Embedded registration of visible and infrared images in real time for noninvasive skin cancer screening

Fabién Inostroza, Silvana Díaz, Javier Cárdenas, Sebastián E. Godoy, Miguel Figueroa*

Department of Electrical Engineering, Universidad de Concepción, Concepción, Chile

ARTICLE INFO

Keywords:
Long-wave infrared
Image registration
Embedded systems
Video processing
FPGA

ABSTRACT

We present an embedded system architecture that implements real-time multimodal registration to enable dual-camera spatio-temporal feature extraction in a skin cancer screening application. We test the system on a combination of visible and long-wave infrared image sequences, but it can be easily extended to setups operating in different sections of the spectrum. Image registration is performed by matching common features between each frame of a visible image to each frame of an infrared image sequence to estimate a projective transformation between them. The parameters of this transformation are estimated recursively on line with the video, thus enabling image registration in real time. The algorithm is implemented using a combination of embedded software and dedicated hardware units on a heterogeneous reconfigurable system-on-a-chip. The hardware performs feature detection and extraction, while the software estimates the transformation parameters and maps each visible video frame onto the infrared image coordinates. Implemented on an FPGA, our prototype runs at 540 frames per second with a 135 MHz clock, consumes 1.8 W and utilizes 29% and 54% of the logic and multiplier resources of the chip, respectively.

1. Introduction

The incidence of skin cancer has increased steadily worldwide, and is currently the most frequent kind of cancer in the United States and Australia [1]. As of today, the lack of noninvasive and reliable techniques to detect it at an early stage has become one of the critical barriers for opportune diagnosis. This has motivated the search for new screening methods, of which infrared (IR) imaging has proved to be a suitable technique due to the angiogenesis process that occurs in cancerous cells [2]: Unlike healthy tissue, a cancerous lesion has a higher density of capillaries due to the increased growth of blood vessels as the cancer progresses. This implies that the thermal response of a cancerous lesion differs from the thermal response of healthy tissue, thus creating an opportunity for thermal IR imaging to become a key step in cancer diagnosis [3].

A dual visible/IR camera setting was proposed in [4] to extract spatial and spectral (color) features from the lesions using a red-green-blue (RGB) image in the visible spectrum, whereas thermal and temporal features are extracted from the IR video. The use of these features from the perspective of statistical inference enables the detection of cancerous lesions with high sensitivity and high specificity [5]. Because patients with malignant lesions have a survival rate of almost 100% if they are diagnosed early, this screening method creates a powerful tool to diminish the impact of skin cancer on society. However, one of the first problems to address in this specific application is the spatial and temporal alignment of the images acquired simultaneously with both cameras. Although precise optomechanical alignment is possible for a dual-camera setup, it significantly increases the cost of the system and restricts the movements of the patient. In this application, subject movement is limited and the procedure time is in the order of only a few minutes, therefore digital image registration is a preferable solution. However, traditional registration methods do not typically produce good results in this case, because the RGB and IR images are acquired at different wavelengths and close distance to the skin, thus they lack distinguishable common features to guide the registration process.

There are several articles describing the image registration problem and some solutions, such as those based on the sum of squared-differences [6] or in mutual information [7,8]. The limitation of these methods is that they rely on correlations that exploit texture similarities, which are not always present, especially in multimodal images. Other algorithms are based on image features, such as Scale Invariant Feature Transform (SIFT) [9,10] or Speeded-Up Robust Features (SURF) [11]. A few methods were applied in the analysis of visible and thermal images to detect plant illnesses [12] or to estimate the optical
flow from the affine transformation [13]. Some applications have been implemented in the Compute Unified Device Architecture (CUDA) framework [14] and in commercial devices such as FLIR ONE [15], which fuses visible and infrared images. However, most of these works rely on manually-selected features between the images of different spectra [16] (which can be inefficient in many applications) or use edge detection in both images and compute control points for the image registration [17]. This last approach is limited to images based on straight linear segments, which are not prevailing for many applications, and are not very accurate because they are applied one frame at a time, thus the number of matching points is often insufficient to obtain an accurate result.

Due to their high computational cost, image registration algorithms are frequently executed off-line, first acquiring the video and then processing the data on a computer. Some works have explored the use of Graphics Processing Units (GPUs) [14] to accelerate image registration, but the processing is still performed off-line with the added problem of having high power requirements, which complicate the design of a portable screening instrument. These problems become more prominent for multimodal registration, because algorithms that find and correlate features in different sections of the spectrum require more computational power.

