Understanding 3-D World by Kalman Filter

Fai'z Ali Rashed AL Ma'mouri
College of science for women , Department of Computer Science

1-Abstract

Human beings can form perceptions about their 3-D world through many years of their experience. Building perception for machines to recognize their 3-D world by visual information is difficult, as the images obtained by cameras can only represent 2-D information. This work demonstrates approach for understanding the 3-D world from 2-D images of a scene by Kalman filtering.

The work ends with reconstructing the 3-D world from multiple 2-D images by using Kalman filtering.

الخلاصة

محاولة الانسان لبناء معرفة حول العالم ذو ثلاثة ابعاد كان منذ زمن طويل. لبناء معرفة للماكنة كي تميز العالم ذو الثلاثة ابعـاد باستخدام المعلومات المرئية يكون صعب لان الصور التي يتم الحصول عليها بواسطة الكامرات تمثل معلومات ذات بعدين فقط.

يستعرض العمل طريقة لفهم العالم ذو ثلاثة ابعاد من الصور ذات البعدين باستخدام مرشح كالمان.

2-Introduction

Visual perception' generally refers to construction of knowledge to understand and interpret 3-dimensional objects from the scenes that humans and machines perceive through their visual sensors. The human eye can receive a wide spectrum of light waves, and discriminate various levels of intensity of light reflected from an object.

In order to construct the models of visual perception, in this work, will presume the camera to be the sensor. Though there exists some similarity between the human eye and a camera with respect to image formation, the resolution of a camera cannot be compared with the human eye. In fact, the resolution of the human eye is more than 100 times the resolution of a good camera. Another basic difference between the human eye and a camera lies in the dimensionality of the devices. While a human eye can feel the third dimension (the depth), a camera can extract only the 2-dimensional view of an object.

Getting the third dimension requires integration of the multiple camera images taken from different directions. This is generally referred to as the 3-D reconstruction problem, which is a frontier area of research in imaging. There are many approaches to re-construct 3-D objects from their 2-D partial images. The 3-D reconstruction problem and its solution will be covered in this work by Kalman filtering.

3- Minimal Representation of Geometric Primitives

For estimation of parameters of 2-D lines, 3-D points, 3-D lines and 3-D planes, we first represent them with minimal parameters. Further the selected representation should be differentiable, so that we can employ the principles of Linear Kalman filtering.

Representation of Affine lines in R^2: A 2-D line can be represented by at least two independent parameters. The simplest form of representation of a 2-D line is given by the following expressions.

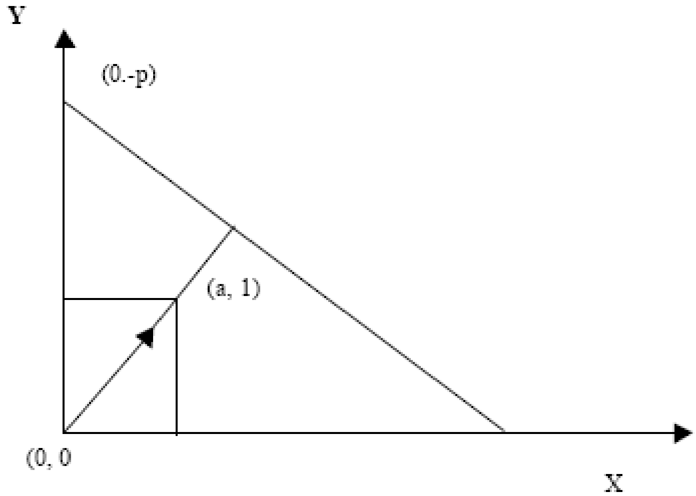
Case 1: When the lines are not parallel to the Y-axis, they are represented by

$$a x + y + p = 0 \qquad (1\text{-a})$$

Case 2: When the lines are not parallel to the X-axis, they are represented by:

$x + a\,y + p = 0$ (1-b)

In brief, the representation of a 2-D line is given by a vector (a, p), where the line passing through (0, 0) and (a, 1) is normal to the line under consideration, which also passes through the point (0, -p). This is illustrated in figure(1).

