Research Article

Implementation of a Low-Cost Mobile Devices to Support Medical Diagnosis

Carlos García Sánchez, Guillermo Botella Juan, Fermín Ayuso Márquez, Diego González Rodríguez, Manuel Prieto-Matías, and Francisco Tirado Fernández

Department of Computer Architecture and Automation, Universidad Complutense de Madrid, 28040 Madrid, Spain

Correspondence should be addressed to Guillermo Botella Juan; gbotella@fdi.ucm.es

Medical imaging has become an absolutely essential diagnostic tool for clinical practices; at present, pathologies can be detected with an earliness never before known. Its use has not only been relegated to the field of radiology but also, increasingly, to computer-based imaging processes prior to surgery. Motion analysis, in particular, plays an important role in analyzing activities or behaviors of live objects in medicine. This short paper presents several low-cost hardware implementation approaches for the new generation of tablets and/or smartphones for estimating motion compensation and segmentation in medical images. These systems have been optimized for breast cancer diagnosis using magnetic resonance imaging technology with several advantages over traditional X-ray mammography, for example, obtaining patient information during a short period. This paper also addresses the challenge of offering a medical tool that runs on widespread portable devices, both on tablets and/or smartphones to aid in patient diagnostics.

1. Introduction

Medical imaging [1] as a diagnostic technique in medicine requires complex image analysis of image sequences obtained by a plethora of variety, such as ECG, X-ray, MRI, ultrasound, CT, and so forth. Magnetic resonance imaging (MRI) [2] technology is one of the most promising tools over other methods, like conventional X-ray mammography, regarding breast cancer diagnosis. Nowadays, X-ray images still have a higher spatial resolution than MR images, but this technique has the advantages of producing natural tridimensional images and being able to noninvasively monitor the contrast agent concentration in the depicted tissue over time.

On other hand, motion estimation is still an open problem with important applications to medical imaging. Attending to the estimation of a pixel motion inside the image sequence, there are many models and algorithms that could be classified as belonging to the matching domain approximations [3], energy models [4], and gradient models [5]. Related to this last family, different studies [6–8] show that this represents an admissible choice for keeping a tolerable trade-off between accuracy and computing resources. For

designing systems operating efficiently, many challenges must be dealt with, such as robustness, static patterns, illumination changes, different kinds of noise, contrast invariance, and so forth.

Regarding the hardware platform used, the portable industry in recent years has dramatically increased the importance of the processing hardware elements. The iPhone 3GS offered more than twice the performance of the first- and second-generation iPhones. As the market becomes more demanding, many processor manufactures have specialized in using smartphones for its own solutions, such as Cortex A8, Snapdragon, ARM11, Tegra, Armada, OMAP, and more. Nowadays, device manufacturers boast about their phones' computer-like capabilities, from desktop-like Internet browsing to HD video playback and, of course, enough computing capability to face up scientific processing [9, 10].

The ARM instruction set has become the reference architecture in low-power devices, so there are many general CPUs able to run compatible ARM code; this fact creates tendency lines in reference processors that fit with these instruction sets, and so the companies produce the chipset following the ARM directives. Some companies, such as Texas Instruments,

Samsung, and Nvidia, adopt the ARM CPU reference design, whereas others, like Qualcomm and Marvell, license only the instruction set and create their own processors to fit them.

The popularity of laptops overtook desktop PCs some time ago; mobile devices are currently among the most ubiquitous devices around. Mobile devices are now emerging in two directions: on one hand, they are pushing upward into the territory of Intel Atom-powered netbooks; on the other hand, they are trickling downwards in price, replacing high-end feature phones near the $100 mark.

More than 2 billion ARM chips [11] are shipped every year, beating Intel's Atom devices. The role of GPUs (Graphic Processing Units) is to provide hardware acceleration for tridimensional graphics applications like games, CAD, and so forth. However, in recent years, their role has also become responsible for drawing the main user interface for desktop Operating Systems (OS). On most modern smartphone platforms (iPhone OS, Android, and Palm WebOS, with Windows Mobile as an exception), the OS's user interface itself is composited, meaning it is rendered by the GPU [12]. This makes the interface feel a lot smoother than it would if it were displayed calculations on the already resource-constrained CPU.

Regarding the motion estimation for embedded systems, there are other gradient optical flow models implemented in hardware; some belong to the Lucas and Kanade algorithm [13, 14], and some belong to Horn and Schunk approximations [15, 16].