In this manuscript, we propose an embedded implementation of a semisupervised algorithm for multimodal image registration between visible and long-wave infrared (LWIR) video, intended to be used in an on-going research project to design a portable and low-cost skin cancer screening device. The paper describes a hybrid hardware/software architecture and its implementation on a Xilinx Z-7010 System on a Chip (SoC). The hardware component of the proposed implementation performs feature detection and extraction in both images, while the software component estimates the geometric transformation that maps each image onto the other, and applies the transformation to the infrared video frames. A graphical interface implemented in software and hardware enables a user to operate the system and visualize the resulting images in real time.

The manuscript is organized as follows: Section 2 describes the general setup of the proposed skin cancer screening instrument, and provides the context where image registration must take place. Section 3 describes the registration algorithm that is the basis of the proposed solution. In Section 4, we present the architecture of our multimodal image registration system, and analyze the numerical tradeoffs of our implementation. In Section 5 we discuss experimental results, and finally summarize our conclusions in Section 6.

2. Skin cancer detection based on thermal recovery

Fig. 1 shows the device setup and data flow of the skin cancer screening application that motivates our multimodal registration system. In this approach, both thermal and visible images are used to classify a lesion as malignant or benign. First, a cooling unit is used to lower the temperature of the lesion and the surrounding skin tissue, and then the infrared camera (previously calibrated using black-body radiators) is used to estimate the temperature of the skin while it recovers its normal temperature. Finally, a classification algorithm uses the thermal recovery curve of the lesion and its surrounding tissue to determine whether the lesion is benign or malignant.

Because the lesion is not clearly visible in the thermal image, a visible RGB image is first used to automatically detect the region of interest, and a segmentation algorithm labels each pixel as belonging to the lesion or to the surrounding tissue. As Fig. 2 shows, the cameras are positioned close together in a rigid platform to minimize their difference in perspective. However, because the images are taken from different angles and with different devices, image registration is necessary to establish a correspondence between the pixels in the RGB image (which provide the location of the lesion) and the infrared image (which provides thermal information during recovery). Moreover, because the patient typically experiences involuntary movements during the experiment, this registration needs to be performed for each frame of the infrared video using a projective transformation.

Fig. 3 shows that the difference between the thermal recovery curves of the lesion and the surrounding tissue are quantifiably different when the lesion is benign and when it is malignant [4]. This difference is not always evident as the one observed in the present figures. Thus, we will perform the classification of a measured curve as belonging to a benign or a malignant lesion following the continuous-time Karhunen–Loève approach in conjunction with a Neyman–Pearson decision rule [18], in which an optimal comparison threshold is selected so as the theoretically calculated detection probability is maximized while keeping the false-alarm probability below a prescribed level of our choice [19].

The RGB and IR images shown in Fig. 1 show a white plastic marker placed around the skin area to be analyzed. This marker introduces visual features common to both images, which are used by the registration algorithm during the procedure. The marker is necessary because the multimodal registration algorithms discussed in Section 1 require distinguishable features in the RGB and IR images in order to assess the quality of the alignment. However, as Fig. 4 shows, this is not the case for skin images acquired from a short distance. Fig. 4(a) and (b) show two images acquired in a large indoors scene, using an RGB and an IR camera, respectively. Even though the images were acquired at different wavelengths, it is possible to find distinguishable features that are common to both images, which multimodal registration algorithms can use effectively to guide the registration process. In contrast, Fig. 4(c) and (d) show close-up images of human skin taken with the same cameras. In this case, the skin tissue does not exhibit prominent features common to both images that could be used to perform registration. In fact, even a black dot painted on the skin is partially lost in the IR image, and its contrast varies with temperature. The white plastic marker shown in Fig. 1 frames the region of interest and introduces new features in the scene, which can be easily detected in both the visible and IR spectra due to the shape, color and temperature of the marker.

3. Registration algorithm

Our algorithm registers an RGB image and an LWIR video frame by matching corresponding points of interest in both images. The algorithm registers each image in the LWIR video to a single RGB image acquired at the beginning of the procedure, compensating for global motion of the area inside the plastic marker that defines the region of interest. The registration algorithm uses the features provided by the marker; more precisely, it identifies the inside corners of the marker in the RGB and LWIR images, and uses the spatial coordinates of these corners to estimate the parameters of a projective transformation that aligns the images. In our experimental setup, the subjects were required to sit in a comfortable position during image acquisition in an environment with controlled diffuse illumination. The images acquired with the camera setup of Fig. 2 are processed to estimate the projective transformation. Then, multimodal registration is performed by mapping each pixel in the RGB image to the LWIR image space using the projective transformation. Because the transformation normally yields noninteger coordinates in the LWIR image, we use bilinear interpolation on the measured pixels in the thermal image to estimate the skin temperature at each point of interest.

