CSE 4084
Multimedia Systems
Project

COMPARING MOTION ESTIMATION ALGORITHMS

Aybüke ÖZKAN - 150115005

Celal BAYRAK - 150114044

Abstract

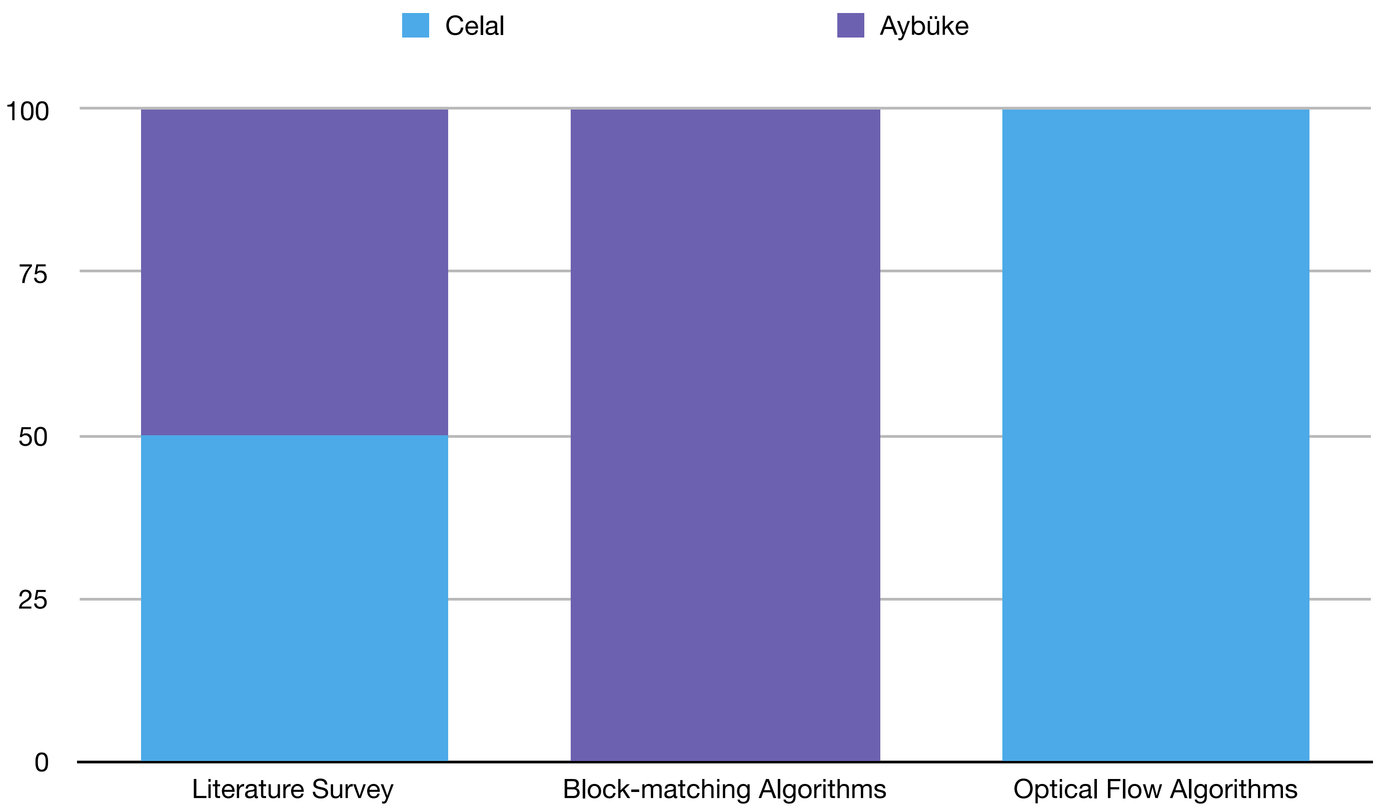
Video signal processing an important research area and has many applications in the industry and computer science such as online monitoring of assembly processes, robot navigation, medical treatment, multimedia broadcasting, remote sensing, and military [1]. Motion estimation is used for determining the motion vectors used to detect the transformation from one sequence to another. Motion estimation has many applications and has been proven essential for video coding and compression [2].

There are different approaches to motion estimation, and these can be categorized into region matching, gradient based methods and transform methods. The most commonly use motion estimation technique in video coding is the block matching algorithm due to its simplicity [3].