It is worth paying attention to previous implementations [17–19] of the sophisticated and complex Multichannel Gradient Model (McGM) algorithm [20]; this bioinspired algorithm is required to deal with many challenges, such as robustness, static patterns, illumination changes, different kinds of noise, contrast invariance, and so forth. Also the McGM is able to detect correct motion related to optical illusions or avoid operations like matrix inverse or iterative methods that are not biologically justified [15]. We must carefully select a model that carries out these kinds of requirements. There are many implementations for this algorithm [17–19], although we will focus on the Carma board-based [21] implementation. Despite this system's ability to manage complex situations better than others algorithms in real environments [20, 22–25] and mimic some behaviors of mammals [22, 25], its computational complexity is high and still not appropriate for the kind of microprocessors considered.

Under the assumptions mentioned above, in this paper we present a tool to aid medical diagnosis developed *ad hoc* for mobile devices like tablets and/or smartphones. Not only has their acceptance and reduced price driven diffusion but their ever-increasing performance computations offer the opportunity to use them in tasks such as medical diagnosis. This paper studies the feasibility of using this modern technology in a medical scenario, where medical images are processed to aid in medical decisions.

The present paper is organized as follows. First, the stages of the Lucas and Kanade model and Otsu's method basic equations for segmentation based on histogram are explained very briefly. After that, the basic equations for Otsu's method are explained. Next, the implementation is analyzed using two kinds of microprocessors—an Intel Atom and an ARM processor from the Nvidia Carma board. Several comparisons are performed using existing optical flow implementations in other embedded devices commonly used for mobile platforms. Additionally, output images for the final segmented system are shown. Finally, quality results and associated costs are presented for the hardware.

2. Optical Flow Gradient-Based Computation and Segmentation

2.1. Optical Flow Estimation and Lucas and Kanade. Assuming that an object moves relative to an imaging device, its 2-dimensional projection usually moves within the projected image. The projection of the 3-dimensional relative motion vectors onto the 2-dimensional detector yields a projected motion field often called the "image flow" or "motion field." Unfortunately, it is impossible to access the velocity field directly, since optical sensors collect luminance distributions and no speeds. However, it is feasible to compute the motion of local regions of the luminance distribution, and it is this motion field that is referred to as optical flow. The optical flow provides an approximation to the velocity field, but it is uncommonly equal. There are a number of problems to overcome in order to compute the optical flow from the changes in the luminance distribution. First, we can only calculate the motion of patterns, not isolated points, in the luminance distribution. This means that the luminance information must be combined in some way over a finite spatial neighborhood around each point, where we want to measure motion. The so-called aperture problem arises when we try and measure the two components of image velocity using a neighborhood that does not keep enough luminance structure [26–28].

In such situations, we are unable to constrain the measurement to a single solution (Figure 1). Increasing the size of the neighborhood permits us to constrain the measurement, but collecting information over a large region increases the likelihood of pooling over motion boundaries and over smoothing the results, which has been referred to as the general aperture problem [29]. The motion algorithms seek to recover the optical flow field that is the best approximation to the projected velocity field. However, using this information to draw conclusions about the 3-dimensional environment is a difficult process.

The estimation of the velocity field using optical flow is an ill-posed problem, since there are an infinite number of velocity fields that can cause the observed changes in the luminance distribution. Additionally, there are an infinite number of 3-dimensional motions in the real world that could yield a particular velocity field. External knowledge about the behavior of objects in the real world, such as rigid body motion constraints, are required in order to make use of optical flow. Despite the problems, the optical flow information is a rich array of vectors that has both local and global properties [30]. The optical flow field can thus be subjected to many higher-level interpretations [31, 32].
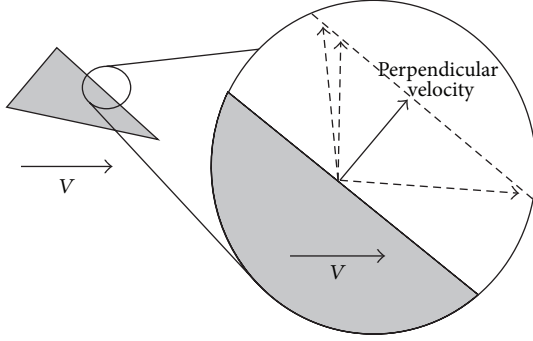
FIGURE 1: The aperture problem. There are infinite solutions for this problem.

The Lucas and Kanade method [33] is a well-known algorithm, and we have applied the original description of the model [34] while adding several variations to improve the viability of the hardware implementation. We present a simplified scheme of the algorithm, as follows.

