Self-Calibration of a camera from video of a walking human *

Fengjun Lv Tao Zhao Ram Nevatia
University of Southern California
Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089-0273
{flv|taozhao|nevatia}@iris.usc.edu

Abstract

Analysis of human activity from a video camera is simplified by the knowledge of the camera's intrinsic and extrinsic parameters. We describe a technique to estimate such parameters from image observations without requiring measurements of scene objects. We first develop a general technique for calibration using vanishing points and vanishing line. We then describe a method for estimating the needed points and line by observing the motion of a human in the scene. Experimental results, including error estimates, are presented.

1 Introduction

Observation of human activity from a stationary camera is becoming of increasing interest for many applications. This problem can be aided greatly if the camera is calibrated meaning that its intrinsic parameters and its position and orientation with respect to some reference in the scene are known [5]. Camera parameters can be obtained by using standard methods if a calibration object or measurements of enough 3-D points in the scene are available [6]. Such measurements, however, are not always available and this has inspired research in *self-calibration* methods. *Vanishing points* of parallel lines have proven to be useful features for this task. Caprile and Torre described a method to use vanishing points to recover intrinsic parameters [1]. Liebowitz *et al* have developed a method to estimate intrinsic parameters by Cholesky decomposition [2] and to reconstruct scene geometry by using projective geometry methods; they do not explicitly compute the extrinsic parameters. Cipolla *et al* present a method [4] to compute both intrinsic and extrinsic parameters by using three orthogonal vanishing points and one reference point.

We describe an approach similar to [2] and [4] for estimating both intrinsic and extrinsic parameters from three orthogonal vanishing points and an object of known height. Our method differs in deriving a direct solution of the parameters without using matrix manipulations, which should result in greater numerical stability. One vertical vanishing point and a horizontal *vanishing line* provide an approximate solution with some assumptions.

Many scenes of interest may not contain parallel lines or these may be hard to find automatically. We propose a method to estimate the vertical vanishing point and the horizontal vanishing line by observing the motion of a single human in a video sequence. Human motion should be available naturally in videos acquired for human activity observation.

We first describe our method for calibration from three orthogonal vanishing points and then its use by observing a walking human. Some experimental results, including error estimates are presented.

2 Calibration from three vanishing points

The perspective projection model can be represented by a 3×4 matrix M. Let m_1, m_2 and m_3 represent the three row vectors. Then the image coordinates (u, v) of a point P expressed in homogeneous coordinate system, are given by:

$$\begin{cases} u = (m_1 \cdot P)/(m_3 \cdot P) \\ v = (m_2 \cdot P)/(m_3 \cdot P) \end{cases} \quad (1)$$

M is determined by eleven camera parameters. We use a simpler model by assuming that the camera has zero skew and unit (or known) aspect ratio, which is common for a CCD camera [2]. Let a World Coordinate System (WCS), defined by three orthogonal vectors X, Y and Z be placed such that the origin of the Camera Coordinate System (CCS) is *vertically* above (i.e. along the Y-axis) the WCS origin as shown in Fig.1. We will call the X-Z plane as the *ground plane*.

To calibrate the camera, we need to know the three intrinsic parameters (the focal length f, the principal point (u_0, v_0)) and four extrinsic parameters (the tilt angle β, the pan angle α and the yaw angle γ that describe the rotation transform between the WCS and CCS, and the height of the

[1]This research was supported, in part, by the Advanced Research and Development Agency of the U.S. Government under contract No. MDA-908-00-C-0036.

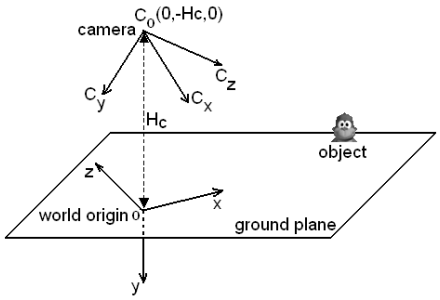
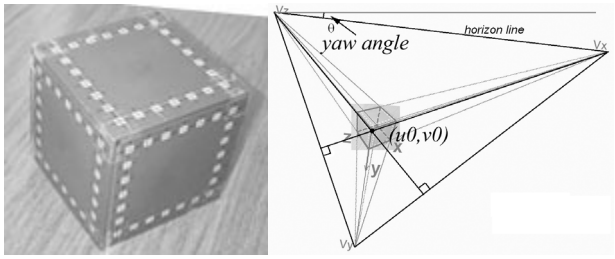


