PRECISE MOTION DESCRIPTORS EXTRACTION FROM STEREOSCOPIC FOOTAGE USING DAVINCI DM6446

Muhammad Asif, and John J. Soraghan

CeSIP, Department of Electronic and Electrical Engineering, University of Strathclyde
204 George Street, G1 1XW, City, UK
phone: +44 (0) 141-548-2921, fax: +44 (0) 141-552-2487, email: masif@eee.strath.ac.uk
web: www.eee.strath.ac.uk

ABSTRACT

A novel approach to extract target motion descriptors in multi-camera video surveillance systems is presented. Using two static surveillance cameras with partially overlapped field of view (FOV), control points (unique points from each camera) are identified in regions of interest (ROI) from both cameras footage. The control points within the ROI are matched for correspondence and a meshed Euclidean distance based signature is computed. A depth map is estimated using disparity of each control pair and the ROI is graded into number of regions with the help of relative depth information of the control points. The graded regions of different depths will help calculate accurately the pace of the moving target and also its 3D location. The advantage of estimating a depth map for background static control points over depth map of the target itself is its accuracy and robustness to outliers. The performance of the algorithm is evaluated in the paper using several test sequences. Implementation issues of the algorithm onto the TI DaVinci DM6446 platform are considered in the paper.

1. INTRODUCTION

Video Analytics (VA) for surveillance systems has evolved, to automatically detect and recognize objects. Several computer vision based techniques were introduced into surveillance systems in an attempt to extend their functionalities [1][2][3]. The range of video analytics for surveillance systems, includes all known fields of image and video processing, ranging from object detection/recognition to behaviour analysis. The work in this paper proposes a novel approach to extract useful target information in the form of MPEG7 motion descriptors [6][7]. It is assumed that the system has multiple cameras with partial overlapped FOV. To compute motion descriptors, the algorithm makes use of background modelling [2][3], feature extraction and matching [5][6] and depth estimation using disparity [9][10].

The paper is organized into five sections. Section 2 presents the novel approach of precise motion descriptor extraction using depth information of selected feature points. The algorithm comprises four stages: (i) Feature Detection (ii) ROI Marking (iii) Depth Estimation and (iv) Motion Descriptors Extraction. Each stage is discussed in-detail in sub-sections of section 2. Section 3 outline the main features of the test-bed selected for implementation: the DM6446 EVM, which is a member of Texas Instrument *DaVinci* family [11]. In section 4 results of the algorithm on selected test video sequence is discussed. Finally in section 5 some concluding remarks are presented.

2. MOTION DESCRIPTOR EXTRACTION USING STEREO-PAIR SURVEILLANCE FOOTAGE

The focus of this work is to develop efficient algorithms for video surveillance systems. The processing can be on-line or off-line thus achieving effective usage of stored data respectively. In a surveillance environment the resources, e.g. cameras, are normally not limited. More often one installation is covered with number of cameras, capturing different parts with partial overlapped field of view (FOV). Our algorithm assumes two cameras capturing partially overlapped FOVs to extract information about the target. The algorithm comprising the following four stages is illustrated in Figure 1: (i) Detect features (ii) marking ROI (iii) Estimating depth of selected feature points and (iv) Extracting motion descriptors. The following sub-sections discuss the details of each of above-mentioned stages.

2.1 Feature Detection

The footage acquired from the two cameras is processed separately to acquire a background model (BGM). There are a number of good approaches that can be used to construct a BGM with and without post-processing. For this work two choices for BGM were tested. The first is a recursive technique called a Median Filtering (MF) [2] approach and the second is the Approximated Median Filter (AMF) [2]. MF sets each pixel in the BGM to be the median value as determined from the buffer of video frames. This technique provides very robust BGM at the cost of high memory usage. On the other hand AMF performs equally good with the use of post-processing [2]. The advantage of using AMF is two-fold. Firstly the background image constitutes only the static part of the scene and secondly the image is robust to transient noise and fluctuations in intensity. AMF is computational efficient and simple to implement. Its limitation is that it does not model the variance of a pixel [2]. Another advantage of using AMF with post-processing is that the features extracted from the background are robust to noise and other short time abrupt changes in intensity.

