ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Human Pulse Detection Based on Eulerian Video Magnification on Arm Cortex A Processors on Linux Platform

N.Narayana Murty
M.Sc. , AMIE , M.Tech.
Sri Sivani Institute of Technology,
Chilakapalem Jn, Etcherla M,
532402, Srikakulam District.

A.Venkateswara Rao, M.Tech
HOD,
Department of ECE,
Sri Sivani Institute of Technology,
Chilakapalem Jn, Etcherla M,
532402, Srikakulam District.

Abstract:

At present, Embedded Linux has become research focus in embedded system field. In this paper, we focus on the analysis of Eulerian Video Magnification algorithm implementation on ARM NEON architecture. Eulerian Video Magnification is a method, recently presented at SIGGRAPH 1 2012, capable of revealing temporal variations in videos that are impossible to see with the naked eye. Using this method, it is possible to visualize the flow of blood as it fills the face [WRS + 12]. And to assess the heart rate in a contact-free way using a camera [WRS + 12, PMP10, PMP11]. We use linux on arm cortex A series processors . Through the device demonstrates human pulse and heartbeat graph on the display.

1 Introduction:

The human visual system has limited spatio-temporal sensitivity, but many signals that fall below this capacity can be informative. For example, human skin color varies slightly with blood circu-lation. This variation, while invisible to the naked eye, can be ex-ploited to extract pulse rate [Verkruysse et al. 2008; Poh et al. 2010;Philips 2011]. Similarly, motion with low spatial amplitude, while hard or impossible for humans to see, can be magnified to reveal interesting mechanical behavior [Liu et al. 2005]. The success of these tools motivates the development of new techniques to revealinvisible signals in videos. In this paper, we show that a combination of spatial and temporal processing of videos can amplify subtle variations that reveal important aspects of the world around us. This is called Eulerian video magnification algorithm. Because of complexity of algorithm computational efficiency is very low on normal computing. So in order to carry out this algorithm on RISC processors like ARM cortex A series processors is bit challenge.

Cortex A series processors consists a SIMD engine called NEON. NEON is a floating point SIMD unit in replacement of VFP present in ARMv6 architecture. This unit will accelerate the DSP based algorithm implementations easy on Cortex A processors. We use Embedded linux platform for realizing the system to development of devise, as well as demonstration of heart beat findings like ECG graph .Other fields include medical applications, software development for mobile devices, digital signal processing.In this work, an Linux application for monitoring vital signs based on the Eulerian Video Magnification method will be developed, which should include the following features:

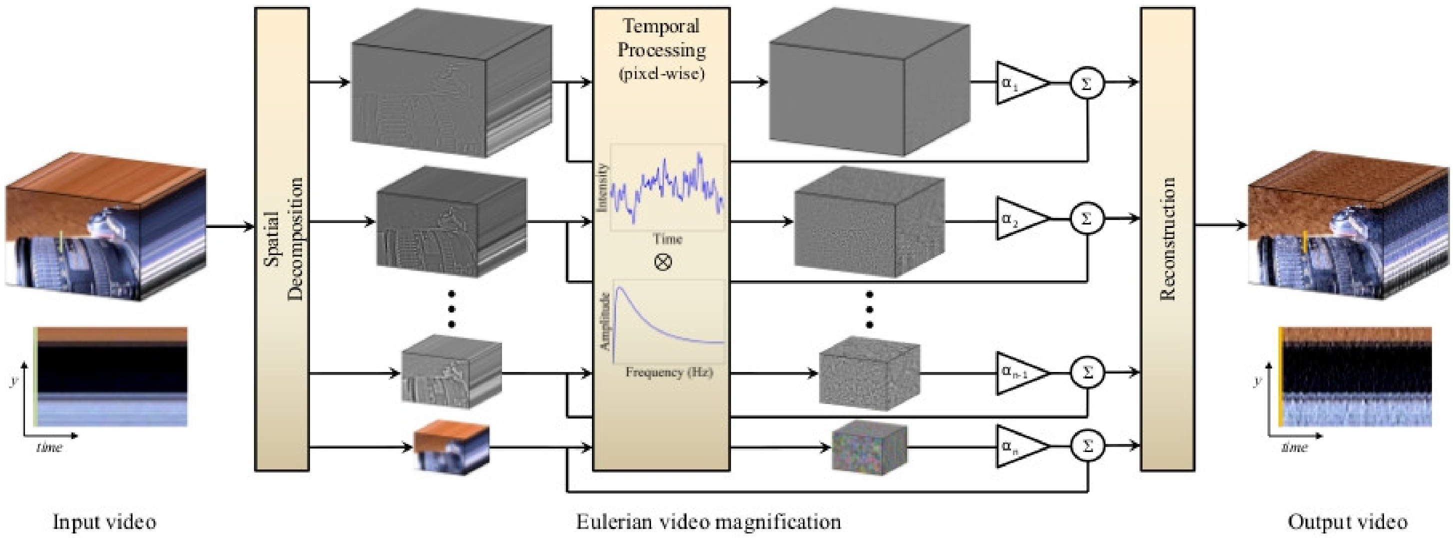
(1)heart rate detection and assessment based on the Eulerian Video Magnification method;
(2)display real-time changes, such as, the magnified blood flow, obtained from the Eulerian Video Magnification method;
(3)deal with artifacts' motion, due to, person and/or webcam movement.

