Enhanced Object Tracking using Davinci Processors

J. Navin Sankar, S. Mary Joans, S. J. Grace Shoba, A. Arun

***Abstract*— Modular tracking methodologies have shown the promises of great versatility and robustness. In a similar way, the proposed paper, Enhanced Object Tracking Using Davinci Processors, will also possess major challenge for emerging computer vision technology. The Continuously Adaptive Mean Shift [CAMSHIFT] Algorithm used here is based on the Mean Shift Algorithm for object tracking for a perceptual user interface.**

The main aim of this proposal is to determine the effectiveness of the CAMSHIFT Algorithm as a general purpose object tracking approach in the case where a small portion of image is assumed as region of interest. Then the object within the corresponding region of interest is tracked using CAMSHIFT algorithm. The algorithm performs well mainly on moving objects in video sequences and it is robust to changes in shape of the moving object.

The Digital Video Development Platform [DM6437 EVM] is used to obtain the video from the camera and will use the Ethernet media access control address and video processing back end drivers for the real time transmission of the video captured. The video is received and processed at DM6446, where the CAMSHIFT algorithm is implemented and the video object tracking takes place.

The experimental results obtained from the proposal proves the consistency and efficiency of the proposed algorithm..

***Index Terms*— CAMSHIFT, CCS, DM6437, DM6446, LINUX, Ubuntu.**

I. INTRODUCTION

Object tracking from a video sequence is the process of locating moving objects in real time through a camera. An algorithm is used to scrutinize the video frames and spot the moving targets within that video frame. The main steps involved in this process are object detection, tracking and analysis of the tracked object. Object detection from the video sequence is the process of detecting the moving objects in the frame sequence using digital image processing techniques. Background subtraction is the most common used technique for the object detection.

Object tracking is a serious assignment in the field of image processing. The mercurial increase of robust computers, availability of eminent thrifty video cameras and the increased use for automated video analysis has generated great interest in object tracking algorithms.

The major steps involved in video analysis are:

1. Detection of the interested moving objects.
2. Tracking the interested objects from frame to frame.
3. Analyzing the object tracks to recognize the behavior of the interested object.

In its clear form, tracking is defined as the estimation of the trajectory of an object in an image plane while it is moving around the scene. By other means, a tracker assigns labels to the tracked object consistently in different frames of the video. In addition, with respect to the tracking domains, a tracker also provides information about the object, such as orientation, area or shape of that object. One can simply track by forcing constraints on the motion including or without including the appearance of the object. Most of the systems developed so far is based on the system integrated not with wireless whereas in our proposal we intend to build an efficient wireless vision based integration for tracking object from remote plane using CAMSHIFT algorithm.

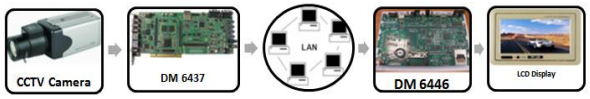

Figure -1: System Overview

II. THE TRANSMITTER SECTION

The proposed system is a portable, low power and network based system thus making it possible to be applied to a variety occasions and replace the traditionally used systems. The system is divided into two parts, The Transmitter Section and the Receiver Section. The Transmitter Section has the CCTV camera as the capturing device and TMS320DM6437 EVM as the core processor. The camera and the core processor constitute the hardware parts of the section. A computer with Windows XP and Code Composer Studio [CCS] constitutes the software section.

A. DESIGNING WITH THE XP AND CCS

Even as several advanced options are available with Windows 7 and Windows 8 nowadays, why we still go for Windows XP is mainly because of its stability and reliability. All the development softwares whether old or new are supported by XP because the companies know that they must build their development softwares with backward compatibility. When we go for Windows 7, “Windows XP” mode is available to support all the development softwares.

Figure -2: Transmitter Section

Manuscript received on March, 2013.

Navin Sankar J, PG Student/Applied Electronics, Velammal Engineering College, Chennai, India.

