A Video Stream Processor for Real-time Detection and Correction of Specular Reflections in Endoscopic Images

Stéphane Tchoulack, J.M. Pierre Langlois and Farida Cheriet
Département de génie informatique et génie logiciel
École Polytechnique de Montréal, Canada
{ralph-stephane.tchoulack-ngounou, pierre.langlois, farida.cheriet}@polymtl.ca

***Abstract*— This paper presents the architecture and FPGA implementation of a video processor for detection and correction of specular reflections in endoscopic images by using an inpainting algorithm. Stream processing and parallelism are used to exceed real-time performance on NTSC format video without the need for an external memory. The system was implemented in a XC2VP30 FPGA and uses 91% of available slices. Image quality is significantly enhanced.**

I. INTRODUCTION

Advances in video imaging have played a key role in bringing forth the widespread use of Minimally Invasive Surgery (MIS) in a variety of procedures such as cardiology, neurosurgery, orthopaedics, urology and oncology. However, the inherent difficulties of MIS techniques have traditionally imposed limitations on their applicability. Reduced instrumental control and freedom, combined with unusual hand-to-eye coordination and a limited view of the operating field, enforce restrictions on the surgeon and require considerable dexterity and skill. On the other hand, these procedures entail several benefits for the patient and the healthcare system: smaller incisions, minimal blood loss, preservation of normal tissue, reduced pain, and shortened recovery and rehabilitation times.

The use of video-assistance to facilitate MIS has been the focus of increasing attention since the early 1980s. In typical video-assisted MIS, a small camera called an endoscope is inserted into the surgical site via a small incision on the surface of the patient's body. The surgeon will exclusively use the endoscope video displayed on a monitor to view the surgical site and control the position of his instruments, which are also inserted through small incisions. Several sophisticated 3D navigation systems are under development in cardiology and neurosurgery. However, most of the current systems devised for spinal surgery still rely on rigidly fixed dynamic reference (or fiducial) markers on the instrumented vertebrae for the registration of preoperative patient data with the intra-operative data. This allows the surgeon to localize precisely the anatomical structures of interest while minimizing damage to adjacent critical structures.

Our team is working on the development of an augmented reality surgical environment using an image-based approach -- instead of using visible markers in the pre- and intra-operative images -- to achieve a non-contact, automated method for elastic 2D-3D registration. Unfortunately, the reflection of the light on specular surfaces such as metallic tools and moist tissues, as shown in Fig.1, produces artifacts in the images that render the task of automatic segmentation of endoscopic images very challenging. Improving the quality of endoscopic video is an important goal in itself, especially for augmented reality applications [8]. Any kind of processing must operate in real-time on regular sized video to be usable in a hospital, in an operating room. Previous work has shown that endoscopic images can be significantly improved, but with significant memory and processing requirements [1].

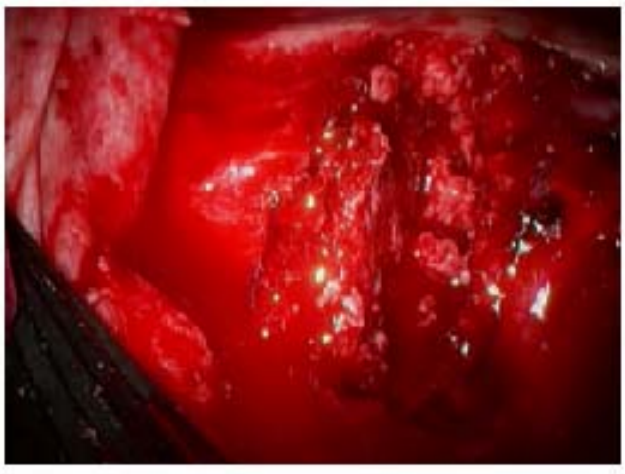

Figure 1. An endoscopic image with specular reflections [1]

In this paper, we present particularized algorithms for automated detection and correction of specular reflections in an endoscopic context, together with their real-time implementation in hardware. The paper is organized as follows. Section II presents the system architecture and some implementation considerations that affect algorithm development. Section III deals with the problem of detecting the specular reflections, and section IV is concerned with their correction. Section V presents results and a discussion.