Fig. 5 shows that each marker corner in the RGB image is paired to a corresponding corner in the thermal image. We use a minimum distance criterion to perform this association, referencing the corner coordinates to their centroid in each image. We use the Manhattan distance in our hardware architecture to simplify its implementation. The rest of this section describes the algorithm used to detect the corners and estimate the parameters of the transformation.

Let \( y_j \) and \( x_j \) denote the spatial coordinates of a point on the RGB image and an LWIR video frame, respectively, for the \( j \)th point of interest at the \( k \)th frame of the LWIR video. As illustrated in Fig. 5, we use
Fig. 1. Data flow of the skin cancer screening application.
a plastic marker that has the property of having high contrast in both the visible and the LWIR images. Therefore, the identification of the corners of the marker on each image enables us to estimate the mapping between the spatial coordinates $y_j$ and $x_j (k)$. The mathematical definition of $\phi^k$ is given later. For each LWIR frame, this mapping must be updated. In this work, we compute this update using the Stochastic Gradient Descent (SGD) method, which converges to a neighborhood of the direct least-squares solution, uses comparatively low computational resources, and can be implemented online.

Algorithm 1 describes the registration algorithm, which is based on the work presented in [4]. The mapping is initialized as the identity transformation. We used a Harris Corner Detector (HCD) [20] to detect the inside corners of the marker in both images. The HCD selects a pixel as a corner if the spatial gradient of the image at that pixel location exceeds a threshold in two orthogonal directions. In the HCD, the intensity at location $E_{i,j}$ in the image can be estimated by the matrix equation:

$$E_{i,j} \approx [i \ j]M_{\ [i \ j]},$$

where

$$M = \left[ \begin{array}{cc} \langle \frac{\partial I}{\partial u} \rangle & \langle \frac{\partial I}{\partial v} \rangle \\ \langle \frac{\partial I}{\partial u} \rangle & \langle \frac{\partial I}{\partial v} \rangle \end{array} \right] = \left[ \begin{array}{cc} A & C \\ C & B \end{array} \right].$$

Here, $\langle \cdot \rangle$ indicates the weighted sum of a Gaussian window $w_{u,v}$ and $I_{u,v}$ is a window centered at location $(u,v)$ of the image $I$ under test. In order to quantify the changes in intensity, we use the trace and the determinant of the matrix $M$ to compute the response $H$ of the HCD.
F. Inostroza et al.  

Microprocessors and Microsystems 55 (2017) 70–81

\[
H = \det(M) - \epsilon \cdot \text{trace}(M)^2 = AB - C^2 - \epsilon/(A+B)^2,
\]

where \(\epsilon\) is a positive parameter with a suggested value of 0.04 [21]. A pixel is declared a corner if the value of \(H\) is a local maximum within a 3 \(\times\) 3-pixel neighborhood. The HCD has the important property of being insensitive to scaling, translation and rotation of images. In order to guarantee that the corners detected by the HCD are indeed the inside corners of the marker, the algorithm runs on a region of interest (ROI) manually selected by the user in the RGB and the first IR video frame. The ROI must enclose the inside corners of the marker, and exclude other parts of the image that could render undesired results, such as the outside corners.

During registration, the projective transformation \(A^k\) maps the coordinates of each pixel in the RGB image onto the \(k\)th frame of the LWIR video. In homogeneous coordinates, \(A^k\) is defined as:

\[
A^k = \begin{bmatrix}
\theta_0^k & \theta_1^k & \theta_2^k \\
\theta_3^k & \theta_4^k & \theta_5^k \\
\theta_6^k & \theta_7^k & 1
\end{bmatrix},
\]

where the parameters \(\theta = [\theta_0^k \theta_1^k \theta_2^k \theta_3^k \theta_4^k \theta_5^k \theta_6^k \theta_7^k]^T\) are estimated by the SGD algorithm from the output of the HCD, and \(T\) represents the matrix transpose operation. The elements \(\theta_0^k, \theta_1^k, \theta_2^k\) and \(\theta_3^k, \theta_4^k, \theta_5^k\) specify translation, \(\theta_6^k\) and \(\theta_7^k\) specify linear scaling, \(\theta_4^k\) define the shear, and \(\theta_1^k\) and \(\theta_2^k\) define the nonlinear effect of projection. At the \(k\)th video frame, we map the position of each pixel of the initial RGB image at coordinates \(t = [t_x \ t_y]^T\) using \(A^k\) to obtain their coordinates \([s_x \ s_y]^T\) in the IR video space:

\[
s_x = t_x + t_y s_0 - t_y s_1,
\]

\[
s_y = t_y + t_x s_0 - t_x s_1.
\]

