ECE 6258 Fall 2014

Digital Image Processing

Project Report

Joint Framework for Motion Validity and Estimation using Block Overlap

Submitted by:

Ashwin A Shenoi (GTID 902919640)

Guru Das Srinagesh (GTID 902878096)

Table of contents

1 ABSTRACT

Motion estimation is the process of estimating the motion vectors which transform one image to another. This problem is inherently an ill-posed problem. The ill-posedness is because the motion is being estimated on a 2D image which is a projection of a 3D scene. Also factors such as untextured regions, small structures, occlusions and other types of complex motions contribute to the ill-posedness. This means that there is a need for a validity metric to evaluate the optimality of motion vectors (MV).

The aim of this project is to understand and reproduce the work of Santoro et al, in which a new framework for motion vector validity and estimation based on block-overlap is introduced.

2 INTRODUCTION

2.1 Motion Estimation

The very nature of the motion estimation problem is such that there are bound to be inconsistencies in the motion vectors obtained. Since we are projecting a 3D scene onto two dimensions, there is ample room for error. In addition, the presence of complex motion, occlusions, etc. further complicate the problem. The motion estimation problem is generally solved using either the computationally intensive but state-of-the-art optical flow algorithms or block matching based algorithm which have have low computational requirement. For practical and real time applications the block matching based methods are used as they give a flexible trade-off between complexity and MV quality. Because of the ill posedness of the problem and the non-convex nature of the problem, there is a need for a validity metric to evaluate the possible set of solution for its optimality.

2.2 Regularization and basic refinement

Having chosen to employ the block matching algorithm, the authors define addendums to regularize and refine its results:

- Because the block matching algorithm is sensitive to block size, there needs to be a careful choice of the block size employed. In order to mitigate this, a hierarchical block matching framework is used. This is a well-known technique in which the search for motion vectors is first conducted on a downsampled version of the image using a large block size, obtaining an initial rough estimate. Then the block size is reduced as the search continues on higher resolution versions of the image.

- The Sum of Absolute Differences (SAD) metric is used for determining matching blocks. Since this problem is non-convex, this gives rise to multiple local minima. This is the main motivation behind the introduction of the novel block-matching algorithm.
- In order to smoothen the motion field, the authors seek to penalize deviations amongst MVs while retaining the overall naturally-obtained motion vector field.

The main contribution of this paper is the development of a validity metric and algorithm that improves the quality of the motion field.

3 THE BLOCK OVERLAP BASED METRIC

3.1 Energy Minimization

In the block matching based algorithm, a smoothness of MV's constraint is imposed. This makes the problem an energy minimization problem. This function is not convex in nature. This paper essentially tries to address this problem by introducing a block overlap based regularizing term in the energy minimization framework. The term for energy minimization term used for images I_0 and I_1 is as follows (Eqn. 12 from the paper):

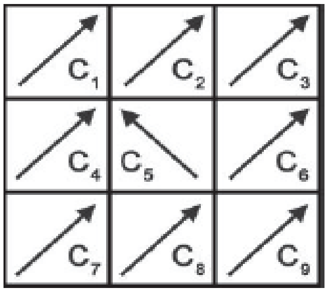
$$\mathrm{E} = \min_i \left\{ D(I_0, I_1, v_i) + \lambda R(v_i) \right\}$$

Here, λ is the Lagrange multiplier, which is used to weight the regularization term over the data term. The data term, in turn, is defined as follows (Eqn. 13 from the paper):

$$D(I_0, I_1, v_i) = \sum_{x \in B} |I_0(x) - I_1(x + v_i)|$$

The Regularization term is defined as (Eqn. 14 from the paper) :

$$R(v_i) = \sum_{j \in C^S} \|v_i - v_j\|_1$$



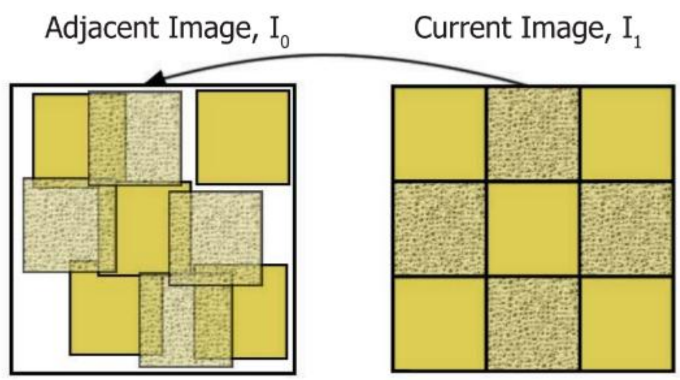
where C^S is the candidate set of 8 motion vectors v_j surrounding the motion vector of interest v_i and $i \neq j$.