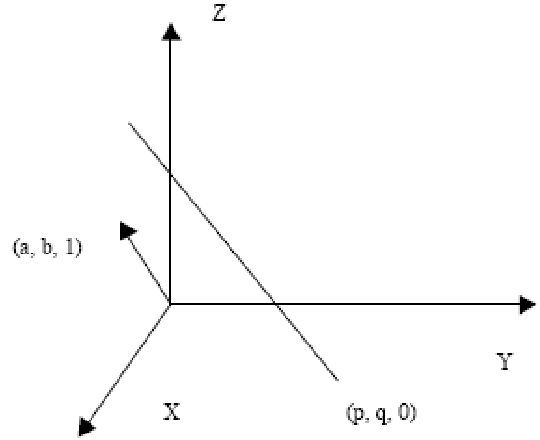


Figure(1): A 2-D line represented by (a, p).

Representation of Affine lines in R^3: The 3-D affine line can be represented minimally by four parameters given by (a, b, p, q). Other minimal representations are possible but there exists scope of ambiguity in other representations [1]. For example, when the line is parallel to the direction vector $(a, b, 1)^T$ and touches the x-y plane at $(p, q, 0)^T$, it can be represented by the following two expressions (vide figure(2)).

$x = a\,z + p$

$y = b\,z + q$



Figure(2): A 3-D representation of a line by (a, b, p, q).

This, however, is a special case. In general, we have the following three cases representing 3-D lines

Case I: Line not orthogonal to the Z axis:

$x = a z + p$ (2-a)

$y = b z + q$

Case II: Line not orthogonal to the X axis:

$y = a x + p$ (2-b)

$z = b x + q$

Case III: Line not orthogonal to the Y axis:

$z = a y + p$ (2-c)

$x = b y + q$

The representation is preferred by the following counts:

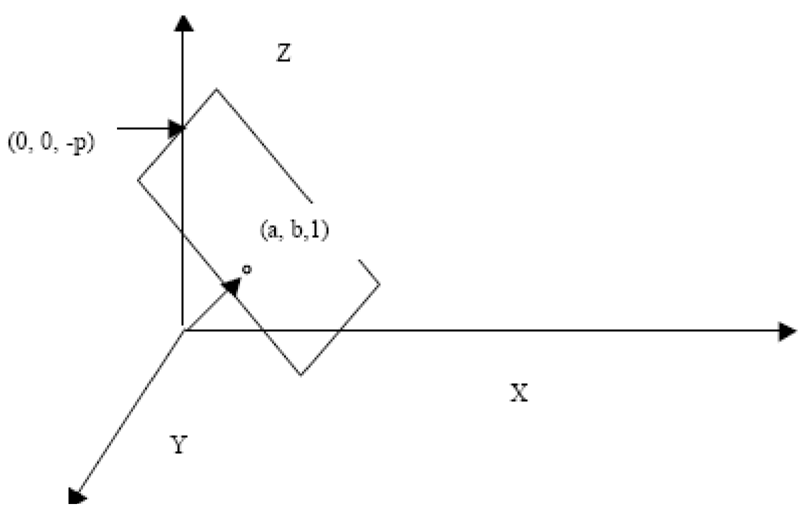
i) It imposes no constraints on the parameters (a, b, p, q).

ii) Parametric representations of the lines remain linear, which are advantageous to Kalman filtering optimization.

Representation of Affine planes in R^3: One way of representing 3-D planes is by a 3-D vector (a, b, p) such that points (x, y, z) of the plane are defined by the following equation.

$a x + b y + z + p = 0$

Here the vector $(a, b, 1)^T$ is the normal to the plane and the point $(0, 0, -p)^T$ is the point of intersection of the plane with the Z-axis.



Figure(3): A 3-D plane representation by (a, b, p).

The limitation of this notation is that planes parallel to the Z axis can not be represented. More formally, we have three cases:

Case I: Planes not parallel to the Z axis

$a x + b y + z + p=0$ (3-a)

Case II: Planes not parallel to the X axis

$x + a y + bz + p = 0$ (3-b)

case III: Planes not parallel to the Y axis

$b x + y + a z + p = 0$ (3-c)

4-Kalman Filtering

A Kalman filter is a digital filter that attempts to minimize the measurement noise from estimating the unknown parameters, linearly related with a set of measurement variables. The most important significance of this filter is that it allows recursive formulation and thus improves accuracy of estimation up to users' desired level at the cost of new measurement inputs.

