RWTH Aachen University
Media Computing Group
Prof. Dr. Jan Borchers
Post-Desktop User Interfaces
WS 2005/06

Motion Estimation

Canan Bicer, Andreas Pancenko
Matrikelnr 233826, 225140

Advisor: Tico Ballagas

Contents

1 Introduction

Originally motion estimation were developed for video compression. Motion estimation examines the movement of objects in an image sequence to try to get vectors forming the estimated motion. To obtain compression the similarities between successive frames known as temporal redunancy, can be utilized. Compression is used to reduce the valume of data required to describe the sequence.
Today, motion estimation used for a sort of interactions. If a sequence of images is given we can ask:

- what the moving objects in the scene are,
- what sort of motion they are undergoing,
- where they will be in the future?

To answer these questions one must measure the motion. In motion estimation there are problems like ambiguity, efficiency (because of a lot of data) and complexity (some tasks involve complex motion). In this paper interaction techniques which uses motion estimation will be presented and explain the certain algorithms.

2 Motivation

2.1 Interaction Techniques

An interaction technique is a way to conduct an interactive task, e.g. choosing one of several objects shown on a display screen. It is defined in the binding, sequencing and functional levels, and is depend on using an amount of input and output devices or technologies.

Mobile devices serve optimally as input device for interactions by using the mobile keyboard. In addition, for the controlling of a cursor the joystick of the mobile phone can be used. Thus, mobile phones became meanwhile constant companions of human[1],[2]. They have the abilities of a small computer. Most of the new generation devices possess a joystick, a built-in camera, a memory of several megabytes and support different wireless transfer mode systems like bluetooth, built-in wireless-LAN-modules and GSM. You can interact with a mobile phone e.g. for large public displays. Public displays are big display systems which are advisible on public places most on situation with high number of foot passengiers and queuetime e.g. in metros or airports. An approach for mobile phones with built-in cameras is the interaction technique *Sweep*[4],[5]. By the Sweep technique, which uses the optical flow algorithm, images are received in fast displacements, where the built-in camera of the mobile phone is used. A processing program running in the device compares these pictures sequentially, computed from the differences the relative movement of the telephone and transmit to the display. Thus, as with an optical mouse, a cursor can be steered for a display (see figure). The Sweep function is activated as the joystick of the mobile phone into a certain direction is pressed and held. In order to bring the

a) 

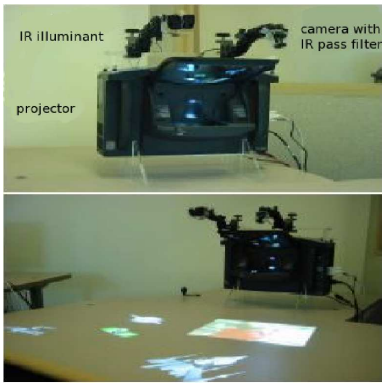
b) 

Figure 2.1: a)Sweep techniuqe, b)PlayAnywhere

arm into a new position, without moving the cursor simply the joystick is released briefly. Is thus only one hand necessary, the camera can into any direction be kept comfortable and the users can concentrate on the large display. There the movement computation directly in the mobile phone and not in the computer of the display takes place. A high number of simultaneous users is possilbe. The problem of this prototype is for the moment still the weak processor achievement of mobile phones, so that a delay of approximately 200ms arises to the effect of a movement on the display becomes visible. However, furture developments will provide here for the necessary achievement.

Playanywhere[6] which is developed in this year, is a projected computer vision-based system. With this device it is possible to project interactive images on any surfaces without any mounted cameras. It changes an ordinary tabletop into an interactive appearance. PlayAnywhere presents the objects into a single movable unit that does not require calibration. So it is possible to display and sense over a large surface area. For this manipulation, an optical flow algorithm is be used. The system consist of a computer, a projector, and an image-processing system that analyse the movement at the table (see figure). The system senses when a hand or a piece of papaer is placed at the table. The user are able to use their hands to move the virtual objects, or the playanywhere project an image or a video onto the paper. The aim of this system is to use alternative surface for input and output, thus no keyboard or mouse.

