Embedded and Modular Video Processing Design Platform

Thomas Gonnot and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*— Research and industry increasingly rely on image processing to analyze an environment. Most image processing requires significant computing power and consequently a complex processing unit. This paper presents a hardware specific platform that computes common image processing algorithms and extracts the information from a video stream. By this arrangement, the image processing can be customized for a specific application, using the same embedded system. This platform also simplifies the development of image processing systems and algorithms by focusing on higher level algorithmic operations.**

I. INTRODUCTION

Automation systems depend increasingly on image processing since the images from cameras contain enormous amount of information about the environment. Complex autonomous robots are often built with one or more cameras, increasing their capabilities to interfere with the surrounding environment. Industries also rely on cameras to improve their manufacturing process by checking the quality of their products at a very high speed, beyond human capabilities. Finally, video processing systems bring hope to people with visual disabilities by extracting all the information from the environment and describe it to the visually impaired, improving their daily lives. Furthermore, the security and surveillance of the environment depends mainly on the information acquired using cameras.

The consequence of expansion of the visual analysis of the environment is the pressing needs for more computing power. In the last decade, a number of advancements have been made by bringing parallel computing to computers through the use of graphical processing units (GPUs). In practice, this approach usually requires specific tools and knowledge. The power requirements of GPUs are also very high compared to other devices such as FPGAs [1]. This issue can be problematic when using an embedded processing component with power limitations.

This paper focuses on the architecture of a design platform for an Embedded and Modular Video Processing (EMVP) system. This modular unit can be fully optimized to operate with a maximum efficiency and integrate massive parallel processing elements using devices such as FPGAs or GPUs [1]. Furthermore the integration of complex processing components is also addressed in this paper. This approach allows the designer of video processing algorithm to use a complex system without extensive knowledge of the devices within the system.

Certain video processing systems already exist, for example, the popular Kinect platform from Microsoft. Its algorithm implementation provides the developer the necessary data for 3D reconstruction and human body detection and tracking. This explains why the Kinect is extensively used in research, since it reduces the time and resources needed to implement different video processing algorithms. However, this platform is limited because of its resolution, frame rate and scalability. Furthermore, video processing algorithms provided with the software development kit [2] are actually running on the host computer and not on the Kinect device.

The industry provides what is known as smart cameras, which includes a powerful processor within the camera to implement custom algorithms. Such cameras offer more flexibility to the developer, but also require more knowledge about image processing and the processor architecture. The sequential execution nature of the processor also reduces the potential efficiency of the image processing algorithms. The EMVP platform described in this paper is a design tool intended to include the algorithms required to execute basic image processing operations, such as color or feature extraction, motion field calculation, and cascade classifier implementation. This platform allows a standardized implementation of basic operation as blocks to enable fast prototyping of complex algorithms in hardware/software codesign. It is also designed to be implemented on a system-on-chip integrating programmable logic arrays to process the algorithms in parallel with maximum efficiency.

The proposed EMVP platform includes implementation of different algorithms that can be used for color, features and motion extraction (Section II), cascade classifier (Section III) and depth map reconstruction (Section IV). In this platform, a computer with OpenCV and CUDA is used to run the algorithm. Figure 1 shows the internal architecture of the

978-1-4799-4774-4/14/$31.00 ©2014 IEEE

EMVP platform on an embedded system combining hardware and software image processing components.

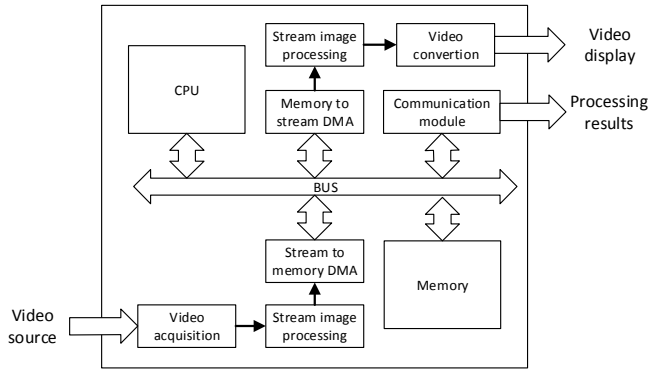

Figure 1. The proposed EMVP platform architecture

II. COLOR, FEATURES AND MOTION EXTRACTION

The EMVP platform presented will be able to extract information from the images, such as specific colors and features, and also the motion vectors from one frame to another.

