13

Real-Time Motion Processing Estimation Methods in Embedded Systems

Guillermo Botella and Diego González
Complutense University of Madrid,
Department of Computer Architecture and Automation
Spain

1. Introduction

Motion estimation is a low-level vision task which manages a high number of applications as sport tracking, surveillance, security, industrial inspection, robotics, navigation, optics, medicine and so on. Unfortunately, many times it is unaffordable to implement a fully functional embedded system for real-time operation which works in a while with enough accuracy due the nature and complexity of the signal processing operations involved.

In this chapter, we will introduce different motion estimation systems and their implementation when real-time is required. For the sake of clarity, it will be previously shown a general description of the motion estimation paradigm. Subsequently, three systems regarding low-level and mid-level vision domain will be explained.

The present chapter is thus organized as follows, after an introductory part, the second section is regarding the motion estimation and optical flow paradigms, the similarities and differences to understand the real-time methods and algorithms. Later will be performed a classification of motion estimation systems according different families and enhancements for further approaches. This second section is finished with a plethora of real-time implementations of the systems explained previously.

The third section focuses on three different case studies of real-time optical flow systems developed by the authors, where also are presented the throughput and resource consuming data from a reconfigurable platform (FPGA-based) used in the implementation.

The fourth section analyses the performance and the resources consumed by each one of this three specific real-time implementations. The rest of the sections, fifth to seventh, approaches the conclusion, the acknowledgements and the bibliography, respectively.

2. Motion estimation

The term "optic flow" was named by James Gibson after the Second World War, when he was working on the development of tests for pilots in the U.S. Air Force (Gibson, 1947). His basic principles provided the basis for much of the work of computer vision 30 years later (Mar, 1982). In later publications, Gibson defines the deformation gradient image along the motion of the observer, a concept that was dubbed "flow".

When an object moves on, for example, a camera, the two-dimensional projection image moves with the project as shown in Figure 1. The projection of three-dimensional motion vectors on the two-dimensional detector is called the apparent field of flow velocities or image. The observer does not have access to the velocity field directly, since the optical sensors provide luminance distributions, not speed, and motion vectors may be decidedly different from this luminance distribution.

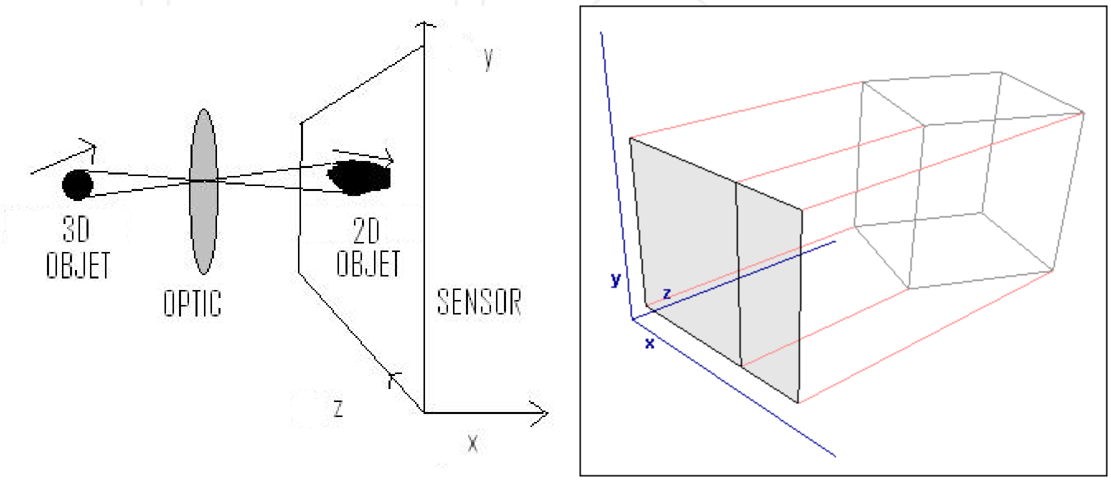

Fig. 1. Projection from 3D world to 2D detector surface.

The term "apparent" refers to one of the capital problems in computer vision, in the absence of speed never real, but a two dimensional field called motion field. However, it is possible calculate the movement of local regions of the luminance distribution, being known as the field of optical flow motion, providing only an approximation to the actual field of speeds (Verri & Poggio, 1989).

As an example, it is easy to see that the optical flow is different from the velocity field. As shown in Figure 2, there is a sphere rotating with constant brightness, which produces no change in the luminance of the image. The optical flow is zero everywhere in contradiction with the velocity field. The reverse situation yields with a static scene with a moving light in which the velocity field is zero everywhere, even though the luminance contrast induces no zero optical flow. One additional example can be appreciated with the rotational poles announcing the barbershops of yesteryears, with the velocity field perpendicular to the flow.

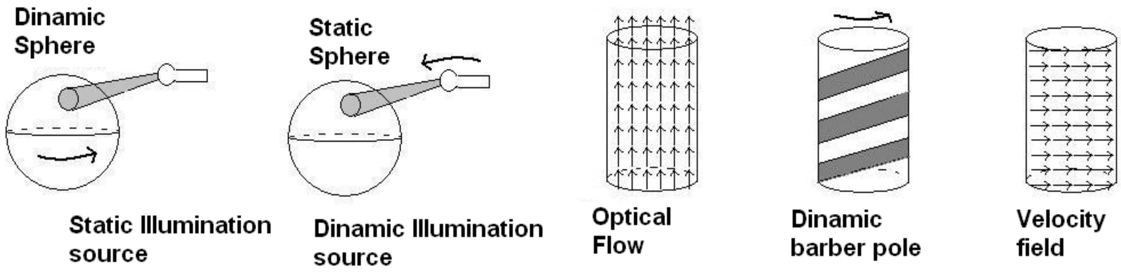

Fig. 2. Difference between velocity field and optical flow.

Among these atypical situations, under certain limits it is possible to recover motion estimation. Motion algorithms described in this chapter recover the flow as an approximation of the velocity field projected flow information. They deliver two dimensional array of vectors that may be subject to several high-level performances. In the literature appears several applications in this regard (Nakayama, 1985; Mitiche, 1996).

Optical flow is an ill-conditioned problem since it is based on imaging three dimensions on a two-dimensional detector. The process removes information, and its recovery is not trivial. Therefore, the flow is a measure of the ill-posed problem, since there are infinite velocity fields that may cause the observed changes in the luminance distribution, in addition, there are infinite three-dimensional movements that can generate a given field of velocities.

Thus, it is necessary to consider a number of clues or signals to restore the flow. The so-called problem of aperture (Wallach, 1976; Horn & Schunck, 1981; Adelson & Bergen, 1985) appears when measuring the two-dimensional velocity components using only local measurements. It is possible recovering the component of the velocity gradient perpendicular to the edge, forcing adding external conditions that usually require obtaining information from a finite neighbourhood area, see Figure 3a. The region must be large enough to get a solution as the search for a corner to resolve this problem, for example. However, the collection of information across a region increases the probability of also taking on different motion contours and hence, to truncate the results, needing a trade-off solution, this last question is named as the aperture general problem (Ong & Spann, 1999).

2.1 State of art estimating real-time motion

There are many algorithms and architectures frequently used for real-time optical flow estimation, emanating from artificial intelligence, signal theory, robotics, psychology and biology. There is an extensive literature, and it is not the purpose of this section to explain all the algorithms. It will be reviewed the state of the art as descriptive as possible, for the sake of clarity, in order to justify the real-time implementations presented specifically at the end of this chapter.

We can classify motion estimation models in three different categories:

- Correlation based methods. They work comparing positions from image structure between adjacent frames and inferring the speed of the change in each location. They are probably the most intuitive methods (Oh, 2000).
- Differential or gradient methods. They are derived from work using the image intensity in space and time. The speed is obtained as a ratio from the above measures (Baker and Matthews, 2004; Lucas & Kanade, 2001).
- Energy methods. They are represented by filters constructed with a response oriented in space and time to work at certain speeds. The structures used in this processing are parallel filter banks that are activated for a range of values (Huang, 1995).

The different approaches to motion estimation are appropriate under each application. According to the sampling theorem (Nyquist, 2006), a signal must be sampled at a sampling rate that is at least twice the highest frequency that has such this signal. Therefore, it ensures us the motion between two frames is small compared to the scale of the input pattern.

When this theorem is no longer fulfilled, it appears the phenomenon of sub-sampling or aliasing. In space-time images, this phenomenon produces incorrect inclinations or structures unrelated to each other, as an example of temporal aliasing, we can observe a rotation of the propeller of the planes in the opposite direction to true as shown in Figure 3b. In short, no long displacements can be estimated from input patterns with small scales. In addition to this problem, we have the problem of aperture, discussed previously. These two problems (aliasing and aperture) fulfill the general problem of correspondence as shown in Figure 3.

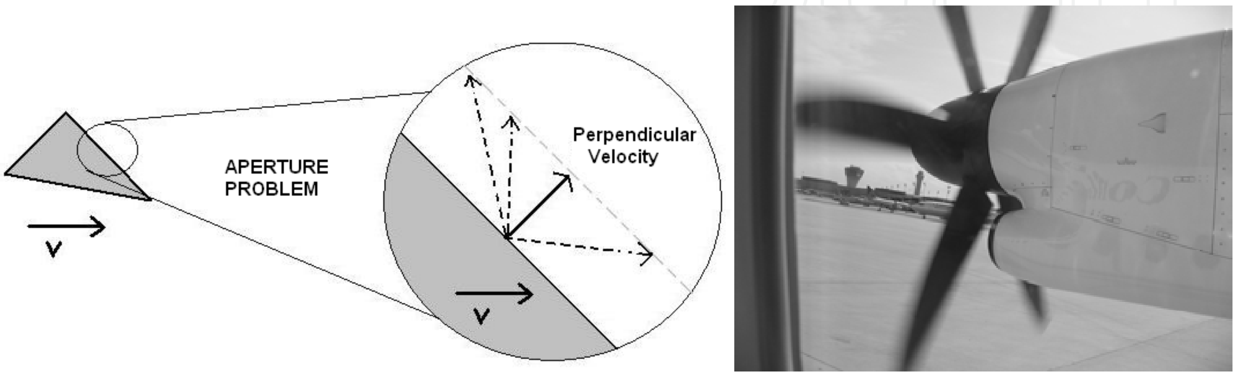

Fig. 3. (3a. left). So-called "Aperture problem". (3b. right) "Aliasing problem". The two problems conform the "correspondence problem" for motion estimation.

Therefore, the movement of the input patterns does not always corresponds to features of consecutive frames in an unambiguous manner. The physical correspondence may be undetectable due to the problem of aperture, the lack of texture (example of Figure 2), the long displacements which commute between frames, etc. Similarly, the apparent motion can lead to a false correspondence. For such situations, it is possible using matching algorithms (tracking and correlation), although currently there is much debate about the advantages and disadvantages of using these techniques rather than those based on gradient and energy of motion.

The correlation methods are less sensitive to changes in lighting, they are able to estimate long displacements that do not meet the sampling theorem (Yacoob & Davis, 1999). However, they are extremely sensitive to cyclical structures providing various local minima and when the aperture problem arises, the responses obtained are unpredictable.