Each algorithm has a different approach to estimation of the motion, like gradient-based or block-based estimation, single-layer or multi-layer estimation. In our project, we compared performance of two different algorithms: Block Matching and Optical Flow. In block matching we used different searching algorithms to achieve the best performance. In Optical Flow we used two different approaches, which are named: Horn-Schunck and Lucas-Kanade. We compared their performance by using PSNR value.

Overview

- *Literature survey about motion estimation* **(November 22, 2019)**: We looked for the previous researches and papers that is related to different methods for motion estimation.
- *Searching for different algorithms* **(November 22, 2019)**: We searched about the different algorithms and selected Three-step Search, Exhaustive Search, Horn-Schunck method, Lucas-Kanade method, and Farneback method.
- *Preparing the midterm report* **(November 24, 2019)**
- *Implementation and training of different algorithms* **(January 1, 2020)**: We implemented the algorithms mentioned about. We compared their performances by using their PSNR values.
- *Final presentation* **(January 2, 2020)**
- *Preparing the final report* **(January 6, 2020)**



Accomplishments

<u>Motion Estimation</u>

Motion, in physics, is the positional change of an entity with respect to time in reference with some position which is pre-recognized and assumed invariant. Motion is usually specified in vector form i.e. with magnitude and direction called the motion vector.

Motion estimation can be described as any method for obtaining the motion vectors that specifies the transformation of a two dimensional image frame into the next frame in an image sequence [4].

Motion estimation algorithms can be classified into time-domain and frequency-domain algorithms (Fig. 1).

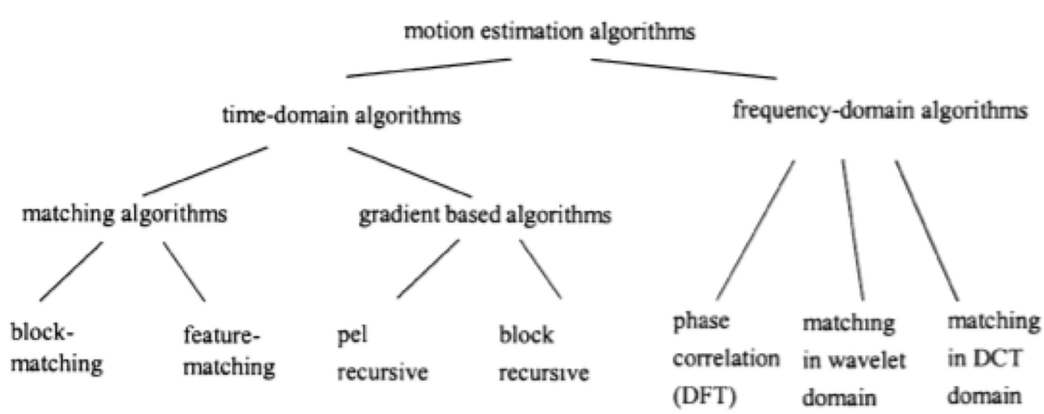


Fig. 1: Classification of motion estimation algorithms [5]

The time-domain algorithms comprise matching algorithms and recursive (gradient based) algorithms, whereas the frequency-domain algorithms comprise phase correlation algorithms [5].

Also the motion estimation methods can be broadly classified into pixel based methods (direct methods) as well as feature based methods (indirect methods). These methods are further subdivided as shown in the Fig. 2.

Direct Methods	**Indirect methods**
Block-matching algorithm	Colour tone based
Phase correlation and frequency-domain methods	Corner detection based
Pixel recursive algorithms	
Optical flow based	

Fig. 2: Classification of motion estimation algorithms [4]

In our study we focused on Block Matching and Optical Flow algorithms both of which are pixel based or direct motion estimation techniques.

1. Block Matching Algorithm

The key idea behind BM algorithm is the division of current working frame into a matrix of macro blocks and the comparison of each macro block with the corresponding block and its adjacent neighbors in the previous frame. This results in the generation of a set of motion vectors that specify each macro block's movement from its position in previous frame. For getting a precise macro block match, the search area should be extended up to p pixels on all fours sides of the corresponding macro block in previous frame. The constraint variable ‘p’ is called as the search parameter.

A cost function is employed for matching one macro block with another. The output of this cost function gives a numerical indication for the amount of mismatch (or similarity in some other cases) between the macro blocks that are compared. The macro block which results in the least cost is the one that matches closely with the current block. A large number of cost functions are available, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) (Fig.3).

$$MAD = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

Fig. 3: MAD formula

Peak Signal to Noise Ratio (PSNR) is a cost function that characterizes the motion compensated image which is created from the motion vectors using the macro blocks of the reference frame(Fig. 4) [4].

$$PSNR = 10 \log_{10} \left[\frac{(peak\ to\ peak\ value\ of\ original\ data)^2}{MSE} \right]$$

Fig. 4: PSNR formula

The motion vector is selected within a suitable set of candidate vectors. This set is called search window and represents the area where the most similar block will be searched. In the most common case, the search window corresponds to a rectangular area centered in the block B of the reference image.