The Lucas and Kanade model computes optical flow using a gradient technique [35] that makes use of space-temporal derivative filters. The model comes from the basic intensity conservation over the time (1), where x, y, and t are the coordinates of the sequence. Developing the expression (1), we reach expression (2),

$$\frac{dI\left(x\left(t\right),y\left(t\right),t\right)}{dt} = 0, \tag{1}$$

$$\frac{dI\left(x\left(t\right),y\left(t\right),t\right)}{dt}$$

$$= \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}\frac{dx}{dt} + \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y}\frac{dy}{dt}$$

$$+ \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial t}$$

$$= \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}u + \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y}v$$

$$+ \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial t} = 0. \tag{2}$$

The Lucas and Kanade model assumes that the motion vector really does not change in the studied vicinity V. Considering the error to minimize the motion constraint expression (2):

$$E\left(u,v\right) = \sum_{\text{pixel}\in V}\left(\frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}u \right.$$

$$+ \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y}v \tag{3}$$

$$\left. + \frac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial t}\right)^2.$$

Solving $\partial E/\partial u(t) = 0$; $\partial E/\partial v(t) = 0$ and grouping them all together, we find an algebraic system expressed by (4), which means the LMS estimation of the optical flow in the centered pixel of the vicinity V. The symbol $\hat{}$ denotes the estimator of the corresponding magnitude. The resulting optical flow estimated is dense:

$$\begin{bmatrix} \sum_{\text{pixel}\in V}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x} & \sum_{\text{pixel}\in V}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y} \\ \sum_{\text{pixel}\in V}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y} & \sum_{\text{pixel}\in V}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y} \end{bmatrix}\begin{bmatrix}\widehat{u}\\\widehat{v}\end{bmatrix}$$

$$= \begin{bmatrix} -\sum_{\text{pixel}\in V}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial x}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial t} \\ -\sum_{\text{pixel}\in V}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y}\dfrac{\partial I\left(x\left(t\right),y\left(t\right),t\right)}{\partial y} \end{bmatrix}. \tag{4}$$

So the final notation is

$$\begin{bmatrix}\widehat{u}\\\widehat{v}\end{bmatrix}A = B;$$

$$A = \begin{bmatrix} \sum_{\text{pixel}\in V}I_x^2\left(\text{pixel}\right) & \sum_{\text{pixel}\in V}I_xI_y\left(\text{pixel}\right) \\ \sum_{\text{pixel}\in V}I_xI_y\left(\text{pixel}\right) & \sum_{\text{pixel}\in V}I_y^2\left(\text{pixel}\right) \end{bmatrix} \tag{5}$$

$$\times B = \begin{bmatrix} -\sum_{\text{pixel}\in V}I_xI_t\left(\text{pixel}\right) \\ -\sum_{\text{pixel}\in V}I_yI_t\left(\text{pixel}\right) \end{bmatrix}.$$

The subindex in the equation means the derivatives computed by separable filtering (Gaussian derivatives or Gabor function).

2.1.1. Segmentation by Histogram Using the Otsu Method. The Otsu Method [36, 37] applies an automatic threshold in order to efficiently segment the image; it is based on a discriminant criterion to optimize the function of separation of obtained classes in gray levels. We describe very briefly the method used: if supposing every pixel forms an image represented by gray levels $[1, 2, \ldots, L]$, the number of the pixels at level I is denoted by n_i and the total number of pixels by N.

The gray-level histogram is normalized regarding a probability distribution expression:

$$p_i = \frac{n_i}{N}, \quad \sum_i^L p_i = 1. \tag{6}$$

Assuming a classification of pixels in two classes C_0 and C_1 (objects and background) by a threshold level k, where pixels with levels $[1, \ldots k]$ belong to C_0 and pixels with levels $[k + 1, \ldots, L]$ belongs to C_1, the probabilities of the class occurrence and class means levels are written as

$$w_0 = \Pr(C_0) = \sum_{i=0}^k p_i; \qquad w_1 = \Pr(C_1) = \sum_{i=k+1}^L p_i, \tag{7}$$

$$\mu_0 = \sum_{i=1}^k i\Pr(i \mid C_0) = \sum_{i=1}^k \frac{ip_i}{w_o};$$

$$\mu_1 = \sum_{i=k+1}^L i\Pr(i \mid C_1) = \sum_{i=k+1}^L \frac{ip_i}{w_1}. \tag{8}$$