Alternatively, the other methods are better in efficiency and accuracy; they are able to estimate the perpendicular optical flow (in the presence of the aperture problem).

Typically in machine vision CCD cameras are used with a discrete ratio, where varying this modifies the displacement between frames, if these shifts are too large, so gradient methods fail (since it fractures the continuity of space-time volume). Although it is possible using an anti-aliasing spatial smoothing to avoid temporal aliasing (Christmas, 1998; Zhang & Wu, 2001), this is the counterpart to degrade spatial information. Therefore, for a given spatial resolution, one has to sample at a high temporal frequency (Yacoob & Davis, 1999).

On the other hand, it is quite common, for real-time optical flow algorithms remain a functional architecture, as shown in Figure 4, via a hierarchical process.

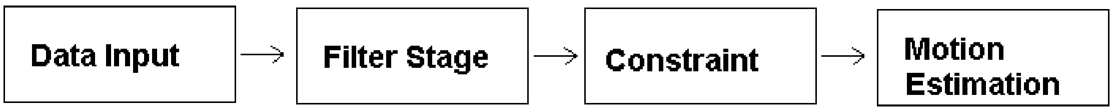

Fig. 4. Functional real-time architecture found in most optical flow algorithms

Previously, it filters the sequence of images or the temporary buffer to obtain basic measures through convolutions, Fast Fourier Transform (FFT), extraction of patterns, arithmetic, and so on.

The measures are then recombined by through various methods to reach a basic evaluation of speed (usually incomplete and deficient in these early stages). Subsequently, the final flow estimation is done by imposing a set of constraints on action and results. These are generated by assumptions about the nature of the flow or change (such as restrictions of a rigid body), but even with these restrictions, the retrieved information is often not robust enough to get a unique solution for optical flow field.

At the beginning of this section, was explained that the optical flow motion is estimated from the observable changes in the pattern of luminance over time. In the case of a no detectable movement, situation such as a sphere rotating with the same brightness (Figure 2), the estimated optical flow is zero everywhere, even if current speeds are not zero. Another attention is given by the non-existence of a unique movement of the image, in order to justify a change in the observed brightness, therefore, the visual motion measurement is often awkward and always have to be associated with a number of physical interpretations.

2.2 Basic movement patterns

Despite the difficulties in the recovery of flow, the biological systems work surprisingly well in real-time. In the same way that these systems have specialized mechanisms to detect color and stereopsis, are also devoted to visual motion mechanisms (Albright, 1993). As in other areas of research in computer vision, they are formed models of such natural systems to formalize bio-inspired solutions.

Thanks to psychophysical and neurophysiological studies, it has been possible to build models that extract the motion from a sequence of images, usually characterized these biological models to be complex and designed to operate poorly at high speed in real-time.

One of the first models based on real-time bio-inspired visual sensor was proposed by Reichardt (Reichardt, 1961). The detector consists of a couple of receptive fields as shown in Figure 5, where the first signal is delayed with respect to the second before nonlinearly combined by multiplication.

Receptors 1 and 2 (shown as edge detectors) are spaced a distance ΔS, imposing a delay on each signal, after the signal C1 and C2 are operated via multiplication. In the final stage, the result of the first half of the detector is subtracted from the next, and then estimated what each contributes to increased directional selectivity. The sensors shown in Figure 5 are Laplacian detectors, although it is possible using any spatial filter or feature detector.

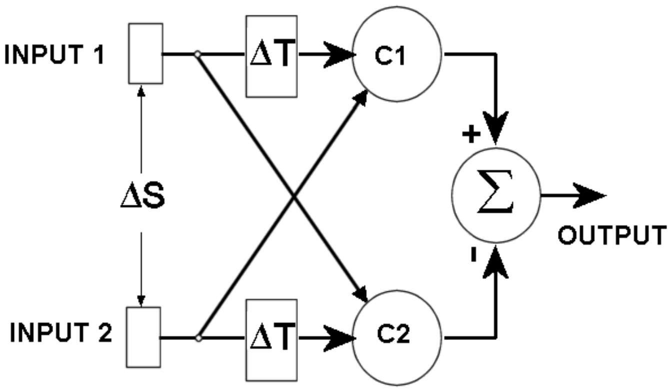

Fig. 5. Reichardt real-time correlation model.

One of the main disadvantages of this detector, is that the correlation is dependent on the contrast, in addition, no speed can be retrieved directly. It requires banks of detectors calibrated at various speeds and directions, and its interpretation at least ambiguous. However, despite these drawbacks, the Reichardt detector can be easily applied by biological systems and is used successfully to explain the visual system of insects. The detector continues to be used as a starting point for more sophisticated models of vision (Beare & Bouzerdoum, 1999; Zanker, 1996), detectors can be implemented in real-time CCD sensor using VLSI technology (Arias-Estrada *et al.*, 1996).

2.2.1 Change detection and correlation methods

Considering the basic case of a segmented region with motion in static regions, it comes up as result a binary image which shows regions of motion. The process may seem easy, so they are simply looking for changes in image intensity over a threshold, which is supposed to cause the movement of an object in the visual field. However, the number of false positives that stem from sources such as noise sensors, camera movement, shadows, environmental effects (rain, reflections, etc.), occlusions and lighting changes make extraordinarily difficult to detect robust movement.

Biological systems again despite being highly sensitive to movement are also robust to noise and visual effects uninteresting. This technique is used in situations where motion detection is an event that should be taken into account for future use. Currently, the requirements of estimation for these algorithms are minimal reaching a satisfactory result with little more than an input buffer, arithmetic signs and some robust statistics, as the supervisory systems have to be particularly sensitive and not normally available in large computing power (Rosin, 1998; Pajares, 2007).

When the differential approaches are subject to errors due to noncompliance with the sampling theorem (Nyquist, 2006) or inconvenience lighting changes, it is necessary to apply other strategies. The methods of correlation or pattern matching are the most intuitive to regain speed and direction of movement, work characteristics in selecting a frame in the sequence of images and then looking for these same characteristics in the next as shown in Figure 6. Changes in the position indicate movement in time, i.e speed.

These algorithms are characterized by a poor performance due to its exhaustive search and iterative operations, usually requiring a prohibitive amount of resources. If the image is of size M^2, the search template is size N^2, and the search window size is L^2, then the whole estimate of computational complexity required would be around $M^2N^2L^2$. By way of example, with an typical image of 640x480 points, a template window size 50x50 and search one of 100x100, would be required to compute 0.8 billion (long scale) operations. The current trend is to try to reduce the search domain (Oh & Lee, 2000; Anandan *et al.*, 1993; Accame *et al.*, 1998), although still the need for resources is too high. One of the most common application models used is the encoding of video in real time (Accame *et al.*, 1998; Defaux & Moscheni, 1995) increasing the amount of effort devoted to research in these algorithms.

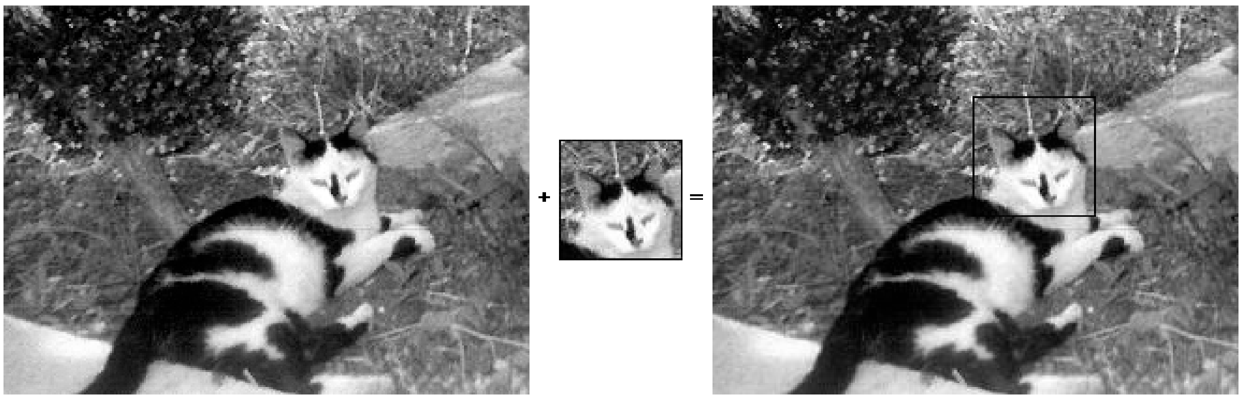

Fig. 6. Block-Matching technique.

A key in video compression is the similarity between adjacent images temporarily in a sequence. This technique, demand less bandwidth to transfer the differences between frames than to transfer the entire sequence. It is possible even further reduce the data amount transmitted if known a priori the movement and deformation needed to move from one frame to the next.

This family of algorithms can be classified deeply, and there are two prominent approaches:

- Correlation as a function of 4 variables, depending on the position of the window and displacement, with output normalized between 0 and 1, independent of changes in lighting.
- Minimizing the distance, quantifying the dissimilarity between regions. Many optimizations have been on the line to reduce the search space (Oh & Lee, 2000) and increase their speed (Accame *et al.*, 1998).

Adelson and Bergen (Adelson & Bergen, 1985) advocate no biological evidence of such models, since they are not able to make predictions about complex stimuli (as example randomly positioned vertical bars), for which, experimental observers perceive different moves in different positions. These techniques are straightforward, have spent many years researching and dominate in industrial inspection and quality control.

The ability to work in environments where the displacements between frames are longer than a few points is one of the main advantages. Though this requires extensive processing search spaces.

2.2.2 Space-time methods: Gradient and energy

The movement can be considered as an orientation in the space-time diagram. For example, Figure 7a presents a vertical bar moving continuously from left to right, sampled four times over time. Examining the space-time volume, we can observe the progress of the bar about the time axis thus a stationary angle oriented which shows the extent of movement.

The orientation of the space-time structure can be retrieved through low-level filters. There are currently two dominant strategies: the gradient model and the model of energy as shown in Figure 7b, where the ellipsis represents the negative and positive lobes.

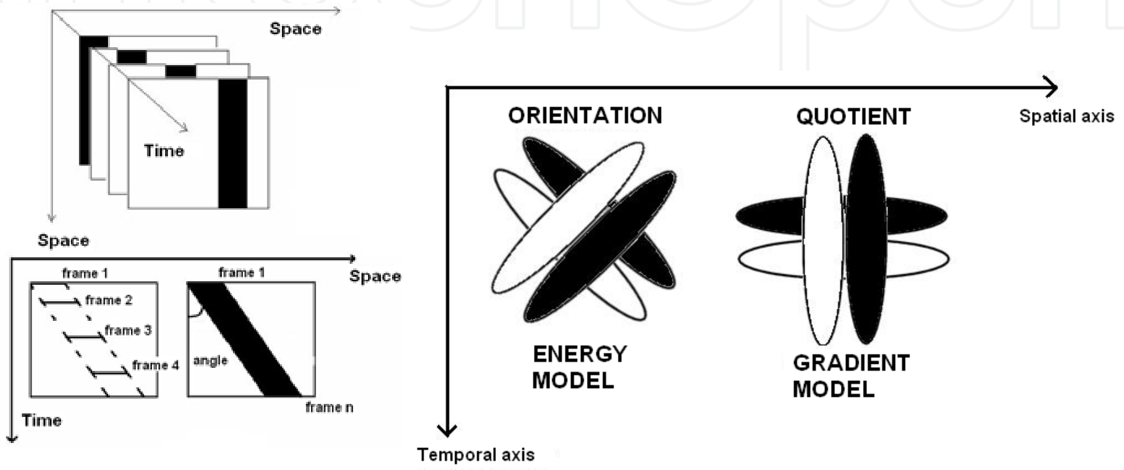

Fig. 7. (7a.left). Motion as orientation in the space-time (*x-t* plane) where the angle increases with the velocity. (7b.right). Space-time filtering models.