Eq. (4) shows that we require at least four points (i.e., four pairs of coordinates) to solve for \(\theta\). Assuming we have four of the corners in our marker in both the visible and the kth frame of the LWIR video, which we have denoted as \(y_j\) and \(x(k)\), respectively, with \(j = 1, ..., 4\), Eq. (4) can be recast as

\[
\begin{bmatrix}
x_1(k) \\
x_2(k) \\
x_3(k) \\
x_4(k)
\end{bmatrix} =
\begin{bmatrix}
x_1^j(k) \\
x_2^j(k) \\
x_3^j(k) \\
x_4^j(k)
\end{bmatrix} +
\begin{bmatrix}
x_1(k) - x_1^j(k) \\
x_2(k) - x_2^j(k) \\
x_3(k) - x_3^j(k) \\
x_4(k) - x_4^j(k)
\end{bmatrix} =
\begin{bmatrix}
x_1^j(k) \\
x_2^j(k) \\
x_3^j(k) \\
x_4^j(k)
\end{bmatrix} +
\begin{bmatrix}
x_1(k) - x_1^j(k) \\
x_2(k) - x_2^j(k) \\
x_3(k) - x_3^j(k) \\
x_4(k) - x_4^j(k)
\end{bmatrix} =
\begin{bmatrix}
x_1^j(k) \\
x_2^j(k) \\
x_3^j(k) \\
x_4^j(k)
\end{bmatrix} +
\begin{bmatrix}
x_1(k) - x_1^j(k) \\
x_2(k) - x_2^j(k) \\
x_3(k) - x_3^j(k) \\
x_4(k) - x_4^j(k)
\end{bmatrix}.
\]

It was mentioned before that the estimate of the mapping parameters \(\theta\) are updated using the SGD algorithm [22] in each new video frame, in order to minimize the estimation error. At video frame \((k - 1)\), the estimation error function \(I\) is

\[
I(\theta^{k-1}) = \frac{1}{2} \|b - A(\theta^{k-1})\|_2^2.
\]

where \(b\) denotes the coordinates of the corners in the RGB image. Consequently, the update rule for \(\theta^k\) for iteration \(k\) is

\[
\theta^k = \theta^{k-1} + \eta(A^{k-1})b\|
\]

where \(\eta\) is a diagonal matrix which contains the empirically-obtained learning rates \(\eta_i\) for each parameter \(\theta_i\), defined as \(2 \times 10^{-4}\) for \(\eta_x, \eta_y, \eta_z\) and \(\eta_{translations}\) for \(\eta_{translations}\) and \(\eta_{shear}\) for \(\eta_{shear}\).

Finally, the position of the ROI in the IR video must be updated for each video frame, in order to track interframe marker displacements.
cause by involuntary patient movements. Otherwise, the position of the inside corners of the marker could shift outside of the ROI selected at the beginning of the procedure. To update the position of the ROI, we simply align its centroid with the centroid of the corners detected in the previous video frame, as expressed by:

\[ c(k) = (1 - \alpha)c(k-1) + \alpha \mathbf{x}(k-1), \]

where \( \alpha \) is a parameter equal to 0.5 in our case, \( c(k) \) is the centroid of the ROI, and \( \mathbf{x}(k-1) \) is the centroid of the corners \( \mathbf{x}/(k-1) \) detected in the previous frame.

### 4. Heterogeneous system architecture

The registration algorithm described in Section 3 requires high computational performance to operate in real time (i.e., at the frame rate of the IR camera). This poses a serious challenge to the design of an architecture that combines a special-purpose pipeline that implements the HCD algorithm. The processor uses the corner data to estimate the registration parameters and map each LWIR pixel to the RGB image coordinates. We first discuss the fixed-point arithmetic used in functional units of the HCD pipeline, then we present the system architecture and discuss the implementation of the key hardware blocks.