Prof. S. Mary Joans, HOD, Electronics and Communication Department, Velammal Engineering College, Chennai, India.

Mrs. S. J. Grace Shoba, Professor, Electronics and Communication Department, Velammal Engineering College, Chennai, India.

Mr. A. Arun, Assistant.Proffessor.-II, Electronics and Communication Department, Velammal Engineering College, Chennai, India.

The Code Composer Studio [CCS] is an integrated development specifically for the Texas Instruments products. It includes the source code editor, compiler, profiler, project build environment, debugger, simulators and many other features.

The development flow in the CCS includes four basic phases:

- Application Design
- Code Creation
- Debug
- Analysis/Tuning

After installing CCS, an icon will be created in the desktop. To launch the CCS, click the icon in the desktop. A simulator will be automatically configured by default. The project settings will be stored in the project file that is created. We can add files by selecting the files required for the project.

B. PROGRAMMING USING EMBEDDED C

Embedded C are the predetermined language extensions for the C Programming language for address commonalty for different embedded systems. In this project, the first step is the inclusion of the codec engine files.

C. INITIALISING THE CODEC ENGINE

The codec engine determines the format of the input that is going to be given to the core processor, TMS320DM6437. The core processor supports three types of codecs, MPEG 2, MPEG 4 and H.264. The first step is the installation of the dvsdk software. This installs the directories required for the core processor. In this project, we are using a PAL [Phase Alternating Line] camera. So, the CIF [Common Intermediate Format] has a resolution of 352 x 288 pixels and a frame rate of about 25 frames per second. A jumper in the TMS320DM6437 is used to select NTSC or PAL type of camera used. The next step is to connect the camera to the board and set the board status as "connected" in the system. Then we have to set the switch SW7 to the encode/decode mode, since we have to decode the video captured. The mode is then set as "decode from file", to decode the captured video. In the video settings, the codec is given as "H.264" to initialize the codec to the board. By adding the header file, the codec is initialised in the program. The codec engine provides a robust and consistent interface for dynamically creating and deleting algorithms and accessing and controlling algorithm instances. It also allows the same application code to be used across a variety of platforms without modification.

D. INITIALISING THE NDK ENGINE

As the proposed paper is the transmission of the image through wireless medium, networking is needed to transmit the image to the receiver. To enable networking protocols in the EVM, "NDK Engine" [Network Development Kit] has to be installed in the EVM. The Ethernet port in TMS320DM6437 supports 10/100 Mbit interface, sourcing the 25MHz clock. The NDK's Embedded File System [EFS] which supports the RAM files, is the file Input / Output API [Application Programming Interface] that is used by the HTTP server and several other programs. Some of the examples of NDK's EFS are, EFS_createfile (create a file from a RAM array), EFS_destroyfile (remove a file), EFS_fopen (open a file), etc,.

E. INITIALISING THE PSP ENGINE

In the TMS320DM6437 EVM, we need some physical layer to connect the input devices such as cameras and output devices such as LCD or LED displays and to process the given input video. The Video Processing Subsystem (VPSS) in the EVM consists of these physical layers. This VPSS can be initialized using the PSP include files. VPSS is generally comprised of two parts, Front End (VPFE) and the Back End (VPBE). The VPFE includes the CCD Controller (CCDC), Statistics Engine (H3A), Previewer and the Resizer. The VPBE includes the On-Screen Display (OSD) and Video Encoding (VENC). In this paper, we are using the VPFE for processing, while VPBE is used to check the working condition of the EVM.