In this step, we are ready to define the following relation for the choice of k and the variance based on first-order statistics (class means) as

$$\mu_T = w_o\mu_0 + w_1\mu_1; \qquad w_1 + w_o = 1;$$

$$\sigma_B^2(k) = \frac{[\mu_T w(k) - \mu(k)]^2}{w(k)[1 - w(k)]}. \tag{9}$$

The optimal threshold k^* that maximizes σ_B^2 is selected by a sequential search using the cumulative quantities expressed in (6) and (7)

$$\sigma_B^2(k^*) = \max_{1 \le k \le L} \sigma_B^2(k);$$

$$S^* = \{k; w_0 \, w_1 = w(k)[1 - w(k)] > 0\}. \tag{10}$$

3. System Implemented

3.1. Patients and MR Imaging.
Breast MRI was performed on patients with indeterminate mammographic breast lesions. All patients were consecutively selected after clinical examination, mammography in standard projections (craniocaudal and oblique mediolateral projections), and ultrasound. Only lesions classified BIRADS 3 and 4 in mammography were selected. In addition, at least 1 of the following criteria had to be present: nonpalpable lesion, previous surgery with extensive scarring, and location difficult for biopsy (e.g., close to chest wall). Histologic findings were malignant in 14 and benign in 17 lesions. Lesion size was derived from mammography images. Mean size of malignant lesions was 1.2 cm (median = 1.0 cm, range = 0.4–3.5 cm); mean size of benign lesions was 1.1 cm (median = 0.9 cm, range = 0.3–3.0 cm).

MRI was performed with a 1.5 T system (Magnetom Vision, Siemens, Erlangen, Germany) equipped with a dedicated surface coil to enable simultaneous imaging of both
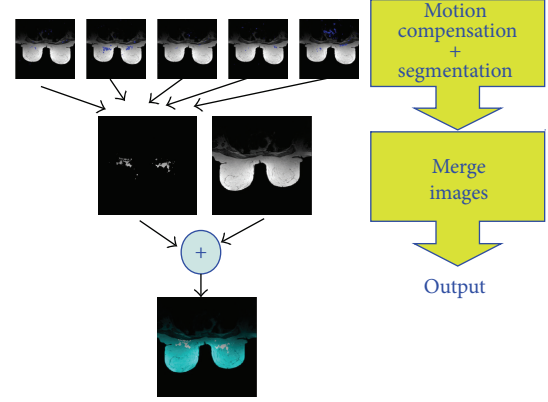

FIGURE 2: Scheme of the implemented system.

breasts. The patients were placed in a prone position. Transversal images were acquired with a STIR (short TI inversion recovery) sequence (TR = 5600 ms, TE = 60 ms, FA = 90°, TI = 150 ms, with a matrix size of 256 × 256 pixels, sliced 4 mm thick).

Then, a dynamic T1 weighted gradient echo sequence using a 3D FLASH (fast low-angle shot pulse sequence) was performed (TR = 12 ms, TE = 5 ms, FA = 25°) in transversal slice orientation with a matrix size of 256 × 256 pixels and an effective slice thickness of 4 mm. FA (Flip Angle), STIR (Short Tau Inversion Recovery), TE (Echo Time), and TR (Pulse Repetition Interval) are abbreviations for MRI modalities. The dynamic study consisted of 6 measurements with an interval of 83 s. The first frame was acquired before injection of paramagnetic contrast agent (gadopentetate dimeglumine, 0.1 mmol/kg body weight, MagnevistTM, Schering, Berlin, Germany) immediately followed by the 5 other measurements.

3.2. Scheme of the System.
Figure 2 shows the algorithm implemented in the embedded system. In the first stage, dense motion is estimated, and the zone corresponding to the range of motion is determined, segmented, and valued using the histogram-based system. It is important to note that this system is adjustable and configurable since the segmented motion values can fit the diagnostic needs, as determined for each individual case. The first system used here was designed using the Carma platform [38] from the SECO company [21], which integrates an Nvidia Tegra CPU with 3 Quad-Core ARM Cortex-A9 CPUs. The second is based on an Intel Atom (2x Intel(R) Atom (TM) CPU D510). In Figure 3, prototyping boards that contain the processors to be programmed are shown.

4. Results

In this section, results are shown in the boards based on ARM and ATOM. First we discuss the results in terms of performance (execution time) in both boards and then evaluate the visual results obtained in the proposed medical

(a)   (b)
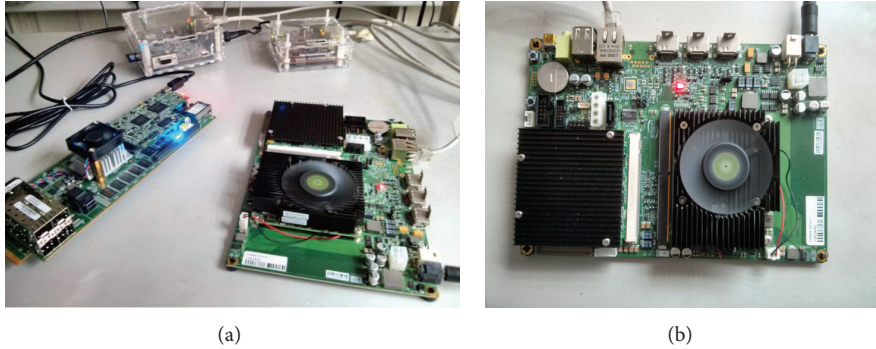
Figure 3: Scheme of the low-cost system implemented at different commercial microprocessors.

aid system. We have chosen two systems that incorporate low-power processors which are the base of many mobile devices. Additionally, we have measured how much is affecting the attention windowed motion-zone to be monitored depending on the specific medical diagnostic to be computed.