#### 4.1. Fixed-point analysis

Eqs. (2) and (3) show that the HCD detects the corner based on image gradients. The minimum number of bits needed to represent these gradients is:

\[ \text{bits}_g = \log_2(\max(\max[g_{xy}, 1], \max[g_{yx}, 1])) + 1, \]

where \( g_{xy} \) and \( g_{yx} \) are the values of the spatial derivatives computed on a set of IR images acquired at ambient temperature using cold objects. Our experiments show that using a 10-bit two’s complement representation for the gradients and a 22-bit format for the product between them renders a numeric accuracy comparable to a floating-point implementation of the algorithm.

During the gradient filtering stage, we use a Gaussian kernel that is insensitive to corner orientations [23]:

\[
W = \frac{1}{256} \begin{bmatrix}
1 & 4 & 6 & 4 & 1 \\
4 & 16 & 24 & 16 & 4 \\
6 & 24 & 36 & 24 & 6 \\
4 & 16 & 24 & 16 & 4 \\
1 & 4 & 6 & 4 & 1
\end{bmatrix} = \frac{1}{256} \mathbf{w}^T \mathbf{w},
\]

where \( \mathbf{w} = [1 \ 4 \ 6 \ 4 \ 1]^T \). Therefore, the filter can be implemented as:

\[
\hat{g} = \mathbf{G}^T \mathbf{w} = \frac{1}{256} \mathbf{w}^T \mathbf{G} \mathbf{w},
\]

where \( (\ast) \) denotes the convolution operator, \( \hat{g} \) is the output of the filter, and \( \mathbf{G} \) is a matrix that contains the spatial gradients \( (g_{xy}, g_{yx}, g_x, \) and \( g_y) \) in a \( 5 \times 5 \) window. As stated above, we used a 10-bit fixed-point format for \( g_x \) and \( g_y \), and a 22-bit representation for \( \hat{g} \).

Eq. (3) shows the output \( \mathbf{H} \) of the detector, where \( A, B, \) and \( C \) are the outputs of the three filters. We define the error of the fixed-point implementation of the HCD with respect to a software floating-point implementation reference as:

\[
\text{Error}(z) = \frac{1}{NM} \sum_{i} |H_{nm}(z) - H_{nm}(z)\text{ref}|.
\]

where \( N \) and \( M \) are the number of pixel rows and columns of the ROI \( R \) in each image, respectively, \( H_{nm}(z) \) is the output of the HCD for row \( n \) and column \( m \) in the floating-point reference, and \( H_{nm}(z)\text{ref} \) is the output of the HCD in our fixed-point implementation of Eq. (3) using \( z \) bits to represent \( \mathbf{H} \). Fig. 6 plots the error, averaged over 700 video frames of experimental data, as a function of \( z \). Using a 7-bit representation for \( \mathbf{H} \), the error is more than three orders of magnitude smaller than the magnitude of the HCD output in the detected corners.

Finally, we use 32 bits to store \( \mathbf{H} \), which we determined by using three video sequences in normal and extreme temperature conditions and recording the maximum detected values. During the computation of \( \mathbf{H} \), we use fixed-point representations of up to 44 bits to preserve the accuracy of the results.

#### 4.2. System architecture

Fig. 7 depicts the complete system architecture, which was implemented on a Xilinx Z-7010 SoC. This chip integrates traditional field-programmable gate array (FPGA) logic fabric with a dual-core ARM Cortex-9 32-bit processor, and a bus and peripheral system. The hardware HCD pipeline was designed using the Verilog hardware description language at the register-transfer level (RTL) to maximize efficiency and performance, while the software component was implemented in C. The software algorithms were programmed directly on the processor, which does not run an operating system. The SoC acquires video frames from the LWIR camera through a digital LVDS port, and acquires the initial RGB image from a camera connected through an Ethernet interface. Both interfaces are controlled by specialized logic on the FPGA. The system implements a graphical user interface that allows the user to select an ROI on both the LWIR and RGB images separately using a PS/2 mouse. Registered images are fused and displayed on an external monitor through a VGA interface.

The architecture uses a single processor core to read video frames from both cameras and stores them on a 512 MiB DDR3 memory external to the chip. Two internal memory buffers locally store data accessed from the external memory. The size of each memory buffer is 8.8 Kib and each can store six image lines with up to 500 pixel each, which are used to implement a 7 x 7-pixel window used by the HCD. The Image Register module, implemented using reconfigurable logic, reads the ROI data from the RGB and LWIR images using the memory buffers, and estimates the four corner coordinates of each ROI using the HCD algorithm. The processor uses the corner data to estimate the registration parameters and map each LWIR pixel to the RGB image coordinates. The processor also controls the graphical interface that allows the user to select the ROI on both images, and displays either the separate images or a combined version that superimposes the registered LWIR video frames over the reference RGB frame in real time. Specialized modules implemented in hardware assist the processor by fusing the superimposed images and driving the peripheral interfaces. The rest of this section describes the implementation of the Harris detector.

