V. Experiment

We derived a sequence of 33 frames from a commercially available CAV laserdisc of a film shot from a flying aircraft. The focal length was assumed to be 8 mm. The digitization was done with a resolution of 640 by 480. This is a challenging sequence to our algorithm since the images contain partially or completely occluded vanishing lines and there is reflection of the ground on the bottom of the airplane.

Lines are detected using a modified version of the method described in [8]. The threshold value of δ_{sc} in Eq. (14) was set to one.

We apply the integrated 3D analysis method in [6] to compute R, $\vec{T}$ and the relative orientation of the ground with respect to the camera-centered coordinates for each pair of successive frames. Then, the estimates are used to compute $\vec{H}_{k,F}$, $\vec{V}_{k,F}$ and $\vec{N}_{k,GND}$ from the input image sequence.

From the displacement field obtained from the 3D estimates, we measured the length of displacement vectors for comparison. Their values at PHD and FOE are 1.15 pixels and 9.18 pixels, respectively. Therefore, we see that the displacement vector is close to zero at PHD for the forward screw motion. By applying Eq. (14), only those frames between $t_0,\ldots,t_7$ are classified as having a forward screw motion. For the frames between $t_7,\ldots,t_{32}$ where the motion consists of a dominant translation, PHD is the same as FOE.

In Fig. 8(a) and (b), we show 3 frames of the input

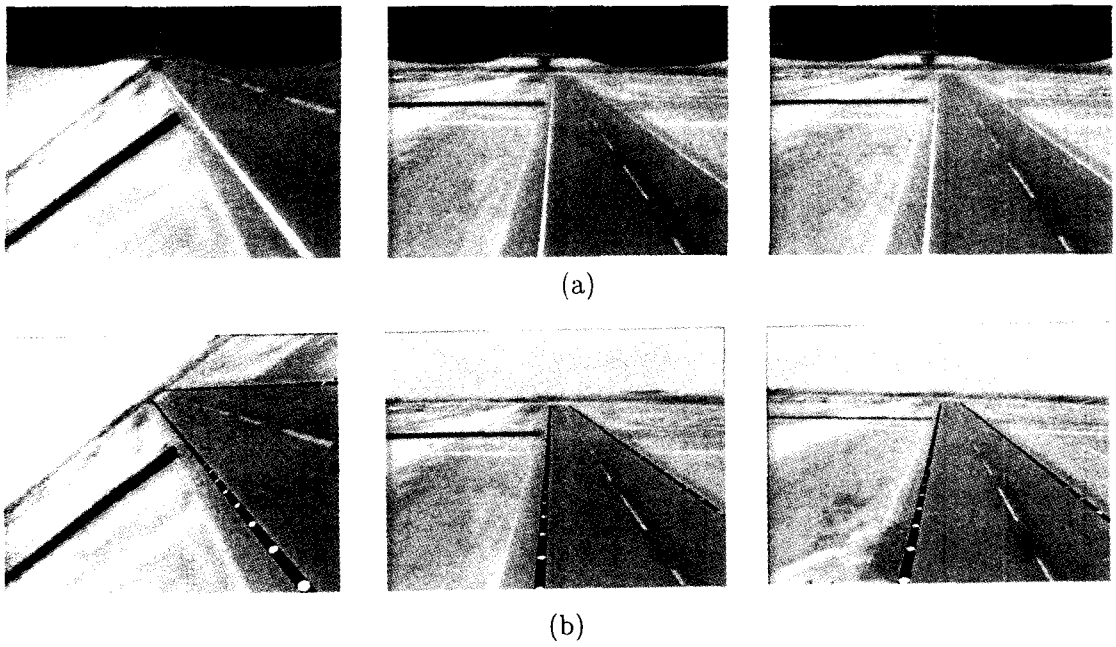


Fig. 8. Runway Video :(a) Original sequence (b) Same as the original sequence but the recognized runway edges are enhanced by repainting them and placing yellow disks in them. As results of the 3D analysis :the airplane could be removed from the image sequence, the occluded scene parts filled in (by interpolating the estimated nearby structure), and the scene part above the vanishing line colored (blue).

