Report: Optical Flow estimation & Global motion estimation in the image plane with RANSAC algorithm

Hunor Laczkó

I. INTRODUCTION

The goal if this lab is to explore the global motion estimation calculation method with RANSAC. For this, first the optical flow has to be calculated. This can be done in multiple ways which are compared in this report. To evaluate the different methods several metrics were used like MSE, PSNR, Entropy. For the implementation OpenCV and python were used.

II. METHOD AND IMPLEMENTATION

In this section the different methods and the implementation details will be discussed.

A. Optical Flow

Optical flow is the motion of objects in a video between two consecutive frames. This can be caused by the movement of the camera or the object. It can be represented by a 2D vector field where to each pixel a corresponding vector is assigned showing the displacement, meaning the movement of the pixel from the first frame to the second one. Several methods exist to calculate optical flow, a few of them are presented below. Because it will be used for the global motion estimation, dense optical flow methods had to be used to have information about each pixel. It is also worth to mention that the calculations are done using the grayscale versions of the video frames.

1) Farneback Method: This method was introduced by Gunnar Farnebäck in a paper from 1893 [1]. It proposes an effective algorithm to estimate the motion of interesting features by comparing the two consecutive frames. In the first step the windows of the image frames are approximated by quadratic polynomials with the help of polynomial expansion transform. The transformations of these polynomials is observed under translation and a method to estimate the displacement vector field from polynomial expansion coefficients is defined. This is done several times in order to refine the results and the dense optical flow is calculated from this.

An implementation of this method can be found in the OpenCV library and can be called with *calcOpticalFlowFarneback()* function. It has several parameters to fine-tune its operation:

- **prev**: previous image frame
- **next**: next/current image frame
- **pyr_scale**: the scale parameter for building the pyramid (to detect at different scales)
- **levels**: number of layers in the scale pyramid
- **iterations**: number of iterations the algorithm does at each pyramid level.
- **poly_n**: size of the pixel neighborhood used to find polynomial expansion in each pixel
- **poly_sigma**: standard deviation of the Gaussian that is used to smooth derivatives used as a basis for the polynomial expansion

Since this was not the main aim of the lab, these parameters were not fine-tuned.

2) Dual TV L1: This is a variational method to calculate the optical flow. It is based on total variation regularization and the L1 norm. It preserves discontinuities in the flow field and provides an increased robustness against illumination changes, occlusions and noise. This method is slightly slower than the Farneback method.

3) Dense RLOF Optical Flow: Fast dense optical flow computation based on robust local optical flow (RLOF) algorithms and sparse-to-dense interpolation scheme. The RLOF is a fast local optical flow approach similar to the pyramidal iterative Lucas-Kanade method as proposed by [3].

4) Note: Only the Farneback method was evaluated fully, the other methods were only experimented with in some cases. Because of high computational need for the Person-Convergence video, the Dual TV method was only evaluated on the LongJump video.

B. Global motion estimation

Global motion estimation was the second task of this lab. It is determined with the help of the previously calculated optical flow. First, a specific image is generated which will be used as a helper variable. It is constructed such that the middle has a value of zero, going to the right and down the values increase by one at each pixel while in the opposite directions they decrease similarly. This assures that the estimation will be centered to the center of the image. This image will represent the source image. Adding to this source image the optical flow values provides the destination image. Every pixel has a corresponding flow value so this operation is a simple elementwise addition. Between these two images, the source and destination, the homography is calculated using OpenCV's built in *findHomography()* method. This uses the least square method coupled with RANSAC algorithm. Using the RANSAC algorithm assures that the outliers are not taken into consideration during the calculation. These outliers are the objects that move differently in the frame, meaning they have a different motion or flow value than the majority of the pixels. So by using RANSAC the individual objects get ignored and only the majority motion, meaning the global motion is considered. To get the values of global motion, the previously calculated homography is applied to the source image which gives us the destination image where the outlier objects' motions are omitted. Since this source image was specially generated, applying the homography resulted in an image that is equal to the source image plus the global motion's flow. By subtracting the source image from this, the remaining part will be global motion itself.