II. SYSTEM ARCHITECTURE AND IMPLEMENTATION CONSIDERATIONS

Fig. 2 shows the block diagram of our system. The input is a stream of pixels and synchronization signals from a video decoder connected to an endoscope. The video format is deinterlaced NTSC with an effective resolution of 720 × 480 pixels and a frame refresh rate of 60 Hz. The output includes pixel values and synchronization signals of the same format and refresh rate, transmitted to a VGA port for real-time display.

978-1-4244-2332-3/08/$25.00 © 2008 IEEE

We require real-time video processing, which is difficult to achieve on a serial processor because of the great amount of data involved. In order to perform an operation on every pixel in real-time, the processor must execute 40 million operations per second. If we take into account the time to store and load data, the corresponding number of instructions rapidly increases. This is not outside the capabilities of FPGA implementations, however, where massive parallelism can be exploited. However, this choice implies special constraints related to the mode of processing data: streaming, offline and hybrid [2].

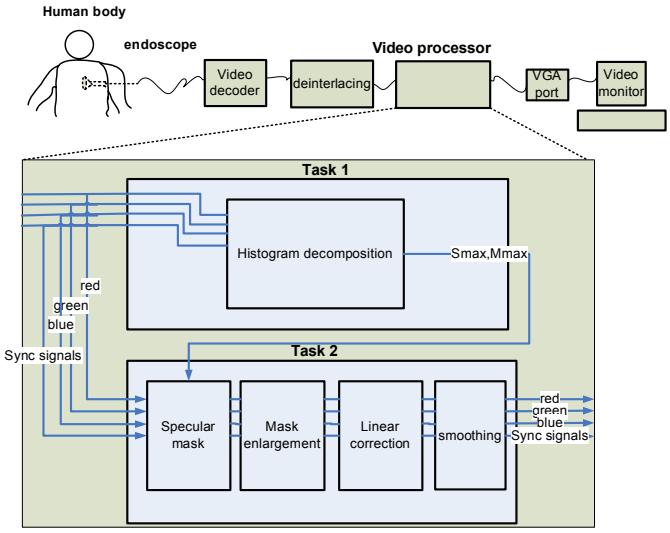

Figure 2. System block diagram

Storing one video frame requires approximately 1 MB. The limited amount of memory resources in a FPGA would normally require that an external memory be used, especially if several frames must be accessed for processing. However, this implies timing constraints related to the access time to the external memory. It must be shorter than the incoming rate of the pixels. Generally speaking, external memory doesn't allow multiple port access. The resultant memory bandwidth limitations make it desirable to avoid all external memories. Consequently, we favor the use of a streaming mode with as little memory storage as possible. We have to take the challenge to use a small amount of memory. In fact by using the internal memory of an FPGA we can avoid all the constraints due to the use of an external one.

There are two kinds of parallelism: data and pipeline [3]. Generally both of them are used at the same time: first we divide the frame into different sections (data parallelism), then each section passes through a processor which executes different tasks sequentially (pipeline parallelism). The algorithm described in this paper uses pipeline parallelism because each frame is entirely computed by one processor; but it's quite particular because two processors run on the same frame at the same time, and the results of one processor are used by the next one, as shown in fig. 2.

III. DETECTION OF SPECULAR REFLECTIONS

A diffuse reflection occurs when the incident ray is reflected in a multitude of angles. In this case, the incident energy is distributed in all the directions of the reflection. It generally occurs at the contact of a granular surface. That is the kind of reflection that permits us to see objects and their shape. A specular reflection, also called specularity, occurs when the incident light is reflected in only one direction. In this case, both the incident and reflected lights have the same energy, in principle without any loss. This energy can glow, especially when the light source is near the surface. This kind of reflection occurs when the surface is smooth. A reflection generally has both specular and diffuse components.

A. Histogram decomposition and criteria for detection

Specularities are by definition regions of an image where pixel intensity is very high and where the color matches the illumination source. Building an image histogram therefore has the potential to assist us in identifying these regions. Three separate histograms can be generated for each of the three colors in the image. For endoscopic images taken inside the body, the dominant color is red and the light source is white. Consequently, intensely white regions generally correspond to specular reflections. Analyzing the red, green and blue histograms for matching high intensity zones has the potential to point to specular regions [1].