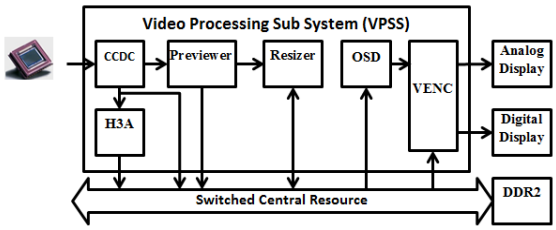

Figure – 3: VPSS

F. IMAGE CAPTURING AND DECODING

In this paper, we use the Hi-Focus Digital CCD camera, which is a PAL camera, to capture the moving object. This captured image is decoded using the TVP5146 Digital Video Decoder available in the EVM. The TVP5146 device is a high quality, single chip video decoder, that digitizes and decodes the analog video formats into digital component video. By initiating the VPFE parameter as vpfechannelparams.id = PSP_VPFE_CCDC, the CCD camera is initiated to capture the video. Now, by initiating the TVP5146 parameters using tvp5146params .enable656sync=true, the image format can be changed from the analog to digital.

III. IMAGE RESIZING

Image Resizing is nothing but altering the image size at which it will be printed without affecting the number of pixels in the image, instead the pixels are printed further separated or closer in sync with eachother. Resizing the image does not affect the screen display. In this project, the resizer coefficients are given as, PAL D1 704 x 576i → PAL CIF 352 x 288p. This is because we are using a PAL camera. In this EVM, we can resize the image according to our required size in the horizontal and vertical directions. They can be changed in the resizing parameters as paramsHfiltCoeffs[i] =local_ resizerCoeffsPalD1PalCifhorz[i] and params .VfiltCoeffs[i] =local_resizerCoeffsPalD1PalCifVer[i]. Now, the resizer file is created as, rszfd = GIO_create ("resizer",iom_inout,null,(void*)&memsegid,gioattrs). For each operation in the transmitting end or in the receiving end, buffer must be allocated in order to temporarily store the result of any particular process. Buffer is allocated for the video as, result = FVID_alloc (hGiovpfeccdc & framebufftable[i]).

IV. IMAGE ENCODING

Image encoding/compression is the conversion of the image from one format into another, otherwise it is cutting down the image size without squashing the image quality to objectionable standards.

In this paper, we are using H.264 video compression standards to encode the video captured. Since the main application of this proposal is for the surveillance purpose, JPEG standard cannot be used as it is best used for digital still photographs. When we go for the MPEG-4 standards, which is most commonly used nowadays, it can provide us both intra-frame compression and inter-frame compression at a low bit rate but it requires a higher bandwidth to transmit the frames. So we go for H.264 which provides high quality video at a much lower bit rate and lower bandwidth.

The first step in the encoding process is the initialization of the encoder parameters. These parameters include maximum height, maximum width, frame rate, bit rate, endianness and the input format. An example for the initialization is given as follows, vencparams → size = sizeof(VIDENC_params), vencparams → maxheight = resizeroutheight. Then we have to allocate the buffer for the encoder output. This allocation of the output buffer is given as, encodedbuf = (XDAS_int8*)memory_contigalloc(FRAMESIZE, BUFALIGN). Now, we have to create the encoder as, enc = VIDENC_create (ce ,encodername,&vencparams). Now the video is set for streaming frame by frame.

V. THE RECEIVER SECTION

The receiver section has the TMS320DM6446 EVM as its core processor which is a dual-core processor with an ARM processor operating upto 300 MHz and a C64xx DSP operating upto 600 MHz. Similar to the transmitter section, the receiver section also has two parts, hardware part and the software part. The hardware part constitutes the core processor and the LCD monitor for display. The software section constitutes the MontaVista Linux LSP and Open Source Linux. We use TMS320DM6446 here as it can playback, communicate and record videos when compared with TMS320DM6443, which only has the operations of either playback or playback and communicate.

Figure – 4:Receiver Section

A. Designing With The Ubuntu And Minicom

The core processor used here is compatible with Linux. So we go for Ubuntu, which is based on Linux as it is more secure and easy to change the options. In order to make the computer with ubuntu to communicate with the core processor, we use Minicom. Minicom is serial communication program which is free with source code and runs under most unices.