Figure 1: WCS and CCS



(a) (b)

Figure 2: (a) Original image; (b) Vanishing points, horizon line and principal point of the image.

camera H_c). M is then given by:

$$M = \begin{pmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

$$= \begin{pmatrix} c\alpha c\gamma + s\alpha s\beta s\gamma & -c\beta s\gamma & -s\alpha c\gamma + c\alpha s\beta s\gamma \\ c\alpha s\gamma - s\alpha s\beta c\gamma & c\beta c\gamma & -s\alpha s\gamma - c\alpha s\beta c\gamma \\ s\alpha c\beta & s\beta & c\alpha c\beta \end{pmatrix}$$

$$T = (t_1, t_2, t_3)^T = R\begin{pmatrix} 0 & H_c & 0 \end{pmatrix}^T \quad (2)$$

$c\alpha$, $s\alpha$, $c\beta$, $s\beta$, $c\gamma$ and $s\gamma$ denote $cos(\alpha)$, $sin(\alpha)$, $cos(\beta)$, $sin(\beta)$, $cos(\gamma)$ and $sin(\gamma)$ respectively.

2.1 Geometrical properties of vanishing points

Property 1: Images of a family of parallel lines pass through a common point in the image plane called their vanishing point. From Equ.1 and 2, the image coordinates of the vanishing point can be derived as (see Appendix A):

$$\left(f\frac{r_{11}a+r_{12}b+r_{13}c}{r_{31}a+r_{32}b+r_{33}c} + u_0, \quad f\frac{r_{21}a+r_{22}b+r_{23}c}{r_{31}a+r_{32}b+r_{33}c} + v_0 \right) \quad (3)$$

Where (a, b, c) is the direction vector of the parallel lines. Given three families of parallel lines in mutually orthogonal directions, we can choose these to be the direction vectors for the axes of the WCS (X, Y and Z). The vanishing points of lines along these directions, say V_x, V_y, V_z can be obtained from Equ.3 by setting (a, b, c) to be $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ respectively and are given by:

$$\begin{array}{l} \left(f\frac{r_{11}}{r_{31}} + u_0, \quad f\frac{r_{21}}{r_{31}} + v_0 \right) \\ \left(f\frac{r_{12}}{r_{32}} + u_0, \quad f\frac{r_{22}}{r_{32}} + v_0 \right) \\ \left(f\frac{r_{13}}{r_{33}} + u_0, \quad f\frac{r_{23}}{r_{33}} + v_0 \right) \end{array} \quad (4)$$

Property 2: The orthocenter of the triangle with the three orthogonal vanishing points as vertices is the principal point (see Fig.2(b)), as shown in [1].

Property 3: Vanishing points of families of parallel lines parallel to a plane lie on a line in the image plane called the *vanishing line* (see Appendix B). If the lines are parallel to the ground plane, the vanishing line is called the *horizon line*. Vanishing points, horizon line and the principal point are illustrated in Fig.2.

Property 4: The angle from horizontal line to the horizon line (θ in Fig.2(b)) is equal to yaw angle γ (from Equ.2 and 4, see Appendix C).

2.2 Computing camera parameters from three vanishing points

Step 1: Given three orthogonal families of parallel lines, we associate a WCS with them and compute (V_x, V_y, V_z) by the intersection of two or more parallel lines in the family. We use an M-Estimator technique with the following weighting function (5) for robust estimation [8].

$$w_i = \begin{cases} 1 & |r_i| \le \sigma \\ \sigma/|r_i| & \sigma < |r_i| \le 3\sigma \\ 0 & 3\sigma < |r_i| \end{cases} \quad (5)$$

where r_i is the residue error of the i-th point and σ is the standard deviation of all residue errors. The initial estimate is given by LMS method.