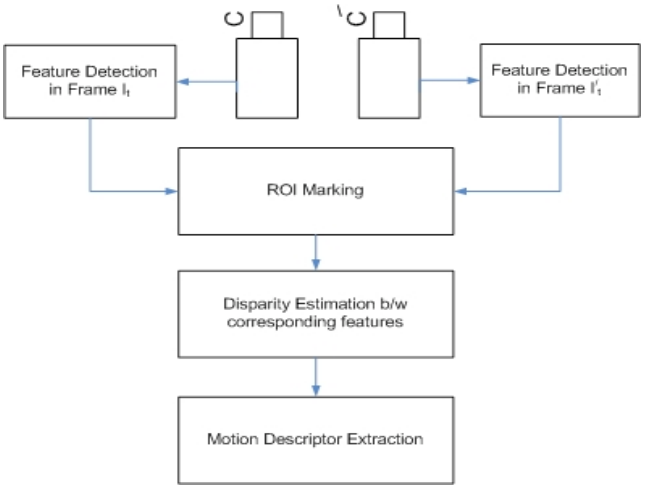

Figure 1 – Block diagram of motion descriptor extraction using stereo-pair footage

Computing the correspondence between stereo pairs depends on the accuracy and strengths of features extracted. Establishing correspondence between two views of a scene involves either finding a match between the location of points in the two images or finding a transformation between the two images that maps corresponding points between the two images into one another [8]. The former is a feature based matching technique, whereas the latter is a direct method using optical flow. The work here uses feature-based method. The features used are corners and are extracted using the *Harris corner detection* technique [8]. The number of features is filtered down to few using repeatability and accuracy criteria that we have published previously [3].

2.2 Marking ROI

The FOV of footage acquired from the two cameras can be divided into three segments. One segment in both images is non-overlapping, the second is overlapping with significantly poor correspondence and the third is an overlapping segment with good correspondence. The last segment constitutes the ROI. Normalized Cross-Correlation (NCC) is a well known technique to establish correspondence between matched pair of features. For this work NCC is used to mark the ROI between two partially overlapped video frames. To mark the ROI accurately in two cameras footage, the images are firstly divided into four blocks of equal size as illustrated in Figure 2. Search windows are defined in both images. As indicated the search windows in the reference image (Fig 2(b)) are significantly larger than those in the Fig 2(a), for example, a video stream of frame size 640x512 is divided into four blocks each of size 320x256. The search window for the reference frame is selected of size 150x150 and the search window for stereo-pair image is selected to be 60x60. In the worst case scenario, where there is no constraint on cameras FOV and no prior information is available, there could be as many as a maximum of 16 cross-region searches needed to find a seed for the ROI.

However once the ROI is acquired the computational demands drops significantly as only fine-tuning is needed to keep it up-to-date, as the two cameras are static. Also in the real world scenario there is prior information about the location of cameras, which usually minimizes the cross-region search to 2 or 4. Also the fine-tuning of ROI is performed using a sub-set of matched feature point extracted in section 2.1. The total number of feature points is filtered down to a few on the basis of repeatability, accuracy and trust vector as we have described in [3]. The resulting feature points are termed control points.

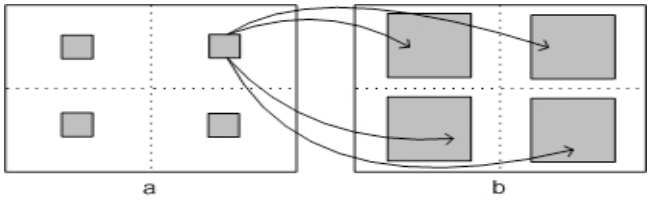

Figure 2 - Stereo frames (a) 4-blocks with search windows of 60x60 (b) showing reference frame with search windows of 150x150

2.3 Depth Estimation

The purpose of structure analysis in stereo-pair images is to determine accurately 3D location of image control points, as described in the last two sections. Figure 3 illustrates the geometry for disparity estimation using stereo-pair images. Assume complete knowledge of the perspective projection transformation matrix Q [8] is available for a point m = [u, v] in image I, as shown in Figure 3, which corresponds to point M = [X, Y, Z] in world coordinate, then the relationship between m & M can be written as [8]:

$$s\begin{bmatrix} u & v & 1 \end{bmatrix}^T = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \quad (1)$$

where $\mathbf{q_i}$ is the i^{th} row vector in perspective projection matrix **Q.** The scalar *s* can be eliminated from Eq(1) to become [8]:

$$\begin{bmatrix} q_1 - q_3 u \\ q_2 - q_3 v \end{bmatrix} \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T = 0 \quad (2)$$