B. Decoding The Image And Implementation Of Camshift Alorithm

After the video is transmitted from the TMS320DM6437 EVM, it should be received at the TMS320DM6446 EVM. So, the EVM should be loaded such that the video format is supported by the EVM. Thus the DVSDK [Digital Video Software Development Kit] driver is installed at the receiver end computer. The DVSDK consists of the Video and Imaging Evaluation Codecs, Digital Media Algorithm Standards, Multimedia APIs, Real Time Kernel and the MontaVista Pro Linux demonstration software. Moreover, the DVSDK allows the system integrators to blend discrete software modules and synthesize them into a single executable output for the systems. Then we have to install the codecs for decoding the received video. The codecs available for this EVM are, MPEG2, MPEG4 and H.264 and we use H.264 for the process since it supports both main and high profile solutions, provides frame rates upto 60fps and bit rates upto 100Mbps with simultaneous multi-channel decoding.

C. Loading And Decoding

The first step at the receiver section is the decoding the received video and then applying the Improved CAMSHIFT algorithm to track the required object. To receive the video, the files required for the EVM should be loaded. As the first step, the UDP socket is created since we are using it to communicate between the transmitter and the receiver side. The encoded video is now received and the received frames are arranged before being decoded. Before processing, the parameters of the video are defined. Then, the streamer arguments such as, frame number, address length and buffer are initialized. A video decoder is created to decode the received video as, hvd2=vdec2 _create(hengine,"h264dec"/envp→videodecoder, params, dynparams).

D. Implementation Of The Camshift Algorithm

There are two major groups of tracking algorithms, state space approach and the kernel based approach. Since the state space approach consume high computational costs, we go for kernel based approach that includes the CAMSHIFT object tracking algorithm. The CAMSHIFT is an adaptation of the mean shift algorithm. The mean shift algorithm is a non-parametric density gradient estimator that is basically an iteration based clustering algorithm executed within the local search regions and is computationally feasible. Some drawbacks of the mean shift algorithm are that it fails to track multi-hued targets, it may fail to track the target when its shape and orientation changes and it fails to track the fast moving objects. In this paper, we use a tracker that represents the center of the target. Here, the coordinates of the centroid of the search window is also calculated to show that the moving objects can also be tracked using this algorithm.

The steps involved in the CAMSHIFT algorithm inorder to track the object that we are interested are as given below:

1. Tap the original location of the search window.
2. Select the search window location in the Region Of Interest [ROI] slightly larger than the mean window size.
3. Run the Mean Shift algorithm to find the movement of the centroid of the search window and store the zeroth moment Z_{00} area or size and centroid location which is given as follows:

$$Z_{00} = \sum_x \sum_y A(x, y)$$

4. For the subsequent video frame, center the search window at the greedy location stored in the step-3 and set the window size as a function of Z_{00}.

$$Z_{10} = \sum_x \sum_y cA(x, y)$$

While programming, the CAMSHIFT algorithm is created as h = camshaft _ create (hengine, params1). The buffer size for the algorithm to be executed is also defined. While we use the CAMSHIFT algorithm, a probability distribution of the desired colour (ie, the colour of the selected object in the select window) in the video sequence is created. Now the colour system of the selected or the captured video is changed from RGB to the HSV [Hue Saturation Value]. Inorder to avoid the problems due to the size of the image during the implementation of the CAMSHIFT algorithm, the image size is set to the maximum. The received video is copied to the decoder buffer and if any encoded data is missing, the next frame cannot be found. The buffer where the processed frame has to be stored is resized after the first frame is processed using the command numbufs = resizebuftab (hvd2, display _ pipe _size) but we may not know its requirements before it is decoded. After selecting the ROI, the CAMSHIFT process is called as status = universal_process (hvd, inbufdesc,outbufdesc,inoutbufdesc,inargs,outargs). Inorder to get the X-coordinates and Y-coordinates of the centroid, the flags for the X-coodinates is defined as coord_flag and coord_flag1 for the Y-coordinates. After the tracking of the interested object begins, the X and Y coordinates of the centroid of the search window will be calculated and printed at the output.