Step 2: Principal point (u_0,v_0) is computed from property 2.

Step 3: Yaw angle γ is computed from property 4.

Step 4: **(a)** If $\gamma = 0$, from Equ.2 and 4, we have

$$\begin{cases} u_{V_x} = f * ctan(\alpha)/cos(\beta) + u_0 & (6.1) \\ v_{V_x} = -f * tan(\beta) + v_0 & (6.2) \\ u_{V_y} = u_0 & (6.3) \\ v_{V_y} = f * ctan(\beta) + v_0 & (6.4) \\ u_{V_z} = -f * tan(\alpha)/cos(\beta) + u_0 & (6.5) \\ v_{V_z} = -f * tan(\beta) + v_0 & (6.6) \end{cases} \quad (6)$$

From Equ.6.2 and 6.4, f is derived as:

$$f = \sqrt{-(v_{V_x} - v_0)(v_{V_y} - v_0)} \quad (7)$$

From Equ.6.2 and 7, β is derived as:

$$\beta = tan^{-1}((v_0 - v_{V_x})/f) \quad (8)$$

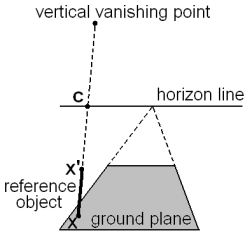


Figure 3: Computing the height of camera

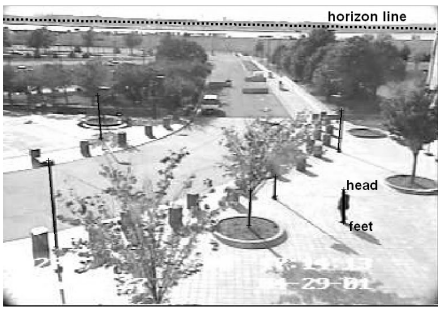


Figure 4: Any vertical structure like pole, trunk or human body can be used to get the vertical vanishing point. The horizon line is the apparent intersection of the earth and sky.

From Equ.6.1, 7 and 8, α is derived as:

$$\alpha = ctan^{-1}((u_{V_x} - u_0)cos(\beta)/f) \tag{9}$$

As shown in Fig.3, by the cross ratio invariance [3], we have

$$\frac{H}{H_c} = 1 - \frac{d(x', c)d(x, v)}{(d(x, c)d(x', v))} \tag{10}$$

If H, the height of a reference object is known, Equ.10 can be used to infer the height of the camera H_c.

(b) If $\gamma \neq 0$, we can rotate the image around the image origin by γ such that the horizon line is aligned with the horizontal line. Now since the yaw angle of the new image is zero, we can apply steps above to this image. Suppose the result is $(f', u_0', v_0', \alpha', \beta', \gamma' = 0, H_c')$, it can be easily derived that the following are the parameters computed for the original image: $(f = f', u_0 = u_0'cos\gamma - v_0'sin\gamma, v_0 = u_0'sin\gamma + v_0'cos\gamma, \alpha = \alpha', \beta = \beta', \gamma, H_c = H_c')$.

2.3 Approximate calibration using a vertical vanishing point and the horizon line

Use of the above method requires finding three orthogonal families of parallel lines. Such lines may not be readily available in many cases. Fortunately, in many such cases, the vertical vanishing point and the horizon line can still be known, as shown in Fig.4.

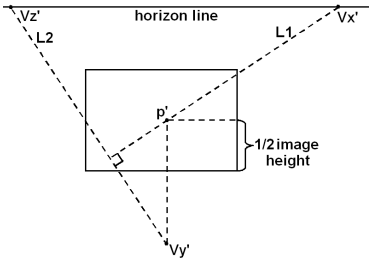


Figure 5: Get the third vanishing point V_z' from V_x', V_y' and p'.

Once the vertical vanishing point and horizon line are known, we can use the following approximate algorithm to compute the camera parameters.

Step 1: Choose arbitrarily a point on the horizon line as V_x.

Step 2: If $\gamma \neq 0$, we rotate the image around image origin by γ such that the horizon line is aligned with the horizontal line. V_x' and V_y' are two vanishing points of the new image.

