

Design and Simulation of Robot Vision System Using Simulink

Nagarathna H S, Sushma P S, Ramesh Kumar K R

Abstract--Vision-based technique is an effective approach for robot navigation in the close region. This paper shows development of video processing algorithm for robot vision system to process the online 2-D videos and images. The configuration and some important characteristics of this robot vision system, which not only complete the work of video capturing but, can also process images and videos using different DSP algorithms in real-time. For this application was used the Matlab's toolbox of Simulink to implement and simulate Video processing algorithms.

Keywords---DSP, Video processing, MATLAB, Simulink. Robotics.

1. INTRODUCTION

For local navigation, vision-based techniques are an effective way to solve the close-range navigation problem. Vision-based navigation comprises a variety of tasks that include pattern recognition, target identification, etc. There are many applications in space such as autonomous vehicles and planetary rovers based on this. In practical applications, the optical sensor and image processing algorithms are also critical factors to affect the accuracy of navigation.

The traditional robot vision system is useful to image processing only because of image grabber. But for video signal processing is not possible with this system. The image quality of the old robot vision system is also less, because of low level image processing algorithms are used to develop the image filtering and image segmentation. To improve and enhance the applications of earlier system intelligent frame grabber is designed to capture online video signal and processing video signal to produce high quality of videos and images.

This paper describes the development of video processing algorithms for robot vision system. Since MATLAB and SIMULINK having beautiful feature of by integrating these two we can analyze, simulate, and revise our models in either environment at any point. So this system is developed by implementing high level video processing algorithm using MATLAB Simulink.

The implementation of video processing algorithms for intelligent frame grabber is the backbone for robot vision system using MATLAB SIMULINK. The implementation is carried out by following sub implementations,

*Designing the Simulink model for converting color images into a binary image.

*Designing the Simulink model for Spatial filter.

*Designing the Simulink model for the 2D-Gaussian filter.

*Designing the Simulink model for Sobel operator.

2. BLOCK DIAGRAM OF ROBOT VISION SYSTEM

Fig.1 shows the block diagram of the robot vision system. An intelligent frame grabber is the main building block of the robot vision system. This system is based on DSP.

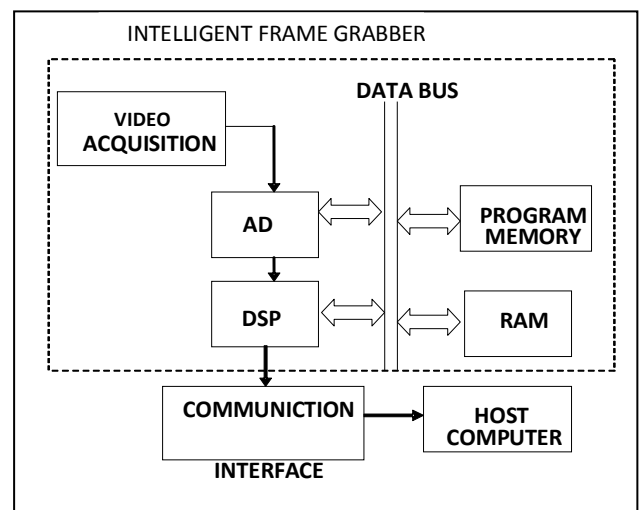


Fig. 1 Block Diagram of robot vision system

The intelligent frame grabber can not only play the role of video capturing but can also do the work of high speed video processing according the command from its host computer so that the computing burden on the host computer can be reduced greatly.

Video acquisition device captures and record the video from external source. That video signal fed to the AD converter to digitize the analog video signal. Then that digitized signal is sent to DSP which is the heart of the intelligent frame grabber, the main task of this block is applying the high level video processing algorithms to capture the videos to produce high quality of videos and images.

There are two types of memories, program memory and frame memory. The program memory is used to store the

DSP programs and RAM memory consists of frame buffers to save the video frame data temporally for the DSP. Finally the processed video frames are transmitted to host computer by serial communication. Since host computer receives processed video signal, there is no need for applying algorithms to get good quality of the videos. Hence the computational burden to the host computer is reduced greatly.