As we can see in Tables 1 and 2, the performance in terms of seconds/slides (breast cancer stimuli is 256×256 and its brain fMRI counterpart is 95×69) and final power consumption for both input-stimuli are shown. The results observed demonstrate that system implementation is totally competitive in terms of response time and completely feasible as a tool for medical help. Table 1 reflects the execution times observed in the sequential application (one processor Atom versus one ARM) and the best configuration obtained in terms of performance (best computing times) corresponding to the exploitation of parallelism with the use of multiple processors. The exploitation of several processors is performed by means of task-level parallelism. Parallelization scheme is based on the uniform distribution of the computational workload among the available processors by means of OpenMP programming paradigm. Accelerations achieved range between 2.2x and 3.3x times faster. For the fMRI brain test, task-level parallelism reports are hardly beneficial; this fact is motivated by the granularity of the problem and the lack of parallelism available to be exploited. The degree of parallelism available in the test considered (lower and middle ranges) makes it unsuitable for the exploitation of additional hardware such as GPU as in the Carma board (the cost of startup, exchange information does not outweigh the benefits that could be achieved in accelerator or GPU). This table also includes a comparison of consumption (peak energy demand) in both systems, so in global terms, we can conclude that a mobile system based on ARM processor reports better performance rates with less power requirements. We would like to highlight that every stage belonging to the system has been designed as customizable, scalable, and modular, containing this system a processing scheme belonging to the most gradient-based optical flow models. As a conclusion, we can affirm that the platforms considered are feasible to process at high scale motion and segmentation attending to the performance obtained at different scale levels.

From the standpoint of the system usefulness and visual results provided, they are also showing some screenshots of

Table 1: Summary of the final performance (in seconds/slide) for both processors considered and three different motion attention zone selected. (Window) for breast cancer stimuli.

Breast cancer stimuli			
Performance (secs/slide)	ARM v7	Intel ATOM	Final density
Window size = 5			
1 CPU	1,22	0,35	100,00%
Best config.	0,75	0,16	
Window size = 7			
1 CPU	2,12	0,61	100,00%
Best config.	1,18	0,24	
Window size = 9			
1 CPU	3,36	0,92	100,00%
Best config.	1,85	0,28	
Power consumption	8 W	13 W	

Table 2: Summary of the final performance (in seconds/slide) for both processors considered and three different motion attention zone selected. (Window) for fMRI Brain Stimuli.

fMRI brain			
Performance (secs/slide)	ARM v7	Intel ATOM	Final density
Window size = 5	0,02	0,02	100,00%
Window size = 7	0,04	0,01	100,00%
Window size = 9	0,06	0,01	100,00%
Power consumption	8 W	13 W	

the medical analysis generated in a mobile device. Figure 4 shows a collection of slides from the MRI breast cancer test described in Section 3. The output image displayed on the mobile device is colored for the sake of clarity in recognition, meaning clear-red zones high motion density. Meanwhile Figure 5 illustrates the motion vector map of one slide; Figure 6 addresses a zoom for motion estimation + segmentation output image, where it is possible to recognize the flow vectors corresponding to the adjustable window motion attention zone at different scales in the upper-right and -left part of the image. Additionally, Figure 7 shows