The gradient model applies the ratio of a spatial and a temporal filter as a measure of speed, however, the energy model uses a set of filter banks oriented in space-time. Both models use a bio-inspired tightly filters (Adelson & Bergen, 1985; Young & Lesperance, 1993). The debate about which is the scheme adopted by the human visual system remains open, and there are even gateways to go from one model to another because it is possible to synthesize the filters oriented in the pattern of energy through space-time separable filters (Fleet & Jepson, 1989; Huang & Chen, 1995). It is also interesting to note that independent component analysis (ICA) notices these spatial filters are those that cover the majority of components of the structure of the image (Hateren & Ruderman, 1998). In the gradient model, the main working hypothesis is the conservation of intensity over time (Horn & Schunk, 1981). Assuming this, over short periods of time, the intensity variations are due only to translation and not to lighting change, reflectance, etc. The total derivative of image intensity with respect to time is zero at every point in space-time, therefore, defining the image intensity as I (x, y, t), we have:

$$dI(x, y, t) / dt = 0 \qquad (1)$$

Differentiating by parts, we obtain the so-called motion constraint equation:

$$\frac{\partial I}{\partial x}\frac{dx}{dt}+\frac{\partial I}{\partial y}\frac{dy}{dt}+\frac{\partial I}{\partial t}\frac{dt}{dt}=0 \qquad (2)$$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{3}$$

where u=dx/dt and v=dy/dt. The parameters (x, y, t) are omitted for the sake of clarity. Since there is only one equation with two unknowns, (two unknown velocity components), it is possible to recover only the velocity component v_n, which lies in the direction of the gradient of luminance.

$$v_n = \frac{-\partial I/\partial t}{\sqrt{\left(\partial I/\partial x\right)^2 + \left(\partial I/\partial y\right)^2}} \tag{4}$$

There are several problems associated with the motion constraint equation, because it is an equation with two unknowns, therefore, insufficient for estimating the optical flow. Using just only equation (3), is possible only to obtain a linear combination of velocity components, this effect, moreover, is fully consistent with the aperture problem mentioned before in the present section. A second problem arises if I_x or I_y spatial gradients become very small or zero, in which case, the equation becomes ill-conditioned and estimated speed tends asymptotically to infinity. Furthermore, the stable realization of the spatial derivatives is something problematic by itself, applying a differential filter convolution, as operators of Sobel, Prewitt or difference of Gaussians.

As they are using numerical derivatives of a function sampled, they are best suited for space-time intervals small, the problem of aliasing will appear every time the sample in the space-time is not enough, especially in the time domain, as commented before. There are several filtering techniques to solve this problem, such as a spatiotemporal low-pass filtering as noted by Zhang (Zhang & Jonathan, 2003).

Ideally, the sampling rate should be high enough to reduce all movements within one pixel/frame, so the temporal derivative is well-conditioned (Nyquist, 2006). Moreover, the differential space-time filters that are used to implement gradient algorithms seem reasonably to those found in the visual cortex, although there is no consensus on the optimal from the point of view of functionality (Young and Lesperance, 2001). One advantage of models on the energy gradient, is that they provide a speed from a combination of filters, however energy models provide a population of solutions.

2.3 Improving optical flow measures

We have seen that the equation of motion constraint (MCE) has some anomalies have to be addressed properly to estimate optical flow. There is a wide range of methods to improve it. Many restrictions apply to resolve the two velocity components *u* and *v*, collecting more information (through the acquisition of more images or getting more information for each image) otherwise, applying physical restraints to generate additional MCEs:

- Applying multiple filters (Mitiche, 1987; Sobey & Srinivasan, 1991; Arnspang, 1993; Ghosal & Mehrotra, 1997).
- Using a neighborhood integration (Lucas & Kanade, 1981; Uras *et al.*, 1988; Simoncelli & Heeger, 1991).

- Using multispectral images (Golland & Bruckstein, 1997).

There are different general methods for restricting MCE and improve optical flow measures:

- Conducting global optimizations, such as smoothing (Horn & Schunck, 1981; Nagel, 1983; Heitz & Bouthemy, 1993).
- Restricting the optical flow to a model known, for example, the affine model (Liu *et al.*, 1997; Ong & Span, 1997; Fleet & Jepson, 2000).
- Using multi-scale methods in a space and time domain (Anandan, 1989; Webber, 1994; Yacoob & Davis, 1999).
- Exploiting temporal consistency (Giaccone & Jones, 1997, 1998).

2.3.1 Applying physical restraints to generate additional MCEs

A motion constraint equation alone is not sufficient to determine the optical flow, as indicated previously. It is proposed an refinement of the same given that its partial derivatives provide additional solutions to the flow working as multiple filters. Nagel (Nagel, 1983) was the pioneer in applying this method uses second-order differentiates, in fact, the differential operator is one of many that could be used to generate multiple MCES. Usually these operators are used numerically by convolutions as linear operators.

This process works because the convolution does not change the orientation of the space-time structure. On the other hand, it is important to use filters that are linearly independent otherwise the produced MCES will degenerate and will not have won anything. The filters and their differentials can be estimated previously to achieve efficiency and, due to the locality of the operators, a massively parallel implementation of these structures.

It is possible also using neighborhood information from local regions to generate motion constraint equations extras (Lucas & Kanade, 1981; Simoncelli & Heeger, 1991). It is assumed, therefore, that the movement is a pure translation in a local region, where these constraints are modeled using a weight matrix so that the results are placed centered within a local region as for example, following a Gaussian distribution. It is rewritten then, the MCE as a minimization problem. The error term is minimized, or solved the set of equations generated by numerical methods.

Working with multicolored images, can be generated different functions of brightness. For example, the planes of red, green and blue of a standard camera can be treated as three separate images, producing three MCES to solve. As a counterpart to this multispectral method, it should be noted that the color planes are usually correlated, a fact, moreover, that is exploited by most compression algorithms. In these situations, the linear system of equations can be degenerate, so that ultimately there is no guarantee that the extra cost in computing lead to an improvement in the quality of flow.

A variation of this method is using additional invariance with respect small displacements and lighting changes, basing these measures in the proportion of different planes of color (spectral sensitivity functions) such as RGB or HSV commonly used. Using this last variant is obtained significant improvements over the use of a single plane RGB (Golland & Bruckstein, 1997).

2.3.2 Different general methods for restricting MCE and improve optical flow measures

Due to the lack of information and spatial structure of the image, is not easy to estimate a sufficiently dense velocity field.

To correct this problem, several restrictions are applied, as for example, that the points move closer together in a similar way. The general philosophy is that the original flow field, once estimated, is iteratively regularized with respect to the smoothing restriction.

The first constraint was proposed by Horn and Schunk (Horn & Schunk, 1981). Optic flow resulting from the global constraints, is quite robust due to the combination of results, and is also flattering to the human eye. Two of the biggest drawbacks are its iterative nature, requiring large amounts of time and computing resources, and motion discontinuities are not handled properly, so that erroneous results are produced in the regions surrounding the motion edges. To address these latter gaps are proposed other techniques that use global statistics such as random Markov chains (Heitz & Bouthemy, 1993).

In all MCE estimation techniques, appear significant restrictions on a neighborhood where it is assumed constant flux. To meet this requirement when there are movement patterns, this neighborhood has to be as small as possible, but at the same time it must be large enough to obtain information and to avoid the aperture problem. Therefore, we need a trade-off compromise.

A variety of models uses estimations related to this neighborhood, such as least squares. If using a quadratic objective function, is assumed a Gaussian residual error rate, but if having multiple movements in the neighborhood, these errors can no longer be considered Gaussian. Even if these errors were independent (very usual situation) the error distribution can be modeled as a bimodal.

There are approximate models can be incorporated into the flow range of techniques that are being exposed. These approaches also model spatial variations of multiple movements. The neighborhood integration techniques, as mentioned, assume that the image motion is purely translational in local regions. Thus, more elaborate models (such as the affine model) can extend the range of motion, and provides additional restrictions. These methods recast the MCE with an error function that will resolve or minimize least squares (Campani & Verri, 1992; Bergen & Bart, 1992; Gupta & Kanal, 1995, 1997; Giaccone & Jones, 1997, 1998).

Large displacements between frames that originate gradient methods, behave inappropriately, since the image sequences are insufficient or that the time derivative measures are inaccurate. As a workaround it is possible using larger spatial filters than early model (Christmas, 1998).

The use of multi-scale Gaussian pyramid can handle high movements between frames and fill the gaps in large regions where the texture is uniform, so that estimates of coarse-scale motion are used as sources for a finer scale (Zhang, 2001).

The use of temporal multi-scale (Yacoob & Davis, 1999) also allows the accurate estimation of a range of different movements, but this method requires using a high enough sampling rate to reduce movements about a pixel/frame.

The schemes discussed so far, consider the calculation of optical flow as a separate problem for each frame, without any feedback (the results of motion of a frame does not cover the analysis of the following). Giaccone and Jones (Giaccone & Jones; 1998, 1999) have designed an architecture capable of dealing with multiple motions (keeping the temporal consistency) which segments moving regions by using a method of least squares.

This algorithm has proven to be robust for a given speed limits, also works well when compared to similar models. The cost calculation is overwhelmed by the generation of a projected image, PAL sizes needed for about 40 seconds/image, with SPARC 4/670MP.

However, this time consistency constraint is only used sporadically today. The objects in the real world must obey physical laws of motion and the inertia and gravity, so that there is predictability in their behavior, and at least surprising, that most real-time algorithms do not implement a flow based feedback.

For this purpose, it is used the fact that it is possible to create an additional constraint equation from the velocity field for use in the next iteration, managing the problem as an evolutionary phenomenon. The use of probabilistic models or Bayesian (Simoncelli & Heeger, 1991) may be an alternative to using real world information and update results of previous estimates integrating temporal information.

We have seen that the perception of motion can be modeled as an orientation in space-time, where the methods of extracting this orientation gradient across the filter ratio oriented. The so-called motion energy models are often based or similar in many respects to models of gradient, since both systems use filter banks to obtain this time-space orientation, and therefore the motion. The main difference is that the filters used in energy models, are designed to meet time-space directions, rather than a ratio of filters. The design of space-time oriented filters is usually performed in the frequency domain.

The methods of energy of motion are biologically plausible, but the implementations have an extra computer associated high due to the large number of filtering required being difficult its implementation in real-time. The resultant velocity of energy methods is not obtained explicitly, unlike gradient methods, only using a solution population, being these last Bayesian models.