The structure of the search windows has a huge impact on both the complexity of the motion estimation algorithm and on its precision. Therefore the choices of the search window and of the associated search strategy are critical for a motion estimation algorithm. There are several search strategies:

1.1) Three Step Search

This algorithm calculates the cost function at each possible location in the search window. This leads to the best possible match of the macro-block in the reference frame with a block in another frame. The resulting motion compensated image has highest PSNR ratio as compared to any other block matching algorithm. However this is the most computationally expensive block matching algorithm among all. A larger search window requires greater number of computations.

Algorithm runs as follows:

1. Start with search location at center
2. Set step size S = 4 and search parameter p = 7
3. Search 8 locations +/- S pixels around location (0,0) and the location (0,0)
4. Pick among the 9 locations searched, the one with minimum cost function
5. Set the new search origin to the above picked location
6. Set the new step size as S = S/2
7. Repeat the search procedure until S = 1

The resulting location for S=1 is the one with minimum cost function and the macro block at this location is the best match.

There is a reduction in computation by a factor of 9 in this algorithm. For p=7, while ES evaluates cost for 225 macro-blocks, TSS evaluates only for 25 macro blocks. [7]

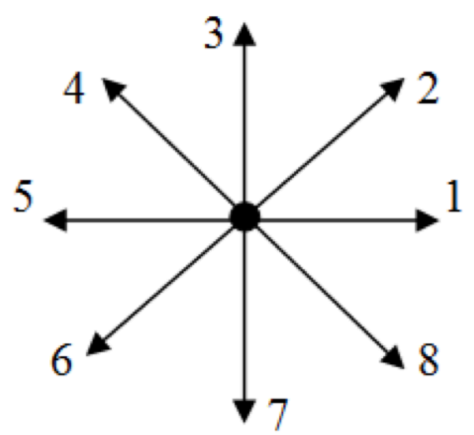


Fig.5: Search direction in a search pattern of the TSS [8].

1.2) Exhaustive Search

This algorithm calculates the cost function at each possible location in the search window. This leads to the best possible match of the macro-block in the reference frame with a block in another frame.

The resulting motion compensated image has highest PSNR ratio as compared to any other block matching algorithm. However this is the most computationally expensive block matching algorithm among all. A larger search window requires greater number of computations.

2. Optical Flow Algorithm

The idea of optical flow was first introduced by the American psychologist James J. Gibson in the early 90's for describing the visual stimulus in moving animals. When an object moves in 3D, it induces a 2D motion in the imaging plane. This 2D motion is called the Optical Flow. The OF can also be used to compute 3D motion i.e., translation and rotation in 3D.

Today, OF estimation is one of the key problems in computer vision application. It is used to estimate the displacement field between two images and is applied to identify the similarity between pixels. These types of problems are not restricted only for motion estimation but they are also present in 3D reconstruction as well as image registration [4].

There are two classical optical flow algorithms: Horn-Schunck and Lucas-Kanade.

2.1) Horn-Schunck

The method uses a minimization of a cost function, which combines the optimization of the optical flow constraint with a constraint on the smoothness of the motion vector field. The cost function is defined as follow:

$$J_{HS}(V) = \iint_{\Re} \left[u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y} + \frac{\partial f}{\partial t} \right]^2 dxdy + \lambda \iint_{\Re} \left[\|\nabla u\|^2 + \|\nabla v\|^2 \right] dxdy$$

The first term in the above expression is a mean square error on the motion constraint, while the second one is a regularization term: it allows ensuring that the gradient of the motion vector field takes small values ("smoothness" of the solution). In this criterion, is the regularization constant, which allows to trade off the influence of the regularization term and the minimization of the motion constraint. The integrals are on an arbitrary region where the motion is homogeneous, meaning that this technique can be adapted to region-based motion estimation or to a joint segmentation-motion estimation solution.

After some mathematical developments involving the minimization of the cost function, we arrive at the following solution for the two components of the motion vector field:

$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator. The following observations can be made concerning this method:

This global optimization is quite complex, involving the resolution of a system of partial differential equations.