2 Eulerian Video Magnification:

Our approach combines spatial and temporal processing to empha-size subtle temporal changes in a video. The process is illustrated in Figure 1. We first decompose the video sequence into different spatial frequency bands. These bands might be magnified differently because (a) they might exhibit different signal-to-noise ratios or (b) they might contain spatial frequencies for which the linear approx-imation used in our motion magnification does not hold .In the latter case, we reduce the amplification for these bands to suppress artifacts. When the goal of spatial processing is simply to increase temporal signal-to-noise ratio by pooling multiple pixels, we spatially low-pass filter the frames of the video and downsample them for computational efficiency.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In the general case, however, we compute a full Laplacian pyramid [Burt and Adelson 1983]. We then perform temporal processing on each spatial band. We consider the time series corresponding to the value of a pixel in a frequency band and apply a bandpass filter to extract the frequency bands of interest. For example, we might select frequencies within 0.4-4Hz, corresponding to 24-240 beats per minute, if we wish to magnify a pulse. If we are able to extract the pulse rate, we can use a narrow band around that value. The temporal processing is uniform for all spatial levels, and for all pixels within each level. We then multiply the extracted bandpassed signal by a magnification factor α. This factor can be specified by the user, and may be attenuated automatically according to guidelines . Possible temporal filters are discussed Next, we add the magnified signal to the original and collapse the spatial pyramid to obtain the final output. Since natural videos are spatially and temporally smooth, and since our filtering is performed uniformly over the pixels, our method implicitly maintains spatio-temporal coherency of the results.



This process mainly consists spatial filtering, temporal filtering.

Spacial Filtering:

As mention before, the work of [WRS + 12] computes the full Laplacian pyramid [BA83] as a general case for spatial filtering. Each layer of the pyramid may be magnified differently because it may exhibit different signal-to-noise ratios, or contain spatial frequencies for which the linear approximation used in motion magnification does not hold [WRS + 12, Section 3]. spatial filtering may also be used to significantly increase signal-to-noise ratio, as previously mention on section 2.1 and demonstrated on the work of [VSN08] and [WRS + 12]. Subtle signals, such as, a person's heart rate from a video of its face, may be enhanced this way. For this purpose the work of [WRS + 12] computes a layer of the Gaussian pyramid which may be obtained by successively scaling down the image by calculating the Gaussian average for each pixel. However, for the signal of interest to be revealed, the spatial filter applied must be large enough.

Section 5 of [WRS + 12] provides an equation to estimate the size for a spatial filter needed to reveal a signal at a certain noise power level:

S(λ) = S(r) = σ 02 = k
σ 2
r 2
(2.3)

where S(λ) represents the signal over spatial frequencies, and since the wavelength, λ , cutoff of a spatial filter is proportional to its radius, r, the signal may be represented as S(r). The noise power, σ 2 , can be estimated using to the technique of [LFSK06].

Finally, because the filtered noise power level, σ 02 , is inversely proportional to r 2 , it is possible to solve the equation for r, where k is a constant that depends on the shape of the low pass filter

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Temporal Filtering:

Temporal filtering is used to extract the motions or signals to be amplified. Thus, the filter choice is application dependent. For motion magnification, a broad bandpass filter, such as, the butter-worth filter, is preferred. A narrow bandpass filter produces a more noise-free result for color amplification of blood flow.

An ideal bandpass filter is used on [WRS + 12] due to its sharp cutoff frequencies. Alternatively, for a real-time implementation low-order IIR filters can be useful for both: color amplification and motion magnification. These filters are illustrated on 2.2.

3 Embedded Linux device :

The device hardware consists of S5PV210 the high performance Cortex-A8 MCU of 1GHz clock speed ,supports up to 1GBytes DDR2 on 0.65mm pitch, 17X17mm2 FPGA package. This MCU consists of NEON SIMD engine.

OS Linux :

Kernel :

we used kernel 2.6.35 because it is proven on single core arm cortex A processors. And u-boot as system bootloader and ex2 based filesystem on mmc provided on board. And uvcvideo driver we ported to interact with usb based webcam (Logitech) to capture video for processing.

Application :

we used QT frame work on linux frame buffer, for application GUI. QT 4.8 is used for development of application and HMI cross-compilation . Gnu tools from linaro arm-none-linux-gnueabi is used for application compilation. The application is devicde into three parts such as video recording , processing and display results.