Both the above terms are non-convex in nature. The paper introduces a new block overlap based term into this equation in this in order to choose from the solution set arising from the non-convex problem.

3.2 Block Overlap Minimization framework:

The motivation behind the introduction of a overlap based metric is based on the assumption that given two images for motion estimation, the MVs should map each pixel to a unique pixel in the other image. Based on this assumption, MVs which lead to the minimum overlap is the best for images.

The algorithm used for calculating overlap is as follows:



Calculate Volume for Each L_x (y)

For all blocks in I_1 , set $\mathbf{y} = \mathbf{x} + \mathbf{v}$

for $k = y_i$, $k < y_i + B_S$, k++ do

for $l = y_j$, $l < y_j + B_S$, l++ do

L_x (k,l) += 1

end for

end for

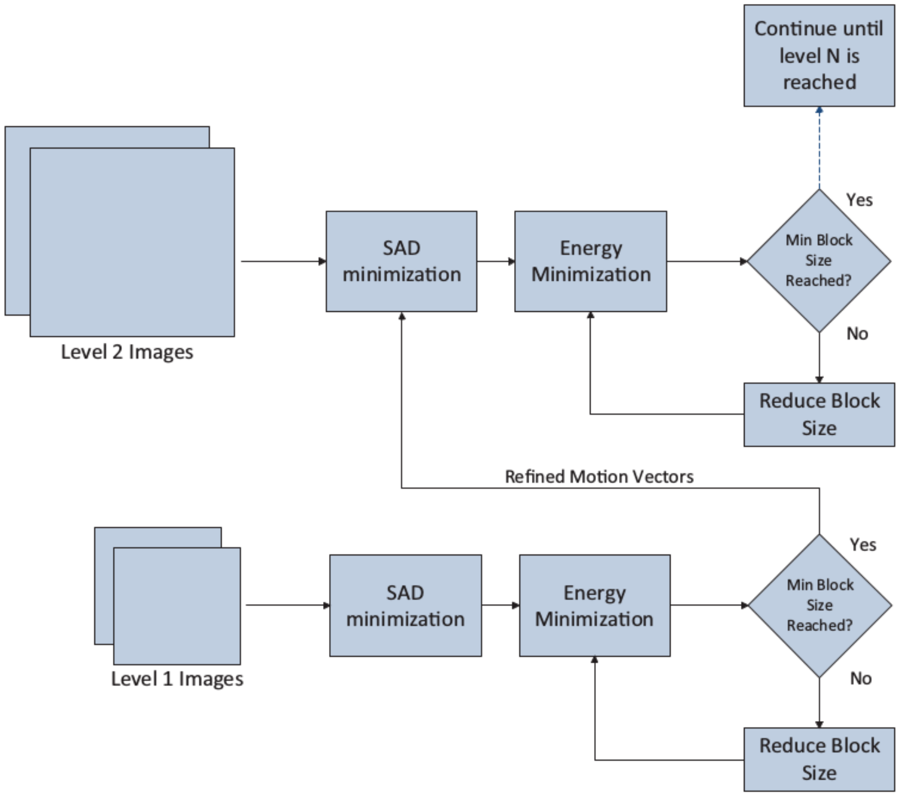
Here, the **L_x (y)** term is the overlap at position y mapped from position x. This term is now integrated with the energy term (Eqn. 19 from the paper):

$$E = \min_i \left\{ \left(\sum_{x \in B} |I_0(x) - I_1(x + v_i)| + 1 \right) \times \left(\frac{L_x(x + v_i)}{B_S^2} + 1 \right) + \lambda \sum_{j \in C^S} \|v_i - v_j\|_1 \right\}$$

Minimization of the above term gives not only a set of motion vectors which are smooth but also the ones which have a minimum overlap.

4 THE MOTION ESTIMATION ALGORITHM

A hierarchical algorithm is used for estimating the vector. The algorithm is as shown in the following diagram.



The algorithm works as follows. First, downsample the image N times and perform SAD minimization to obtain initial estimate of motion vectors. Then perform energy minimization using smaller and smaller block sizes to increase accuracy. Next, pass these initial estimates of the motion vectors to the (N-1) times downsampled image and continue the process till the image in its original resolution is reached.