#### 4.3. Implementation of the HCD algorithm

The HCD was implemented using a fully-pipelined architecture that can output a result on each clock cycle. Fig. 8 depicts the first stage of
the detector, which computes the spatial derivatives, multiplies them, and applies a set of Gaussian filters to produce coefficients A, B, and C in Eq. (2).

Our implementation of the Harris detector uses a window of 7 × 7 pixels, which requires six line buffers to store the image lines previous to the current one. Using this window, we compute five pairs of derivatives \( (g_x, g_y) \) and three products for each derivative \( (g_x g_x, g_y g_y, \text{and } g_x g_y) \).

We compute the derivatives using addition, subtraction and bit shifts using the Sobel operators:

\[
G_x = \frac{1}{4} w_1 g_x^T, \quad G_y = \frac{1}{4} w_2 g_y^T
\]

where \( w_1 = [1 \ 2 \ 1]^T \) and \( w_2 = [-1 \ 0 \ 1]^T \). Therefore, the image gradients are computed as

\[
g_x = I^T g_x = \frac{1}{4} w_1^T I^T g_x
\]

\[
g_y = I^T g_y = \frac{1}{4} w_2^T I^T g_y
\]

where \( I \) is a 3 × 3-pixel window in image \( I \).

Figs. 9 and 10 depict the functional units implementing Eqs. (13) and (14), respectively. Each pair of derivatives requires the same three image lines to implement a window of 3 × 3 pixels. The figures show that each pair of derivatives is computed using two line buffers and data from the current line. For a 7 × 7-pixel window, the HCD block computes five pairs of derivatives in a pipelined fashion, requiring a total of six line buffers because each derivative pair shares two buffers with the previous pair. The products between the derivatives shown in the third block of Fig. 8 are implemented using embedded hardware multipliers.

Fig. 11 shows the implementation of the Gaussian filter in Eq. (10), where the five inputs are the gradient products. The architecture takes advantage of the separability and symmetry properties of the kernel in Eq. (9) to implement the filter by computing two sequential convolutions and right-shifting the result of each by four bits. Moreover, because the values of the coefficients \( w_0, w_1, \) and \( w_2 \) are 1, 4 and 6, respectively, the products can be computed with simple bit-shifts and additions. The pipeline registers in the circuits are omitted for clarity.

Fig. 12 shows the second stage of the HCD. The first block computes the response of the detector, the second block preserves only the highest-ranked corner in a 3 × 3-pixel window (to avoid detecting the same corner more than once), and the third block selects the four highest-ranked corners in the ROI.

Fig. 13 depicts the circuit that computes the output \( H \) of the HCD in Eq. (3). The multiplications and additions are performed in a 4-stage pipeline to output a result on each clock cycle. The first stage computes \( A + B \), the second computes \( (A + B)^2 \), \( A \times B \), and \( C^2 \). The third stage of the pipeline produces \( (A + B)^2 A B - C^2 \), and the final stage outputs \( H \).

Fig. 14 shows the circuit that determines whether \( H \) is the maximum value in a 3 × 3-pixel window. It uses two line buffers to store the values of \( H \) computed for the previous two lines in the image, registers to implement the moving window on the line data, and a set of comparators and multiplexers to output a logic signal that indicates whether \( H \) is a local maximum in the window.

Finally, Fig. 15 shows the sorting stage that selects the four corners of the current ROI. This block reads each new value of \( H \) that the previous block labeled as a local maximum, and compares it against the
four values currently stored in local registers. A priority circuit stores the new value of \( H \) in the first register that contains a value of smaller magnitude. Each stored corner has two associated registers that store the coordinates of the pixel that produced that HCD response.

The size of the ROI where the HCD computes the corners can change from frame to frame in the LWIR video because of subject movement. Consequently, at the end of each video frame, a state machine estimates the maximum size of the ROI and updates a register that controls the effective size of the circular memories used to implement the line buffers. The buffers can store up to 500 values, because we restricted the ROI to a maximum size of 500 \( \times \) 500 pixels in our current implementation.

