International Journal of Scientific & Engineering Research, Volume 3, Issue 5, May-2012

ISSN 2229-5518

Transformation Technique

M.Venkatesh, P.Vijayakumar

Abstract— of the we increase the scope of extracting information from images. Different perspective view is one of the key requirements in WSN image based monitoring systems. This paper

using a simple pin hole camera model. To develop a real time product and evaluate real time performance of this technique, we implemented it on a 32-bit floating point VLIW TMS320C6713 DSP processor.

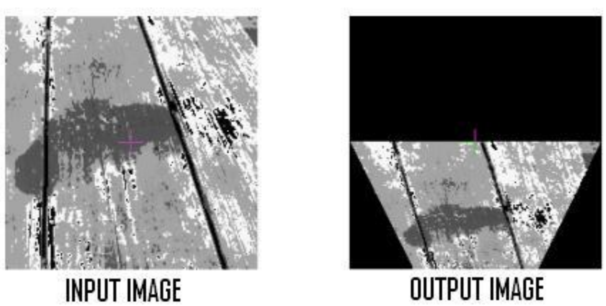
Index Terms— g system, digital image processing.

◆

1 Introduction

B Eye View transformation technique is to generate a top view perspective of an image as shown in Fig. 1. This technique can be classified under digital image processing as geometrical image modification. Basically the ed into three steps .First we have to represent the image in a shifted coordinate system, next perform rotation of image, and then project the image on a two dimensional plane. The block diagram for transformation is given in Fig. 2.

Image in general is represented as two dimensional matrix, where each location represents a pixel value which is between 0 to 255 (for a 8-bit grayscale image). We are interested in the location of each pixel which will be basically relocated to a new position.



eye view transform

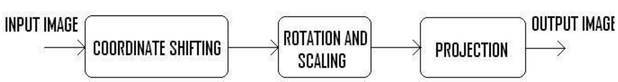


Fig. 2 Block Diagram

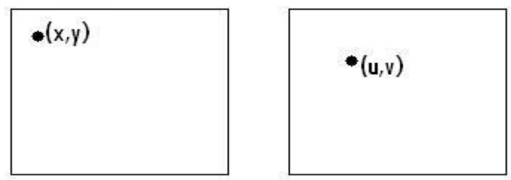


Fig. 3. Change in pixel position

2 Coordinate Shifting

Image is generally represented as two dimensional plane i.e. x and y plane. So if the size of an image is 100 × 100 then each pixel location is represented as shown in Fig. 4.

To perform rotation of image first we need to apply a change in coordinate position such that the image centre is located at (0, 0), thus in a 100 × 100 image starting point is (-50, -50) and ending point at (49, 49) as shown in Fig. 5.

If (x, y) represent the coordinates of an image then shifted coordinate is given by (X, Y) as shown in (1) and (2).

$$X = x - (\text{IMAGE SIZE} / 2) \quad (1)$$

$$Y = y - (\text{IMAGE SIZE} / 2) \quad (2)$$

ISSN 2229-5518

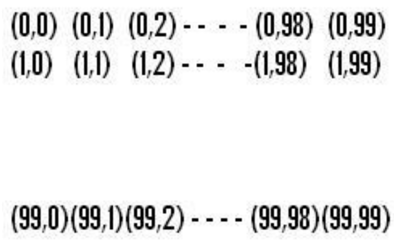


Fig. 4. Representation of image pixel locations

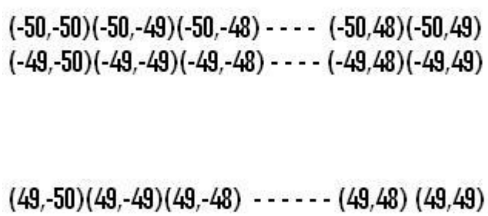


Fig. 5. Coordinate change of input image

3 ROTATION AND SCALING OF AN IMAGE

The rotation of an image can be performed by simple matrix multiplication operation, if (X, Y, Z) is the location of the pixel in input image then (p, q, r) represents the location for the rotated image. The Z is introduced as third coordinate, representing the image in three dimensions (3D). The initial value of Z can be assigned as zero for each pixel change followed by projection operation. The mathematical representation of rotation operation is given by (3).

$$\begin{bmatrix} p \\ q \\ r \\ 1 \end{bmatrix} = \mathrm{R} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

Where R is 4x4 rotation matrix [1]. We can perform rotation around x, y and z axis by using different rotation matrix given by (4), (5) and (6). Where Θ, α and β are the

Rz matrix.

$$\mathrm{R_x} = \begin{bmatrix} cos\,(\Theta) & \sin(\Theta) & 0 & 0 \\ -\sin(\Theta) & \cos(\Theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\mathrm{R_y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$\mathrm{R_z} = \begin{bmatrix} cos(\beta) & 0 & -\sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

If the camera is mounted at 45^0 angle to ground and we try to rotate the image by 45^0 then what we get is the top view of the given image as shown in Fig. 6. Hence our

manual observation or by knowing the angle at which camera is mounted, this process can also be automated by techniques represented in [6]. After rotation is performed the height of the image decreases hence we need to stretch the height of the image, which is done by using matrix SH as in (7). If Sx = 2 then the height will become twice that of original.

$$\mathrm{S_H} = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

We can combine (6) and (7) to get a single equation for ransform matrix B as given by (8).

