

Efficient Hardware Implementation of Real Object Tracking System on FPGA

Abhishek Sinha¹, Sachin Bandewar²

¹Mtech Scholar, ²Assistant Professor, Department of EC, SSSCE, RKDF University, Bhopal, M.P, India

¹abhi.sinha2050@gmail.com, ²sachin.bandewar9@gmail.com

Abstract- Various attempts have been made to implement human vision in real-time computer vision. In most cases, however, this capability has been found to be too complex to implement on a practical systems. In short, human vision capability looks like a simple system but in reality it is difficult to model. One of the difficulties is the computational complexity required to process large amounts of data in real-time, and consequently, the design of real-time object tracking using vision remains a challenging problem. However, in this paper, proposal and implementation of hardware architecture for real-time object tracking system. In this paper Mean-Shift algorithm is chosen for implementation. Firstly implementation of this algorithm on MATLAB and simulated in different environments to verify the performance. After approving its robustness FPGA implementation of the the algorithm on hardware is done.

Keywords -object tracking; Mean-Shift Algorithm ;FPGA;

INTRODUCTION: Computer vision can be defined as a methodology that provides vision capabilities to a machine. Various attempts are being made in the computer vision industry to model human vision and to apply the same techniques to machine vision. In most of these cases researchers have had difficulty with the complexity of implementing such a capability on practical systems. These implementation problems are due to various constraints, including the lack of needed computational capability to process large amounts of data in a real-time, object motion, noise, complex

background and the inability to model human vision exactly. Real-time object tracking remains a challenging problem in the computer vision industry and hence there is a huge interest on the market to make technical equipment “smart” and “self learning”[12].

Research Objective

The objectives of the current dissertation research are

1. Verification of tracking results by software implementation of Mean-Shift Algorithm on MATLAB.

2. To propose a novel hardware Architecture for Mean-Shift algorithm. The algorithm should be easy for hardware implementation.

To confirm experimentally soundness of the proposed hardware Architecture implemented through Hardware Description Language.

Mean-Shift Algorithm

We assume in the sequel the support of two modules which should provide (a) detection and localization in the initial frame of the objects to track targets [5] and (b) periodic analysis of each object to account for possible updates of the target models due to significant changes in color.

Color Representation

Target Model :- Let $\{x_i^*\}_{i=1\dots n}$ be the pixel

locations of the target model, centered at zero. We define a function $b:R^2 \rightarrow \{1\dots m\}$ which associates to the pixel at location x_i^* the index $b(x_i^*)$ of the histogram bin corresponding to the color of that pixel. The probability of the color u in the target model is derived by employing a convex and monotonic decreasing kernel profile k which

assigns a smaller weight to the locations that are farther from the center of the target. The weighting increases the robustness of the estimation, since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or background. The radius of the kernel profile is taken equal to one, by assuming that the generic coordinates x and y are normalized with h_x and h_y respectively. Hence, we can write [2]

$$\hat{q}_u = C \sum_{i=1}^n k \left(\|x_i\|^2 \right) \delta [b(x_i^*) - u] \quad (1)$$

Where δ is Kronecker delta function.

$$\delta(a) = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The standardization constant C springs by imposing the condition $\sum_{u=1}^m \hat{q}_u = 1$, from where

$$C = \frac{1}{\sum_{i=1}^n k \left(\|x_i^*\|^2 \right)}, \quad (3)$$

Since the summation delta functions for $u=1 \dots m$ is equal to one.

Target Candidates: Let x_i $i = 1 \dots n_h$ be the pixel locations of the target candidate, centered at y in the current frame. Using the same kernel profile k , but with radius h , the probability of the color u in the target candidate is given by

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k \left(\left\| \frac{y-x_i}{h} \right\|^2 \right) \delta [b(x_i) - u] \quad (4)$$

where C_h is that the standardization constant. The radius of the kernel profile determines the number of pixels (i.e., the scale) of the target candidate. By

imposing the condition that $\sum_{u=1}^m \hat{p}_u = 1$ we obtain

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k \left\{ \left\| \frac{y-x_i}{h} \right\|^2 \right\}}, \quad (5)$$

Note that C_h doesn't rely upon y , since the picture element location x_i is organized during a regular lattice, y being one in every of the lattice nodes. Therefore, C_h may be recalculated for a given kernel and totally different values of h . [2]

Kernel

- Flat Kernel

$$k(x) = \begin{cases} 1 & \text{if } \|x\| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

- Gaussian Kernel

$$k(x) = \exp \left(-\|x\|^2 \right) \quad (7)$$

- Epanechnikov Kernel

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1-x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Where d is that the range of dimension and C_d is that the space of a unit circle in d dimension.