Let f_i ($\mathbf{x_i}$, $\mathbf{a}$) = 0 be a set of equations describing relationships among a parameter vector a and measurement variable vector x_i ,

$\mathbf{X_I^* = x_i + l_i}$, such that E [$\mathbf{l_i}$] =0, E [$\mathbf{l_i\ l_i}^{\mathbf{T}}$] = positive symmetric matrix $\mathbf{A_1}$ and E [$\mathbf{li\ lj^T}$] =0,

$\mathbf{a_{i-1}^* = a + s_{i-1}}$, such that E[$\mathbf{s_{i-1}}$] = 0,

E [$\mathbf{S_{i-1} S_{j-1}^T}$] = positive symmetric matrix $\mathbf{S_{i-1}}$, E [$\mathbf{S_{i-1}\ S_{j-1}}$] =0.

Expanding f_i ($\mathbf{x_i}$, $\mathbf{a}$) by Taylor's series around ($\mathbf{x_i^*}$, $\mathbf{a_{i-1}}$), we find

f_i ($\mathbf{x_i}$, $\mathbf{a}$)

= f_i ($\mathbf{x_i^*}$, $\mathbf{a^*}$) + $(\boldsymbol{\sigma}\ \mathbf{f_i} / \boldsymbol{\sigma}\ \mathbf{x})$ ($\mathbf{x_i - x_i^*}$) + $(\boldsymbol{\sigma}\ \mathbf{f_i} / \boldsymbol{\sigma}\ \mathbf{a})$ ($\mathbf{a - a_{i-1}^*}$)=0.

After some elementary algebra, we find

$\mathbf{Y_i} = \mathbf{M_i\ a + w_i}$

where $\mathbf{y_i}$ = - f_i ($\mathbf{x_i^*}$, $\mathbf{a_{i-1}}$) + $(\boldsymbol{\sigma}\ \mathbf{f_i} / \boldsymbol{\sigma}\ \mathbf{a})$ ($\mathbf{a - a_{i-1}^*}$)

is a new measurement vector of dimension (pi x 1).

$\mathbf{M_i} = (\boldsymbol{\sigma}\ \mathbf{f_i} / \boldsymbol{\sigma}\ \mathbf{a})$ and

$\mathbf{W_i}$ = $(\sigma\ \mathbf{f_i} / \sigma\ \mathbf{x})$ ($\mathbf{x_i - x_i^*}$) is a measurement noise vector of dimension (pi x 1).

We also want that E [$\mathbf{w_i}$] =0 and define

$\mathbf{W_i}$ = E [$\mathbf{wi\ wi}^{\mathbf{T}}$] = $(\sigma\ \mathbf{fi} / \sigma\ \mathbf{x})$ $\mathbf{A_1}$ $(\sigma\ \mathbf{fi} / \sigma\ \mathbf{x})^{\mathbf{T}}$.

Let **Si** = E **[(ai – ai*) (ai – ai*)T]**

An attempt to minimize **Si** yields the filter equations, given by:

$$\mathbf{ai^* = a_{i-1}^* + Ki\ (yi - Mi\ a_{i-1}^*)}$$
$$\mathbf{Ki = S_{i-1}\ M_i^{\ T}\ (Wi + Mi\ S_{i-1} Mi^T) - 1} \quad (4)$$
$$\mathbf{Si = (I - Ki\ Mi\ )\ S_{i-1}.}$$

Given **S0** and **a0**, the Kalman filter recursively updates **ai**, **Ki**, **Si** until the error covariance matrix **Si** becomes insignificantly small, or all the number of data points have been submitted. The **ai** obtained after termination of the algorithm is the estimated value of the parameters.

The Kalman filter has been successfully used for determining

i) affine 2-D lines from a set of noisy 2-D points,

ii) 3-D points from a set of noisy 2-D points,

iii) affine 3-D lines from noisy 2-D points and

iv) 3-D planes from 3-D lines.