Similarly for m' in the image plane I':

$$\begin{bmatrix} q_1' - q_3' u' \\ q_2' - q_3' v' \end{bmatrix} \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T = 0 \quad (3)$$

By combining Eq(2) and Eq(3) we get:

$$\mathbf{AM} = 0 \quad (4)$$

where, $A = \begin{bmatrix} q_1 - q_3 u & q_2 - q_3 v & q_1' - q_3' u' & q_2' - q_3' v' \end{bmatrix}^T$ is a 4x4 matrix that depends only on the camera parameters and the coordinates of the image points. M is the location of 3D point, which has to be calculated. In conventional baseline stereo systems, as shown in the Figure 3 the solution to Eq(4) becomes simple, with the assumption that the two cameras are coplanar and by ignoring the intrinsic parameters of both cameras. The 3D parameter of object M can be calculated as [8]:

$$X = \frac{b(u - u')}{2d}, \quad Y = \frac{b(v + v')}{2d}, \quad Z = \frac{bf}{d} \quad (5)$$

where $d(u,v) = \sqrt{(u - u')^2 + (v - v')^2}$ is the disparity and *b* is the baseline.

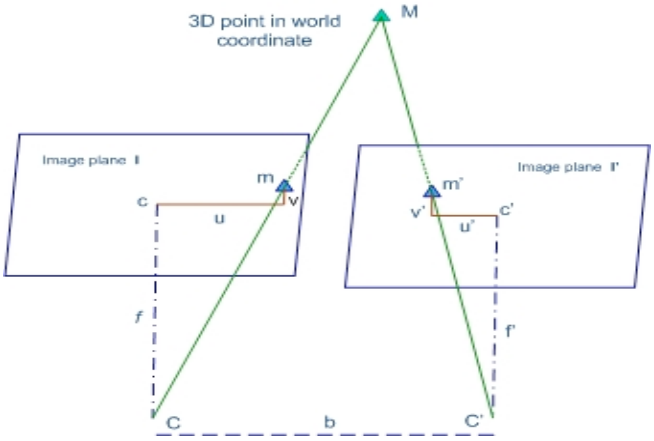

Figure 3 - Disparity estimation using Stereo-pair images

2.4 Motion Descriptors

We now have a depth map of our control points distributed over the entire ROI. The XZ plane is now quantized into 2^n layers, with n number of bits used to store depth information. For example for n=2 produces an XZ plane quantized into 4 layers. As these layers shows only relative depths, it is quite likely that these layers will be non-uniform. Again the control points lie within two boundaries of the layers. Having control points with relative distances from a viewpoint we can accurately access the depth of foreground object, human or vehicle and extract motion descriptors, motion activity, motion trajectory etc. Here we will consider the example of only motion activity.

2.4.1. Motion Activity

For surveillance systems one of the most important descriptors is motion activity, which provides the information about 'intensity of action' or 'pace of action' in surveillance footage. For efficient application a few additional attributes of motion activity are provided. Here we outline only two attributes of motion activity: *Intensity of Activity* – this is expressed by an integer in the range 1-5. A high value of intensity indicates high activity while a low value show low activity. *Direction of activity* – while a video sequence may have several objects with different activity, we will identify a dominant direction of activity for one object, which represent objects of interest e.g. human or vehicle.

If there are two or more foreground objects identified, the dominant direction of activity is computed for each and tagged along with it. Motion vectors provide the easiest approach to represent the gross motion characteristics of the video segment. Since the motion vector magnitude is an indication of the magnitude of motion itself, it is natural to use statistical properties of the motion vector magnitude of Macroblock (*MBs)* defined earlier to measure intensity. We are interested in measuring motion activity of the foreground, for this we take the *MBs* of foreground motion mask to compute motion vectors. Also the effect of global motion is compensated to get the absolute motion activity. Both the standard deviation and average of the motion vector magnitude reasonably match the ground truth after proper quantization and *scaling* [7]. However, it is observed [7] that the standard deviation of the motion vector magnitude provides a slightly better approximation of the ground truth and hence the quantized and scaled standard deviation of motion vector magnitude is used to compute the intensity of motion activity shown in Table 1. This shows the thresholds used for quantization of standard deviation σ. Our proposed empirically scaled standard deviation is:

$$\sigma_i' = \frac{N\sigma_i}{2^n} \qquad N = \{1,2,...2^n\} \tag{6}$$

where 2^n are the number of depth map levels being identified.