As stated before, the processor uses the corner coordinates to estimate the parameters of the projective transformation and maps the pixels in the LWIR video frame to the RGB reference image coordinates. Because these computations are outside the critical path, this implementation decision does not degrade the performance of the system.

5. Experimental results

Our experimental setup consists of a prototype implementation of the registration architecture using a Xilinx Z-7010 integrated circuit, connected to a FLIR Tau 2 LWIR camera core of 640 \( \times \) 512-pixel resolution at 30 frames per second (fps) with a digital LVDS interface. The RGB camera has a resolution of 640 \( \times \) 480 pixels and a frame rate of 30 fps. The RGB camera used in the current version of the prototype does not have direct interface to our system board, therefore we transfer the RGB images to the board on an Ethernet network in real time using a personal computer. Because our algorithm uses only one image from the RGB camera as a reference, the RGB and LWIR video streams do not need to be synchronized. Thus, the latencies introduced by the network and the computer do not affect the registration results. We also use the network to send experimental data to be analyzed on the computer.

In order to evaluate the numeric precision achieved by our prototype, we used the estimated transformation to map the corners detected in the RGB reference frame to the coordinate space of the LWIR image. We then computed the root mean square error (RMSE) between these coordinates and the coordinates of the actual corners detected by the HCD in the LWIR image. The RMSE is defined as

\[
RMSE(k) = \frac{1}{\sqrt{4}} \left\| b - A^{(k)}\theta^{(k-1)} \right\|^2,
\]

where \( b \) are the coordinates of the corners detected in the LWIR image, \( A^{(k)} \) is the matrix in Eq. (5) built with the coordinates of the corners detected in the RGB image in frame \( k \), and \( \theta^{(k-1)} \) are the parameters of the transformation estimated using the data of frame \( (k-1) \).

Fig. 16(a) shows the evolution of the RMSE in a double-precision software implementation of the registration algorithm, which estimates the transformation parameters using the least-squares method. The data uses 700 video frames from the LWIR camera. The RMSE is less than one pixel until frame 400. At this frame, the subject started a sequence of movements that produced significant variations in the registration parameters from frame to frame. Even during this phase, the RMSE remains less than 6 pixels.

As stated in Section 3, our architecture estimates the transformation parameters on line using the SGD update of Eq. (7) instead of least squares, centering the corner coordinates with respect to their centroid to obtain a rough estimate of the transformation and thus speed up the convergence of the algorithm [24]. Fig. 16(b) shows the evolution of the RMSE during the same experiment in Fig. 16(a), this time using SGD. The algorithm converges to an RMSE of around one pixel, comparable to the performance of the least-square estimate, in less than 25 iterations. Because each iteration corresponds to one video frame of the camera, the convergence time is less than 834 ms. Using an affine transformation instead of a projective one would allow the algorithm to converge in only 5 frames (less than 167 ms) [25] because it uses fewer parameters, but we lose the ability to correct changes in perspective.
Fig. 12. Harris detector, second stage.

\[ H = AB - C - \ell (A + B)^{\ell} \]

Fig. 13. Computation of HCD output.

Fig. 14. Selection of local maximum.

Fig. 15. Corner selection.
caused by involuntary patient movement.

Fig. 17 compares the RMSE performance of our algorithm against registration performed using the Matlab Image Processing toolbox, which uses mutual information as similarity measurement and 1+1 evolution as the optimization criterion. The experiment uses the same 700 video frames used to generate the results shown in Fig. 16.

Fig. 17(a) shows the same data as Fig. 16(b), but the range of the vertical axis of the graph was increased to an RMSE value of 40 pixels to facilitate the comparison to the results of mutual information with 1+1 evolution, which are shown in Fig. 17(b). As the graphs show, our approach produces an average RMSE of 1 pixel with a peak of 6 pixels.
when the patient suddenly moves. In contrast, mutual information produces an average RMSE of 3 pixels, with peaks as large as 33 pixels throughout the experiment. Mutual information operates globally on the two images, and often fails to accurately estimate their similarity. In contrast, our approach computes the marker corner coordinates and uses these features to produce a much more accurate and stable measure of similarity. Also, the computational requirements of our algorithm are significantly lower, requiring 112 ms to register a pair of images, compared to 7 s for mutual information, both running in Matlab on the same desktop computer.