Step 3: Because the line passing through V_y' and the principal point p' is a vertical line, the x coordinate of p' is equal to that of V_y'. The y coordinate of p' approximates to half of the image height, which is reasonable because in many situations, the principal point is located near the image center [2].

Step 4: Suppose $L1$ is the line passing through V_x' and p', and $L2$ is the line which passes through V_y' and is orthogonal to $L1$. The intersection of $L2$ and the horizon line gives another orthogonal vanishing point V_z', as shown in Fig.5.

Step 5: Since all of three vanishing points are now known, we can use the algorithm in Sec.2.2 to compute the camera parameters.

3 Self-calibration by Walking Human

When points and lines needed to compute vanishing points and horizon line are not readily available (or easy to find automatically) from static scene structure, observing a walking human can provide us an estimate of the vertical vanishing point and the horizon line to perform camera calibration as described below.

Suppose that we can observe a human in N ($N \geq 2$) different locations in an image sequence, and that the head and feet positions can be located. If we assume that the human's body is vertical, all the principal axes of the body should intersect at the vertical vanishing point V_y. The computation of the principal axes will be explained in Sec.3.1.

For any two locations, if we assume that the height of the human remains constant, the line connecting the heads and the line connecting the feet should intersect at a point on the horizon line. If the N ($N \geq 3$) locations of the human are not all on a straight line, the horizon line can be fixed, as shown in Fig.6.

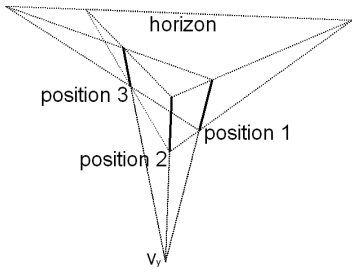


Figure 6: Using walking human to get the vertical vanishing point and the horizon line.

There are $\begin{pmatrix} n \\ 2 \end{pmatrix}$ possible intersections, some of which may be noisy due to the bad localization of head or feet positions. Therefore, a robust estimation technique is needed to compute the accurate horizon line. Again we use an M-Estimator to estimate the horizon line. First all the $\begin{pmatrix} n \\ 2 \end{pmatrix}$ intersections are computed and the initial weights are given by the angles between the two straight lines. The initial estimate is computed by a weighted LMS method and is refined using the same weighting function as in Sec.2.2.

After we compute the vertical vanishing point and the horizon line, the algorithm in Sec.2.3 is employed to derive all the camera parameters.

3.1 Computing the principal axes, head and feet positions

To apply above method, we need to accurately estimate the principal axes and the head/feet positions of a walking human in the image.

Human walking is a periodic motion. At different phases of a walking cycle, the shape of the body is different and the height (i.e., the distance from the head top to the feet bottom) also varies (see Fig.7.a and b for an example). But at the same phase, the shape and the height of a human is the same. Particularly, at the phase when the two legs cross, the two feet positions are nearby and the body is nearly vertical which makes the principal axes and the position of the feet easier to compute. Thus, we aim to detect a walking human from the background, detect the leg-crossing phases, compute the principal axes of the human's body and locate the human's feet and head positions at those phases.

Moving foreground can be extracted fairly accurately with a statistical background model (e.g., [7]). When the moving objects (e.g., humans) are sparse in the image and there is no shadow or reflection, the moving blobs (connected components extracted from moving foreground) correspond well to the moving objects. The moving blobs can be tracked with a blob tracker by spatial proximity and size similarity [5].