3. VIDEO PROCESSING ALGORITHM IMPLEMENTATION

The video processing algorithms for DSP block are developed in following sequence as shown in Fig.2

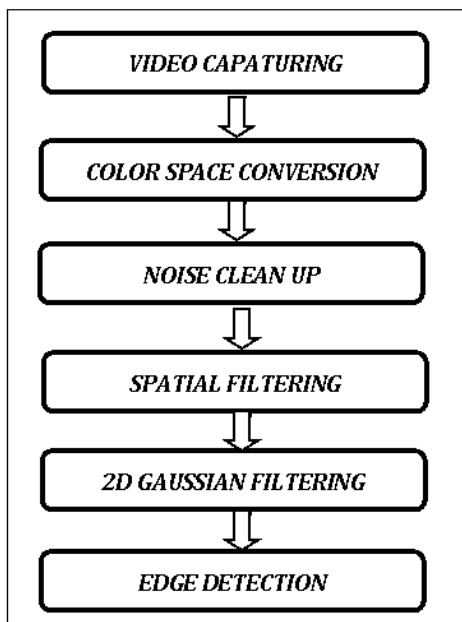


Fig. 2 Algorithms for robot vision system

3.1 Video Capturing

Video capturing is the process of acquiring the live video data streams from video acquisition devices as shown in Fig.9.1. For Video Device block is used to connect external video acquisition devices. This block is taken from Image Acquisition Toolbox from MATLAB Simulink Library. The From Video Device block as shown in Fig.5, lets to acquire images and video data streams from image acquisition devices, such as cameras and frame grabbers, in order to bring the image data into a Simulink model. The block also used to configure and preview the acquisition directly from Simulink.

3.2 Color Space Conversion

It converts color images to a binary image as shown in Fig 9.2. A true color image is an image in which each pixel is specified by three values one each for the red, blue, and green components of the pixel's color. That true images are converted to binary images, then each pixel assumes one of

only two discrete values 1 or 0. A binary image is stored as a logical array.

The Simulink MATLAB block used to perform conversion is Color Space Conversion is shown in Fig.5. The Color Space Conversion block converts color information between color spaces. Use the Conversion parameter to specify the color spaces we are converting between. The choices are R'G'B' to Y'CbCr, Y'CbCr to R'G'B', R'G'B' to intensity, R'G'B' to HSV, HSV to R'G'B'.

3.3 Noise Cleans Up

Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges as shown in Fig 9.3. The value of an output pixel is determined by the median of the neighboring pixels. The Median Filter, Simulink block used to perform this operation. This is taken from Filtering / Analysis & Enhancement from MATLAB Simulink Library. The Median Filter block replaces the central value of an M-by-N neighborhood with its median value. If the neighborhood has a center element, the block places the median value there. If the neighborhood does not have an exact center, the block has a bias toward the upper-left corner and places the median value there.

3.4 Spatial Filtering

The spatial filter operation is to use a fixed size window to scan over an image. The output value of the pixel is estimated by the convolution operation on the input pixels within a window of the given size. Mathematically, a convolution operation in image processing algorithm is represented by the following equation:

$$g(x,y) = \frac{1}{s} \sum_{i=-a}^a \sum_{j=-b}^b w(i,j) f(x+i,y+j) \quad -(1)$$

Where f is the input image, w is the filter mask, g is the convolved output image and s is the scaling factor.