One advantage, is that the bimodal velocity measurements as invisible movements, can be treated by these structures (Simoncelli & Heeger, 1991). The correct interpretation of the processed results is not an easy task when dealing with models of probabilistic nature. Interesting optimizations have been developed to increase the speed of these methods, combined with Reichardt detectors (Franceschini *et al*, 1992) as support.

2.4 Configurable hardware algorithms implemented for optical flow

There are several real-time hardware systems founded on the algorithms mentioned herein, proceeding to do a quick review.

- Some algorithms used are of matching or gradient, such as Horn and Schunck algorithm (Horn & Schunk, 1981) that has been carried out using a FPGA (Zuloaga *et al*., 1998; Martin *et al*., 2005). The model used is straightforward, is not robust and does not provide optimal overall results in software, but the implementation is efficient and the model is capable of operating in real time. The design uses a recursive implementation of the constriction of applying a smoothing iteration in each frame.

- There is also an implementation of Horn and Schunck algorithm (Horn & Schunk, 1981) by Cobos (Cobos *et al*., 1998) on a FPGA platform, but with the same counterparty recorded earlier on the reliability of the model used and grazing real time.
- ASSET-2 algorithm is based on features and has been implemented to run in real time using custom hardware (Smith, 1995). The algorithm is easy and determines the position of the axes and corners trying to solve the problem of the correspondence between neighboring frames in time. The system was implemented using a PowerPC with custom hardware to extract features in real time. This system does not provide continuous outcomes, these being scarce, but cluster groups of similar speeds to segment objects according to their movement.
- Niitsuma *et al*. (Niitsuma & Maruyama, 2004) apply a model-based optical flow correlation, with a joint operation of a stereoscopic system also measures the distance of moving objects. This system is based on a Virtex 2 XC2V6000 scheduled at 68 MHz and delivers results in real time resolution of 640x480 points.
- Tomasi & Diaz (Díaz *et al*., 2006) implemented a real-time system based on Lucas and Kanade algorithm (Lucas & Kanade, 1981; Díaz *et al*., 2006) with satisfactory results in terms of performance. This is an algorithm within the so-called gradient that is used as a didactic introduction to optical flow in most colleges. It has shown a significant relationship between performance and implementation effort. The problem with this implementation is given by abrupt changes in light and heavy dependence on the aperture problem. Its advantage is the extensive documentation and the experience gathered with this algorithm (more than 25 years) from the scientific community. After that, Tomasi has implemented in 2010 and 2011 a fully real-time multimodal system mixing motion estimation and binocular disparity (Tomassi *et al*., 2010, 2011) combining low-level and mid-level vision primitives.
- Botella *et al*. implemented a robust gradient based optical flow real-time system and its extension to mid-level vision combining orthogonal variant moments (Botella *et al*., 2009, 2010, 2011). Also the block matching acceleration motion estimation has been implemented in real-time by Gonzalez and Botella (González *et al*., 2011). All these models will be analyzed thoroughly in this chapter.

3. Case studies of real-time implementation performed

They are presented several case studies of the real-time implementation performed in these last years by the author of the present chapter and other authors.

3.1 Multichannel gradient Model

Multichannel gradient Model (McGM), developed by Johnston (Johnston *et al*., 1995, 1996), has been recently implemented and selected due its robustness and bio-inspiration. This model deals with many goals, such as illumination, static patterns, contrast invariance, noisy environments. Additionally it is robust against fails, justifies some optical illusions (Anderson *et al*., 2003), and detects second order motion (Johnston, 1994) that is particularly useful in camouflage tasks, etc. At the same time, it avoids operations such as matrix inversion or iterative methods that are not biologically justified (Baker & Matthews, 2004; Lucas & Kanade, 1981). The main drawback of this system is its huge computational complexity. It is able to handle complex situations in real environments better than others algorithms (Johnston *et al.*, 1994), with its physical architecture and design principles being based on the biological neural systems of mammalians (Bruce *et al.*, 1996). Experimental results are provided using a Celoxica RC1000 platform (Alphadata, 2007).

This approach is based on the gradient one commented previously, the starting point is the motion constraint equation (MCE) shown in the expression 2. The luminance variation is assumed negligible over time. In this approach, velocity is calculated dividing the temporal derivative by the spatial derivative of image brightness, thus a gradient model can be obtained applying couples of filters, one of them being the spatial derivative and the other one a temporal derivative. If for the sake of clarity we only consider x variable in expression 2 and we get the velocity taking the quotient of the output filters:

$$v = \frac{dx}{dt} = -\frac{\partial I}{\partial t} \Big/ \frac{\partial I}{\partial x} \tag{5}$$

Since velocity is given directly via the ratio of luminance derivatives, one potential problem appears when the output of the spatial filter is null, being the velocity undefined. This can be solved applying a threshold to the calculation or restricting the evaluation value (Baker & Matthews, 2004). Our approach is based on the fact that the human visual system measures at least three orders of the spatial derivative (Johnston *et al.*, 1999; Koenderick & Van Doorn, 1988) and three orders of temporal differentiation (Baker & Matthews, 2004; Hess & Snowden, 1992). Therefore, it is possible to build low level filters for calculating the speed with additional derivatives, although it may be still ill-conditioned:

$$\mathrm{v} = -\frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^{n-1}\mathrm{t}} \Big/ \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^n} \tag{6}$$

Two vectors X and T, containing the results of applying the derivative operators to the image brightness, can be built:

$$\mathrm{X} = \left(\frac{\partial \mathrm{I}}{\partial \mathrm{x}}, \frac{\partial^2 \mathrm{I}}{\partial \mathrm{x}^2}, \ldots, \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^n}\right) \quad \mathrm{T} = \left(\frac{\partial \mathrm{I}}{\partial \mathrm{t}}, \frac{\partial^2 \mathrm{I}}{\partial \mathrm{x}\partial \mathrm{t}}, \ldots, \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^{n-1}\partial \mathrm{t}}\right) \tag{7}$$

For extracting the best approximation to the speed from each of the measurements, a Least Squares Formulation is applied, thus recovering a value v'. The denominator is a sum of squares and therefore it is never null, so a spatial structure is provided:

$$\mathrm{v}' = \frac{\sum_n \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^n} \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^{n-1}\partial \mathrm{t}}}{\sum_n \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^n} \frac{\partial^n \mathrm{I}}{\partial \mathrm{x}^n}} \tag{8}$$

In this framework, we represent the local image structure in the primary visual cortex as a spatial-temporal truncated Taylor expansion (Johnston *et al.*, 1996, 1999; Koenderick & Van

Doorn, 1988), in order to represent a local region by the weighted outputs of a set of filters applied at a location in the image as shown in the next expression (9). The weights attached depend on the direction and length of the vector joining the point where the measurements are taken and the point where we wish to estimate image brightness:

$$\begin{aligned} \mathrm{I}(\mathrm{x}+\mathrm{p},\mathrm{y}+\mathrm{q},\mathrm{t}+\mathrm{r}) &= [\mathrm{I}(\mathrm{x},\mathrm{y},\mathrm{t})]+[\mathrm{I}_{\mathrm{x}}(\mathrm{x},\mathrm{y},\mathrm{t})+\mathrm{I}_{\mathrm{y}}(\mathrm{x},\mathrm{y},\mathrm{t})+\mathrm{I}_{\mathrm{t}}(\mathrm{x},\mathrm{y},\mathrm{t})] \\ &+\frac{1}{2}[\mathrm{I}_{2\mathrm{x}}(\mathrm{x},\mathrm{y},\mathrm{t})\mathrm{p}^2+\mathrm{I}_{\mathrm{xy}}(\mathrm{x},\mathrm{y},\mathrm{t})2\mathrm{pq}+\mathrm{I}_{\mathrm{xt}}(\mathrm{x},\mathrm{y},\mathrm{t})2\mathrm{pr}+\mathrm{I}_{\mathrm{yt}}(\mathrm{x},\mathrm{y},\mathrm{t})2\mathrm{qr}]+\ldots \end{aligned} \tag{9}$$

These filters are generated by progressively increasing the order of the spatial and temporal differential operators applied to the following kernel filter:

$$K(r,t)=\frac{1}{4\pi\sigma}e^{-\frac{r^2}{4\sigma}}\frac{1}{\sqrt{\pi}\tau\alpha\ e^{r^2/4}}e^{-\left(\frac{\ln(t/\alpha)}{\tau}\right)_2} \tag{10}$$

with σ=1.5, α=10 and τ=0.2. This expression is originally scaled so that the integral over its spatial-temporal scope is equal to 1.0. Also, it is tuned assuming a spatial frequency limit of 60 cycles/deg and a critical flicker fusion limit of 60 Hz, following evidences from the human visual system (Johnston *et al.*, 1994).

The Taylor representation requires a bank of linear filters, taking derivatives in time, t, and two spatial directions, x and y, with the derivatives lobes in theirs receptive fields tuned to different frequencies. There are neurophysiological and psychophysical evidences that support this (Johnston *et al.*, 1994; Bruce *et al.*, 1996). This algorithm can be built by neural systems in the visual cortex since all operations involved in the model can be achieved by combining the output of linear space-temporal orientated filters through addition, multiplication and division.

A truncated Taylor expansion is built eliminating terms above first order in time and orthogonal direction for ensuring that there are no more than three temporal filters and no greater spatial complexity in filters (Hess & Snowden, 1992).

The reference frame is rotated through a number of orientations with respect to the input image. Several orientations (24 in the original model spaced over 360°) are employed. For each orientation, three vectors of filters are created by differentiating the vector of filter kernels with respect to *x*, *y* and *t*. From these last measurements, speed, orthogonal speed, inverse speed and orthogonal inverse speed are calculated with the local speed by rotating the reference frame:

$$\hat{\mathrm{s}}_{\mathrm{II}}=\text{speed}=\frac{\mathrm{X}\cdot\mathrm{T}}{\mathrm{X}\cdot\mathrm{X}}\cos^2\theta=\frac{\mathrm{X}\cdot\mathrm{T}}{\mathrm{X}\cdot\mathrm{X}}\left(1+\left(\frac{\mathrm{X}\cdot\mathrm{Y}}{\mathrm{X}\cdot\mathrm{X}}\right)^2\right)^{-1} \tag{11}$$

$$\hat{\mathrm{s}}_{\perp}=\text{orthog. speed}=\frac{\mathrm{Y}\cdot\mathrm{T}}{\mathrm{Y}\cdot\mathrm{Y}}\sin^2\theta=\frac{\mathrm{Y}\cdot\mathrm{T}}{\mathrm{Y}\cdot\mathrm{Y}}\left(1+\left(\frac{\mathrm{X}\cdot\mathrm{Y}}{\mathrm{Y}\cdot\mathrm{Y}}\right)^2\right)^{-1} \tag{12}$$

$$\breve{\mathrm{s}}_{\mathrm{II}} = \text{inverse speed} = \frac{\mathrm{X}\cdot\mathrm{T}}{\mathrm{T}\cdot\mathrm{T}} \tag{13}$$

$$\breve{\mathrm{s}}_{\perp} = \text{inverse orthog. speed} = \frac{\mathrm{Y}\cdot\mathrm{T}}{\mathrm{T}\cdot\mathrm{T}} \tag{14}$$

Where *X*, *Y*, *T* are the vector of outputs from the x, y, t differentiated filters. Raw speed ($X \cdot T$ / $X \cdot X$) and orthogonal speed ($Y \cdot T$ / $Y \cdot Y$) measurements are ill-conditioned if there is no change over x and y, respectively.