Other plays, which use motion estimation technique are for example the mobile game *Mozzies* or *Attack of the killers virus* [1]. These are the first plays based on using the camera on the mobile phone. Such games are called augmented reality games. Company Siemens has put on the market the mobile phone SX1. On this device Mozzies is preinstalled. Nokia has the similar Game Attack of the killers virus on its devices (e.g. Nokia 6600). The purpose of these games is to shoot the mosquitoes (in Siemens) or viruses (in Nokia). The games use real-time images from the mobile camera as a background. And the mosquitoes or viruses are introduced by the game. One should get the impression that one is surrounded by vermin or microbes, and destroy them. To shoot the mosquitos or viruses you must navigate the mobile phone with your hand. The mosquito hunter uses a crosshairs in the middle of the phone's display to target the insects before shooting them by pressing the appropriate button at the right moment. Players need a bit of space to play

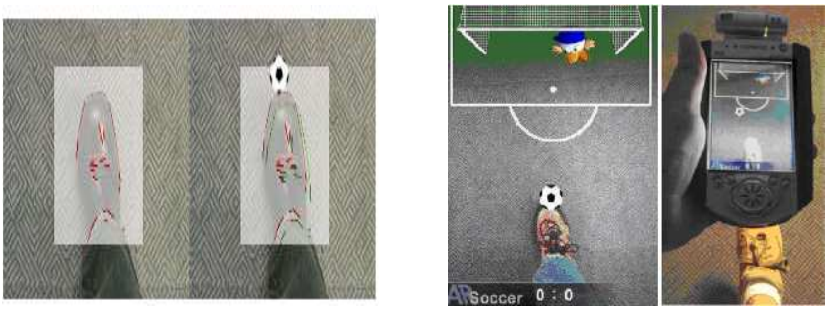


Figure 2.2: Soccer

the game - they sometimes have to turn around wildly in circles with arms outstretched in order to get all of the insects.

Another game called *AR-Soccer* was developed by the university Paderborn [9]. In this game the player has to kick a virtual ball (interaction object) with his real foot into a virtual goal. The game uses the camera of the mobile phone to determine the speed and position of the user's foot. The algorithm (Computer Vision algorithm CV) calculates the motion of a "kicking object" (e.g. user's foot) from the incoming video-stream and detects the collision between it and the interaction object. The results of the collision detection are used to calculate the new direction and speed of the interaction object, e.g. the ball. This game uses the simple and fast algorithm that combines 2D edge extraction and tracking and operates only in regions of interest (ROI) around the interaction objects.

The game *SymBall* is implemented for Symbian OS/Series 60 camera phones [8]. By SymBall the players use the camera phones as rackets, with the camera view to control the virtual racket's location. The users may play the game against a virtual wall, or against each other via Bluetooth connection. Two players are playing against each other using one object (e.g. a red vase) to control their virtual rackets, or the method detects the largest, userdefined color area from the camera view by region growing. Moving or tilting of the phone shows how the user's virtual racket in the display moves. Besides you can see the opponent's racket at the other side of the table and the virtual ball. The ball trajectory is calculated in both phones independently, based on the information received from the opponent. The display is moving with the mobile device and it is not very good for concentrating at the game.

3 High-Level Approaches

Motion estimation describes a lot of algorithms, which detect the motion in a consecutive image sequence. This chapter describes the most important algorithms.