Table 1 Standard Deviation thresholds

Activity value	Threshold Range for σ
1	$0 \le \sigma < 3.9$
2	$3.9 \le \sigma < 10.7$
3	$10.7 \le \sigma < 17.1$
4	$17.1 \le \sigma < 32$
5	$32 \le \sigma$

Table 2 Scaled Standard Deviations

σ	N	σ'
32	1	8.0
10.0	3	7.5
15.5	2	7.75
3.0	4	3.0
18.0	2	9.0

An example is illustrated in Table 2. The measured standard deviation vector σ_i, *i=1,2,3,4,5* for respective layers N_i, *i = 1,2,3,4,5* is scaled using Eq(6) to form σ'_i, *i=1,2,3,4,5*. If two objects at different depths cover the complete FOV in the same time then the motion of the object furthest from the observation point must be faster than the closer object. The same outcome can be observed from Table 2 where for example when the standard deviation value σ_1 equals to 32, suggesting very fast motion intensity. However when it is scaled using the layer information (N_1 equals to 1, closest layer), the scaled standard deviation value, σ' equals to 8 that suggests medium motion activity.

There are two sources of error when computing the scaled standard deviation value: (i) the use of linear scaling as shown in Eq(6) and (ii) relative assessment of depth regions. The first can be reduced by increasing the number of depth levels (like quantization error) while the second source of error can be minimized by using a datum point with known depth from the camera and computing the relatively accurate depth maps for different regions.

2.4.2. Motion Direction

The directional characteristics for each foreground object are represented with an average angle and variance of the angle [6][7]. The directional angle is computed over the foreground motion mask. For more then one motion mask, separate di-

rectional angles are computed. For this work we define the angle matrix A as [7]:

$$A(i,j) = \{ang(i,j)\} \tag{7}$$

where $$\text{ang(i,j)} = \tan^{-1}\left(\frac{y_{i,j}}{x_{i,j}}\right),$$

and $(x_{i,j}\ y_{i,j})$ are the pixel values of motion vector within the (i,j)th MB. The average angle, A^{avg} of the object and variance of A is given as:

$$A^{avg} = \frac{1}{MN}\sum_{i}^{M}\sum_{j}^{N} A(i,j)$$
$$\sigma^{ang} = \frac{1}{MN}\sum_{i}^{M}\sum_{j}^{N}(A(i,j) - A^{avg})^2 \tag{8}$$

where M and N are the width and height of foreground object under-consideration in the MB.

3. TESTBED & IMPLEMENTATION

To implement the algorithms requires a processing unit with high computational power, large memory, and an efficient video acquisition front end. These requirements are available through the use of DM6446 EVM. The DM6446 leverages TI's DaVinci technology to meet the network media application processing needs [11]. The dual-core architecture provides benefits of both DSP and RISC technologies, incorporating a high performance C64x+ DSP core and an ARM9 core. Using this dual core has the advantage of full DSP computational power available for certain algorithm components and an ARM core for dealing with all peripherals to fetch media data. The ARM core runs at 300 MHz clock and C64x+ CPU runs at 600 MHz clock rate [11]. The processor has a powerful feature of parallel instruction execution and can execute up to 8- instruction per cycle. The cache plays a very important role for real time implementation. DM6446 DSP has separate program and data cache, as shown in the Figure 4. The EVM also has a large L2 cache of 128 KByte that is mapped into RAM. The algorithm presented in this paper needs several previous time frames, for BGM and for disparity estimation. This requires fast accessible memory [12]. DM6446 EVM has a large external 256 MByte memory with access data rate of 333-MHz. Finally the algorithm needs to display different depth marks onto the video sequence. This can be achieved with the use of On-Screen Display (OSD) feature of evaluation board [11].

The DaVinci platform employs an on chip operating system, such as Linux. The operating system can be used to debug DSP more effectively. Serial or network connection is used to upload source code and also used for debugging.