To avoid degrading the final velocity estimation, they are conditioned by measurements of the angle of image structure relative to the reference frame (θ). If the speed is large, (and inverse speed is small) direction is led by the speed measurements. However, if the speed is small (and inverse speed is large) the measurement is led by inverse speed.

The use of these antagonistic and complementary measurements provides advantages in any system susceptible of having small signals affected by noise (Anderson *et al.*, 2003). There is evidence of the neurons that built inverse speed (Lagae *et al.*, 1983), also it provides an explanation to sensitivity to static noise for motion blind patients.

The calculus of additional image measurements in order to increase robustness is one of the core design criteria that enhances the model. Finally, the motion modulus is calculated through a quotient of determinants:

$$\mathrm{Modulus}^2 = \left[\frac{\begin{matrix}\widehat{\mathrm{s}}_{\mathrm{II}}\cos\theta & \widehat{\mathrm{s}}_{\mathrm{II}}\sin\theta \\ \widehat{\mathrm{s}}_{\perp}\cos\theta & \widehat{\mathrm{s}}_{\perp}\sin\theta\end{matrix}}{\begin{matrix}\widehat{\mathrm{s}}_{\mathrm{II}}\breve{\mathrm{s}}_{\mathrm{II}} & \widehat{\mathrm{s}}_{\mathrm{II}}\breve{\mathrm{s}}_{\mathrm{II}} \\ \widehat{\mathrm{s}}_{\perp}\breve{\mathrm{s}}_{\perp} & \widehat{\mathrm{s}}_{\perp}\breve{\mathrm{s}}_{\perp}\end{matrix}}\right] \tag{15}$$

Speed and orthogonal speed vary with the angle of reference frame. The numerator of (12) takes a measure of the amplitude of the distribution of speed measurements that are combined across both speed and orthogonal speed. The denominator is included to stabilize the final speed estimation. Direction of motion is extracted by calculating a measurement of phase that is combined across all speed related measures, since they are in phase:

$$\mathrm{Phase} = \tan^{-1}\left(\frac{(\breve{\mathrm{s}}_{\mathrm{II}} + \widehat{\mathrm{s}}_{\mathrm{II}})\sin\theta + (\breve{\mathrm{s}}_{\perp} + \widehat{\mathrm{s}}_{\perp})\cos\theta}{(\breve{\mathrm{s}}_{\mathrm{II}} + \widehat{\mathrm{s}}_{\mathrm{II}})\cos\theta - (\breve{\mathrm{s}}_{\perp} + \widehat{\mathrm{s}}_{\perp})\sin\theta}\right) \tag{16}$$

The model can be degraded to an ordinary Gradient (Baker & Matthews, 2004; Lucas & Kanade, 1981) by suppressing the number of orientations of the reference frame, taking only one temporal derivative and two spatial derivatives, not considering the inverse velocity measures, etc. The computations of speed and direction are based directly on the output of filters applied to the input image, providing a dense final map of information.

The general structure of the implementation is summarized in Figure 8, where operations are divided in conceptual stages with several variations to improve the viability of the hardware implementation.

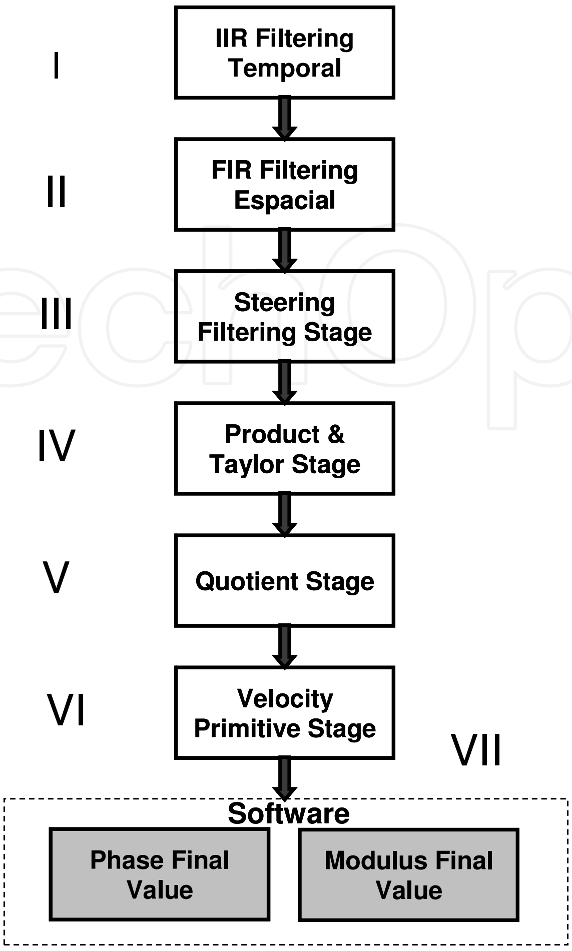

Fig. 8. General structure of the model implemented.

Stage I contains temporal differentiation through IIR filtering, being the output of this stage the three first derivatives of the input. Stage II performs the spatial differentiation building a pyramidal structure of each temporal derivative. Figure 9 represents what the authors (Botella *et al.*, 2010) call as "Convolutive Unit Cell" which implements the separable convolution organized in rows and columns. Each part of this cell will be replicated sufficiently to perform a pyramidal structure.

Stage III steers each one of space-time functions calculated previously. Stage IV performs a Taylor expansion and its derivatives over x, y and t delivering at the output a sextet which contains the product of those. Stage V forms a quotient of this sextet. Stage VI forms four different measurements corresponding to the direct and inverse speeds (8-11), which act as primitives for the final velocity estimation. Finally, Stage VII computes the modulus and phase values (12-13) on software.

Stage VI does not calculate the final velocity estimation due to the bioinspired nature of the model (combining multiple speed measurements, so if direct speed does not provide an accurate value, inverse speed will do that, and *vice versa*). However, every measurement of the speed has entity of velocity, so it could be used as final velocity estimation, even though it would degrade the robustness of the mode

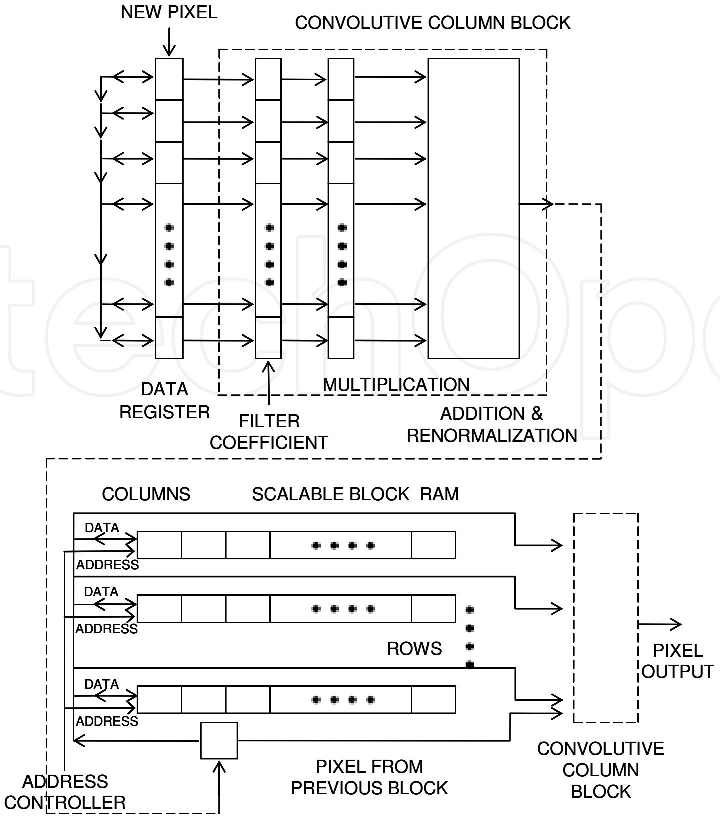

Fig. 9. Unit cell to perform the convolution operation.

3.2 Low and mid-level vision platform. Orthogonal variant models

One of the most well established approaches in computer-vision and image analysis is the use of Moment invariants. Moment invariants, surveyed extensively by Prokop and Reeves (Prokop & Reeves, 1992) and more recently by Flusser (Flusser, 2006), were first introduced to the pattern recognition community by Hu (Hu, 1961), who employed the results of the theory of algebraic invariants and derived a set of seven moment invariants (the well-known Hu invariant set). The Hu invariant set is now a classical reference in any work that makes use of moments. Since the introduction of the Hu invariant set, numerous works have been devoted to various improvements, generalizations and their application in different areas, e.g., various types of moments such as Zernike moments, pseudo-Zernike moments, rotational moments, and complex moments have been used to recognize image patterns in a number of applications (The & Chin, 1986).

The problem of the influence of discretization and noise on moment accuracy as object descriptors has been previously addressed. There are proposed several techniques to increase the accuracy and efficiency of moment descriptors. (Zhang, 2000), (Papakostas *et al.*, 2007, 2009). (Sookhanaphibarn & Lursinsap, 2006).

In short, moment invariants are measures of an image or signal that remain constant under some transformations, e.g., rotation, scaling, translation or illumination. Moments are applicable to different aspects of image processing, ranging from invariant pattern recognition and image encoding to pose estimation. Such Moments can produce image descriptors invariant under rotation, scale, translation, orientation, etc.

The implementation of these systems combines the low-level vision optical flow model (Botella *et al*., 20010) with the orthogonal variant moments (Martín.H *et al*., 2010) as Area, Length and Phase for the two cartesian components (A, L_X, L_Y, $P_{X,}$ P_Y) to built a real-time mid-level vision platform which is able to deliver output task as tracking, segmentation and so on (Botella *et al*., 2011).

The architecture of the system can be seen in the Figure 10. Several external banks have been used for different implementations, accessing to them from both the FPGA and the PCI bus. Low-level optical flow vision is designed and built through an asynchronous pipeline (micropipeline), where a token is passed to the next core each time one core finish its processing. The high level description tool Handel-C is chosen to implement this core with the DK environment. The board used is the well-known AlphaData RC1000 (Alphadata, 2007) which includes a Virtex 2000E-BG560 chip and 4 SRAM banks of 2MBytes each one as shown in the Figures 11a and 12a. Nevertheless Low-level moment vision platform is implemented in a parallel way, being independent the wielding of single one. Each orthogonal variant moment and the optical flow scheme advances the final Mid-Level Vision estimation. The Multimodal sensor core integrates information from different abstraction layers (six modules for optical flow, five modules for the orthogonal moments and one module for the Mid-Level vision tasks). Mid-Level vision core is arranged in this work for segmentation and tracking estimation with also an efficient implementation of the clustering algorithm, although additional functionality to this last module can be added using this general architecture.