Let $(\vec{e}_t^1, \vec{e}_t^2)$, (v_t^1, v_t^2) be the first and second eigen vectors and eigen values of the covariance matrix C_t of a detected

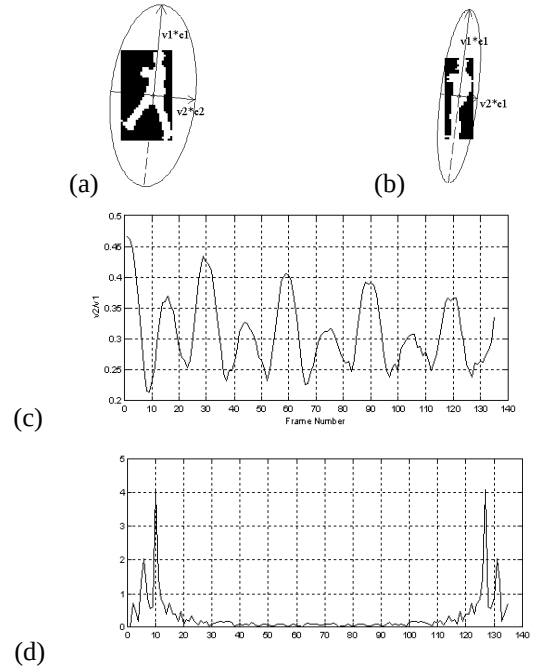


Figure 7: (a), (b) Eigen analysis on the human shape at different phases. The arrows are $\sqrt{v_t^1}\vec{e}_t^1$ and $\sqrt{v_t^2}\vec{e}_t^2$ respectively; (c) Plot of q_t over time; (d) FFT of (c).

blob in frame t respectively. Fig.7.a and b show two example of the computed eigen vectors and eigen values. We define a quantity $q_t = v_t^2/v_t^1$ whose maximum value is achieved when the walker's two legs are most apart and whose minimum value is achieved when the legs cross. q_t is also invariant to the image size and insensitive to the orientation of the walker. Across time, q_t exhibits periodic change (see Fig.7.c for an example of the sequence in Sec.4.2). The principal frequency of q_t is computed by FFT (fast Fourier transformation, as in Fig.7.d). FFT of $q_t, t \in [0, T-1]$ shows a peak at f_{max} (in the example of Fig.7, $f_{max} = 9$), and the frequency of q_t is $F = f_{max}/T$. According to the computed frequency of q_t, we can segment the q_t signal into cycles (starting from the first frame). An average cycle is computed and the leg-crossing phase is located in the average cycle by finding the phase P that gives the minimum average value of q_t. Applying the F and P, all the leg-crossing phases during $t \in [0, T-1]$ are located by:

$P_n = n/F + P$, n are integers so that $P_n \in [0, T-1]$

The value of $\vec{e}_t^1$ gives the principal axis of the body at frame t, which will be used to compute the vertical vanishing point. We find the positions of the head and the feet, at the leg-crossing frame, by analyzing the histogram of the projection of human pixels on the direction of $\vec{e}_t^1$, as shown in Fig.8.a. The head and feet are assumed to be located along the principal axis of the body. Their locations along the axis are determined by finding the first point whose projection is

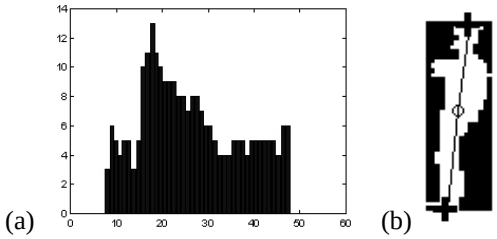


Figure 8: (a) The histogram of the human pixels projected on the first eigen vector; (b) The computed head and feet position as marked with cross.

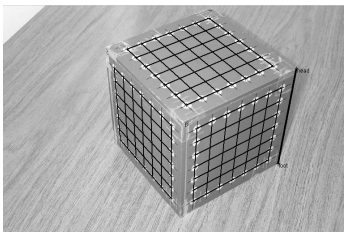


Figure 9: Box image marked with three families of parallel lines in mutually orthogonal directions. The reference height (the right most line) is also shown.

above a threshold.

In deciding the camera height, a reference height needs to be specified. We assume that we know the height of the human or assume an average height ($\bar{H}_h$). In this case, the camera height computed $\hat{H}_c = \frac{\bar{H}_h}{H_h} H_c$, where H_h and H_c are the true heights of the human and the camera respectively. This scale factor only affects H_c and the result of using this camera model is that all the measured heights are scaled by this factor.

When the human walks fast, the forward shift of his/her center of gravity may cause the principle axes to lean forward slightly. This could result in some inaccuracy in the computation of the vertical vanishing point. Considering multiple walkers in opposite directions can solve this problem partially.