The main interest of the color extraction is to discard the effects of the light and shadows on the object's colors. There are three principal color planes used in image processing. The first, and the most widely used, is the RGB format. The camera codes the image in three primary color planes. Such a color space is very convenient for displaying pictures on the monitors supporting 3 primary color matrices of red, green and blue pixels. However, the coding of the color is mixed on three planes, which makes it very difficult for any color operation. The second color space is the YUV, composed of a luminance plane Y, and two other planes representing the chrominance U and V. This color space is useful in image processing in the sense that the light has less influence on the color map. The third color space is HSV which is used for EMVP platform. In HSV, the information is divided in one plane for the color H for Hue, one for the Saturation S and one for the Value V (or intensity of the color). HSV compensates the differences of luminosity in the images, and also the deviation of color induced, for example, by the shadows.

For the HSV color space, we need to compute:

$$M = max(R,G,B), \quad m = \min(R,G,B) \tag{1}$$

And then compute the HSV values:

$$H = \begin{cases} 0, & if\ M = m \\ \left(60° \times \frac{G-B}{M-m} + 360°\right) mod\ 360°, & if\ M = R \\ 60° \times \frac{B-R}{M-m} + 120°, & if\ M = G \\ 60° \times \frac{R-G}{M-m} + 240°, & if\ M = B \end{cases} \tag{2}$$

$$S = \begin{cases} 0\ , & if\ \mathrm{M} = 0 \\ 1 - \frac{m}{M}\ , & otherwise \end{cases} \tag{3}$$

$$V = M \tag{4}$$

To extract a specific color, and discard the background, a binary map is generated from the thresholding of the Value and Saturation planes. This binary image is then filtered using morphological operations to reduce the effect of the noise. The platform can either return the binary map of the extracted color, or just the boxes surrounding the color areas. In practice, the color information is not sufficient for scene analysis, and thus requires other algorithms for complex object recognition and characterization.

In addition to color analysis of a scene, the EMVP platform is designed to extract other features. In particular, the Shift Invariant Feature Transform (SIFT) is used, which consists of two functional parts called extraction and matching [3] [4]. In the extraction step, the image is blurred using Gaussian functions with different blurring intensities. The blurred images are sequentially subtracted from each other and the resulting subtracted images are called Difference of Gaussians (DoGs). These DoG images are used to extract keypoints defined as all minimum and maximum points in the DoG images. Then, the original image is scaled and the process of extracting keypoints for the scaled images is repeated. Figure 2 shows how the DoG images are calculated. The pattern of keypoints is thresholded in order to keep and emphasize the strong keypoint features of the image.

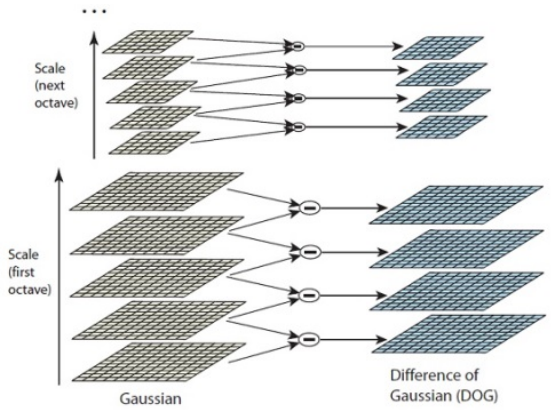

Figure 2. Difference of Gaussian (DoG) used in the SIFT algorithm

Next, keypoints are analyzed in order to generate feature vectors. The feature vector represents the intensity of the different gradient orientations around each keypoint. Finally, the system matches the extracted features with other known features and can recognize objects. In this study, the EMVP platform performs the features extraction, and then the features are used for feature matching operations.

In addition to the colors and keypoint features, the system is also designed to acquire information about the motion of object of interest within images. In this platform, an algorithm is integrated to extract the motion field between two frames. A method to extract the motion is to focus on certain features of the image and then find the same feature on the next frame to determine its position, and consequently to determine the direction of movement. The problem with this approach is fewer extracted features results in lower motion field

resolution. Another popular method, derived from MPEG video encoding, divides the image in small parts of a certain shape and size, and then search for them in the next frame. The resulting vectors form a motion field of the same resolution regardless of the content of the image, but might give inaccurate results in certain cases [5].