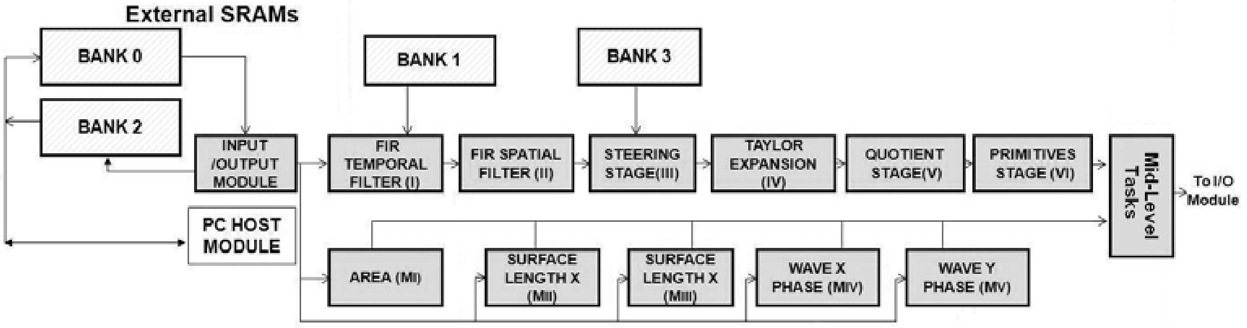

Fig. 10. Scheme of the real-time architecture of low-level and mid-level vision.

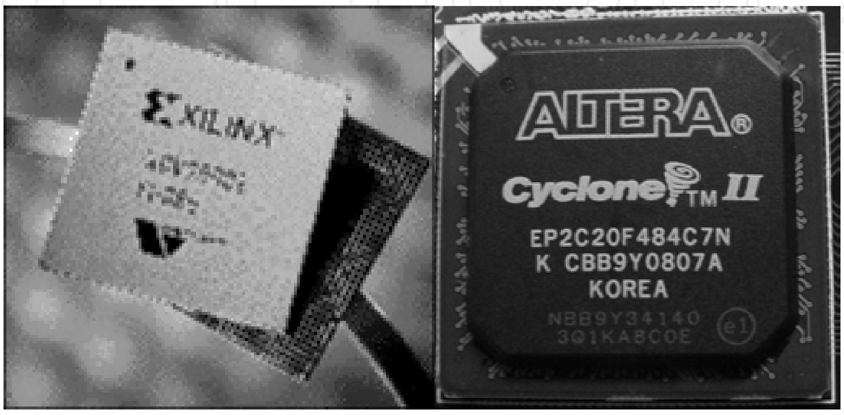

Fig. 11. (11a.left)Xilinx Virtex 2000E. (11b.right) Altera Cyclone II.

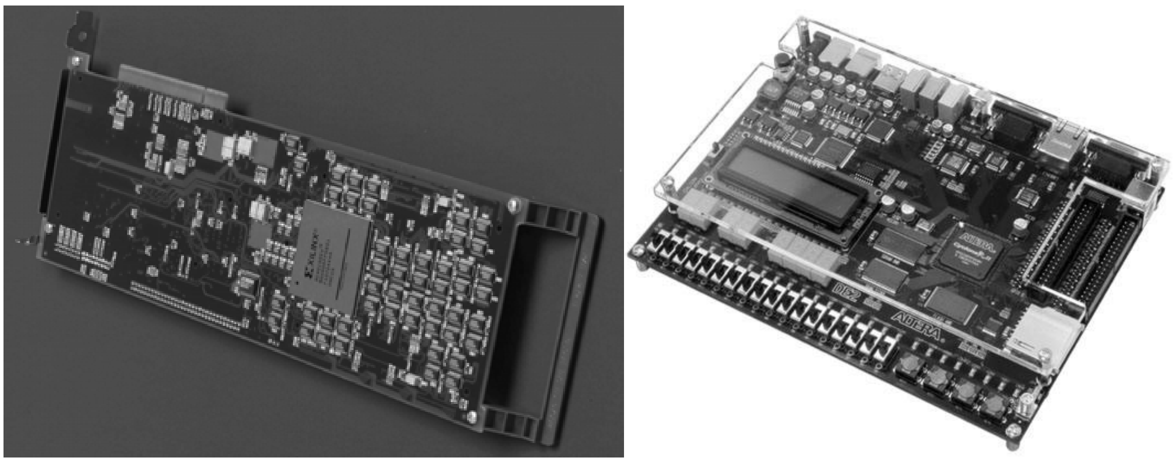

Fig. 12. (12a.left) ALPHADATA RC1000 branded by Celoxica. (12b.right) DE2 branded by Altera.

3.3 Block Matching models

The full search technique (FST) is the most straightforward Block Matching Method (BMM) and also the most accurate one. FST (Figure 13a) matches all possible blocks within a search window in the reference frame to find the block with the minimum Summation of absolute differences (SAD), which is defined as:

$$\mathrm{SAD}(x,y;u,v)=\sum_{x=0}^{31}\sum_{y=0}^{31}\left|I_t(x,y)-I_{t-1}(x+u,y+v)\right|, \tag{17}$$

Where It(x, y) represents the pixel value at the coordinate (x, y) in the frame t and the (u, v) is the displacement of the possible Macro Block (MB). For example, for a block with size 32×32, the FST algorithm requires 1024 subtractions and 1023 additions to calculate a SAD. The required number of checking blocks is $(1+2d)^2$ while the search window is limited within ± d pixels, currently is used for it a power of two. Three Steps Search Technique (TSST) is not an exhaustive search.

The step size of the search window is chosen to be half of the search area (Figure 13b). Nine candidate points, including the centre point and eight checking points on the boundary of the search are selected in each step. Second step, will move the search centre forwarding to the matching point with the minimum SAD of the previous step, and the step size of the second step is reduced by half. Last step stops the search process with the step size of one pixel; and the optimal MV with the minimum SAD can be obtained.

The hierarchical methods as Multi-scale Search Technique are based on building a pyramidal representation for each frame. This representation calculates an image where each pixel's value is a mean of itself with its neighbourhood; after that, the image is under sampled to the half, as shown in Figure 13c. This model is implemented using the Altera DE2 board and the Cyclone II FPGA as shown in the Figures 11b and 12b (González *et al.*, 2011), balancing the code between the microprocessor implemented and the acceleration system which uses Avalon Bus thanks to so-called "C to hardware compiler" from Altera (Altera, 2011).

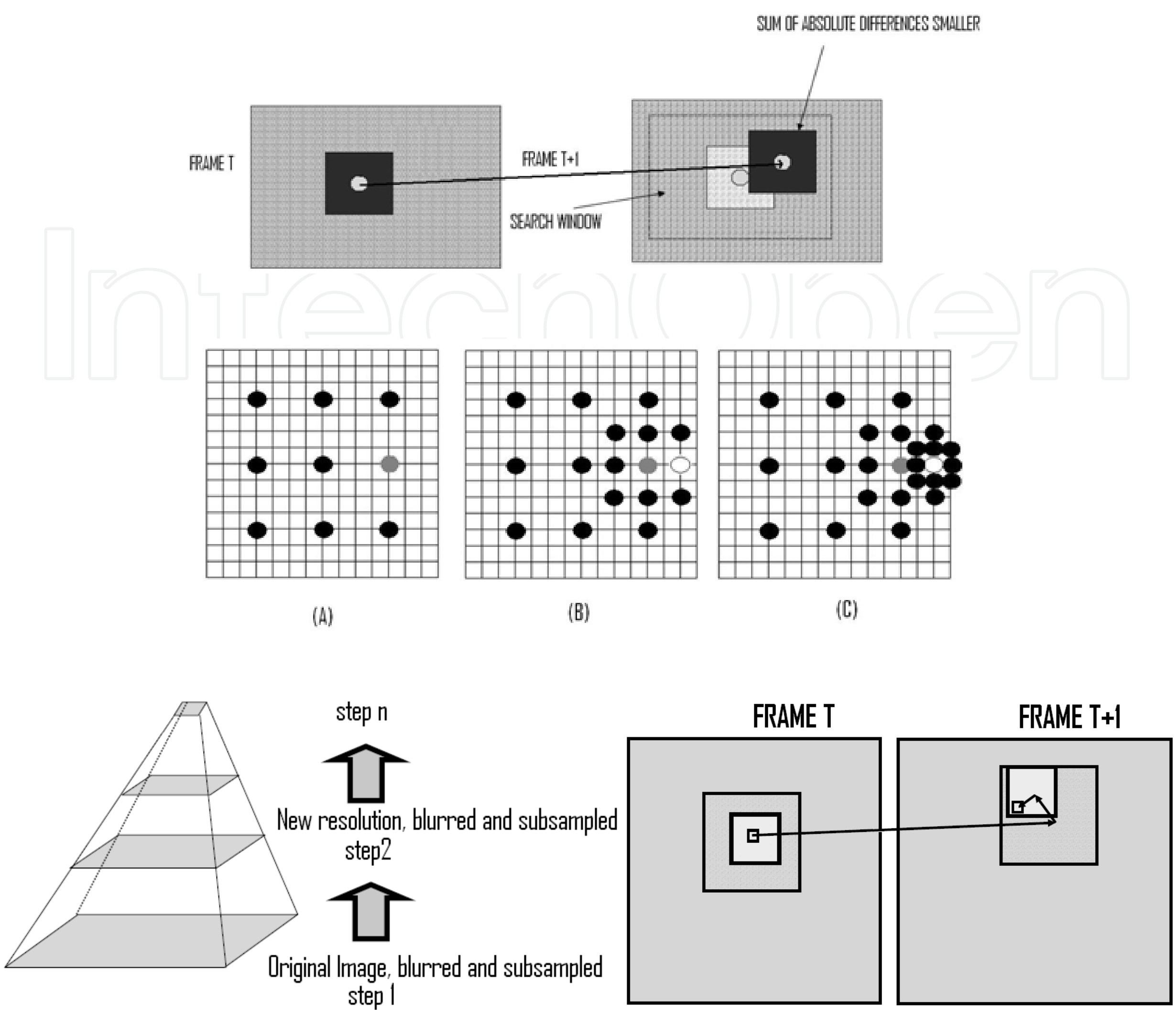
SUM OF ABSOLUTE DIFFERENCES SMALLER
FRAME T
FRAME T+1
SEARCH WINDOW
(A)
(B)
(C)
step n
New resolution, blurred and subsampled
step2
Original Image, blurred and subsampled
step 1
FRAME T
FRAME T+1

Fig. 13. (13a. Upper row) Full Search Technique. (13b. medium row) Three Steps Search Technique. (13c. lower row) Multiscale Search Technique.

4. Computational resources and throughput of the real-time systems

In this section, we analyse each of these real-time motion estimation systems regarding the computational resources needed and the throughput obtained. The resources used in the implementations described in 3.1 and 3.2 are shown in the Tables 1-3 as Slices and Block Ram percentage accounting. MC is the maximum delay in terms of the number of clock cycles needed for each modulus implemented. Finally, the Throughput provides the Kilo pixels per second (Kpps) of the implementations.

Table 1 shows a set of these parameters just for the McGM optical flow implementation, (see case 3.1) but Table 2 manages the multimodal sensor treated in 3.2. For this last case, has been separated the implementation of each one of the orthogonal variant moments (Table 2) and just the whole implementation (all modules working together as shown in Table 3).