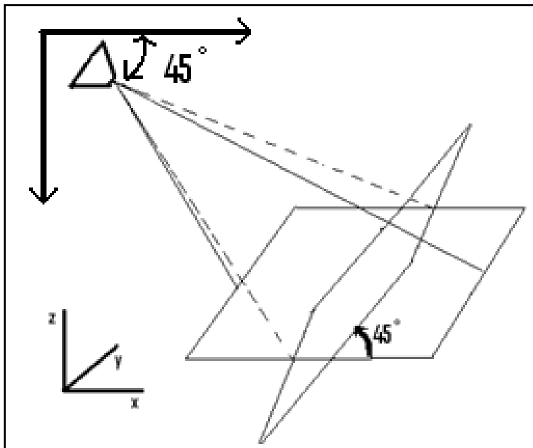


Fig. 6. Rotation of image on z axis

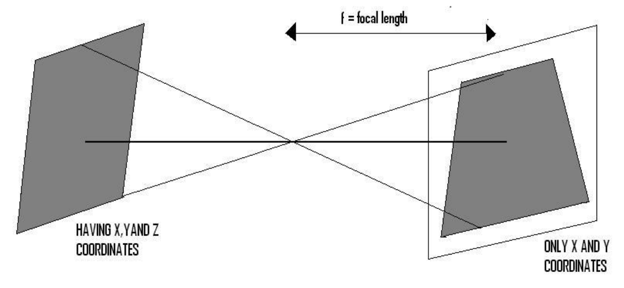


Fig. 7 Projection into a 2D image plane

$$B = \begin{bmatrix} Sx(cos\ (\beta)) & 0 & -\sin(\beta) & 0 \\ 0 & Sy & 0 & 0 \\ \sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

So the final equation that we get is given by (9).

$$\begin{bmatrix} p \\ q \\ r \\ 1 \end{bmatrix} = B \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (9)$$

4 Projection Of Image On Two Dimensional Plane

The image has to be again represented in 2D plane for which the image has to be projected from the 3D plane to 2D plane like a simple pin hole camera model [1].

A simple projection formula can be written using similar triangles, where (p, q, r) are the positional coordinates in 3D plane and (u, v) are the output image coordinates as shown in (10) and (11). Here f represents focal length according to simple pin hole camera model. Focal length can have any arbitrary value.

$$u = \frac{f\,p}{f-r} + (\text{IMAGE SIZE} / 2) \quad 10)$$

$$v = \frac{f\,q}{f-r} + (\text{IMAGE SIZE} / 2) \quad (11)$$

The positional coordinate of the output image will have integer values hence the coordinate shifting is applied to get the coordinates in whole number.

5 Experimental Results

For embedded based applications we implemented the technique on TMS320C6713 DSP Starter Kit (DSK) developed jointly with Spectrum Digital is a low-cost development platform. The kit uses USB communications for true plug-and-play functionality. Code Composer Studio IDE is used where the code is written in simple C language. We got the following results on DSK as shown in Fig. 8 and Fig. 9. For different size of the image, the time required for processing has been calculated as shown in table (1). So if a video has to be generated for about 15 frames /sec then it would require processing time to be around 0.066s per frame. Hence a real time video can be generated by using an image size of 100 x 100 or less. For larger image size we would require a higher clock processor.