The choice of is critical. On the one hand, it leads to a different smoothness of the field. On the other hand, it influences the numerical stability of the system.[6]

2.2) Lucas-Kanade

The Lucas-Kanade method is another classical approach for optical flow estimation. In this method, the assumption is made that the optical flow is approximately constant in a small neighborhood, instead of adding a smoothness constraint as in the Horn-Schunck algorithm.

We also used LKDoG(**Lucas-Kanade Derivative of Gaussian**) method. It uses a Gaussian filter and the derivative of a Gaussian filter to compute the optical flow.

2.3) Farneback

The Farneback algorithm generates an image pyramid, where each level has a lower resolution compared to the previous level. When you select a pyramid level greater than 1, the algorithm can track the points at multiple levels of resolution, starting at the lowest level.

Increasing the number of pyramid levels enables the algorithm to handle larger displacements of points between frames. However, the number of computations also increases [9].

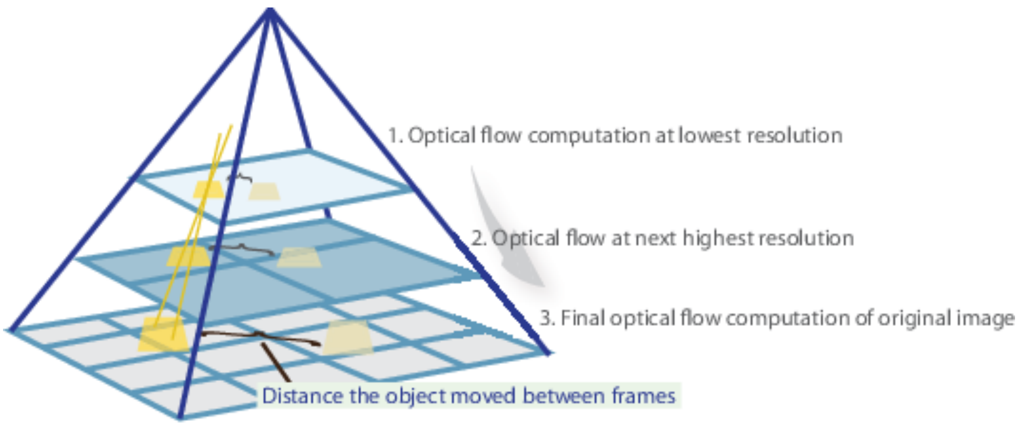


Fig.6: An image pyramid with three levels [9].

Motion Compensation

Motion compensation is an algorithmic technique used to predict a frame in a video, given the previous or future frames by accounting for motion of the camera and objects in the video.

Using motion compensation, a video stream will contain some reference frames; the only information stored for the frames in between would be the information needed to transform the previous frame into the next frame.

Implementation of the Algorithms

Original images

*Fig.7: Original images at time **t** and **t+1***

Block Matching Algorithms

Exhaustive Search Results

The first algorithm we have used in block matching was exhaustive search. In Fig.8 we determined the block size as 16. The blue arrows represents the motion vectors. The number of motion vectors is relatively less because block size is relatively high. Fig. 9 represents the motion compensated image of exhaustive search when block size is 16. The PSNR value between original frame and motion compensated image is 78.8578. It is relatively high but exhaustive search requires more computations. There is a tradeoff between accuracy and computations. The number of average computations for each macro block in exhaustive search when block size is 16 is 204,2828. It is also relatively high.

Fig.8: Block-size=16, Exhaustive Search, Motion Vectors

Fig.9: Block-size=16, Exhaustive Search, Estimation

When we decrease the block size to 8, the number of motion vectors increased too much. It means more computations and more accurate estimation (motion compensated image). Fig.10 represents the motion vectors of exhaustive search, when block size is 8. Fig.11 represents the motion compensated image. The PSNR value between original image and estimated image is 81,2345. The number of average computations for each macro block in exhaustive search when block size is 8 is 214,5177. These are the highest values of our experiments.

Fig.10: Block-size=8, Exhaustive Search, Motion Vectors

Fig.11: Block-size=8, Exhaustive Search, Estimation

Three Step Search Results

The second algorithm we have used in block matching is three step search. Three step search requires lower computations than exhaustive search. But exhaustive search's estimations are more accurate than three step search. Fig.12 represents the motion vectors of three step search algorithm when block size is 16. Fig.13 represents the estimation of three step search algorithm when block size is 16. The PSNR value between original image and estimated image is 78,7232. The number of average computations for each macro block in three step search when block size is 16 is 16,7045. It is very low compared to exhaustive search.