Spatial filtering operation implementation is performed by using two Simulink model blocks, Convert Image to Single and 2-D FIR Filter. The Image Data Type Conversion block changes the data type of the input to the user-specified data type and scales the values to the new data type's dynamic range and scales the values to the new data type's dynamic range. To convert between data types without scaling, use Simulink conversion block. 2-D FIR Filter, it performs 2-D FIR filtering of input matrix I using filter coefficient matrix H . It is taken from Filtering from MATLAB Simulink Library. Spatial Filter design in Fig.5 and output is shown in Fig 9.4

3.5 2-D Gaussian Filter

A Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function as shown in Fig.9.5 Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales

Filters with Gaussian functions are very useful to easily identify the shapes of objects in an image, and suppress the noise in the image effectively. In 2-D space, a circularly symmetric Gaussian has the form of

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad \text{-- (2)}$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. Values from this distribution are used to build a convolution matrix which is applied to the original image. This result in a blur that preserves boundaries and edges better than other, more uniform blurring filters.

- **Hardware implementation**

Fig. 3 shows [1] the design of a generic 2-D Gaussian filter. It consists of line buffers and processing elements (PEs) array. Any 2-D convolution operation involves passing a 2-D template over an image, and carrying out estimations at each template position. Line delays are necessary in order to assure that each pixel is fed only once to the DSP. These line delays synchronize the supply of input pixel values to the processing elements. The incoming pixels are shifted through the line buffers that create delayed lines. The size of the buffer depends on the number of pixels in each line of the frame. These delay lines feed the PEs array simultaneously with pixels from all the relevant video lines. At each filter node, the pixel is multiplied with the appropriate filter coefficients. All the multiplier results are added together in the adder tree to produce the filter middle point output result.

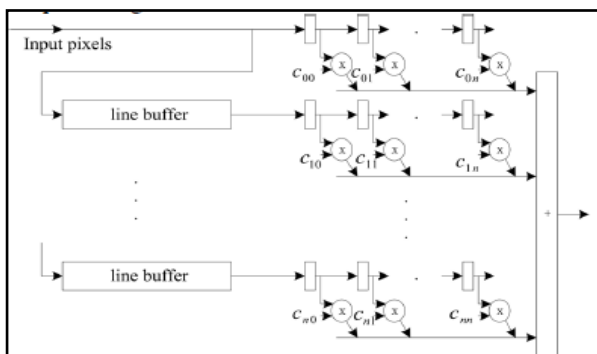


Fig.3[1] Hardware implementation of Gaussian Filter

In Simulink Gaussian filter is developed by using Line Buffer, Delay line, Matrix multiplier and Matrix adder blocks as shown in Fig.6. The line Buffer block used to store the frames of videos. Videos obtain from Gaussian filter is

much sharper than other filters because of continuous noise reduction in videos as shown Fig.9.5.

3.6 Implementation of Sobel operator

- **Edge Detection**

Edge detection refers to the process of identifying and locating sharp discontinuities in a video as shown in Fig 9.8. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

Gradient: The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

Laplacian: The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location.

- **Sobel Edge Detection**

Edge detection is another fundamental operation in image processing applications. Gradient operators such as Roberts, Sobel, and Prewitt are the mainly used operators for edge detection. In the case of the Sobel operator, it performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. The operator consists of a pair of 3x3 convolution kernels as shown in Fig.4 shown one kernel is simply the other rotated by 90°.

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations.

-1	0	+1
-2	0	+2
-1	0	+1

G_x

-1	-2	-1
0	0	0
+1	+2	+1

G_y

Fig. 4 Sobel convolution kernels

The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these G_x and G_y). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{|G_x|^2 + |G_y|^2} \quad \text{---(3)}$$

Typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y| \quad \text{---(4)}$$

This is much faster to compute. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \tan^{-1}(G_y/G_x) \quad \text{---(5)}$$

Sobel Edge detection can be designed by using Edge Detector block in MATLAB Simulink as shown in Fig.7. The Edge Detection block detects the edges in an input video by approximating the gradient magnitude of the video. The block convolves the input matrix with the Sobel, Prewitt, or Roberts's kernel depending on selection of edge detector. The output obtained from the block is, the two gradient components of the video, which are the result of this convolution operation. The output obtained from the Sobel edge detector is as shown in Fig 9.8.