video and the resulting manipulated video, respectively. The parts of the image corresponding to *sky* are colorized with sky blue where the sky is defined as the parts of the image above the vanishing line obtained from the analysis stage. For this video the occluded parts of the earlier images are generated from the information in the later images of the sequence. In the matching algorithm, the values of *thre_length*, *thre_cont* and *thre_angle* were 140 pixels, 19 and 15 deg, respectively. All of the four candidate line segments were successfully identified except at t=(0, 1, 2, 3, 4, 5, 8, 11, 17). Using these candidate line segments, the coefficients for the model were computed. Note that the runway boundaries are located very well. For the supersampling ray tracing technique to paint a yellow circle pattern along the runway boundaries, the size of the radius (R_C) is set to 90% of the width of the runway boundary obtained from the analysis stage. The period of the circle pattern is set to $8 \times R_C$. The number of rays (N_R) used within each pixel is 4. We see that the resulting video is realistically manipulated using the 3D analysis result.

VI. Condusions and Extensions

We have described an approach to synthesizing video sequences by using the outputs of the 3D motion and depth analysis. We focused on the video containing planar runway (or road). A video sequence containing a runway was manipulated by (i) coloring the scene part above a vanishing line, say blue, to show sky, (ii) filling in the occluded scene parts, and (iii) overlaying the identified runway edges and placing yellow disks in them, simulating lights. Experimental results for a real video sequence were presented to demonstrate the feasibility of our 3D analysis-based approach to manipulate video content in a realistic way. We plan to extend our method.

References

[1] C. Toklu, A. T. Erdem, M. I. Sezan, and A. M. Tekal, "2-d mesh tracking for synthetic transfiguratio," in *Proc. Int. Conf. Image Processing*, vol. 3, (Washington, D.C.), 1995.

[2] D. Raviv and M. Herman, "A new approach to vision and control for road following," in *Proc. IEEE Workshop on Visual Motion*, (Princeton, NJ), pp. 217-225, Oct. 1991.
[3] E. Dickmanns and B. D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Trans. Patt. Anal. Mach Intell.*, vol. 14, pp. 199-213, Feb. 1992.
[4] H. H. Chen, "A screw motion approach to uniquness analysis of head-eye geometry," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, (Maui, Hawaii), pp. 145-151, Jun. 1991.
[5] W. Burger and B. Bhanu, "on computing a fuzzy of expansion for autonomous navigation," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, pp. 563-568, 1989.
[6] S. Sull and N. Ahuja, "Integrated 3D analysis and analysis guided synthesis of flight image sequence s," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, pp. 357-372, Apr. 1994.
[7] G. Strang, *Linear Algebra and Its Applications*. New York: Academic Press, 1980.
[8] Y. Liu and T. Huang, "Determining straight line correspondences from intensity images," *Patt. Recog.*, vol. 24, pp. 489-504, 1991.
[9] D. Lowem "Three-dimensional object recognition from single two-dimensional images," Artif. Intell., vol. 31. pp. 355-395, 1987.
[10] T. T. D. Koller, K. Daniilidis and H. Nagel, "Model-based object tracking in tracking in traffic scenes," in *Proc. European Conf. Comput.* Vision, (Italy), pp. 437-452, 1992.
[11] M. Irani and S. Peleg, "Image sequence enhancement using multiple motions analysis," in *Proc. IEEE Conf. Comput. Vision Patt.* Recogn., (Champaign, IL), pp. 216-221, 1992.
[12] A. S. Glassner, *An Introduction to Ray Tracing*. San Diego: Academic Press, 1986.

저 자 소 개

설 상 훈

1981년 서울대학교 전자공학과 학사
1983년 한국과학기술원 전기 및 전자공학과 석사
1993년 University of Illinois, Urbana-Champaign Dept. of Electrical and Computer Eng. Ph. D
1996년 8월 ~ 1997년 8월 IBM Almaden Research Center : Visual Media Managemant Department
1994년 3월 ~ 1996년 8월 NASA, Ames Research Center : Flight Deck Branch
1993년 1월 ~ 1994년 2월 Univ. of Illions, Beckman Institute : Artificial Intelligence/Vision Group
1983년 2월 ~ 1986년 7월 KBS 기술연구소
현 재 고려대학교 전기전자전파공학부 조교수
주관심분야 : 영상처리 및 응용. 멀티미디어 데이터 관리 및 통신