Fig.12: Block-size=16, Three Step Search, Motion Vectors

Fig.13: Block-size=16, Three Step Search, Estimation

When we decrease the block size to 8 in three step search, estimation is more accurate compared to block size 16, but number of computations are higher. Fig.14 represents the estimated image. The PSNR value between original image and estimated image is 80,8816. The number of average computations for each macro block in three step search when block size is 8 is 18,3251. Both of these values are higher compared to block size 16.

Fig.14: Block-size=8, Three Step Search, Estimation

Optical Flow Algorithms

The optical flow algorithms we have used for motion estimation are: Horn-Schunck Method, Lucas-Kanade Method, Farneback Method and Lucas-Kanade Derivative of Gaussian Method. But motion compensation of these algorithms are different from block matching. The optical flows are represented by complex numbers. So when we shift the pixels by these complex numbers the coordinates of blocks will not be integers. It means the blocks will not match the coordinates which they must be. So we have decided to round these complex numbers and we have used interpolation method to determine the pixel intensities of estimated image.

Horn-Schunck Method Results

Fig.15 represents the optical flows of Horn-Schunck Method and Fig.16 represents the estimation of Horn-Schunck Method. The PSNR value between estimated image and original image is 72,0839.

Fig.15: Optical Flow, Horn-Schunck, Motion Vectors

Fig.16: Optical Flow, Horn-Schunck, Estimation

Lucas-Kanade Method Results

Fig.17 shows the optical flows of Lucas-Kanade Method and Fig.18 shows the estimation of Lucas-Kanade Derivative of Gaussian Method. The PSNR value between estimation of Lucas-Kanade and original image is 72,0839. The PSNR value between estimation of Lucas-Kanade Derivative of Gaussian and original image is 73,5683.

Fig.17: Optical Flow, Lucas-Kanade, Estimation

Fig.18: Optical Flow, Lucas-Kanade DoG, Estimation

Farneback Method Results

Fig.19 represents the optical flows of Farneback Method. The PSNR value between estimation of Farneback Method and original image is 71,9988.

Fig.19: Optical Flow, Farneback, Motion Vectors

PSNR Values

Table 1. PSNR values of BM algorithms

	PSNR	Computations
Exhaustive Search Block-size=8	81.2345	214.5177
Three Step Search Block-size=8	80.8816	18.3251
Exhaustive Search Block-size=16	78.8578	204.2828
Three Step Search Block-size=16	78.7232	16.7045

Table 2. PSNR values of OF algorithms

	PSNR
Lucas-Kanade	72.0839
Lucas-Kanade DoG	73.5683
Horn-Schunck	72.2314
Farneback	71.9988

Summary

As a result of our experiments the exhaustive search gives the best result when block size is 8. But as we mentioned, it requires more computations compared to the other algorithms. The block matching algorithms generally give better results compared to optical flow algorithms. But interpolation and rounding processes have affected the results of optical flow algorithms.

References

[1] Yaakob, R., Aryanfar, A., Halin, A. A., Sulaiman, N., A comparison of different block matching algorithms for motion estimation. Procedia Technology, 2013. 11: p. 199-205.

[2] Dufaux, F. and J. Konrad, Efficient, robust, and fast global motion estimation for video coding. Image Processing, IEEE Transactions on, 2000. 9(3): p. 497-501.

[3] Dufaux, F. and F. Moscheni, Motion estimation techniques for digital TV: A review and a new contribution. Proceedings of the IEEE, 1995. 83(6): p. 858-876.

[4] Philip, J.T., Samuvel, B., Pradeesh, K., Nimmi, N.K., A comparative study of block matching and optical flow motion estimation algorithms. Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), 2014 Annual International Conference on. :1-6, Jul, 2014. doi: 10.1109/AICERA.2014.6908204

[5] Kuhn, P., Motion estimation algorithms. Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation: pp 17-60. https://link.springer.com/chapter/10.1007/978-1-4757-4474-3_2

[6] Pesquet-Popescu, B., Cagnazzo, M., Dufaux, F., Motion Estimation Techniques, TELECOM ParisTech: https://pdfs.semanticscholar.org/98ca/f8325abb40aa6bbddd0e7f5d3a6c366d03e6.pdf

[7] Block-matching Algorithm: https://en.wikipedia.org/wiki/Block-matching_algorithm

[8] Bhavsar, D.D., Gonawala, R.N, Three step search method for block matching algorithm, Proceedings of the IRF International Conference, 13th April-2014, Pune, India.

[9] MathWorks, Documentation. [Online]. Available: https://www.mathworks.com/help/vision/ref/opticalflowfarneback.html (Date of Access: 02 / 01 /2020)