4 Experimental results

The presented algorithms have been tested on both a static structure image and a walking human sequence. In order to estimate its accuracy, some points with known 3-D coordinates are given. We use the resulting projection matrix M to compute their image points and measure the average Euclidian distance between the computed image coordinates and the actual image coordinates of each point. Notice that the measured 3-D coordinates may be defined in a different coordinate system than what we choose to be the WCS; we need to also estimate the rigid transformation between the two. If two of the axes of the two systems are in the same plane (say the ground plane), the transformation is determined by the translation (a, b) of the two origins and the angle δ between one axis from each in the ground plane. Given the positions of two points in the ground plane, (a, b) and δ are given by solving three equations.

We compare the result of our algorithm with the classical linear calibration method [6] and method of Liebowitz et al. [2]. Note that the latter algorithm does not explicitly compute extrinsic parameters, so we use the intrinsic parameters from this method and our approach to derive the extrinsic parameters.

4.1 Static structure image

Fig.9 is taken with a KODAK DC280 digital camera and the resolution is 1760×1168. 72 points in the image are used to compute the camera parameters. Their 3-D positions are also given in order to estimate the accuracy of the results. Notice that the linear method computes only the projection matrix and we did not decompose the matrix to get the camera parameters.

As we can see from Table 1, the linear method gives the best result because the error is estimated on the same set of points that are used to compute the projection matrix. In this sense, the linear method is guaranteed to have the least-squares error. The presented method that uses three vanishing points as well as the approximate algorithm generates similar result as that of Liebowitz's method.

	A1	A2	A3	A4
f(pixel)	2782.3	2782.0	2787.5	n/a
u0(pixel)	836.2	837.5	837.2	n/a
v0(pixel)	590.5	584.0	588.7	n/a
pan(rad)	0.5870	0.5873	0.5874	n/a
tilt(rad)	0.7468	0.7477	0.7476	n/a
yaw(rad)	-0.0022	-0.0022	-0.0022	n/a
Hc(mm)	504.09	504.09	504.09	n/a
error(pixel)	1.337	1.377	1.370	0.793

Table 1: Results of static structure image (A1, A2, A3 and A4 denote algorithm presented in Sec.2.2, the approximate algorithm in Sec.2.3, modified method of Liebowitz et al. and the classical linear algorithm respectively).

4.2 Walking human sequence

Fig.10 shows the result of the approach to calibrate the camera by a walking human as described in Sec.3. The sequence was taken with a SONY TRV11 digital camcorder and the image resolution is 360×240. Fig.9.a shows all the located frames of the leg-crossing overlaid on one frame. Fig.9.b shows the head/feet pairs as line segments, the estimated vertical vanishing point,

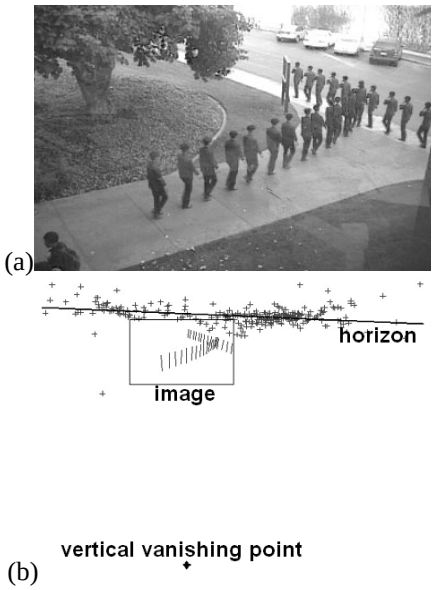


Figure 10: (a) A set of overlaid images of a walking human sequence; (b) Using this sequence to get the vertical vanishing point and the horizon line.

the intersections of the line pairs connecting the human heads and feet, and the estimated horizon line. The result camera parameters are $(f, u_0, v_0, pan, tilt, yaw, H_c) = (332.3, 183, 120, 0.5940, 0.4007, 0.0438, 5247.87)$. Note that f, u_0 and v_0 are in unit pixel, pan, $tilt$ and yaw in rad and H_c in unit mm. We use 12 known 3-D points to compare the error with the result of linear method. The average error of our method is 1.525 pixels and the average error of the linear method is 0.532 pixels.