Fig. 18 shows video captures from the user interface of the prototype. The images in the figure show a visualization of the RGB reference image superimposed over the LWIR frame, after registration. The images were captured at different times of the experiment. The dark area inside the marker and the white frame around it represent the nonoverlapping areas between the RGB and LWIR images, which result from the registration error. The figure confirms that the prototype, implementing the HCD in hardware with a fixed-point numeric representation, is able to align the images inside the marker almost perfectly after 25 frames.

The critical path of the hardware implementation, which determines the clock frequency of the circuit, is located in the HCD module, specifically between the output $g_{kB}^q$ of the Gaussian filter and the register that stores $A \times B$ in the computation of $H$. The maximum clock frequency with this critical path is 135 MHz, which enables the circuit to process a new pixel every 7.4 ns. The circuit can therefore process a ROI of 500 × 500 pixels in as little as 1.85 ms, achieving a frame rate of up to 540 fps. This is two orders of magnitude faster than a software implementation of the algorithm with the rest of the computations needed to perform the skin cancer screening tests.

### 6. Conclusions

We have presented the algorithms, architecture, and implementation of an embedded hardware system that performs multimodal registration on visible and thermal images in real time. Using a combination of software and special-purpose hardware on a configurable SoC, the system processes data from an RGB camera and an LWIR camera core. The architecture uses a hardware implementation of a corner detection algorithm to extract features from an ROI of the image, estimates the registration parameters using stochastic gradient descent, and maps the LWIR ROI onto the coordinates of the RGB image using a projective transformation. The system was implemented on a Xilinx Z-7010 SoC and achieves a maximum frame rate of 540 fps on a 500 × 500-pixel ROI, converging in 25 video frames with a numeric performance comparable to that of a double-precision implementation of the algorithm. The prototype consumes 1.8 W, most of which is used by the ARM Cortex 9 processor embedded in the SoC, while the hardware module that implements the corner detector uses only 84 mW. The prototype uses 29% of the logic resources and 54% of the embedded arithmetic resources of the chip. As future work, we plan to integrate the registration module with other blocks that implement image segmentation, temperature estimation, and classification to achieve a portable, low-cost integrated instrument for skin-cancer screening.

### Acknowledgments

This work was supported by Fondecyt grants 1151278 and 11150702. Simulation and synthesis were done with EDA software donated by Synopsys and Xilinx.

### References


dynamic thermal imaging, Ph.d. dissertation, ECE Dept., The University of New Mexico, Albuquerque, NM, USA, 2015.


Fabían Inostroza received a professional degree in Electronics Engineering from Universidad de Concepción, Chile, in 2015. He was a member of some interest groups, including the Equipo Solar Ka Antu Newen and the flight interest group. His interests include embedded systems and control systems.

Silvana Díaz received an undergraduate degree in Biomedical Engineering from Universidad de Concepción, Chile, in 2017, where she is currently pursuing an M.Sc. degree in Electrical Engineering. Her research currently focuses on the design of an embedded system that integrates image segmentation, multimodal registration, and real-time classification to assist in noninvasive skin cancer detection.

Javier Cárdenas received an undergraduate degree in Electronics Engineering from Universidad de Concepción, Chile, in 2015. He is currently pursuing an M.Sc. degree in Electrical Engineering in the same University. His thesis work focuses on distributed hardware accelerators for multimodal image registration. His other research interests include embedded systems and reconfigurable hardware design.

Sebastián E. Godoy was born in Chile. He received his M.Sc. in Electrical Engineering from Universidad de Concepción in 2009 and his Ph.D. in Engineering from The University of New Mexico in 2015. His research mainly focuses on the application of advanced imaging systems to address biomedical problems such as the early detection of skin cancer, providing guarantees of reliability, accuracy and performance. His current research interests span digital image processing, digital signal processing, detection theory, biomedical engineering and advanced imaging systems. He is currently an Assistant Professor at Universidad de Concepción.

Miguel Figueroa is a Professor in Electrical Engineering at Universidad de Concepción, Chile. He obtained an undergraduate degree in Electronics Engineering (1990) and an M.Sc. degree in Electrical Engineering (1997) from Universidad de Concepción. He also received the M.Sc. (1999) and Ph.D. (2005) degrees in Computer Science and Engineering from the University of Washington in Seattle, WA, USA. Prof. Figueroa has vast experience in the design of high-performance embedded systems for real-time signal processing applications using FPGAs, as well as digital and mixed-signal VLSI. His current research interests include high-performance VLSI systems and reconfigurable computing for video processing and computer vision, and instrumentation and processing for quantum communications. He has authored and co-authored more than 50 scientific publications.