Software development is divided into three areas , (i) Application layer (ii) I/O layer and (iii) Signal Processing layer, as shown in Figure 5. DaVinci platform also provides a collection of optimized image/video processing functions (IMGLIB) [13]. These library functions include C-callable, assembly-optimized image/video processing routines. These functions are typically used in computationally intensive real-time applications where optimal execution speed is critical [13].

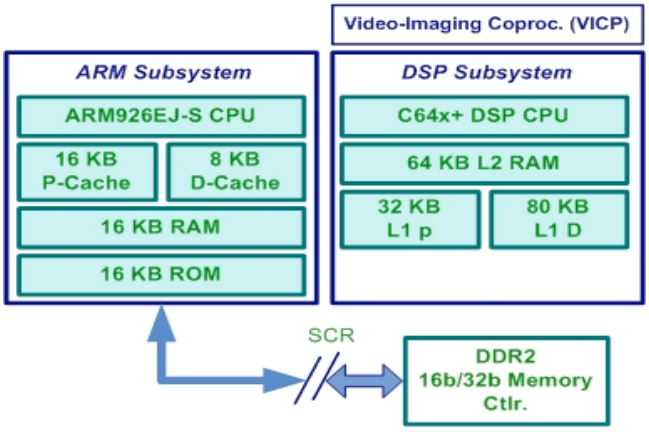

Figure 4 - DM6446 Dual core Architecture

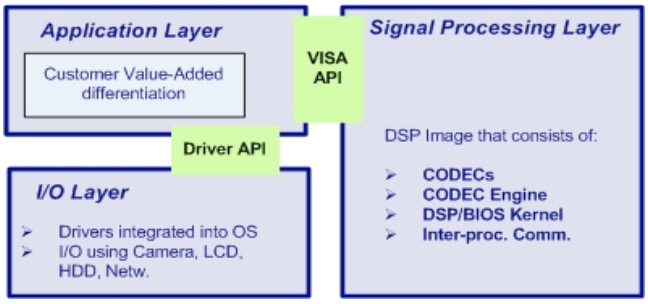

Figure 5 - DM6446 Software development layers

4. RESULTS

For performance testing of the algorithm video streams were captured in a typical multi-camera surveillance environment. For simplicity of reconstruction mathematics, the two cameras image planes are kept roughly parallel to avoid any orientation around the axes. BGM acquired using post-processing and AMF help extract robust feature points in both video streams, most of the feature remain consistence with their strings throughout the test sequence. ROI marking using NCC was very accurate and efficient. Figure 6 shows the result of marking the ROI. The features labelled with red are those that lie within the ROI while those labelled in blue shows features outside ROI. In Figure 7 the result of depth estimation for selected control points are shown. Depths are quantized into four regions. Here the control points marked with red are the farthest and then blue, green and yellow represent respectively decreasing depth levels. These depth levels are used to estimate the distance of foreground object and finally the pace of its motion is computed. A video sequence with different motion pace in different regions is used to test the algorithm and the resultant values of scaled standard deviation (SD), using Eq(6), are shown in Table 3. From Table 3, it can be observed that SD values, in the 2nd column, suggest motion pace: slow, medium, very slow and fast, are scaled to, in 4th column: very slow, slow, very slow and slow respectively. The implementation of this algorithm onto DM6446 has the advantage of its dual core and large memory. The DSP core can be used for algorithm and ARM will perform the task of fetching video frames. The problem of

taking multi-camera feeds can be resolved using a *http* call from a network port. Such a configuration with one IP-camera and one analogue camera is shown in Figure 8. Also, as mentioned earlier, the feature extraction process is robust due to AMF for BGM that provide the flexibility of using every 5th frame for feature extraction and correspondence computation. This will significantly reduce the computational requirements.

Table 3 Scaled SD Values for test sequence

Frame slots	SD σ	Region N	Scaled SD σ'
27-60	5.98	2	2.99
83-92	11.3	3	8.5
119-158	3.8	3	2.85
185-191	18.6	2	9.8

5. CONCLUSIONS

The work presented here provides a feasible and efficient solution for surveillance system problem of real time motion activity monitoring regardless of target distance from the camera. Using the persistent feature points from BGM to acquire depth estimation is more robust compare to depth estimation of the target themselves. From an implementation point of view it is suggested to use an Ethernet port and video port to connect IP and analogue cameras to the DM6446 target board.

