Video Processing SoC for Low-Latency Real-Time Interactive Real-Brush Painting System

Luc Claesen, Peter Vandoren, Tom Van Laerhoven, Andy Motten, Domien Nowicki,
Frank Van Reeth, Eddy Flerackers
EDM Expertise Center Digital Media,
University Hasselt, tUL, IBBT
Wetenschapspark 2, B3590 Diepenbeek Belgium
{firstname.lastname@uhasselt.be}

***Abstract*—To reduce the learning curve and enhance expressiveness, Human Computer Interfaces continuously improve to mimic object interaction in the real environment. A novel digital painting system has been developed that provides the artist with the expressiveness of virtual painting by using real and wet brushes. The kernel of the system is an interactive canvas powered by an SoC-based real-time video processing of the detailed wet brush-canvas contact image. To enable realistic and fast painting interaction, low latency video processing from smart-camera up to painting rendering is key. This paper presents the real-time video processing SoC system.**

Keywords: System-on-Chip, SoC, FPGA, digital painting, active canvas, Video Processing, human-computer interface, real-time, Smart Camera, embedded video, FTIR.

I. INTRODUCTION

Since the introduction of graphic computer displays, software painting applications have been developed [1]. Most current computers provide software where simple paintings can be made by means of a computer mouse, a touch screen or a stylus/tablet input device. Usually a color bit-map file is generated by the input device by adding and combining one or more bit-map layers of the canvas and the virtual brush at the mouse pointer or pen-stylus cursor.

A. Active-canvas digital paint methods.

Realistic models for digital painting have recently been developed which mimic the physical painting process using a detailed simulation of the complex interaction between brushes and the paint canvas. These are so-called "active-canvas" methods [2,3,4,5]. They model the paint as a solvent fluid that can flow and evaporate and that contains color pigment models and fixation binder glue.

B. Artist-Computer Interfaces for Digital Paint systems.

Although computer mice have proven their usefulness in a lot of daily computer tasks, the expressiveness for digital painting is rather limited. They only record relative movements and have little expressiveness for pressure input. Therefore professional artists often use tablets and stylus pens. These systems have a good absolute accuracy with respect to the drawing tablet and also provide a measurement of the drawing force along the axis of the pen stylus shaft. This allows paint programs to model the force exercised on the pen tip while drawing and consequently generating thinner or thicker pen strokes depending on the force employed by the artist.

Mueller [9] describes a real-time painting system based on frustration of internal reflected light in a prism. The light in the prism is generated, via an optical setup, by the scan signal of a CRT (Cathode Ray Tube) image. The frustrated light generated by a drawing utensil can be detected by a (photo multiplier) light sensor in a synchronous way with the CRT scan signal. This enables a quasi real-time brush detection. The display of the painting result is not at the same location as the drawing surface, which is a hindrance for the artist.

Greene [6] introduced the drawing prism, commercialized under the name OptiPaint [7]. The rendering of the painting result on the screen is separate from the location of the drawing surface [6,7] just like the method of Mueller [9].

Carver Mead et al. [8] proposed a paintbrush stylus sensed by a capacitive sensor array. Because of the capacitive sensing mechanism only electrically conductive brushes can be used in this system.

Electro-magnetic tablets are the most widely used input devices for paint-systems. By using a layer of optically transparent wires on top of an LCD display, the Wacom Cintic system [10] integrates the input tablet with the drawing screen. This provides direct feedback of the drawing result under the pen tip. Because tablet based systems use stiff styluses, painting with a stiff stylus is different from painting with real brushes with flexible tufts made of camel, hog, squirrel hair or other natural or synthetic fibers. In Western painting and Chinese calligraphy, the specific movement and deformation of the brush tuft is crucial for achieving specific effects. Although some tablet systems [10] provide co-located drawing input/painting display, they suffer from the distance between the drawing plane and the display plane which causes a parallax effect. Depending on the relative position of the artists, the pen tip and its drawing result on the screen will be different.