It has also been shown that it is possible to use a grey-level image to detect specularities [4]. With a simple thresholding on this image we can detect specularities, because their pixel intensity is independent from other regions. Specularities are more visible in the S (saturation) component of the HSV plan. Consequently, two images are important to achieve good detection: the grey-level image and the saturation image.

The methods described in [1] and [4] use one-dimension histograms. They divide the image into two distinct regions: one where most of the pixels are located and the other where specularities are located. Generally however, these histograms tend to be noisy, which complicates the distinction between the two regions. In other terms, to perform a good thresholding on these histograms they must first be de-noised. Detecting a mass of pixels corresponding to specular reflections can be accomplished by double derivation of the histogram to extract the beginning and end of the specular region (in intensity) [1]. However, this involves many computations and it requires that a complete histogram of an image be stored.

It has also been suggested to use bi-dimensional histograms to perform detection. Specular reflection regions tend to be located in a static region of this histogram [5]. The bi-dimensional histogram is built as follows [6]:

$$m = \frac{1}{3}(r+g+b) \tag{1}$$

$$s = \begin{cases} \frac{1}{2}(2r-g-b) = \frac{3}{2}(r-m), if(b+r) \geq 2g \\ \frac{1}{2}(r+g-2b) = \frac{3}{2}(m-b), if(b+r) < 2g \end{cases} \tag{2}$$

where m is the intensity, s is the saturation, and r, g and b respectively represent the red, green and blue components of the image. Specularities can be identified from the bi-dimensional histogram based on the maximum values of m

and s for the image [6]. They correspond to the region located in the lower right part of the M-S diagram. The relations proposed in [6] tended to produce poor results in the context of endoscopic images. After careful investigation of the parameters with a large quantity of endoscopic images, we found that the following relations reliably identify pixels that are part of a specular reflection. A pixel p will be a part of the specular region if it meets the following conditions:

$$\begin{cases} m_p \geq \frac{1}{2} m_{\max} \\ s_p \leq \frac{1}{3} s_{\max} \end{cases} \quad (3)$$

where $m_{\max}$ and $s_{\max}$ are the maximum intensity of M and S for all pixels in an image, respectively.

Histogram decomposition corresponds to task 1 of our system (fig. 2). Once a pixel is received,

1. its M value is computed by (1);
2. having its M value, its S value is computed(2); and,
3. the $s_{\max}$ and $m_{\max}$ values are updated for each frame.

From a computation point of view, the approach using the bi-dimensional histogram is superior to the one with the one-dimension histogram. No de-noising or histogram post-processing is necessary. This eliminates a further cause of error in the form of rounding in fixed-point calculations. In fact, it is not required to store the bi- dimensional histogram at all, only to compute and track the maximum values of its two components for each frame. Simple thresholding is sufficient to detect specular regions.

B. *Composition of the specular mask*

The composition of the specular mask is part of task 2 in fig. 2; it works as follows. For each pixel p, the values of i_p and s_p are computed with (1) and (2). The relations of (3) are then evaluated to determine whether the pixel is part of the specular mask or not.

In theory, one would have to inspect all pixels from a frame f in order to calculate the $s_{\max}$ and $m_{\max}$ values, then apply (3) to that frame. However, we found that the $s_{\max}$ and $m_{\max}$ values vary little from frame to frame, and hence the values found for frame f allow to find the specular mask of the $f+1$ frame. The specularities tend to be more or less identical between two successive frames. This is true when there is no sudden change of direction of the camera or of the light source. In the worst case the error is limited to a single frame with duration under 17 ms. The obtained mask is a black and white frame, with the white parts indicating the presence of specularities.

C. *Mask enlargement*

The specular mask received from the detection includes only the specular spikes of the frame. It doesn't take into account the specular lobe or camera artifacts at the boundaries of specular regions and the diffuse regions, caused by the direction of the camera. If these components are not included in the specular mask, the correction will be severely compromised. Consequently, mask enlargement is necessary. One approach consists of using an intensity descent [1,5]. However, this algorithm requires a significant amount of memory access and is computationally intensive.