(a) slide = 1   (b) slide = 5   (c) slide = 10

(d) slide = 15   (e) slide = 20   (f) slide = 25

(g) slide = 30   (h) slide = 35   (i) slide = 40

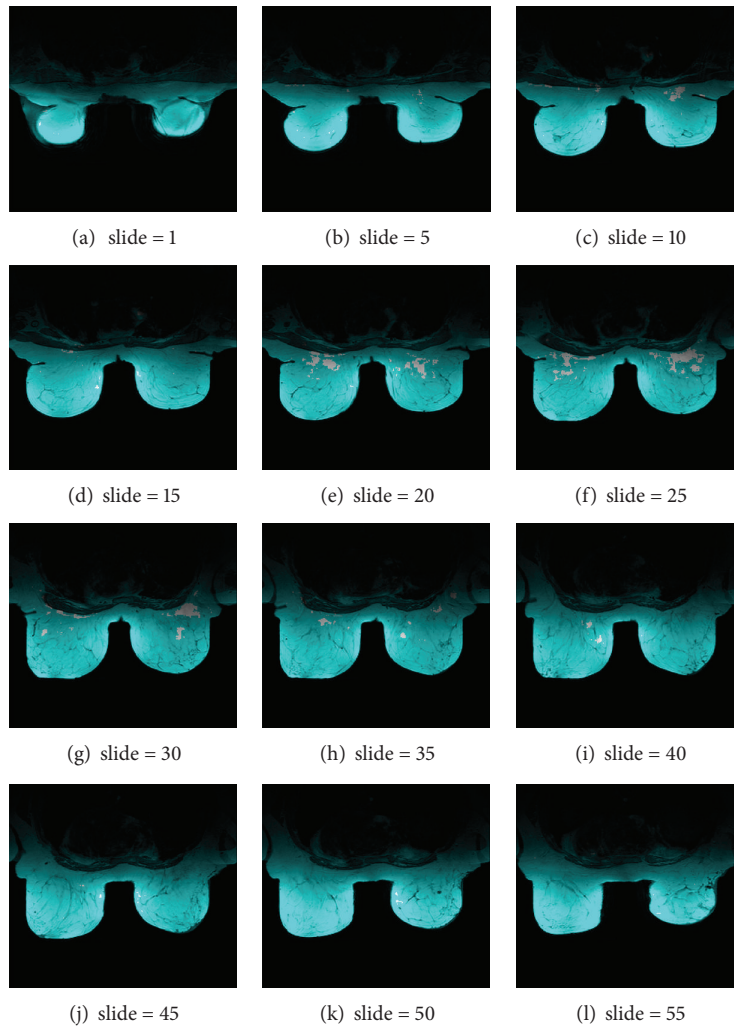(j) slide = 45   (k) slide = 50   (l) slide = 55

FIGURE 4: Twelve different slides from the MRI image described in Section 3. The image output from the system is colored for the sake of clarity in recognition. White zones mean high motion density.
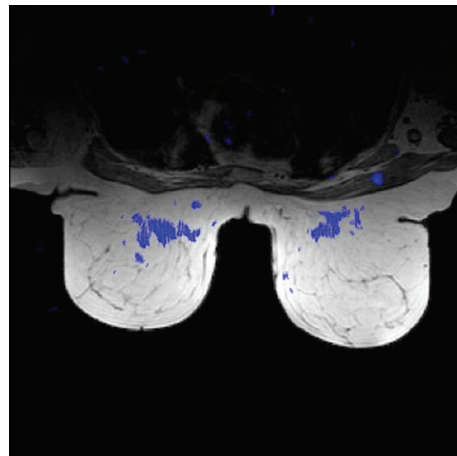

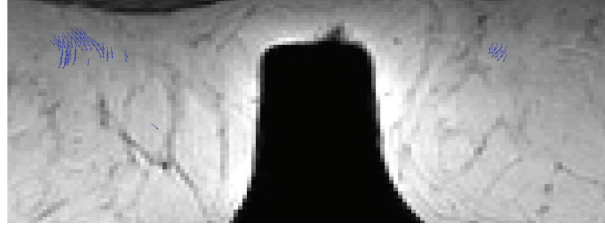
FIGURE 5: Scheme for motion vector map of one slide.