The IntuPaint system [12] uses electronic brushes with bristles made of optically transparent fibers. An infrared light source inside the brush propagates light through the transparent fibers by means of total internal reflection. The light exits at the bristle tips. When IntuPaint brushes are in contact with a diffuser screen, the tuft footprint and position can be imaged by an infrared camera behind the screen. In IntuPaint, the diffuser screen is used to display the result of painting, thereby providing co-located input/display. By using a brush with bristles, an artist can exploit the deformation of the brush tuft during drawing by brush movements, inclination and pressure on the canvas. Because of its infrared light emission operating principle, the IntuPaint system requires specially built brushes and drawing tools.

All previous methods still limit artists in their expressiveness in comparison to traditional painting with brushes and paint. To solve this problem, the authors have introduced the FluidPaint system [13]. Both the IntuPaint and the FluidPaint system are built on top of physical based painting simulation software [4] running on a high-end GPU powered PC. The novelty of FluidPaint is that it uses *real brushes* on a *co-located painting input/display canvas* surface. To enable the real-time and low-latency virtual painting, a dedicated video processing SoC architecture has been developed and is presented in this paper.

In Section II, the system setup of the FluidPaint virtual painting system with real brushes is presented. In Section III the usage of the Video Processing SoC is introduced. The SoC hardware architecture and prototype is presented in Section IV. Section V formulates conclusions and further work.

II. Virtual Painting With Real-Brushes.

FluidPaint is a novel digital painting system that operates with real brushes. In this section the operation principle of FluidPaint is briefly introduced. The reader is referred to [13] for a more in-depth presentation and user tests.

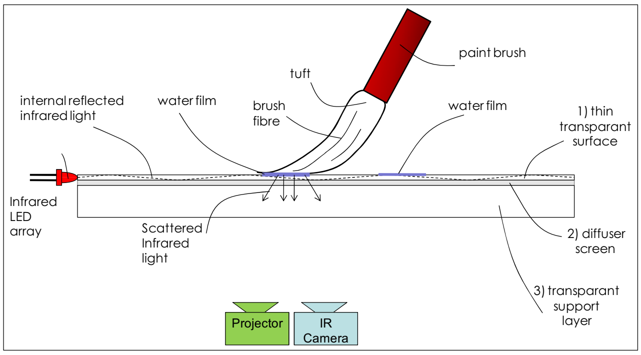

Figure 1. Operation principle of the FluidPaint digital painting system.

The FluidPaint paint canvas constitutes the key component of the system as shown in Fig. 1. The top layer consists of a 0.6mm thick transparent plate. On the four sides there is an array of 950nm LEDs, introducing IR (infrared) light inside the transparent layer. This IR light is propagated inside the layer by means of total internal reflection and normally exits the layer at the other side. The second layer is a diffuser screen that is used for displaying the painting results by means of a projector. A third thick transparent support layer provides the necessary mechanical strength for supporting the canvas.

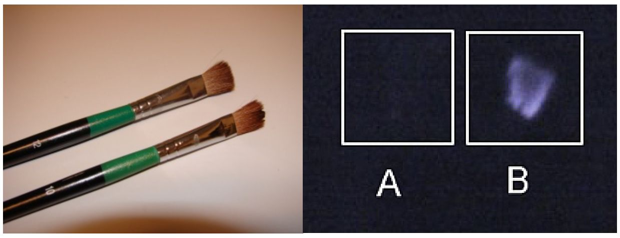

Figure 2. Left: dry brush A (12mm) and wet brush B (10mm). Right: the infrared footprint of the two brushes. Notice the clear footprint image for the wet brush B caused by frustrated total internal reflection. Dry brush A does not generate a footprint image.

When a wet brush makes contact with the top layer, as illustrated in Fig. 1, the IR light inside the top layer is not internally reflected anymore and can propagate outside the layer and propagate inside the water in the wet brush until it arrives at the brush bristles. Here the IR light will be scattered in different directions according to the bristle structure. An IR camera placed below the screen can capture this IR image. It is in fact a footprint of the brush contact surface as illustrated by brush B in Fig. 2.. When a dry brush is put into contact with the surface layer, there is nearly no optical contact and consequently the light inside the layer remains internally reflected and is not frustrated. Consequently no image is visible by the IR camera as is illustrated by brush A in Fig. 4.. When using wet brushes, wet traces are left on the drawing surface. As shown in Fig. 1 these water films do not frustrate the internal IR light reflection. At the interface of the surface layer and the water film the IR light leaves surface and propagates further inside the water film under a similar angle. When it reaches the top of the water film it is internally reflected again and propagates back into the transparent layer.