5- Construction of 3-D Points Using 2-D Image Points

The 3-D object points are mapped onto an image plane by using the principle of perspective projection. Let the 3-D object point be P having co-ordinates $(x,y,z)^T$, which is mapped onto the image plane at point (U, V, S)T.. Let T be the perspective projection matrix. Then

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

where t_{ij} is the (i, j) th element of the perspective projection matrix. Le t u= U/S and v=V/S. Now, after elementary simplification, let us assume for brevity that $t_i = (t_{i1}\ t_{i2}\ t_{i3}\ t_{i4})^T$
and P is (x, y, z)T,Also assume that
$a=(t_1 t_{14} t_2 t_{24} t_3)^T$. For a match of an image point I with an associated scene point P, we now have the following relationships between P, u and v.
$P^T t_1 + t_{14} + -u (P^T t_3 + 1) = 0$ and
$P^T t_2 + t_{24} - v (P^T t_3 + 1) = 0$.
Now suppose we have xi = (ui , vi $)^T$ and we have to evaluate **a** =$(x, y, z)^T$.
The measurement equation is given by
fi(xi , a) =0 yields

$$(t_1^i - u_i t_3^i)\, a + t_{14}^i - u_i t_{34}^i = 0$$

where t_j^i comes from perspective matrix $\mathbf{T_i}$ from camera i.
Further, $\mathbf{y_i = M_i\, a + w_i}$
where

$$y_i = \begin{pmatrix} t_{34}^i u_i * - t_{34}^i \\ t_{34}^i v_i * - t_{24}^i \end{pmatrix}$$

$$M_i = \begin{pmatrix} -(u_i t_3^i - t_1^i)^T \\ -(v_i t_3^i - t_2^i)^T \end{pmatrix}$$

$$\partial f / \partial x_i = \begin{pmatrix} -t_3^{i}\ a_{i-1} * \ -t_{34}^{i} & 0 \\ 0 & -t_3^{i}\ a_{i-1} * \ -t_{34}^{i} \end{pmatrix}$$

The following algorithm may be used for the construction of 3-D points from noisy 2-D points.

Procedure 3-D-Point-Construction (2-D image points: u,v; camera parameters: xo, yo, zo,, A,B,C)

Input: *coordinates of the image points along with the six camera parameters determined by its position (xo, yo, zo)and orientation (A,B,C) w.r.t global coordinate system.*

Output: *the state estimate **a** (3´1), along with the covariance error **S** (3´3), associated with the estimate.*

Begin

For (no. of points: = 1 to n) do

Initialize the Initial Covariance matrix **S** and the state estimate

S0←Very large initial value;

a0←Arbitrary Value preferably $[0\ 0\ 0]^T$;

For (j: =1 to no. of iterations) do

Compute the perspective matrix from the input camera Parameters;

Compute the measurement vector **y** (2×1), the linear transformation **M** (2×3) and **W** (2×2) the weight matrix

obtained after linearizing measurement equation from the input parameters at each iteration ;

Initialize all the matrices involved in matrix multiplication;

Compute gain **K** using previous **S** and **M** values ;

Compute covariance matrix **S** recursively using its value at previous iteration;

Compute the state estimate '**a**' recursively using its value at previous iteration; j:= j + 1;

End For ;

End For ;

End.

Traces of Procedure 3-D-Point-Construction

Input file for the above program: In this input file first two columns contains the (u, v) co-ordinates of the image points. Next three columns correspond to the (x0 , y0, z0) co-ordinates of the camera position and last three columns represent orientation of the camera co-ordinate system (A, B, C) w.r.t. some user selected reference co-ordinate systems. Here, A represents the pan angle of the camera, B represents the tilt angle and C represents the

skew angle of the camera w.r.t. the global co-ordinate system. The whole data set comprises of 6 blocks, each of 6 rows. The first row corresponds to the point1 from image 1, the second row for point1 from image 2 and so on for six different images. Similarly the second block is for point 2 and so on. The output datafile, on the other hand (row-wise), presents the estimated (x, y, z) points. All the input and output data files are given below.

Input file: Set1.dat

-1.8	1.7	9.0	14.0	28.5	-0.262	-1.974	0.0
-1.8	2.0	18.0	15.5	28.5	0.0	-1.974	0.0
-4.4	1.7	24.0	13.5	28.5	0.0	-1.974	0.0
-1.7	1.0	26.5	13.7	28.5	0.227	-1.974	0.0
-3.2	0.8	32.5	14.8	28.5	0.262	-1.974	0.0
-5.0	-0.5	28.0	3.0	28.5	0.0	-1.974	0.0
5.5	0.7	9.0	14.0	28.5	-0.262	-1.974	0.0
6.5	1.7	18.0	15.5	28.5	0.0	-1.974	0.0
3.8	1.4	24.0	13.5	28.5	0.0	-1.974	0.0
6.3	1.7	26.5	13.7	28.5	0.222	-1.974	0.0
4.5	1.8	32.5	14.5	28.5	0.262	-1.974	0.0
1.5	-0.7	28.0	3.0	28.5	0.0	-1.974	0.0