MATLAB Implementation:



Figure 1 Helicopter is completely track



Figure 2 Car is completely track

HDL Implementation

We have taken several different videos with different resolutions and different environmental condition. We have applied the algorithm and results are shown below. We have compared every video taking two bin sizes 16 and 8 and also calculated false detection rate and frame rate

Top Level Architecture for Object Tracking

The Object Tracking deals with tracking of objects which are of interest. The algorithm followed for implementation of tracking is Mean-shift. Mean-shift uses Histogram and Bhattacharyya coefficient for tracking the position of object in subsequent frames of videos by taking the fact into much in continuous frames.

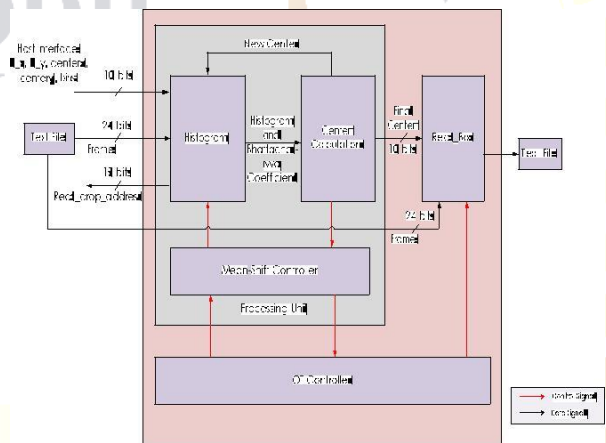


Figure 3 Mean-shift algorithms for Object Tracking [5]

The above is the top level architecture which we have proposed for the hardware implementation of Mean-shift algorithm for Object Tracking. To provide a RGB image as an input to this architecture we have converted an image into a text file representing R,G and B values of each pixel in hexadecimal format[5]. Each pixel is read by histogram block as a frame. A frame is a 24-bit wide R,G,B values, 8-bits each, of a pixel

representing intensity values of a pixel of a colored image. Host interfaces are inputs given by user in order to specify the object to be tracked in an image

Histogram

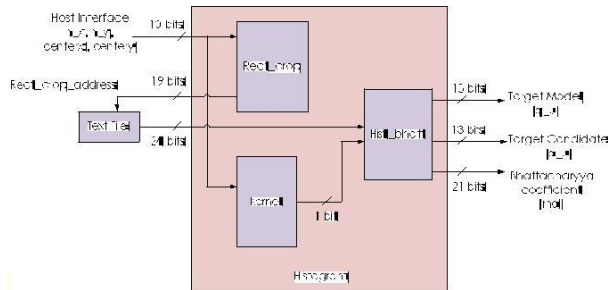


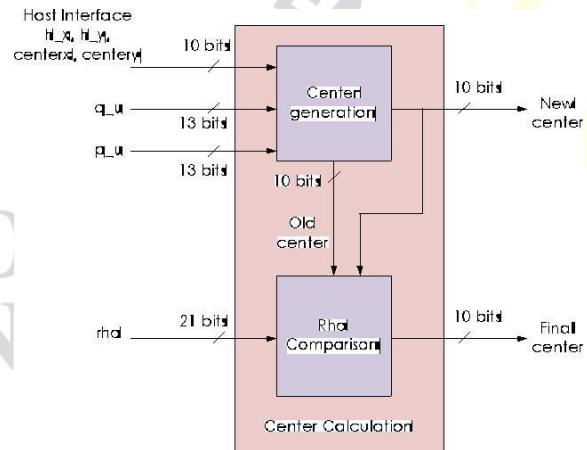
Figure 4 Histogram [5]

Sub modules of Histogram module are – Rect_crop, Kernel, Hist_bhatt. Rect_crop sub module is responsible for generating address of pixels of object of interest using (5.1). As we have explained early that for processing of algorithm we do not require complete image, rather we just want only object of interest. So we need to crop our image and extract object of interest. For that we have designed an equation which calculates addresses of only those pixels related to crop image. [5]

Where x_1, x_2, y_1 and y_2 are corner pixel location of cropped image. When Rect_crop calculates address of the desired pixel at the same time kernel sub module calculates value of the kernel for the same pixel. For our implementation we have taken Flat Kernel (3.6). Flat Kernel is either ‘0’ or ‘1’ so only 1-bit is required. In Hist_bhatt sub module histogram for the desired object and Bhattacharyya Coefficient are calculated. Histogram is stored in histogram buffer. Size of the histogram buffer depends on the number of bins provided via host interface input. In our case we have taken it 8 bins, so the size of the histogram buffer would be $\text{bins} \times \text{bins} \times \text{bins} = 512$ locations. When pixel intensity value and kernel value reaches to the Hist_bhatt sub module, address for the histogram buffer can be calculated

by pixel intensity value. In order to calculate histogram kernel generated is added to the location addressed by Hsit_bhatt sub module in histogram buffer.

Center Calculation



Architecture because it predicts the next possible center which is stored as new_center. In Center Generation sub module weight for each pixel of Object of Interest is calculated. Then every weight is multiplied with normalized pixel location in order to predict new_center.