This input method of painting with real and wet brushes results in a feeling and expressiveness like in real-world painting. The IR camera only images the brush contact surface. During real painting it is also only the wet contact surface that really matters. The image of the contact surface images the real brush and bristle structure in the contact zone. Such a brush footprint can be very well used in physical model based painting systems [4]. This enables an artist to express very small nuances due to the specific brush movements and complex tuft deformation during the act of painting or caligraphy. The artist directly sees the result of the painting under the brush as illustrated in Fig. 3. The bristle structure dependent stroke output is clearly visible.

Although also based on the principle of total internal reflection, multi-touch systems as introduced by Han [17] are not directly usable for painting like the 3-Layer structure of FluidPaint. These multi-touch systems usually consist of a ~1cm thick transparent acrylic layer with IR leds on the side. On top of this acrylic layer there is a "compliant layer", usually made of textured silicone. On top of the compliant layer is a diffuser screen. Touch is detected by the IR light frustration of the contact points of the middle silicone with the bottom acrylic layer. Using a brush on such a system requires an unnatural high force from the brush to make contact through the diffuser screen and through the silicone layer with the acrylic layer. No detailed footprint as in FluidPaint is possible with such a system. Multi-touch systems without compliant layer have also been realized. They consist of a ~1cm thick acrylic layer with a diffuser screen under it. Although here wet brushes could be used like in FluidPaint, the distance (~1cm) between the brush contact surface and the diffuser screen is too high, resulting in a very unclear image below the diffuser screen and would reintroduce an undesired parallax.

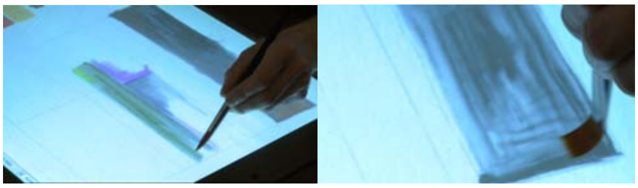

Figure 3. Interactive Painting in FluidPaint with real wet brushes.

III. VIDEO PROCESSING ARCHITECTURE.

The first prototype of the FluidPaint system [13] made use of a standard machine vision camera. It was a PointGrey GRAS-20S4C camera with an IEEE 1394b FireWire interface to the host PC. Using standard cameras has the advantage of fast prototyping. The disadvantage is however that in applications such as digital painting, there are very stringent real-time requirements, both on the overall processing time as well as on the latency between brush stroke input and the processed display reaction. A standard machine vision camera sends full images to the PC, where further image processing is to be done to detect the brush footprint images and positions. It is well known that streaming video data and real-time image processing are very computation intensive. In addition, standard cameras add delays between the capture of the image in the camera sensor and the delivery of the image processed results to the painting application. This delay which usually consists of several frame periods, causes a latency between painting with a brush and displaying the result on the screen. This is noticed by the fact that the paint on the canvas screen does not directly follow fast brush movements.

As the application PC is already very occupied with the physical model based paint simulation software, the combination with the camera image streaming communication and processing limits the real-time simulation effects.

A dedicated real-time video processing SoC architecture can perform the required image processing in hardware and reduce the delay time from image capture to processing. The direct processing of the image data in hardware can avoid the use of unnecessary frame buffers in the camera and the PC. In an SoC, frame buffers can be reduced to the absolute minimum and can directly be employed for the required image processing at hand. In case of a controlled environment lighting, frame buffers could even be avoided.

An SoC architecture also allows for a direct per-frame camera control without lost frames. Hereby the SoC can directly change the camera field of view, shutter times, gains, black level calibration etc.

IV. SOC HARDWARE ARCHITECTURE.

A. *Image Processing Pipeline*

An infrared camera captures the image of the contact of the wet brush with the canvas as shown in Fig. 1. Image processing [14,15] and segmentation enable the accurate determination of the brush location on the canvas and the determination of the brush footprint image. The segmented brush footprint image is the input for the physical model-based paint simulation software [4].