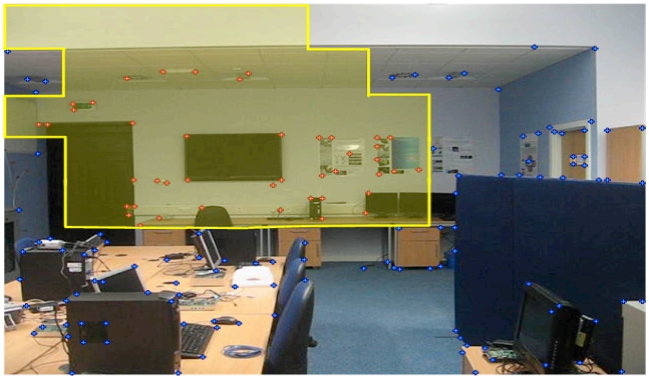

Figure 6 - ROI marked with yellow, feature in red in ROI

Figure 7 - Frame from camera 1, showing depth of selected feature points. Red showing farthest then blue, green and yellow nearest

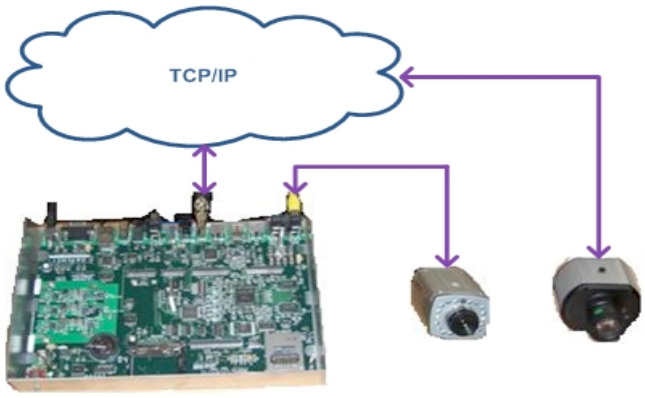

Figure 8 – Proposed DM6446 layout with IP & analogue Cameras

REFERENCES

[1] D. Hall, J. Nascimento P. Ribeiro, "Comparison of Target Detection Algorithms using Adaptive Background Model", IEEE Inter. Workshop on VS-PETS, Beijing, Oct 15-16 2005.

[2] Donovan H., Sidney S. 2008 Fels, Evaluation of Background Subtraction Algorithms with Post-processing, IEEE Inter. Conf. on Advanced Video and Signal Based Surveillance, Sep. 2008.

[3] Muhammad Asif, John J. Soraghan, 2008. MPEG7 Motion Descriptor Extraction for Panning Camera using Sprite Generated. IEEE Inter. Conf. on Advanced Video and Signal Based Surveillance, Sep. 2008.

[4] R. Hartley, A. Zisserman, 2000 Multiple View Geometry in Computer Vision. Cambridge University Press 2000.

[5] Lisa Gottesfeld Brown, 1992 A Survey of Image Registration techniques, ACM Computing Surveys, vol 24 issue 4 pp. 325-376 Dec. 1992.

[6] Standard MPEG7 part-8 ISO/IEC 15938-8; 2002 Information technology -- Multimedia content description interface -- Part 8: Extraction and use of MPEG-7 descriptions.

[7] Manjunath B, Sikora Thomas, *Introduction to MPEG-7: multimedia content description interface*, ISBN 0471486787

[8] Al Bovik, *Hand Book of Image and Video Processing*, Academic press 2005.

[9] Xiaoming Li, Debin Zhao et al. "Fast Disparity and Motion Estimation Based on Correlation for Multi-view Video Coding", IEEE Trans. on Consumer Electronics, vol. 53 issue 2, pp. 712-719. May 2007.

[10] Sung Yeol K. Eum Kyung L et al. "Generation of ROI Enhanced Depth Maps Using Stereoscopic Cameras and Depth Camera", IEEE Trans. on Broadcasting, vol 54, issue 4, pp. 732-740, Dec. 2008.

[11] Texas Ins. Digital Media Processor DM6446 Datasheet, http://focus.ti.com/docs/prod/folders/print/tms320dm6446.html

[12] Sen M. Kue, Digital *Signal Processors Architecture, Implementation and Application*, Prentice Hall 2005.

[13] TMS320C64x+ Image/Video Processing Libr. v2.0.1 http://focus.ti.com/docs/toolsw/folders/print/sprc264.html