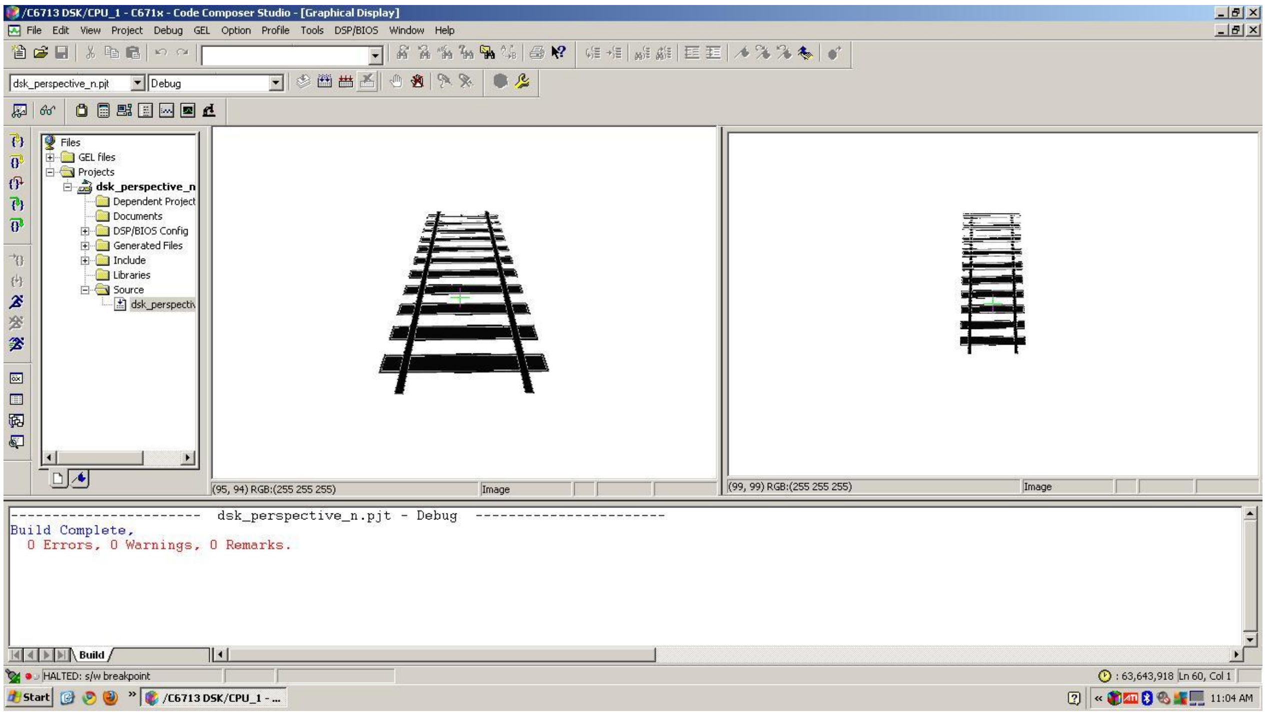
TABLE 1
Time Complexion Analysis

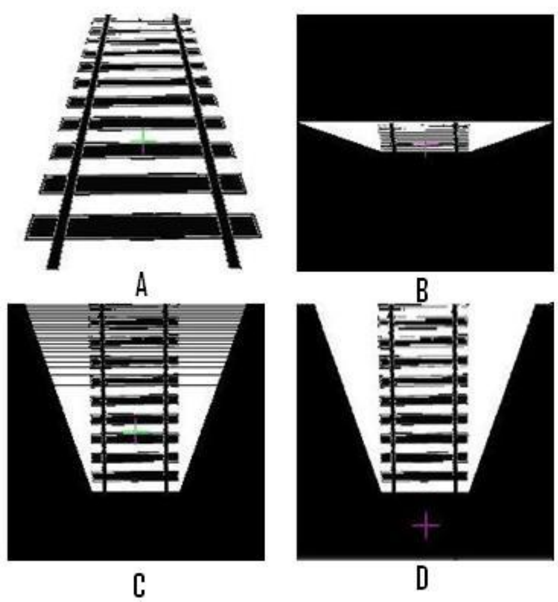
IMAGE SIZE	CLOCK PULSE COUNT	TIME (225 MHz processor)
32 × 32	1,490,356	6 ms
100 × 100	14,723,872	65 ms
200 × 200	63,643,918	282 ms

6 Conclusion

We conclude that the technique is practical feasible technique and can be applied to different applications. This technique applies a simple rotation operation and follows a simple pin hole camera model technique, hence requires a very low processing time hence can be applied for low cost application products.

In future we would like to show applications based on this technique and further reduce the computation cost.





x = 1 (B), x = 8 (C), Output image after filling gaps (D)

REFERENCES

[1] 2010 pp. 387-410
[2] LinBo Luo, InSung Koh, SangYoon Park, ReSen Ahn, JongWha -hardware cooperative implementation of -eye view system for camera-on- -NIDC 2009
[3] Lin-Bo Luo, In-Sung Koh, Kyeong-Yuk Min, Jun Wang, Jong- - -eye View System for Camera-on-
[4] H. Kano, Keisuke Asari, Yo IEICE Trans. Inf. & Syst., vol. E91-D, pp. 1893-1898, 2008.
[5] Yi-Yuan Chen, Yuan-Yao Tut, Cheng-Hsiang Chiul , and Yong- and Intelligent Transportation System 2009
[6] pective Correction Method For Camera-