In this paper, the results are published for N=4. Also the image is pre-processed by upscaling it four times before feeding it to the algorithm, which causes the estimated motion vectors to be of quarter pixel accuracy.

5 EVALUATION

An endpoint error metric is used for evaluating the error. The Endpoint error is the L_2 norm of the difference between the calculated motion vectors and the ground truth motion vectors. It is calculated as follows.

$$EE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2}$$

The dataset used for evaluating the performance of the algorithm is the one hosted on the Middlebury website (http://vision.middlebury.edu/flow/data/)

A few sample images from the dataset are shown below:

Top to bottom, L-R: Dimetrodon, Urban 2, Grove 2, Rubber Whale, Venus, Hydrangea

6 RESULTS

The results we obtained from our implementation of the algorithm described in the paper are tabulated below.

Sequence	Santoro et al.			Our results		
	Endpoint Error Eqn 19	Endpoint Error Eqn 12	Improvement in dB	Endpoint Error Eqn 19	Endpoint Error Eqn 12	Improvement in dB
DIMETRODON	0.215	0.215	**0.00**	0.208	0.208	**0.00**
GROVE 2	0.202	0.254	**0.98**	0.226	0.232	**0.12**
GROVE 3	0.618	0.683	**0.43**	0.649	0.744	**0.59**
HYDRANGEA	0.230	0.230	**0.00**	0.234	0.229	**-0.08**
RUBBER WHALE	0.161	0.161	**0.00**	0.180	0.191	**0.25**
URBAN 2	0.418	0.472	**0.53**	0.500	0.562	**0.50**
URBAN 3	0.662	0.897	**1.32**	0.767	0.980	**1.06**
VENUS	0.315	0.330	**0.20**	0.315	0.340	**0.33**

7 ANALYSIS OF RESULTS

The results we've obtained are consistent with the results reported in the paper, with very little variation. A few observations that are immediately apparent from the results are:

1. Our endpoint errors for both Eqn. 12 and Eqn 19 are similar to the one published in the paper. A possible reason for the difference in the values between the published and our results could be difference in Block sizes and search size windows.
2. With the exception of the Hydrangea sequence, we have obtained a positive improvement (in dB) of Eqn 19's endpoint error over Eqn 12's endpoint error. These dB values are also appreciably close to those reported in the paper.

We also tried to improve the algorithm by trying to find a good initial estimate of the MVs during the SAD minimization. Currently in the algorithm, the initial estimates are found by capturing the first MV which has the minimum SAD value. We tried to modify this and capture more number of MV's and then find the "best" one among them. Currently the "Best" one we chose was the average MV among them. This did not improve the result. We believe that with some amount of effort, a short algorithm may be developed for figuring out the "best".

8 TEAM WORKLOAD BREAKDOWN

The codebase we received to work on did not include code for reading the .flo files obtained from the Middlebury website, or code for calculating end point error. Therefore, we wrote our own code for carrying out these two activities, besides code for the block overlap calculation.

The individual responsibilities of team members were as follows:

1. ***Ashwin A Shenoi***
   a. Setup code development flow for the code by writing Makefile and bash scripts for compilation and batch execution of Middlebury test sequences.
   b. Wrote code to implement the various corner-case cost functions and initial code for block overlap calculation, and Perl scripts to calculate endpoint error and decibel improvement for both the Lagrange case and Block Overlap case.
2. ***Guru Das Srinagesh***
   a. Extraction of Ground Truth from the .flo files obtained from Middlebury's website.
   b. Wrote code to implement the block overlap calculation, integrate the flow extraction code into the main code base and testing the overall codebase by tweaking block size and boundary_min parameters.

Beside these main responsibilities, the code was jointly debugged for fixing errors by both members. The presentation and report were also jointly prepared by the team members.

9 CONCLUSION

In this course project, we were given the opportunity to reproduce and enhance the results obtained by Mr. Santoro, et al. We found this a very instructive experience in that we worked on something fundamental to the concept of motion estimation – motion vector estimation. By working on the C++ code provided, not only did we augment our coding skills but also gained some very valuable insights into the working of motion estimation.

To summarize, we implemented the block matching algorithm, ran it on the Middlebury dataset and observed results very close to the results reported in the paper. The algorithm, while providing good results for most cases, seems to fail in certain cases. Upon further analysis by tweaking certain parameters, we observed that the results of the algorithm seems to be dependent on the kind of image content and the kind of motion present.