3.7 Video Stabilization

Video Stabilization is an important task in Video and Image Processing, to take off the effect of camera motion from a video stream. In the first video frame, the model identifies the target to track as shown in Fig 9.9. It also creates a dynamic search region, whose position is estimated by the last known target location. The model only searches for the target within this search region, which reduces the number of computations required to find the target. In each subsequent video frame, the model determines how much the target has moved relative to the previous frame. It uses this information to remove unwanted translational camera motions and generate a stabilized video.

The model uses the SAD block as shown in Fig.8, to move the target over the search region and computes the sum of absolute differences (SAD) at each location. The location with the lowest SAD value corresponds to the location of the target in the video frame. Based on the location information, the model computes the displacement vector between the target and its original location. The Translate block in the Stabilization subsystem uses this vector to shift each frame so that the camera motion is removed from the video stream.

4. ALGORITHM IMPLEMENTATION USING MATLAB SIMULINK FOR ROBOT VISION SYSTEM

This section deals with the design of the robot vision system algorithm using MATLAB Simulink .

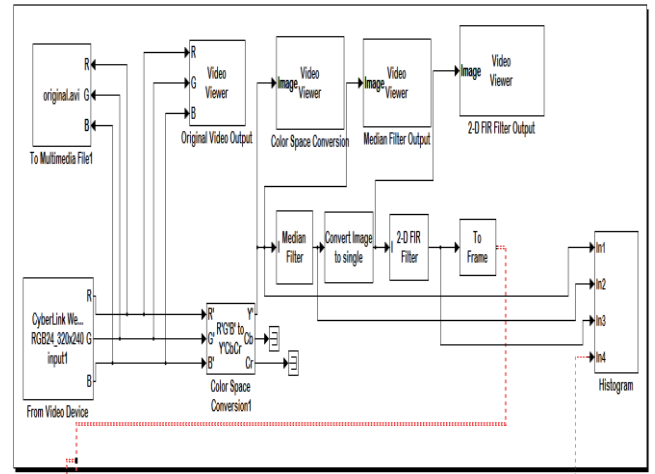


Fig. 5 Video acquisition and Spatial filtering operation

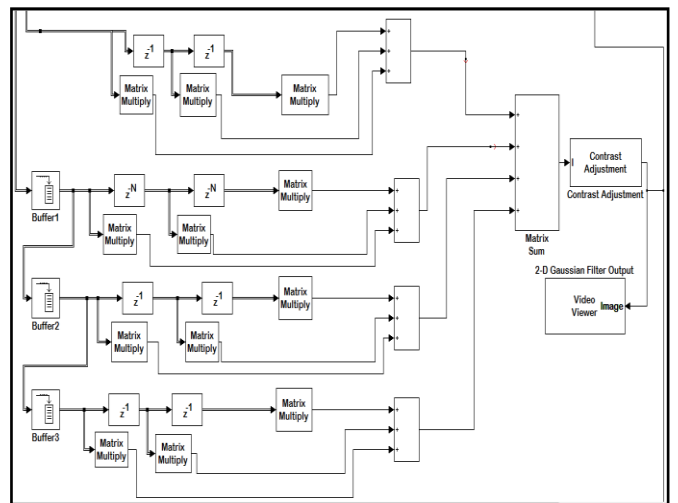


Fig.6 Gaussian Filter design using MATLAB Simulink

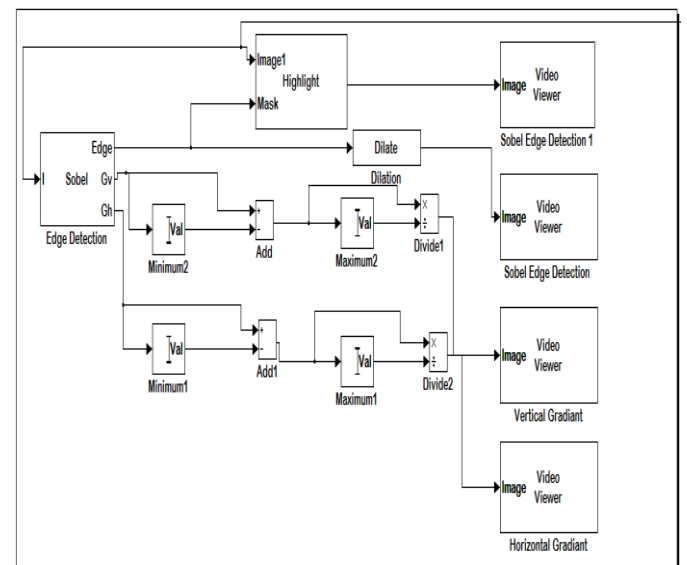


Fig. 7 Sobel edge detection using MATLAB Simulink

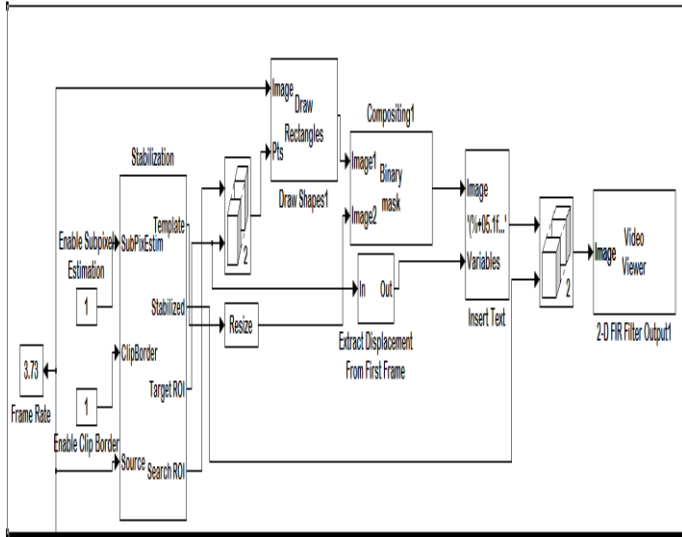


Fig. 8 Video Stabilization using MATLAB Simulink

5. EXPERIMENTAL RESULT

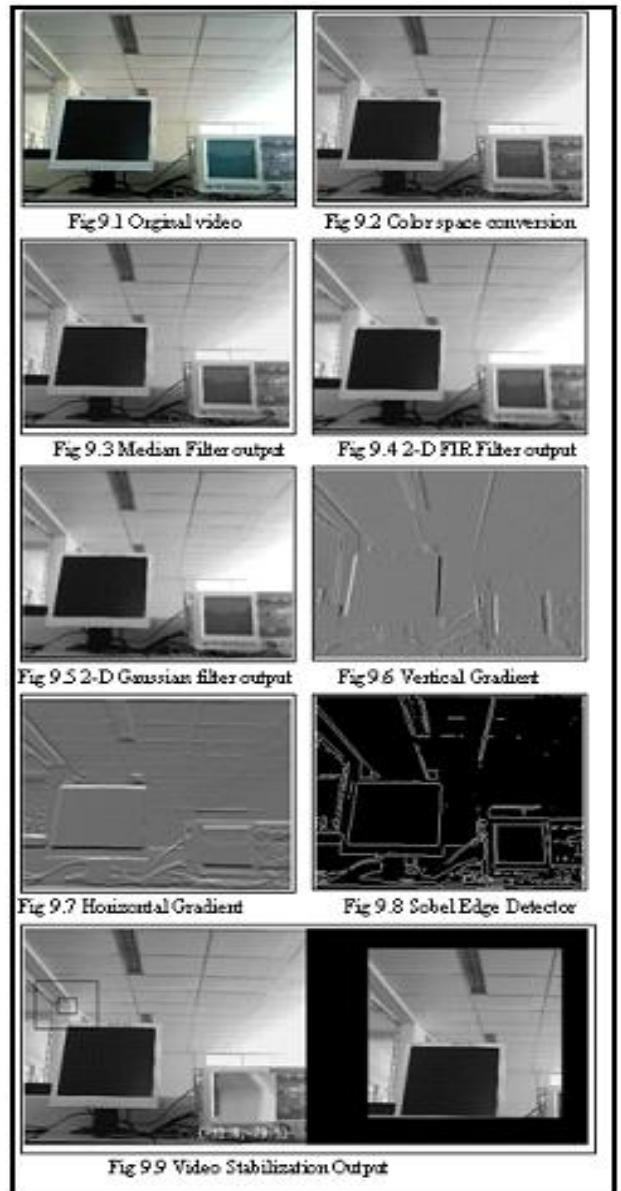


Fig. 9 Experimental results of the proposed robot vision system using MATLAB Simulink