-1.5	4.9	9.0	14.0	28.5	-0.262	-1.974	0.0
-1.5	5.2	18.0	15.5	28.5	0.0	-1.974	0.0
-3.8	4.7	24.0	13.5	28.5	0.0	-1.974	0.0
-1.5	4.2	26.5	13.7	28.5	0.227	-1.974	0.0
-2.95	3.8	32.5	14.8	28.5	0.262	-1.974	0.0
-4.6	2.3	28.0	3.0	28.5	0.0	-1.974	0.0
5.2	3.6	9.0	14.0	28.5	-0.262	-1.974	0.0
6.0	4.5	18.0	15.5	28.5	0.0	-1.974	0.0
3.7	4.4	24.0	13.5	28.5	0.0	-1.974	0.0
6.0	4.5	26.5	13.7	28.5	0.227	-1.974	0.0
4.40	4.6	32.5	14.8	28.5	0.262	-1.974	0.0
1.7	2.0	28.0	3.0	28.5	0.0	-1.974	0.0
3.8	-1.0	9.0	14.0	28.5	-0.262	-1.974	0.0
5.4	-0.5	18.0	15.5	28.5	0.0	-1.974	0.0
3.1	0.6	24.0	13.5	28.5	0.0	-1.974	0.0
5.84	-0.44	26.54	13.74	28.54	0.227	-1.974	0.0
4.7	-0.5	32.5	14.8	28.5	0.262	-1.974	0.0
1.4	-2.15	28.0	3.0	28.5	0.0	-1.974	0.0
-2.5	0.0	9.0	14.0	28.5	-0.262	-1.974	0.0
-1.5	0.0	18.0	15.5	28.5	0.0	-1.974	0.0
-4.0	-0.2	24.0	13.5	28.5	0.0	-1.974	0.0
-0.8	-0.6	26.5	13.7	28.5	0.227	-1.974	0.0
-1.8	-0.8	32.5	14.8	28.5	0.262	-1.974	0.0
-4.0	-1.8	28.0	3.0	28.5	0.0	-1.974	0.0

Corresponding Output file: Out1.dat

13.558480	48.728981	8.969839
33.009026	46.612656	10.612383
13.892322	48.551067	-1.117031
32.706348	46.120842	3.406096
33.843052	57.854473	11.443270
13.166441	60.997398	8.871586

6-Conclusions

vision mainly deals with recognition and interpretation of 3-D objects from their 2-D images. There exist many approaches to interpret a scene from more than one image. Kalman filtering is one of such techniques.

Its main advantage is that it employs a recursive algorithm and thus can update an estimator from input data in an incremental fashion. The vertices of points in a 2-D image can be first mapped to their 3-D locations by supplying the same 2-D points from multiple images. Now, we can construct the equation of 3-D lines from 3-D points by a second stage of Kalman filtering. Lastly, we can determine the equation of the planes containing more than one line. The spatial relationships among the planes are then analyzed to determine the 3-D planer object.

7-References

Ayache, N., *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*, MIT Press, Cambridge, MA, pp. 7-298, 1991.

Das, S. S., *3D Reconstruction by Extended Kalman Filtering*, M.E. Thesis, Jadavpur University, 1999.

Dean, T., Allen, J., and Aloimonds, Y., *Artificial Intelligence: Theory and Practice*, Addison-Wesley, Reading, MA, pp. 409-469, 1995.

Luo, Fa-Long and Unbehauen, R., *Applied Neural Networks for Signal Processing*, Cambridge University Press, London, pp. 188-236, 1997.

Patterson, D. W., *Introduction to Artificial Intelligence and Expert Systems*, Prentice-Hall, Englewood Cliffs, NJ, pp. 285-324, 1990.

Pratt, W. K., *Digital Image Processing*, John Wiley and Sons, New York, pp. 201-279, 1978.

Schalkoff, R. J., *Artificial Neural Networks*, McGraw-Hill, New York, pp. 308-334, 1997.