Rho Comparison is the sub module in which Bhattacharyya coefficient calculated by Histogram module at old_center and new_center is compared. Comparison tells us which center is the correct center and it also tells us that shall we go for iteration loop or not. So from the Rho Comparison sub module, exact center is finalized which is stored as final_center and given to the Rect_box module to draw the rectangular box around tracked moving object. Then Rect_box module writes the image pixel values into a output text file.

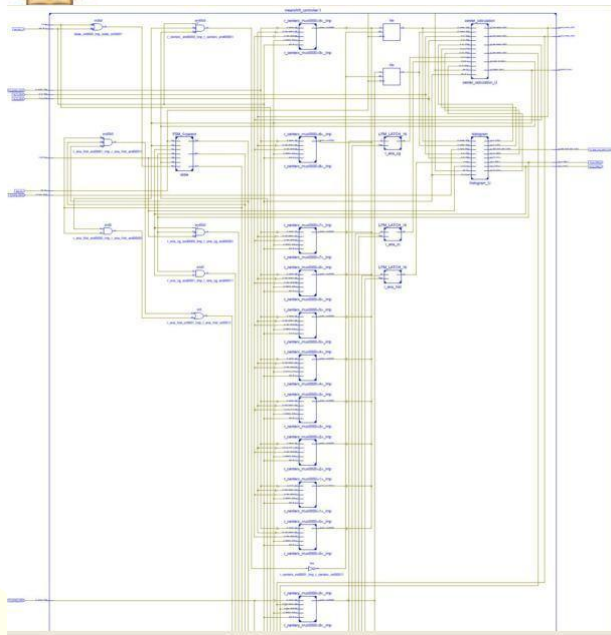


Figure 6 RTL Schematic of Top Module

Resources	Top Module
Slice Registers	20610/28800 (71%)
Slice LUTs	13028/28800 (45%)
IOs	131/480 (27%)
Block RAM	3/60 (5%)

Table 2 Device Utilization Report for architecture

CONCLUSION

The objective of our work was to implement hardware design for real-time object tracking. Several algorithms for detection and tracking of moving objects have been proposed worldwide. Most of the researchers implemented software model of all available algorithms but only few of them attempted for their hardware model. Inspiring by wide range of application and excellent performance results presented by other researches we have chosen Mean-Shift algorithm for tracking. In first phase we implemented software model. We verified the robustness, capability of tracking object under partial occlusion, intensity variation and in random environment. In second phase we have implemented hardware architecture for the design and implemented hardware model using HDL.

Resources	Top Module
RAMs	1
Multipliers	6
Adder/ Subtractor	60
Counter	5
Registers	7427
Latches	1041
Comparators	28
Multiplexers	5
Xors	2

Table 1 Synthesis Report for architecture

REFERENCES

- [1] Su Liu, Alexandros Papakonstantinou, Hongjun Wang¹, Deming Chen 978-0-7695-4448-9/11 2015 IEEE
- [2] J. Aloimonos and A. Badyopadhyay. "Active Vision." In *IEEE intl Conf. on Computer Vision*, pp. 35-54, 2010.
- [4] Comaniciu D, Comaniciu, Ramesh V, and Meer P, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," In Proc. Of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 142-149, 2007.
- [5] VEENMAN, C., REINDERS, M., AND BACKER, E. 2001. Resolving motion correspondence for densely moving points. *IEEE Trans. Patt. Analy. Mach. Intell.* 23, 1, 54-72.
- [6] Ali U, Malik M and Munawar K, "FPGA/SOFT-processor based real time tracking system," In *Fifth Southern Conference on Programmable logic*, pp 33,2009.
- [7] COMANICIU, D., RAMESH, V., AND MEER, P. 2003. Kernel-based object tracking. *IEEE Trans. Patt. Analy. Mach. Intell.* 25, 564-575.
- [8] YILMAZ, A., LI, X., AND SHAH, M. 2004. Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Patt. Analy. Mach. Intell.* 26, 11, 1531-1536.
- [9] BALLARD, D. AND BROWN, C. 1982. *Computer Vision. Prentice-Hall.*
- [10] ZHU, S. AND YUILLE, A. 1996. Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Trans. Patt. Analy. Mach. Intell.* 18, 9, 884-900.
- [11] ELGAMMAL, A., DURAISWAMI, R., HARWOOD, D., AND DAVIS, L. 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of IEEE* 90, 7, 1151-1163.
- [12] Fukunaga K, and Hostetler LD, "The estimation of the gradient of a density function, with application in pattern recognition," *IEEE Trans. Information Theory*, vol. 21, pp.32-40,1975.
- [13] Zhi-Qiang Wen and Zi-Xing Cai, "Mean Shift Algorithm and its Application in Tracking of Objects," In *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August, 2006.*