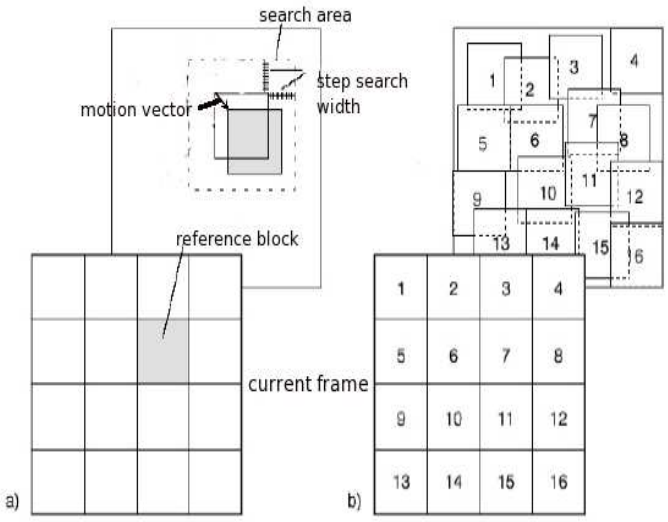


Figure 3.1: Block-matching algorithm

3.1 Block Matching

Block-Matching is frequently used method of the motion estimation. The basic principle of the Block-Matching lies in the pattern comparison. By a pattern you can understand generally the brightness (intensity) within a group of picture points. A group of picture points are square blocks. The movement of a current frame is determined from the previous frame. The image is divided into square blocks (these blocks are mostly 16x16 pixel big). Then will be defined a searching area in previous frame. The block of the actual frame goes through the searching area in a likewise defined searching step width as long as until this has found the greatest possible correspondence. The point of the actual block and the point of the place of the greatest possible correspondence in the searching area build the movement vector of the actual block. This vector is the movement of the looked block between the previous and the actual frame. The given searching step width influences the accuracy of the position of the estimate. The accordance or correspondence determine how similar the actual block is to a reference block from the searching area to which this is just compared. There are a lot of sensible criteria for this. Mainly used in the practical application are the SAD (Sum of the Absolute Differences), MAD (Median Absolute Differences), SSD (Sum of Squared Difference) and the MSE (Means Square Error) criterion.

3.1.1 The full search algorithm

The Full-Search (FS) algorithm is the simplest algorithm of the Block-Matching-Algorithm family [7]. It is simple to implement. But this algorithm is the most compute-intensive solution, because the whole searching area will be through searched, and with it is also very slowly. The advantage of the Full Search algorithm is, if it uses for example the SAD-Method for the similarity of the blocks, to find the certainly absolute minimum. This minimum points at the wanted reference block. The complexity of Full Search is O(p^2)

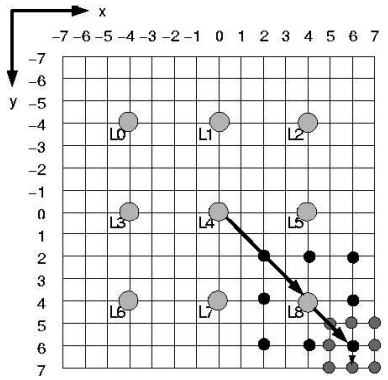


Figure 3.2: Three-Step-Search

and therefore the search area may be not too large. The Block Matching Algorithms can be optimized with reduction of intensity value. This happens as follows, the SAD of a e.g. 16 to 16 reference block will be not calculated for all 256 pixel, but only for e.g. 17 or 32. However, this leads to inaccuracy.

3.1.2 Three Step-Search algorithm

Another way to optimize the block matching is to reduce the search area. The Three-Step-Search algorithm is well-known and distributed too [7]. From the starting point (e.g. in the middle) with the start distance of the step search width the search area will be checked in horizontal, vertical and diagonal direction. The environment of the point with the best correspondence will be examined further. This runs like in the first step, but the step search width becomes smaller. The algorithm ends if the step search width arrive 0 (zero). The TSS is very fast but can not always find the absolute minimum.

3.2 Optical Flow

The definition of optical flow is the representation of the projections of 3D motion on a sequence of 2D images. Optical flow demonstrates visual variation of brightness patterns in sequenced images. By the computation of optical flow one understands a method, which the movement of pixels in a sequence of images are seized and pursued. As result one receives a vector field, which contains its direction of motion for each pixel in form of a vector. In the best case, the optical flow and the movement field of the objects have the same objective. However, there are differences between the terms to movement and optical flow. This is clarified by the following figure depicted above. Image (a) shows an evenly lit up rotary ball. With the fixed source of light there is no perceptive variation recogniziable in the picture itself. However, a rotation takes place, which produces a movement field. Image (b) shows a fixed ball, which is illuminated by a moved source of light. In this case, the movement field is equal to zero, which means that the ball is stationary. Through the changes of brightness in the image, however, an optical flow occurs. With this method one can recognize changes of position of objects and structures.