In order to accelerate computations, we propose the following process. The specular mask is inspected with the help of a sliding window [2]. We define a mask enlargement width of n pixels. When a specular pixel is encountered, all pixels within a $n \times n$ window centered on the specular pixel are included in the mask. The number N of buffered lines in the sliding window is given by:

$$N = 2n + 1 \quad (4)$$

The width n of the enlargement depends on the width of the different artifacts we want to include into the mask. We have found that a value of $n = 3$ was adequate for most images. This means that there is a processing delay of 7 lines between the original mask and the enlarged one. Fig. 3 gives an example of a mask enlargement with $n = 1$. The white pixel represents the specular pixel, the bold lines represent the sliding window, and the light grey pixels represent the new specular pixels after successive iterations. The blue pixel is the actual pixel being computed.

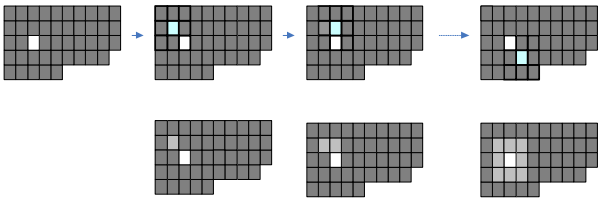

Figure 3. Mask enlargement with *n=1, up: input frame; down:output frame*

IV. CORRECTION OF SPECULAR REFLECTIONS

Correcting a frame consists of removing all the specularities previously detected and replacing them with information obtained from their neighborhood. One of the best ways to perform a good correction is to use an image in-painting algorithm, such as the Navier-Stokes algorithm [7]. However this kind of algorithm uses several loops to pass through a given frame. This requires large amounts of memory and computational effort. We aim to achieve a single frame memory architecture. We therefore propose the following approach, which operates line by line:

- An entire line of the frame is stored.
- For each specular region detected in this line, three data are stored: the value of the pixel before the specular region p_b the value of the pixel after the specular region p_e and the width of the specular region w.
- For each specular region, the linear skew a is calculated:

$$a = \frac{p_e - p_b}{w} \quad (5)$$

- The leftmost pixel in a specular region is given index zero. Pixel p_o is given the value $p_b + a$. The corrected value of all other pixels is given by

$$p_{i+1} = p_i + a \quad (6)$$

At the end of the process there will be a delay of 1 line between the enlarged mask and the corrected frame.

Since linear correction operates only in the horizontal dimension, it is necessary to add correction along the vertical

dimension. This is achieved by passing the corrected frame through a smoothing window which replaces corrected pixels with the average of its neighbors. This is done by using a 3×3 sliding window with 5 passes. Each pass creates a delay of 3 lines, for a total of 15 lines delay.

The total delay of the system is 25 lines of 858 pixels each (with 720 active pixels); this delay of 0.8 ms between the non corrected input frame and the corrected output frame uses 64 KB of the internal memory. The delay is acceptable for real-time operation.

V. Results and Discussion

The system was first developed and implemented with Matlab to adjust parameters and processes and to build a baseline reference. This Matlab implementation included fixed point data types from the start. The system was then described at the register-transfer level with VHDL. Simulation and verification were performed with the help of Modelsim and an automated test bench.

Two systems were in fact implemented with different detection algorithms: the single [1] and bi-dimensional approaches. Table 1 presents resource usage for the implementation of each algorithm after the synthesis process. Bi-dimensional histogram detection uses 10 times fewer resources than the mono-dimensional version and it achieves better results.

TABLE I. Resources used for each hardware implementation on a Xilinx Virtex 2 pro XC2VP30 FPGA

algorithm	Flip-flops (27,392 available)	LUTs (27,392 available)	Brams (2,448 Kb available)
Mono-dimensional histogram detection	6,002 (21%)	10,531 (38%)	162Kb (6%)
Bi-dimensional histogram detection	571 (2%)	1,032 (3%)	108Kb (4%)
Correction	3,641 (13%)	24,044 (88%)	1,018Kb (42%)
Detection and correction	4,212 (15%)	25,076 (91%)	1,126Kb (46%)

Fig. 4 demonstrates the detection and correction of specular reflections. Hardware implementation of mono-histogram decomposition gives an unstable mask because of the computations needed to extract the beginning of the specular region.