VI. RESULT

The algorithm is tested using a yellow colour box as the selected ROI.

Figure – 5: Selection the ROI

After selecting the ROI, the algorithm is applied and the object is tracked as follows:

Figure – 6: Tracking the Object

From the above images, it can be seen that a yellow color box is chosen as the object and its center part is chosen as the region of interest. As the object is moved, let us consider the top left image, where the object is moved towards the right so as the ROI and it is tracked as it moves. Similarly as the object is moved around, the ROI gets tracked using the CAMSHIFT algorithm as proposed.

VII. CONCLUSION

Tracking an interested object while in motion is a difficult. This has been done using an improved CAMSHIFT algorithm which is robust to changes in shape and movement of the object. The experimental results are provided for tracking a single object. As a part of the future development, the algorithm can be developed in such a way that multiple moving objects are tracked simultaneously. By measuring the velocity or speed of the moving object, it can be useful to identify the speeding vehicles. By adding illumination adaptation modules, the algorithm can be made a better tracking algorithm.

REFERENCES

1. "Adaptive object tracking algorithm based on eigenbasis space and compressive sampling", Li. J, Wang. J, Image Processing, IET, November 2012.
2. "A Change Information based fast algorithm for video object Detection and Tracking", Subudhi. B. N., Nanda. P. K, Ghosh. A, Circuits and systems for Video Technology, IEEE Transactions, July 2011.
3. "Real Time People Tracking Using DM6437 EVM", Tomasz Marciniak, Damian Jackowski, Pawel Pawlowski, Adam Dabrowski.
4. Video and Image Processing Blockset, Mathworks.com.
5. "Object Tracking using Improved Camshift Algorithm combined with Motion Segmentation", Emami. E, Fathy. M, Machine Vision and Image Processing 2011.
6. "An Improved Camshaft Algorithm for Target Tracking in Video Surveillance", Chunrong Zhang, Yuansong Qiao, Enda Fallon, Chianqiao Xu, IT &T Conference 2009.
7. First Davinci Products for Digital Video Innovation, www.ti.com.
8. TVP5146 NTSC/PAL/SECAM 4x10-bit digital video decoder with Macrovision detection, www.ti.com.
9. TMS320DM6437 DVDP Getting Started Guide, www.ti.com.
10. TMS320DM6446 DVEVM v2.0 Getting Started Guide, www.ti.com.

AUTHORS PROFILE

J.Navin Sankar, has a Bachelor degree in Electronics and Communication Engineering from Srinivasa Institute of Engineering and Technology, Chennai. During the course, he worked on a project, "Eye Ball Movement Based Wheel Chair For The Physically Challenged With Voice Recognition". Currently, he is pursuing his Masters in Applied Electronics at Velammal Engineering College, Chennai. He is currently working on the project "Enhanced Object Tracking using Davinci Processors".

Prof.S.Mary Joans, is the Head of the Electronics and Communication Engineering Department at Velammal Engineering College, Chennai. She has a Masters in Applied Electronics and is currently pursuing PhD. She is a member of IEEE and a lifetime member of ISTE and IETE. She has published three papers in International Journals and has authored two books. She has over 20 years of taching experience in various reputed institutions.

Prof.S.J.Grace Shoba, is currently working at Velammal Engineering College, Chennai. She is a lifetime member of ISTE and IETE. She is pursuing PhD in the field of Image Processing and has a teaching experience of over 15 years in various reputed institutions. She has over 6 publications to her credit and has presented papers in various national and international conferences.

Mr.A.Arun, is currently working as an assistant professor at the Velammal Engineering College, Chennai. He has a Masters degree in VLSI deisgn and has a teaching experience of over 5 years at various reputed institutions. He is currently pursuing PhD with specialisation in FPGA and has presented various papers in the national and international conferences.