FIGURE 6: Zoom performed in the output image. Flow vectors corresponding to the adjustable window motion attention zone are shown at different scales in the upper-right and -left part of the image.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)
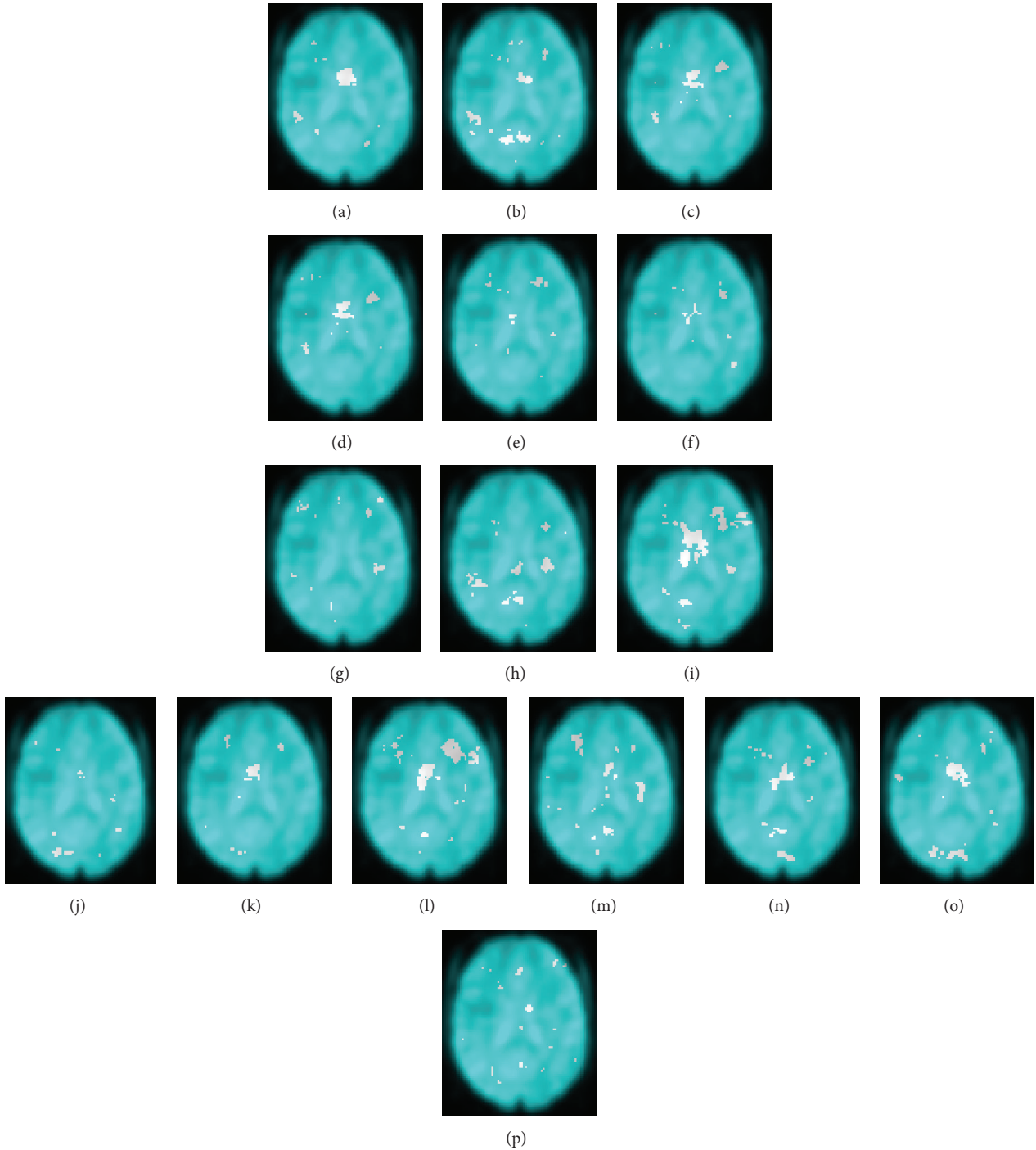
(j)

(k)

(l)

(m)

(n)

(o)

(p)

FIGURE 7: Scheme of the brain image fMRI and motion segmented using the Lucas and Kanade and the Otsu methods.

a collection of the brain image fMRI test displayed on a tablet where motion segmentation have been performed using the Lucas and Kanade and the Otsu methods.

5. Conclusion

This work describes the implementation of a low-cost hybrid system specially designed for mobile devices in medical scenarios where medical images are processed to aid in medical diagnoses and decisions. This system is specifically tuned for breast MRI based on dense motion estimation and segmentation, which can aid specialists in providing rapid attention to breast motion; the present platform can be used as a starting point for motion compensation. The technology can also be utilized for medical diagnosis for remote medicine. These algorithms have been implemented using the same processors as those used in mobile devices, such as tablets, smartphones, and so on.

Our results have shown that the algorithm is able to detect and visualize motion artifacts with high accuracy. We are currently improving the system with the hierarchical multiscale optical flow algorithm, and we will evaluate the achieved motion correction based on receiver operating characteristic (ROC) over different embedded GPUs in order to export that to mobile devices as well.

Acknowledgments

References

[1] A. M. Baese, *Pattern Recognition for Medical Imaging*, Elsevier, New York, NY, USA, 2004.

[2] M. Brammer, "Correspondence: brain scam?" *Nature Neuroscience*, vol. 7, article 1015, 2004.

[3] H. Oh and H. Lee, "Block-matching algorithm based on an adaptive reduction of the search area for motion estimation," *Real-Time Imaging*, vol. 6, no. 5, pp. 407–414, 2000.

[4] C. Huang and Y. Chen, "Motion estimation method using a 3D steerable filter," *Image and Vision Computing*, vol. 13, no. 1, pp. 21–32, 1995.

[5] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[6] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.

[7] H. Liu, T.-H. Hong, M. Herman, T. Camus, and R. Chellappa, "Accuracy vs efficiency trade-offs in optical flow algorithms," *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 271–286, 1998.