5 Summary and Conclusions

We have presented a camera calibration algorithm using vanishing points and lines that only requires trigonometric calculations rather than matrix manipulations. We have also shown how the needed points and line can be estimated automatically from observations of a walking human in the scene. We believe that the results indicate performance that is quite satisfactory especially for the task of human motion analysis in video sequences if not necessarily for architectural reconstruction.

Appendix

A. Proof of Property 1:

Assume a line in WCS has direction vector (a, b, c) and passes point (x_0, y_0, z_0) . The parameterized equation of the line is:

$$x_{wcs} = al + x_0, \quad y_{wcs} = bl + y_0, \quad z_{wcs} = cl + z_0 \tag{11}$$

From Equ.1, 2 and 11, we get the image of the line

$$\begin{cases} u = f\frac{r_{11}(al+x_0)+r_{12}(bl+y_0)+r_{13}(cl+z_0)+t_1}{r_{31}(al+x_0)+r_{32}(bl+y_0)+r_{33}(cl+z_0)+t_3} + u_0 \\ v = f\frac{r_{21}(al+x_0)+r_{22}(bl+y_0)+r_{23}(cl+z_0)+t_2}{r_{31}(al+x_0)+r_{32}(bl+y_0)+r_{33}(cl+z_0)+t_3} + v_0 \end{cases} \tag{12}$$

Take limit of (12), as $l \to \infty$, we can get Equ.3. The position of the vanishing point depends only on the direction (a, b, c) and is independent of (x_0, y_0, z_0).

B. Proof of Property 3:

Suppose in CCS, the plane has the normal vector (n_1, n_2, n_3) and one family of parallel lines has the same direction vector (a, b, c). So we have $(a, b, c)(n_1, n_2, n_3) = 0$ The vanishing point of these lines is $u = fa/c + u_0, v = fb/c + v_0$ Therefore, $n_1 u + n_2 v + n_3 f = n_1(fa/c + u_0) + n_2(fb/c + v_0) + n_3 f = (an_1 + bn_2 + cn_3)f/c + n_1 u_0 + n_2 v_0 = n_1 u_0 + n_2 v_0$ This line equation depends only on the normal vector (n_1, n_2, n_3) of the plane.

C. Proof of Property 4:

$$tan\theta = \frac{v_{V_z} - v_{V_x}}{u_{V_z} - u_{V_x}} = \frac{fr_{23}/r_{33} - fr_{21}/r_{31}}{fr_{13}/r_{33} - fr_{11}/r_{31}} \tag{13}$$

From Equ.2 and 13, we have $tan(\theta) = tan(\gamma)$

References

[1] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, 4, 127-140 (1990).

[2] D. Liebowitz, A. Criminisi and A. Zisserman. Creating Architectural Models from Images. In *Proc. EuroGraphics*, volume 18, pages 39-50, Sep 1999.

[3] A. Criminisi, I. Reid, and A. Zisserman. Single View Geometry. *International Journal of Computer Vision*, 40(2): 123-148, 2000.

[4] Cipolla, R., Drummond, T. and Robertson, D.P. Camera calibration from vanishing points in images of architectural scenes. *Proceedings, British Machine Vision Conference, Nottingham*, 2, 382-391 (September 1999).

[5] T. Zhao, R. Nevatia, F. Lv. Segmentation and Tracking of Multiple Humans in Complex Situations, In *Proc. of IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, Kauai,HA, 2001.

[6] O. Faugeras. Three Dimensional Computer Vision: A Geometric Viewpoint. *MIT Press*, 1993.

[7] Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P. Pfinder: real-time tracking of the human body. *IEEE Trans. on PAMI*, Vol. 19, No. 7, 1997.

[8] Olsen S.I. Epipolar line estimation. In *Proc. of European Conference on Computer Vision, Lecture Notes in Computers Science s.* 307-311. Bind 588. Springer-Verlag, Berlin, Tyskland 1992.