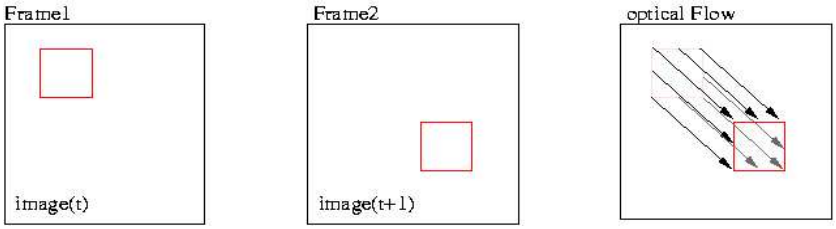


Figure 3.3: Two successive images and their optical flow

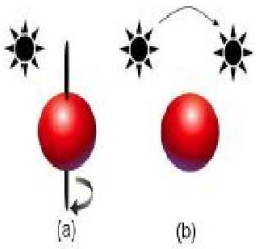


Figure 3.4: Difference between optical flow and motion field

To discuss the mathematical background, first, one regards a point in a 2D picture, which is at the time t at the pixel (x,y). This gets then in the next time step $(t+dt)$ a movement. This translational displacement around the vector (u,v) is to be computed (see figure). A sequence of images with an intensity value I(x,y,t) of the image point (x,y) at the time t are given. To describe the problem, the intensity values $(I(x,y,t))_{x,y,t}$ are adopted to be described by a differentiable function $I : \Omega \times [0,T] \rightarrow R$, where $\Omega \subset R^2$ is the image spatial domain and T is a strictly positve scalar. The partial derivatives from I in x-,y- or z-direction are described by the notation I_x, I_y, and I_z. For the computation of the optical flow vector one assumes for sufficient small time steps a point at the time $t + dt$ the same intensity possesses as at the time t. This leads to the following equation:
(1) $I(x,y,t) = I(x + dx, y + dy, t + dt)$. Using Taylor expansion and another transformation, one gets the optical flow constraint equation (OFCE): (2) $I_x u + I_y v + I_t = 0$. Here

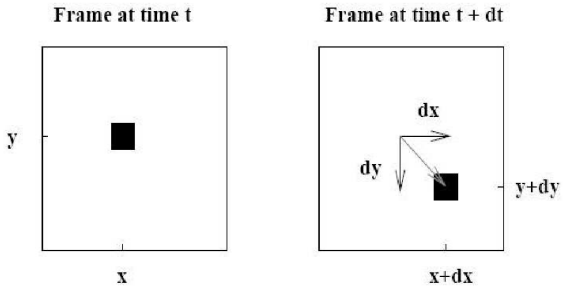


Figure 3.5: The optical flow: left: image at the time t with the point at the pixel (x,y), rechts: o,age at the point t+dt with the velocity vector (u,v)

$(u, v) = (\frac{dx}{dt}, \frac{dy}{dt})$ is called the optical flow vector. (2) represents the first condition for the computation of the optical flow. But these local information are, however, not completely enough to compute the optical flow (u,v) of one point. They only make statements of possible solutions. The optical flow approach produces efficient solutions for a variety of tasks. The optical flow algorithms are divided into *differential technique, energy-based methods, phase-based technique* and *region-based matching.*
Differential approaches, which estimate velocity vectors from spatial and temporal intensity derivations are the most common used techniques in the literature. The approaches are variations of how to minimize an equation that incorporates the gradient constraint equation. Energy-based methods use filters to produce energy estimates based on Fourier transforms. Phase based techniques consider velocity as defined according to the phase behaviour of bandpass filter outputs. Region based matching tries to detect velocities by comparing the position of a region in subsequent images. To read more about the last three techniques we refer to the paper of Baron et al [3].

3.2.1 Differential Technique

Above as introduced, the first constraint is not sufficient. To calculate the exact coordinates a further constraint is necessary. Besides, the intensity preservation, it is assumed that the movement of the pixels runs within the time step smoothly which means that the environment of a regarded point is differentiable, thus without any jump. From this assumption, the so-called smoothness constraint is introduced. It serves as regularizer, which supplements the missing information for the computation of the optical flow and leads with this way a better result. The probably most well-known approach comes from Horn and Schunck. It assumes that the neighbouring pixels are moving with the regarded point. Thus, the velocity of a point does not take place independent from its neighbours perfectly. This means that the neighboring object points have almost the same velocity. Therefore a certain smoothness of the optical flow in a sequence of pictures follows. Mathematically regarded, this means that the sum of the amount of the gradients have to become minimal. Equation (3) formulates the regularizer after Horn and Schunk: $E_r(u, v) := |\nabla_u|^2 + |\nabla_v|^2$. The evolving function which minimizes the absolute gradient of the velocity have to be solved iteratively. Because of the algorithm is iterative you must know when to stop the iteration. If the results at iteration t and t+1 are very similar you stop. This is when the algorithm converges.