Tables 4 is regarding the implementation explained in 3.3. The FPGA used is Altera Cyclone II EP2C35F672C6 with a core NIOS II processor embedded in a DE2 platform.

Low-level Vision Stage (Optical flow)	FIR Temporal Filtering(I)	FIR Spatial Filtering(II)	Steering (III)	Product & Taylor (IV)	Quotient (V)	Primitives (VI)
Slices (%)	190 (1%)	1307 (7%)	1206(6%)	3139(19%)	3646(20%)	2354(12%)
Block RAM (%)	1%	31%	2%	13%	16%	19%
MC	13	17	19	23	21	19
Throughput (Kpixels/s)/Freq.(Mhz)	4846/ 63	3235/55	2526/48	1782/41	1695/39	2000/38

Table 1. Slices, memory, number of cycles and performance for Low-level Optical flow scheme.

Low-level Vision Stage (Orthogonal Variant Moments)	Area (M_I)	L_X (M_{II})	L_Y (M_{III})	P_X (M_{IV})	P_Y (M_V)
Slices (%)	321(2%)	1245(7%)	1245(7%)	658(4%)	658(4%)
Block RAM (%)	1%	4%	4%	3%	3%
MC	7	11	11	5	5
Throughput (Kpixels/sec)/ Frecuency (MHz)	4546/49				

Table 2. Slices, memory, number of cycles and performance for Orthogonal Moment scheme.

COMPLETE Mid-level and Low level Vision	Motion Estimation (Low-Level)	Orthogonal Variant Moments (Low-Level)l	Tracking & Segmentation Unit (Mid-Level)	Multimodal Bioinspired Sensor. (Mid & Low Level)
Slices (%)	4127 (24%)	11842 (65%)	1304 (6%)	17710 (97%)
Block RAM (%)	15%	80%	4%	(99%)
MC (limiting)	29	11	18	29
Throughput (Kpixels/s)/Freq.(Mhz)	4546/49	2000/38	2000/38	2000/38

Table 3. Resources needed and performance for Low and Mid-Level vision. Multimodal system.

Method, Processor / Quality	Logic Cells	Embedded DSPs (9 × 9)	Total memory bits
FST, e / Quality I	4925 (15%)	0 (0%)	62720 (13%)
FST, e / Quality II	7708 (23%)	28 (40%)	
FST, e / Quality III	14267 (43%)	39 (56%)	
FST, s / Quality I	5808 (18%	0 (0%)	99200 (20%)
FST, s/ Quality II	8501 (26%)	20 (28%)	
FST, s / Quality III	15627 (47%)	32 (48%)	
FST, f / Quality I	6488 (20%)	0 (0%)	134656 (28%)
FST, f / Quality II	9225 (30%)	32 (48%)	
FST, f / Quality III	16396 (50%)	43 (61%)	

Table 4. Resources needed and performance for Low level vision Block matching system.

The parameters for measuring the resources for this technology are Logic Cell and Embedded DSPs. For this implementation, we have used different embedded NIOS II microprocessors (E, S, F from “economic”, “standard”, and “full” with different characteristics (González *et al.*, 2011) and for each one of these microprocessors we have applied different parts of the code, distinguishing between “Quality I, II or III” depending of the piece of the code running just into the microprocessor and the part that is accelerated using C2H compiler. (Altera, 2011).

For the multimodal system, it is possible reach up a real-time performance (see Section 3.2 and Table 5). This throughput is between 12 Mega pixels per seconds (Mpps) with basic quality (most of the source code running in Nios II microprocessor) and 30 Mpps (most of the source code being accelerated with C to hardware compiler) which means between 40 and 98 fps at 640x480 resolution.

COMPLETE Mid-level and Low-level Vision	Orthogonal Variant Moments (Low-Level)	Motion Estimation (Low-Level)	Multimodal Bioinspired Sensor. (Mid & Low-Level)
resolution 120x96	395 frames/sec	174 frames/sec	174 frames/sec
resolution 320x240	59 frames/sec	26 frames/sec	26 frames/sec
resolution 640x480	28 frames/sec	14 frames/sec	14 frames/sec
Throughput	4546 Kpixels/sec	2000 Kpixels/sec	2000 Kpixels/sec

Table 5. Throughput in terms of Kpps and fps for the embedded sensor.

COMPLETE Mid-level and Low-level Vision	Quality I	Quality II	Quality III
Throughput	12 Mpixels/sec	20 Mpixels/sec	30 Mpixels/sec

Table 6. Throughput in terms of Kpps and fps for the block matching technique.

5. Conclusion

This chapter has approached the problem of the real-time implementation of motion estimation in embedded systems. It has been descripted representative techniques and systems belonging to different families that contribute with solutions and approximations to a question that remains still open, due the ill-posed nature of the motion constraint equation.

It has been also proposed an overview of different implementations capable of computing real-time motion estimation in embedded systems delivering just low primitives: (gradient model and block matching respectively) and delivering the mid-level vision primitives (combination optical flow with orthogonal variant moments). In the Table 7 are shown the different methods implemented regarding the machine vision domain, the final performance obtained, the robustness implementation and the complexity of the final system.

These systems designed are scalable and modular, also being possible choice the visual primitives involved -number of moments- as well as the bit-width of the filters and computations in the low-level vision -optical flow-. This architecture can process concurrently different visual processing channels, so the system described opens the door to the implementation of complex bioinspired algorithms on-chip.

The implementation of these systems shown offers robustness and real-time performance to applications in which the luminance varies significantly and noisy environments, as industrial environments, sport complex, animal and robotic tracking among others.

Method Implemented	Vision Domain	Throughput FPS (320x200)	Robustness	Complexity of the Implementation (FPGA board)
Orthogonal Variant Moments	Low&Mid	59 fps	Low	Easy
Motion Estimation (McGM)	Low	26 fps	Very High	High
Multimodal Bioinspired Sensor.	Low&Mid	26 fps	Very High	Very High
Block Matching (Full Search Tecnique)	Low	120 fps	Medium	Easy

Table 7. Comparison of the different methods implemented.

6. Acknowledgment

This work is partially supported by Spanish Project project TIN 2008-00508/MIC. The authors want to thank to Prof. Alan Johnston and Dr. Jason Dale from University College London, (UK) for their great help with Multichannel Gradient Model (McGM), explained partially in this chapter.

7. References

A. Anderson, P. W. McOwan. (2003). Humans deceived by predatory stealth strategy camouflaging motion, *Proceedings of the Royal Society*, B 720, 18-20.

A. Johnston, C. W Clifford. (1995). Perceived Motion of Contrast modulated Gratings: Prediction of the McGM and the role of Full-Wave rectification, *Vision Research*, 35, pp. 1771-1783.

A. Johnston, C. W. Clifford. (1994). A Unified Account of Three Apparent Motion Illusions, Vision Research, vol. 35, pp. 1109-1123.

A. Johnston, P.W. McOwan, C.P. Benton. (2003). Biological computation of image motion from flows over boundaries, *Journal of Physiology*. Paris, vol. 97, pp. 325-334.

A. Mikami, W.T Newsome, R.H Wurtz. (1986). Motion selectivity in macaque visual cortex. I. Mechanisms of direction and speed selectivity in extrastriate area MT. *J. Neurophysiol*, vol. 55, pp. 1308-1327.

Accame M., De Natale F. G. B., Giusto D. (1998). High Performance Hierachical Block-based Motion Estimation for Real-Time Video Coding, *Real-Time Imaging*, 4, pp. 67-79.

Adelson E. H., Bergen J. R. (1985). Spatiotemporal Energy Models for the Perception of Motion. *Journal of the Optical Society of America A*, 2 (2), pp. 284-299.

Albright T. D. (1993). *Cortical Processing of Visual Motion*, In Miles & Wallman. Visual Motion and its Role in the Stabilisation of Gaze, pp. 177-201.

AlphaData RC1000 product (2006). , http://www.alpha-data.com/adc-rc1000.html.

Altera (2011). *C to Hardware.*

Anandan P., Bergen J. R., Hanna K. J., Hingorani R. (1993). *Hierarchical Model-based Motion Estimation*. In M.I.Sean and R.L.Lagendijk, editors, Motion Analysis and Image Sequence Processing. Kluwer Academic Publishers.

Arias-Estrada M., Tremblay M., Poussart D. (1996). A Focal Plane Architecture for Motion Computation, *Real-Time Imaging*, 2, pp. 351- 360.

Arnspang J. (1993). Motion Constraint Equations Based on Constant Image Irradiance, *Image and Vision Computing*, 11 (9), pp. 577-587.

Baker, S., Matthews, I. (2004). Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, Vol.56, Nº 3, pp. 221-255.

Beare R., Bouzerdoum A. (1999). Biologically inspired Local Motion Detector Architecture, *Journal of the Optical Society of America A*, 16 (9), pp. 2059-2068.

Bergen J. R., Burt P. J. (1992). A Three-Frame Algorithm for Estimating Two-Component Image Motion, *IEEE T. On Pattern Analysis and Machine Intellig.*, 14 (9), pp. 886-895.

C.-H. Teh, R.T. Chin. (1988). On image analysis by the methods of moments, *IEEE Trans. Pattern Anal. Mach. Intell*igence. vol.10 (4), pp. 496–513.

C.-Y. Wee, R. Paramesran, F. Takeda. (2004). New computational methods for full and subset zernike moments, *Inform. Sci*. vol. 159 (3–4), pp. 203–220.

Campani M., Verri A. (1992). Motion Analysis from First-Order Properties of Optical Flow. CVGIP: Image Understanding 56 (1), pp. 90-107.

Christmas W. J. (1998). Spatial Filtering Requirements for Gradient- Base Optical Flow Measurements, *Proceedings of the British Machine Vision Conference*, pp.185-194.

D. González, G. Botella, S. Mookherjee, U. Meyer-Baese. (2011). NIOS II processor-based acceleration of motion compensation techniques. *SPIE'11*,. Orlando, FL.

Defaux F., Moscheni F. (1995). Motion Estimation Techniques for Digital TV: A Review and a New Contribution, *Proceedings of the IEEE,* 83 (6), pp. 858-876.

Fleet D. J., Black M. J., Yacoob Y., Jepson A. D. (2000). Design and Use of Linear Models for Image Motion Analysis, *Int. Journal of Computer Vision*, 36 (3), pp. 171-193.

Fleet D. J., Jepson A. D. (1989). Hierarchical Construction of Orientation and Velocity Selective Filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 (3), pp. 315-325.

Franceschini N., Pichon J. M., Blanes C. (1992). From Insect Vision to Robot Vision, *Philosophical Transactions of the Royal Society of London B*, 337, pp. 283-294.

G. Botella, A. García, M. Rodríguez, E. Ros, U. Meyer-Baese, M.C. Molina. (2010). Robust Bioinspired Architecture for Optical Flow Computation, *IEEE Transactions on Very Large Scale Integration (VLSI) System*, 18(4), pp. 616-629.

G. Botella, J.A. Martín-H, M. Santos, U. Meyer-Baese. (2011). FPGA-Based Multimodal Embedded Sensor System Integrating Low- and Mid-Level Vision, *Sensors*, 11(8), pp. 8164-8179.