III. Cascade Classifier

The cascade classifier algorithm offers the possibility to recognize objects that can differ on some points from an original model. In this study, we use a special preprocessing cascade classifier called AdaBoost [6]. In this method, the image is segmented in subparts of equal size. Then, features are extracted from each subpart, with a special set of features as shown in Figure 3. Each feature generates a result that enters the classifier. Several classifiers are configured to recognize a specific shape. The algorithm applies the features on each classifier sequentially, which returns a value indicating if there is a match or not. In case of a match, it returns the recognized object; otherwise it exits with a negative result. Once all the subparts are analyzed, the image is downscaled and the process starts again. This continues until only one part forms the entire image. A major drawback is that the cascade classifier needs to be trained with a set of known images of the object to be recognized.

In the EMVP platform, the cascade classifier can be implemented using parallel processing for the computation of the features. Furthermore, the cascade classifier can be implemented in hardware to improve the computation time of each classifier and the overall object recognition performance.

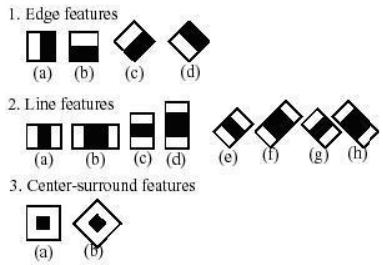

Figure 3. Example of features elements used for the cascade classifier. Each feature corresponds to the difference between the white region and the black region of the feature element [6].

IV. Depth Map Extraction

A system using computer vision may require the calculation of a depth map of the scene. An effective method used to obtain a depth map is by using Kinect system from Microsoft [7]. In Kinect system, an infrared structured light is projected on the scene, and an infrared camera captures this projection. The system can then compute the distance of each object from the deformation of the structured light projected on them. However, this method requires a specific projector, and is complex to assemble.

Another method consists of using two cameras separated by a fixed and known distance. The difference between the two images is used to compute the distance of the objects to the cameras. This process, called stereovision, uses the Epipolar theory for calculating the 3D position of a feature by observing its correspondent point from the two cameras [8].

For the calibration of the stereovision, we must determine the characteristics of the system including the angle formed between the two cameras and the inner optical characteristics of each camera. Furthermore, calibration requires a pattern, typically a chessboard, to be presented to the two cameras with various angles. These measurements are used to adjust the images orientations for accurate stereo matching. The system calibration remains valid as long as the relative position and orientation of the cameras do not change.

In normal operation, the stereovision algorithm is composed of three steps. The first step removes the distortions from the camera optics. It also turns the stereo pair in standard form, projecting the images on the plane formed by the two cameras. The next step is to find the position of each feature of the first picture in the second picture. Since the two images have been transformed to a standard form, the search is restricted to the horizontal lines around the feature. The last step is to determine the distance of the feature from the camera by triangulation using the system position and optic constants. The coordinate disparity of every matching pixel from the two images generates a disparity map that used to extract the depth map.

The EMVP platform can be connected to 2 video cameras to provide the depth map of the scene. An automated calibration can be programmed so the user can use different sets of cameras to get an accurate result.

V. Results

The algorithms discussed above are integrated in the EMVP platform and tested in complex scenes. In order to test the color detection, an image of a complex scene is used and processed with a broad spectrum of colors as shown in Figure 4 with and without background cancelation. As one can see, the colors strike out in the scene in spite of shadow or noise effects. This also shows the efficiency of the algorithm to remove the areas without color, such as windows, tables and floor that generates parasitic color values in the HSV color space. In this specific example, a red ball appears very clearly on the scene.

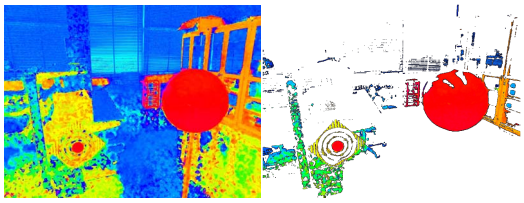

Figure 4. Left image shows the result of color detection. Right image shows the same algorithm for color detection with background cancellation.