4 Conclusion

On the optical flow problem has been much work done. So researchers have studied with different representations of the image sequence, different regularizations techniques and different confidence measures to offer a large amount of flow algorithms. To test each of the optical flow techniques real and synthetic image sequences were used. There are algorthims which provide near real-time frame rates with poorer accuracy while other algorithms offer more accuracy at a higher computational cost. The first-order techniques, local differntial method of Lucas and Kanade, and the local phase-base method of Fleet

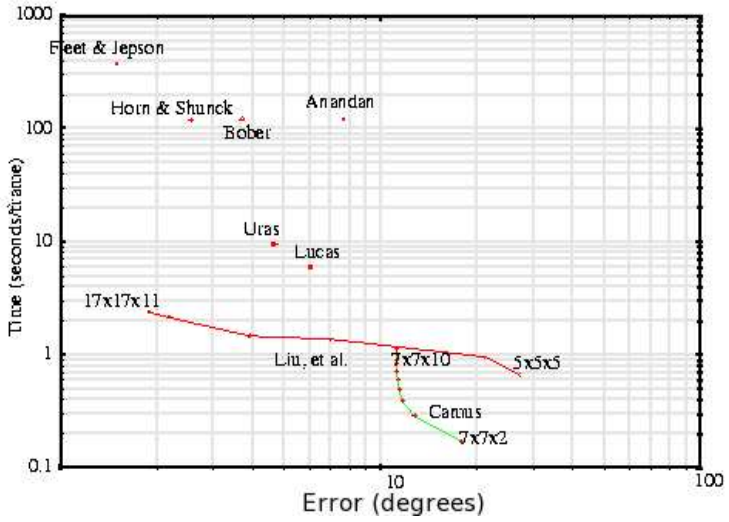


Figure 4.1: 2D performance diagram

and Jepson were the most reliable approaches. The local differential approaches, where v is computed explicitly in term of a locally constant or linear model were very accurate and computationally efficient to global smoothness constraints (used by Horn and Schunck). In the 2D performance diagram depicted below, the x axis uses accuracy (or error) and the y axis uses efficiency (or execution time). A point in the performance diagram corresponds to a certain parameter setting. The closer the performance point is to thr origin (low execution time and small error), the better the algorithm is. Thus, some algorithms (e.g. the phase-based method of Fleet & Jepson) may be very accurate but very slow while some algorithms may be very fast but nocht very accurate. The choice of flow algorithm when implementing a computer vision system is dependent on the applied research. With improvement in computing power and parallel processing, frame rates of these algorithms will continue to advance.

A good compelexity and efficiency trade-off offers the block-matching approach, which divides the image independently of contens into individual blocks. An advantage of the block-matching motion estimation is the simple estimation of the global camera movement from the developed movement vector field.

For small movements the optical flow approach performs better then the block-matching approach. The optical flow technique provides more accuracy then the block-matching technique, because it gives a measure of the motion at every point in the image. However, the two constraints are often violated in real images (e.g. variation of brightness). The iterative approach is computationally expensive, and is not well suited to real-time processing.

References

[1] Motion Detection as Interaction Technique for Games & Applications on Mobile Devices Stephan A. Drab, Nicole M. Artner (Upper Austria University of Applied Sciences, Austria). PERMID Workshop, Pervasive 2005.

[2] Rohs, M. Real-world Interaction with Camera-phones, In 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004), Tokyo, Japan, November 2004.

[3] Barron, J. L., Fleet, D. J., and Beauchemin, S. S. 1994. Performance of optical flow techniques. Int. J. Comput. Vision 12, 1 (Feb. 1994), 43-77. DOI= http://dx.doi.org/10.1007/BF01420984

[4] Rafael Ballagas, Michael Rohs, Jennifer Sheridan, and Jan Borchers. The Smart Phone: A Ubiquitous Input Device. To Appear in IEEE Pervasive Computing, 2005.

[5] Rafael Ballagas, Michael Rohs, and Jennifer Sheridan. Mobile Phones as Pointing Devices.

[6] PlayAnywhere: A Compact Interactive Tabletop Projection-Vision System (Andrew Wilson). Presented at UIST 2005

[7] Lai-Man Po, Wing-Chung Ma "A Novel Four Step Search Algorithm For Fast Block Motion Estimation" IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no.3, pp 313-7, June 1996.

[8] http://www.vtt.fi/multimedia/presents/SymBall-article.pdf

[9] http://wwwhni.uni-paderborn.de/publikationen/download