➤ Histogram

The Histogram block computes the frequency distribution of the elements in a vector input or the elements in each channel of a frame-based matrix input.

The Sobel edge detector is very sensitive to noise. By reducing that noise we can obtain the best quality videos and images. The histograms obtained from different filtering operations are as shown in Fig. 10.1 to 10.4. From the graph it is observed this system produces the best quality of videos.

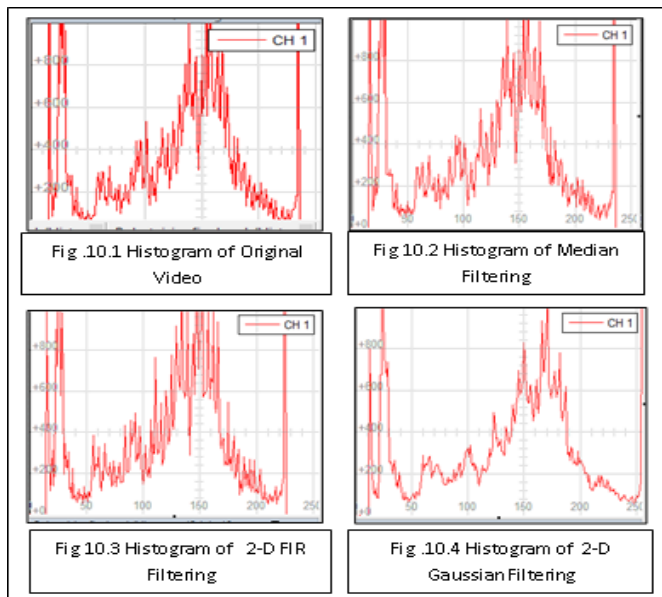


Fig. 10 Histogram graphs of filter

6. CONCLUSION

This system provides good quality of images compare to the traditional robot vision system. Since, MATLAB and SIMULINK are integrated and thus we can analyze, simulate, and revise our models in either environment at any point. This system not only processes the 2-D color image but also process and analyze the online video signal. Edge detection is an important work for object recognition and is also an essential pre-processing step in image segmentation.

These edge detection operators can have a better edge effect under the circumstances of obvious edge and low noise. The Sobel edge detector is differencing and smoothing images. The main advantage of this device is it detects parts of the edges in the original image and suppresses the noise by spatial filtering and Gaussian Filtering.

REFERENCES

- [1] Peng Lu, Kui Yuan, Wei Zou, "A High Performance Low Power Consumption Robot Vision System", *IEEE Computer Society*, vol.5,pp.171-175,2007.
- [2] Sittte, JWinzer, P," Methodic Design of Robot Vision Systems", *IEEE Transactions on Mechatronics and Automation*, vol. 11, pp. 1758 - 1763, 5-8 Aug. 2007.
- [3] Armando Sousa , O Sousa , Paulo Costa , António Moreira," Variable Resolution Vision System in Mobile Robotics",vol,17(2),pp.75-77,2007
- [4] N. D. Guilherme and C. K. Avinash, "Vision for Mobile Robot Navigation: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 237-267, 2002.
- [5] W. James MacLean," An Evaluation of the Suitability of FPGAs for Embedded Vision Systems", *IEEE Transaction on Speech, and*

Signal Pro

- [6] Larry Matties, Mark Maimone, "Computer Vision on MARS", *International Journal of Computer Vision* 75(1),pp. 67-92, 2007