The result of the feature extraction is less trivial to examine. A program has been made to match the features of an object with the scene. The object is defined by its picture and its features are computed and then compared with the features of the consecutive frames (see Figure 5). A rectangle

indicates the area where the selected object has been detected. Figure 5 also represents object match within the original pictures. As one can see, this complex object is correctly recognized, even with a small angle and different lighting conditions and with some positives. In this case, only 16 matches have been retained by the algorithm.

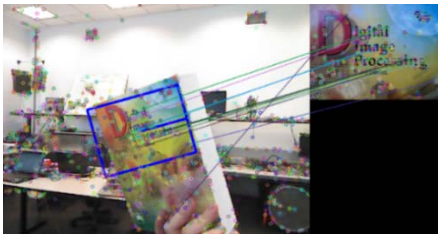

Figure 5. Example of feature extraction and matching

The testing of the cascade classifier is similar to the testing of the features extraction and matching algorithm. The difference is that the classifier is trained using several pictures prior to the object recognition execution, then a video stream is processed by the classifier to find the matching objects. In Figure 6, the classifier is trained to detect a flying helicopter in real-time.

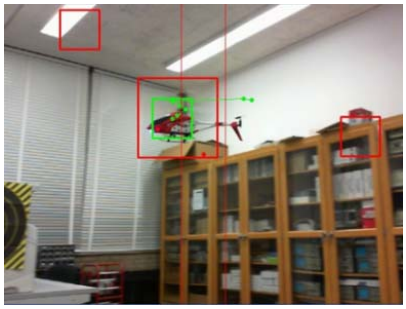

Figure 6. Cascade classifier for the detection of a helicopter

The last algorithm is the depth map reconstitution. Two cameras were positioned side by side and calibrated using a chessboard pattern. The output of the algorithm is a depth map such as the one presented in Figure 7. As we can see on this picture, the depth map reconstruction contains some noise; however, we can clearly distinguish different depths. For example, we can identify the profile of a person reaching the cameras with his hand.

Figure 7. An example of reconstructed depth map [9]

VI. Conclusions

This paper presented a concept of EMVP platform designed to efficiently execute various basic image processing algorithms. We showed the possibility to implement different examples of algorithms, extracting information such as colors, image features, motion, but also more complex algorithms such as classifiers and depth map reconstruction. A set of CAD tools can be considered as part of the EMVP platform to integrate the numerous image processing algorithms from an open library, and reduce the design cycle by allowing block programming of hardware/software algorithms for optimal efficiency.

References

[1] J. Fowers, G. Brown, P. Cooke and G. Stitt, "A performance and energy comparison of FPGAs, GPUs, and multicores for sliding-window applications," *Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays,* pp. 47-56, 2012.

[2] R. A. El-laithy, J. Huang and M. Yeh, "Study on the Use of Microsoft Kinect for Robotics Applications," *2012 IEEE/ION Position Location and Navigation Symposium (PLANS),* pp. 1280-1288, 2012.

[3] T. Lindeberg, "Scale Invariant Feature Transform," 2012. [Online]. Available: http://www.scholarpedia.org/article/SIFT.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* pp. 91-110, 2004.

[5] S. Tsekeridou, F. A. Cheikh, M. Gabbouj and I. Pitas, "Vector rational interpolation schemes for erroneous motion field estimation applied to MPEG-2 error concealment," *IEEE Transactions on Multimedia,* vol. 6, no. 6, pp. 876-885, 2004.

[6] P. Viola and M. Jones, "Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade," *International Journal of Computer Vision,* pp. 1311-1318, 2001.

[7] J. MacCormick, "How does the Kinect work?," [Online]. Available: http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf. [Accessed 2013].

[8] S. Mattoccia, "Stereo Vision: Algorithms and Applications," 2012. [Online] http://www.vision.deis.unibo.it/smatt/Seminars/StereoVision.pdf.

[9] W. Jia, W.-J. Yi, J. Saniie and E. Oruklu, "3D image reconstruction and human body tracking using stereo vision and Kinect technology," *2012 IEEE Iternational Conference on Electro/Information Technology (EIT),* 2012.