The first steps in the image processing isolate and enhance the image of the footprint. In order to obtain an accurate and stable position determination, the center of gravity of the footprint is determined. The footprint image around this center of gravity is transmitted to the painting application on the PC.

The co-location of the brush input/canvas screen requires a transformation of camera coordinates to screen coordinates of the projector. Distortions due to the other placement of the camera, due to lens distortions (cushion effect), due to different pixel densities etc. need to be compensated. This camera image rectification is done by a grid of calibrated control points in which camera coordinates are transformed to projector coordinates by means of bilinear transformation [15].

B. *SoC Video Processor Architecture*

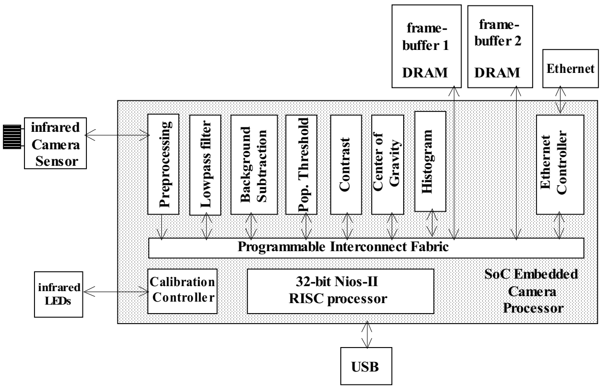

Figure 4. SoC Brush Footprint Video Processing Architecture

The architecture (Fig. 4) consists of a programmable interconnect fabric that allows the flexible arrangement of image processing operations in a pipeline. This architecture supports the operators required by the Image Processing Pipeline: camera normalization preprocessing and calibration, lowpass Gaussian filtering, Background subtraction, Population based Thresholding, Contrast enhancement, Center of Gravity Calculation, Histogram calculation. The 5x5 lowpass filter and the 5x5 population thresholding operators use on-chip line buffer memories. Two independent DRAM frame buffer based memories can be used. A first frame buffer can store the background image for background subtraction. The background image is adaptively updated by means of a temporal IIR (Infinite Impulse Response) image filter. This is useful in environments with changing infrared background lighting. A second frame buffer can be used to store the incoming image. After the location of the brush has been determined, the brush position and footprint image can be sent to the host PC via a direct Ethernet link.

The video processor SoC and all of the image processing operators and communication are controlled by a 32 bit RISC processor. The processor can also communicate via a USB link to the host PC. In this way the application PC can indirectly control all of the functions in the video processing system.

C. *Prototype Implementation*

The SoC architecture has been designed using Verilog and implemented on an Altera Cyclone II EP2C70 FPGA.

A 5 mega pixel (2592x1944) digital camera is used with an infrared sensitive lens and 950nm infrared bandpass filter. The camera can be programmed in resolution and field of view. The camera has on-chip 12-bit ADC and is used in our application at its maximum parallel output rate of 96 MHz. The frame rate is determined by this maximum output speed, by the resolution chosen and by the shutter width. The full resolution frame rate is 15 frames/sec. At 640x480 VGA resolution frame rates of 150 frames/sec are possible. In our application we use a camera resolution of 1280x1024 for the brush image capture. This results in a frame rate of 40 frames/second. This is a tradeoff between footprint resolution and frame rate.

Using the image processing pipeline, described in the previous section, the brush position is determined by a real-time center-of-gravity calculation of the segmented footprint image. This is calculated immediately after the last pixel of a frame has been received. During the vertical blanking period of the camera, the footprint image around the center-of-gravity is retrieved from memory and sent to the application PC as UDP packets over the Ethernet connection.

The synthesis results with Quartus II 9.0 are shown in the Table I:

TABLE I. SYNTHESIS RESULTS OVERVIEW

Description	
Total logic elements	12,867
Total combinational functions	11,547
Dedicated logic registers	4,891
Total registers	5,012
Embedded Multiplier 9-bit elements	7
Total memory bits	724,548

A 50 MHz NiosII/e processor is used as a 32 bit RISC processor for the overall control. For the Gaussian lowpass filtering only power-of-two coefficients are used to economize on multipliers. Completely programmable coefficients would also be possible as in the current prototype architecture only 2% of the available multipliers are used (7 / 300).