The best results come from [7] because the correction is done on two dimensions, instead of one dimension. The specular regions are filled until there is no information to propagate from the boundaries. In other terms, the widths of the boundaries tend to zero. This needs a lot of memory to store multiple frames of the same picture.

The correction algorithm proposed in this paper works well when the specular region is entirely enclosed inside an object (Fig. 4). When it's located at the boundary of two different objects, the information coming from one object can be propagated to the other one.

After the optimization of placing and routing process, the system uses 91% of the available slices of a XC2VP30 FPGA. The maximum operating frequency is 32 MHz, which exceeds the minimum required of 27 MHz for real-time operation.

VI. Conclusion

This paper has presented a method and architecture to implement a processor able to detect and correct specularities in NTSC endoscopic videos. This is done with two parallel tasks in a streaming processing mode. Bi-dimensional histogram decomposition is computed to detect the specularities. It has the advantage of using few memory and computation resources without compromising the quality of the resulting image. The correction is done in two steps, a linear correction and a smoothing process. The system functions in real time and doesn't require an external memory.

References

[1] C. A. Saint Pierre, J. Boisvert, G. Grimard and F. Cheriet, "Dectection and Correction of Specular Reflections for Automatic Surgical Tool Segmentation in Thoracoscopic Images," *Machine Vision and Applications Journal* 2007, in press.

[2] C. T. Johnston, K. T. Gribbon and D. G. Bailey, "Implementing Image Processing Algorithms on FPGAs," *Proc. of the 11th Electronics New Zealand Conference*, pp.118-123, November 2004.

[3] A. Downton and D. Crookes, "Parallel architectures for image processing," *Electronics & Communication Engineering Journal*, vol.10, no.3, pp.139-151, Jun 1998.

[4] M. Gröger, W. Sepp, T. Ortmaier, G. Hirzinger "Reconstruction of Image Structure in Presence of Specular Reflections," Mustererkennung 2001: DAGM2001, Munich, Germany, Sept 2001.

[5] J. Bhattacharyya, "Detecting and Removing Specularities and Shadows in Images", Master's thesis, McGill University, June 2004.

[6] F. Ortiz and F. Torres, "Automatic detection and elimination of specular reflectance in color images by means of MS diagram and vector connected filters," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol.36, no.5, pp. 681-687, Sept. 2006.

[7] M. Bertalmio, A.L. Bertozzi and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol.1, pp I-355- I-362, 2001.

[8] Jing, Chen; Yongtian, Wang; Yue, Liu; Dongdong, Weng, "Navigating System for Endoscopic Sinus Surgery Based on Augmented Reality," *IEEE/ICME International Conference on Complex Medical Engineering,* pp.185-188, May 2007.

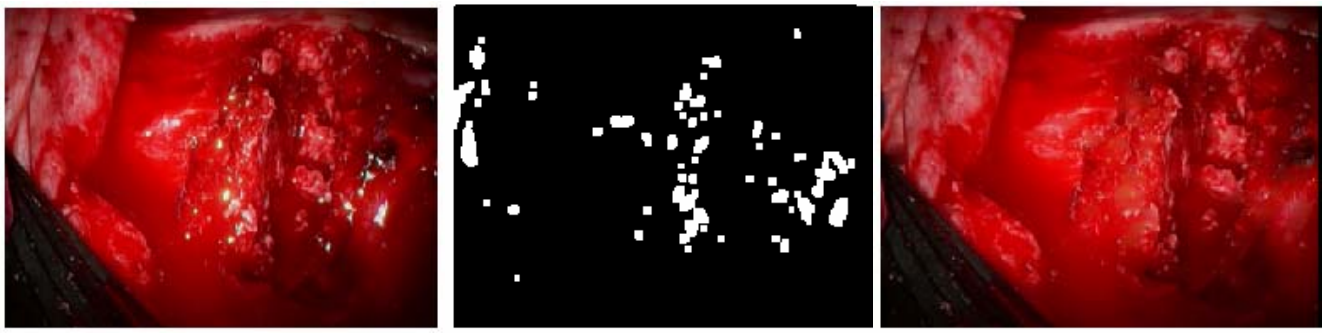

Figure. 4 Hardware implementation of the specular mask (middle) and the correction (right) of the left image