Video recording:

The device is equiped wiith Logitech webcam connected through usb cable. The system detects the cam using uvcvideo driver on linux kernel. The streem is dumped into memory buffer.

Processing :

The buffered video frames are stacked for processing algorthm. Our algorithm process the frames in three stages. The NEON optimized gaussian blur and decimation filter builds pyramid stack and IRR filter 0.4Hz to 2.5Hz that is in order to find heart beats from 24 ~ 150 bpm for human heartbeat range. Temporal processing using IRR filter resultas in blood flow changes in face , and this values are amplified 50times . Then the frames are reconstructed using gaussian interpolation or reconstruction filter. The resulted filter are stacked in ram.

EvmGdownIdeal:

This was the first implementation, thus, its goal was to understand how the method worked, and match the implementation provided, in MATLAB, by [WRS + 12]. In addition, real-time support was implemented by using a sliding window of 30 frames. Resize down. This step applies a spatial filter by calculating a level of the Gaussian pyramid. This Is achieved by looping to the desired level where the input to the next loop is the result from the previous loop, starting with the original frame. A Gaussian pyramid level is calculated by, first, convolving the input frame with the kernel,

$$K: \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

and then, downsampling the frame by rejecting even rows and columns.

Temporal filter:

The temporal filter used is the IIR bandpass filter, as described above for the previous implementation, only this time it is applied to each level of the pyramid. Amplification the amplification method in this implementation is more complex than the one previously used. It is based on the implementation provided by [WRS + 12]. It uses a different α value for each band of spatial frequencies, which corresponds to the Laplacian pyramid levels. The magnification value, α, follows the equation:

$$(1+\alpha)\delta(t) < \frac{\lambda}{8}$$

where δ (t) represents the displacement function and λ the spatial wavelength. Furtherdetails about this equation may be found on [WRS + 12, Section 3.2].

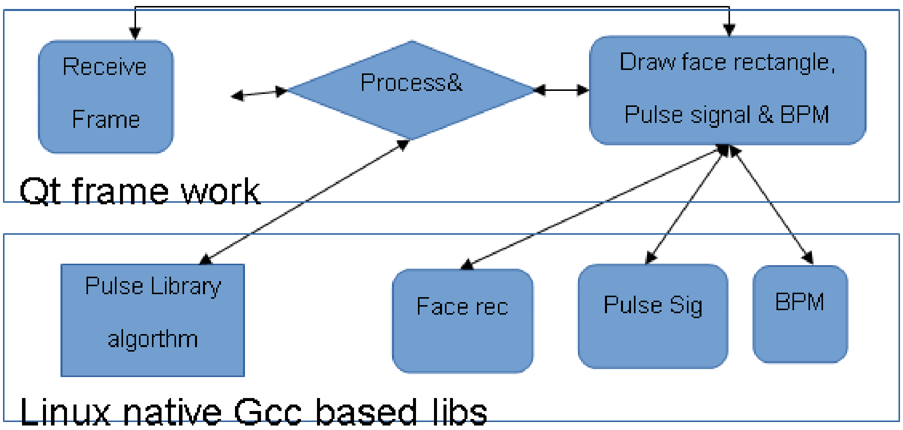
Resize up:

This step reconstructs the original image by iteratively reconstructing each blurred image until the now processed original image is reached.

Heart rate estimation:

Even though the simplification of the fast pulse wave detection algorithm presented on the previous section was capable of counting pulse wave peaks, it would frequently miscount at least by one peak. Thus, since the period analyzed was short, only 5 seconds at 20 frames per second, a miscounted peak would introduce a large error to the final value.

In order to obtain the value of beats per minute from the pulse signal, the method presented on section 2.3.1 was used every time the signal was marked as valid. To prevent big fluctuations, the value was averaged over the values obtained from the power spectrum method every 1 second.



Neon :

There are quite some difference between the two. Neon is a SIMD (Single Instruction Multiple Data) accelerator processor as part of the ARM core. It means that during the execution of one instruction the same operation will occur on up to 16 data sets in parallel. Since there is parallelism inside the Neon, you can get more MIPS or FLOPS out of Neon than you can a standard SISD processor running at the same clock rate.

The biggest benefit of Neon is if you want to execute operation with vectors, i.e. video encoding/decoding. Also it can perform single precision floating point(float) operations in parallel. VFP is a classic floating point hardware accelerator. It is not a parallel architecture like Neon. Basically it performs one operation on one set of inputs and returns one output. It's purpose is to speed up floating point calculations. It supports single and double precision floating point.

Display :

The resulted stack is displayed on scree n as streem of video , the subtle changes can be observed .

4 Results:

The system is efficiently produced the graph and rate of heartbeat almost equal to the results of traditional ECG systems findings. This algorithm makes the revolutionary changes in non-invasive devices for medical application.