The frame buffers for the storage of the current image and background image are implemented in ISSI DRAM memories. In case of controlled lighting environments both frame buffers could be left away.

V. CONCLUSIONS AND FURTHER WORK

By employing a direct hardware implementation of the video processing pipeline, not only real time brush/canvas contact detection can be done, but also a low latency between brush stroke input and resulting painting display is obtained. The SoC architecture has been designed in Verilog, and is amenable to be integrated together with a CMOS image sensor as a smart camera.

REFERENCES

[1] Smith, A.R. . Digital Paint Systems: An Anecdotal and Historical Overview, IEEE Annals of the History of Computing, 23, 2(2001), pp. 4-30.

[2] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer, D.H. Salesin, "Computer-generated watercolor", ACM Siggraph, Proc. 24th ann. conf. on Computer Graphics and Interactive Techniques, pp. 421-430, 1997.

[3] William V. Baxter, Jeremy Wendt, and Ming C. Lin, "IMPaSTo: A realistic, interactive model for paint." In Stephen N. Spencer (ed.), Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, June 5-7, 2004.

[4] T. Van Laerhoven, F. Van Reeth, "Real-time simulation of watery paint", CASA 2005, Volume 16, Issue 3-4, July 2005, pp. 429-439.

[5] Nelson S.-H. Chu and C.-L. Tai, MoXi: Real-Time Ink Dispersion in Absorbent Paper, ACM Transactions on Graphics (SIGGRAPH 2005 issue), Vol. 24, No. 3, August 2005.

[6] R. Greene, "The Drawing Prism: A Versatile Graphic Input Device", ACM SIGGRAPH Computer Graphics, Proceedings of ACM SIGGRAPH 85, San Francisco, CA, July 22--26, 1985, pp. 103-110.

[7] http://www.optipaint.com/

[8] Carver Mead, Ralph Wolf, Timothy P. Allen, "Paintbrush stylus for capacitive sensor pad", US Patent 5,488,204 Jan. 30, 1996.

[9] Robert E. Mueller, "Direct Television Drawing and Image Manipulation System", US Patent 3,846,826, Nov. 5, 1974.

[10] Wacom Cintic. http://www.wacom.com/cintiq

[11] P.F. Ruedi, P. Heim, S. Gyger, F. Kaess, C. Arm, R. Caseiro, J.L. Nagel, S. Todeschini, "An SoC combining a 132db QVGA pixel array and a 32b DSP/MCU processor for vision applications", Proc. IEEE Solid-State Circuits Conference - Digest of Techincal Papers, ISSCC 2009, 8-12 Feb, 2009, pp. 46-47, 47a.

[12] Peter Vandoren, Tom Van Laerhoven, Luc Claesen, Johannes Taelman, Chris Raymaekers, Frank Van Reeth, "IntuPaint: Bridging the Gap Between Physical and Digital Painting" Proc. of TABLETOP 2008, pp. 71 - 78, IEEE, ISBN 978-1-4244-2897-7, Oct. 2008.

[13] Peter Vandoren, Luc Claesen, Tom Van Laerhoven, Johannes Taelman, Chris Raymaekers, Eddy Flerackers, Frank Van Reeth, "FluidPaint: an Interactive Digital Painting System using Real Wet Brushes" Proc. of ITS2009, pp. 53 - 56, ACM, ISBN 978-1-60558-733-2, Nov. 2009.

[14] Rafael. C. Gonzalez, Richard E. Woods, "Digital Image Processing", Prentice Hall, 2002, ISBN 0-201-18075-8.

[15] Kenneth. R. Castleman, "Digital Image Processing", Prentice Hall, 1996, ISBN 0-13-211467-4.

[16] Wayne Wolf, Burak Ozer, Tiehan Lü, "Smart Cameras as Embedded Systems", IEEE Computer, Vol. 35, Issue 9, Sept. 2002, pp. 48-53.

[17] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In Proc. UIST 2005, ACM Press, 115-118.