G. Botella, U. Meyer-Baese, A. García. (2009). Bioinspired Robust Optical Flow Processor System for VLSI Implementation, *IEEE Electronic Letters*, 45(25), pp. 1304-1306.

G.A. Papakostas, D.E. Koulouriotis. (2009). A unified methodology for the efficient computation of discrete orthogonal image moments, *Inform.Sci*. 176, pp. 3619–3633.

G.A. Papakostas, Y.S. Boutalis, D.A. Karras, B.G. Mertzios (2007). A new class of zernike moments for computer vision applications, *I. Science*, vol. 177 (13), pp. 2802–2819.

Ghosal S., Mehrotra R. (1997). Robust Optical Flow Estimation Using Semi-Invariant Local Features. Pattern Recognition, 30 (2), pp. 229- 237.

Giaccone P. R., Jones G. A. (1997). Feed-Forward Estimation of Optical Flow, *Proceedings of the IEE International Conference on Image Processing and its Applications*, pp. 204-208.

Giaccone P. R., Jones G. A. (1998). Segmentation of Global Motion using Temporal Probabilistic Classification, *Proc. of the British Machine Vision Conference*, pp. 619-628.

Gibson, J. J. (1950). Perception of the Visual World *(Houghton Mifflin, Boston)*.

Golland. P., Bruckstein A. M. (1997). Motion from Color, *Computer Vision and Image Understanding*, 68 (3), pp. 346-362.

Gupta N. C., Kanal L. N. (1995).3-D Motion estimation from Motion Field, *Artificial Intelligence*, 78, pp. 45-86.

Gupta N. C., Kanal L. N. (1997). Gradient Based Image Motion Estimation without Computing Gradients. International Journal of Computer Vision, 22 (1), pp. 81-101.

Heitz F., Bouthemy P. (1993). Multimodal Estimation of Discontinuous Optical Flow Using Markov Random Fields, *IEEE Transactions on Patterns Analysis and Machine Intelligence*, 15 (12), pp. 1217-1232.

Horn B. K. P., Schunck B. G. (1981). Determining Optical Flow. *Artificial Intelligence*, 17, pp. 185-203.

http://www.altera.com/devices/processor/nios2/tools/c2h/ni2-c2h.html.

Huang C. and Chen, Y. (1995). Motion Estimation Method Using a 3D Steerable Filter, *Image and Vision Computing,* 13(1), pp.21-32.

J. Díaz, E. Ros, F. Pelayo, E. M. Ortigosa, S. Mota. (2006). FPGA-based real-time optical-flow system, *IEEE Trans.Circuits and Systems for Video Technology*, 16(2), pp.274-279.

J. Díaz, E. Ros, R. Agís, J.L. Bernier. (2008). Superpipelined high-performance optical flow computation architecture., *Comp. Vision and Image Understanding* ,112, pp. 262–273.

J. Flusser, (2006). Moment invariants in image analysis, in: *Proceedings of the World Academy of Science, Engineering and Technology*, vol. 11, pp. 196–201.

J.J. Koenderink, A.J. Van Doorn. (1987). Representation of local geometry in the Visual System, *Biological Cybernetics*, 55, pp. 367-375.

Johnston A., McOwen P. W., Benton C. (1999). Robust Velocity Computation from a Biologically Motivated Model of Motion Perception, *Proceedings of the Royal Society of London B*, 266, pp. 509-518.

José Antonio Martin H., Matilde Santos, Javier de Lope. (2010). Orthogonal Variant Moments Features in Image Analysis, *Information Sciences*, vol. 180, pp. 846 – 860.

K. Sookhanaphibarn, C. Lursinsap. (2006). A new feature extractor invariant to intensity, rotation, and scaling of color images, *Inform. Sci.* vol. 176 (14), pp. 2097–2119.

L. Lagae, S. Raiguel, G.A Orban. (1993). Speed and direction selectivity of macaque Middle temporal neurones, *J. Neurophysiol*, vol. 69, pp. 19-39.

Liu H., Hong T., Herman M. (1997). A General Motion Model and Spatio-Temporal Filters for Computing Optical Flow, *Int.Journal of Computer Vision*, 22 (2), pp. 141-172.

Lucas B. D., Kanade T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pp. 674-679.

M. Tomasi, F. Barranco, M. Vanegas, J. Díaz, E. Ros. (2010). Fine grain pipeline architecture for high performance phase-based optical flow computation, *Journal of Systems Architecture*, vol. 56, pp. 577–587.

M.-K. Hu. (1961). Pattern recognition by moment invariants, in: *Proceedings of IRE (Correspondence) Trans. Inform. Theory*, vol. 49, pp. 14–28.

Marr, D. (1982) *Vision*. Edit. W.H.Freeman. (1982).

Martín J.L., Zuloaga A., Cuadrado C., Lázaro J., and Bidarte U. (2005), *Hardware implementation of optical flow constraint equation using FPGAs*, ComputerVision and Image Understanding, 98, no. 3, pp. 462-490.

Mitiche A. (1987). Experiments in Computing Optical Flow with the Gradient-Based, Multiconstraint Method, *Pattern Recognition*, 20(2), pp. 173-179.

Mitiche A., Bouthemy P. (1996). Computation and Analysis of Image Motion: A Synopsis of Current Problems and Methods, *International Journal of Computer Vision*, 19 (1), pp. 29-55.

Nagel H. (1983). Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences, *Computer Vision, Graphics and Image Processing*, 21, pp. 85-117.

Nakayama K. (1985). Biological Image Motion Processing: A Review. *Vision Research,* 25, pp. 625-660.

Niitsuma, H. and Maruyama T. (2004). Real-Time Detection of Moving Objects, *Lecture Notes in Computer Science*, FPL 2004, vol. 3203, pp. 1153-1157.

Nyquist. (2007). http://en.wikipedia.org/wiki/Nyquist%E2%80%

Oh H., Lee H. (2000). Block-matching algorithm based on an adaptive reduction of the search area for motion estimation, *Real-Time Imaging*, 6, pp.407-414 (2000)

Ong E. P., Spann M. (1999). Robust Optical Flow Computation Based on Least-Median-of-Squares Regression, *International Journal of Computer Vision,* 31(1), pp. 51-82.

Pajares, G. De la Cruz, Jesús. (2003), *Visión por Computador (imágenes digitales y aplicaciones)*. ISBN 84-7897-472-5. RA-MA.

R. F. Hess, R. J Snowden. (1992). Temporal frequency filters in the human peripheral visual field, *Vision Research*, vol. 32, pp. 61-72.

R.J. Prokop, A.P. Reeves. (1992). A survey of moment-based techniques for unoccluded object representation and recognition. *Models Image Process*, vol. 54, pp. 438-460.

Reinery L. (2000). Simulink-based Hw/Sw codesign of embedded neuron-fuzzy systems. *International Journal of Neural Systems*, Vol. 10, No. 3, pp. 211-226.

Rosin P.L. (1998). Thresholding for Change Detection, *Proceedings of the International Conference of Computer Vision*, pp. 274-279.

Simoncelli E. P., Adelson E. H., Heeger D. J. (1991). Probability Distributions of Optical Flow, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Simoncelli E. P., Heeger D. J. (1998). A Model of Neuronal Responses in Visual Area MT, *Vision Research*, 38 (5), pp. 743-761.

Smith S. M. (1995). ASSET-2: Real-Time Motion Segmentation and Object Tracking. Technical Report TR95SMS2b available from www.fmrib.ox.ac.uk/~steve.

Sobey P. and Srinivasan M. V. (1991). Measurement of Optical Flow by a Generalized Gradient Scheme, *Journal of the Optical Society of America A*, 8(9), pp. 1488-1498.

Uras S., Girosi F., Verri A., and Torre V. (1998). A Computational Approach to Motion Perception, *Biological Cybernetics*, 60, pp. 79-87.

V. Bruce, P.R. Green, M.A. Georgeson. (1996). *Visual Perception: Physiology, Psychology & Ecology*. third ed., Laurence Erlbaum Associates, Hove.

Van Hateren J. H., Ruderman D. L. (1998). Independent Component Analysis of Natural Image Sequences Yield Spatio-Temporal Filters Similar to Simple Cells in Primary Visual Cortex. *Proceedings of the Royal Society of London B 265*, pp. 2315- 2320.

Verri A., Poggio T. (1989). Motion Field and Optical Flow: Qualitative Properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 490-498.

Wallach H. (1976). *On Perceived Identity: 1. The Direction of Motion of Straight Lines*. In On Perception, New York: Quadrangle.

Weber K., Venkatesh S., Kieronska D. (1994). Insect Based Navigation and its Application to the Autonomous Control of Mobile Robots, *Proceedings of the International Conference on Automation, Robotics and Computer Vision.*

Y.N. Zhang, Y. Zhang, C.Y. Wen. (2000). A new focus measure method using moments, *Image Vision Comput*. vol. 18, 959–965.

Yacoob. Y. and Davis. L. S. (1999). Temporal Multi-scale Models for Flow and Acceleration. *International Journal of Computer Vision*, 32 (2), pp.1-17.

Yacoob. Y. and Davis. L. S. (1999). Temporal Multi-scale Models for Flow and Acceleration, *International Journal of Computer Vision*, 32 (2), pp.1-17.

Young R. A., Lesperance R. M. (1993). A Physiological Model of Motion Analysis for Machine Vision. *Technical Report GMR 7878*, Warren, MI: General Motors Res. Lab.

Zanker J. M. (1996). On the Elementary Mechanism Underlying Secondary Motion Processing, *Phil. Transactions of the Royal Society of London*, B 351, pp. 1725-1736.

Zhang J. Z., Jonathan Wu. Q. M. (2001). A Pyramid Approach to Motion Tracking, *Real-Time Imaging*, 7, pp. 529-544.

Zuloaga A., Bidarte U., Martin J., Ezquerra J. (1998). Optical Flow Estimator Using VHDL for Implementation in FPGA, *Proc. of the Int. Conf. on Image processing*, pp. 972-976.

REAL-TIME SYSTEMS, ARCHITECTURE, SCHEDULING, AND APPLICATION

Edited by Seyed Morteza Babamir

INTECH

Real-Time Systems, Architecture, Scheduling, and Application
Edited by Dr. Seyed Morteza Babamir

ISBN 978-953-51-0510-7
Hard cover, 334 pages
Publisher InTech
Published online 11, April, 2012
Published in print edition April, 2012

This book is a rich text for introducing diverse aspects of real-time systems including architecture, specification and verification, scheduling and real world applications. It is useful for advanced graduate students and researchers in a wide range of disciplines impacted by embedded computing and software. Since the book covers the most recent advances in real-time systems and communications networks, it serves as a vehicle for technology transition within the real-time systems community of systems architects, designers, technologists, and system analysts. Real-time applications are used in daily operations, such as engine and break mechanisms in cars, traffic light and air-traffic control and heart beat and blood pressure monitoring. This book includes 15 chapters arranged in 4 sections, Architecture (chapters 1-4), Specification and Verification (chapters 5-6), Scheduling (chapters 7-9) and Real word applications (chapters 10-15).

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821