[8] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[9] D. Seal, *ARM Architecture Reference Manual*, Addison-Wesley, Boston, Mass, USA, 2nd edition, 2001.

[10] http://www.forbes.com/sites/darcytravlos/2013/02/28/arm-holdings-and-qualcomm-the-winners-in-mobile/.

[11] http://seekingalpha.com/article/1300481-lilliput-slays-gulliver-arm-vs-intel-and-why-intel-lost-the-war.

[12] J. Owens, "GPU architecture overview," in *Proceedings of the 34th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '07)*, San Diego, Calif, USA, August, 2007.

[13] J. Díaz, E. Ros, F. Pelayo, E. M. Ortigosa, and S. Mota, "FPGA-based real-time optical-flow system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 274–279, 2006.

[14] J. Díaz, E. Ros, R. Rodriguez-Gomez, and B. Del Pino, "Real-time architecture for robust motion estimation under varying illumination conditions," *Journal of Universal Computer Science*, vol. 13, no. 3, pp. 363–376, 2007.

[15] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.

[16] L. Martín, A. Zuloaga, C. Cuadrado, J. Lázaro, and U. Bidarte, "Hardware implementation of optical flow constraint equation using FPGAs," *Computer Vision and Image Understanding*, vol. 98, no. 3, pp. 462–490, 2005.

[17] C. Garcia, G. Botella, F. Ayuso, M. Prieto, and F. Tirado, "GPU-based acceleration of bioinspired motion estimation model," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 8, pp. 1037–1056, 2012.

[18] G. Botella, A. García, M. Rodriguez-Alvarez, E. Ros, U. Meyer-Bäse, and M. C. Molina, "Robust bioinspired architecture for optical-flow computation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 4, pp. 616–629, 2010.

[19] C. Garcia, G Botella, F Ayuso, M. Prieto, and F. Tirado, "Multi-GPU based on multicriteria optimization for motion estimation system," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, article 23, 2013.

[20] A. Johnston, P. W. McOwan, and C. P. Benton, "Robust velocity computation from a biologically motivated model of motion perception," *Proceedings of the Royal Society B*, vol. 266, no. 1418, pp. 509–518, 1999.

[21] http://www.seco.com/en/newselem/carma-devkit-now-shipping.

[22] A. Johnston and C. W. G. Clifford, "A unified account of three apparent motion illusions," *Vision Research*, vol. 35, no. 8, pp. 1109–1123, 1995.

[23] A. Johnston and C. W. G. Clifford, "Perceived motion of contrast-modulated gratings: Predictions of the multi-channel gradient model and the role of full-wave rectification," *Vision Research*, vol. 35, no. 12, pp. 1771–1783, 1995.

[24] P. W. McOwan, C. Benton, J. Dale, and A. Johnston, "A multi-differential neuromorphic approach to motion detection," *International Journal of Neural Systems*, vol. 9, no. 5, pp. 429–434, 1999.

[25] A. Johnston, P. W. McOwan, and C. P. Benton, "Biological computation of image motion from flows over boundaries," *Journal of Physiology Paris*, vol. 97, no. 2-3, pp. 325–334, 2003.

[26] H. Wallach, "On perceived identity: 1. The direction of motion of straight lines," in *On Perception*, H. Wallach, Ed., Quadrangle, New York, NY, USA, 1976.

[27] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.

[28] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America A*, vol. 2, no. 2, pp. 284–299, 1985.

[29] E. P. Ong and M. Spann, "Robust optical flow computation based on least-median-of-squares regression," *International Journal of Computer Vision*, vol. 31, no. 1, pp. 51–82, 1999.

[30] J. J. Koenderink, "Optic flow," *Vision Research*, vol. 26, no. 1, pp. 161–180, 1986.

[31] K. Nakayama, "Biological image motion processing: a review," *Vision Research*, vol. 25, no. 5, pp. 625–660, 1985.

[32] A. Mitiche, Y. F. Wang, and J. K. Aggarwal, "Experiments in computing optical flow with the gradient-based, multiconstraint method," *Pattern Recognition*, vol. 20, no. 2, pp. 173–179, 1987.

[33] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[34] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the DARPA Image Understanding Workshop*, pp. 121–130, April 1981.

[35] A. Bainbridge-Smith and R. G. Lane, "Determining optical flow using a differential method," *Image and Vision Computing*, vol. 15, no. 1, pp. 11–22, 1997.

[36] N. Otsu, "A Threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[37] Z. Jun and H. Jinglu, "Image segmentation based on 2D Otsu method with histogram analysis," in *Proceedings of the International Conference on Computer Science and Software Engineering, (CSSE '08)*, pp. 105–108, Wuhan, China, December 2008.

[38] http://www.seco.com.