After estimating the global motion, the residual motion can be calculated too. The residual motion is the normalized L2 distance between the original motion vectors and the computed global motion vectors. The energy of this residual motion can be visualized, as it can be seen later in the analysis section. This residual motion represents the objects in the frames that moved differently than the global motion.

C. Metrics

To compare the different methods, parameters and videos several metrics were used that are presented below.

1) Mean Squared Error: This metric is given by the following formula:

$$MSE = \frac{1}{N \cdot M} \sum (I(p,t) - I(p, t - \Delta t))^2$$

where p is a pixel coordinate from the frame. This metric measures contrast and motion difference between frames. It is especially high in case of a noisy video.

2) Peak Signal to Noise Ration: This metric is given by the following formula:

$$PSNR = 10 \log_{10}(\frac{255^2}{MSE})$$

As it can be seen from the formula, it is calculated based on the MSE. It is used to describe difference between frames of a video, and it is mostly used for assessing motion compensation and coding methods.

3) Entropy: This metric is given by the following formula:

$$Entropy(I) = -\sum p(x_i) \log_2 p(x_i)$$

where $p(x_i)$ is the probability that a pixel has color x_i. This formula gives the amount of information in the frame. With this en information of the original frame can be calculated, also the entropy of the error image can give information about the proportions of the error.

III. Results and Analysis

The methods described so far were evaluated using multiple videos. These videos and the corresponding results will be presented in this section.

A. PersonConvergence video

This video shows a person walking away from the camera.

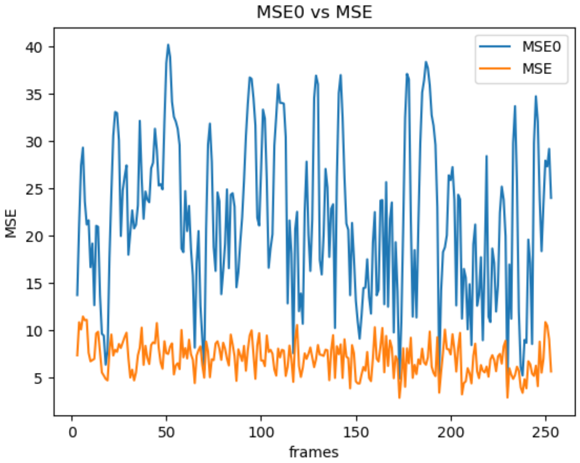


Fig. 1: PersonCOnvergence: MSE0: previous frame and current frame, MSE: compensated frame and current frame

1) Video analysis: The video was taken probably with a handheld camera, thus it has some, although minimal, camera movement. The main object is a person in the center of the frame walking away from the camera, and this person represents the most motion outside of the global motion. There are some vehicles and people further away that also move but much less compared to the main person.

2) Optical flow evaluation: To evaluate the optical flow calculation a compensated frame is generated from the previous frame and optical flow values and this compensated image is compared to the current image. In theory these two images should be the same, but since the optical flow is just an estimation and it is not perfect there is some difference between these two frames. The difference between the compensated image and the current frame is also compared to the difference between the previous frame and the current frame. The latter one acts as a naive approach to the optical flow calculation and provides the baseline. The compensated frame is generated by displacing the pixels in the previous frame with the value of the optical flow vectors. The previous frame is defined by deltaT which is the number of frames before the current frame. This is 3 by default.

First the results are evaluated based on their MSE values. These can be seen in Figure 1. It shows that the Mean Squared Error difference between the previous frame and the current frame (MSE0) is large and highly varied, while the MSE of the compensated frame and current frame (MSE) is significantly smaller and varies in a smaller range. This shows that with the help of the optical flow values a compensated frame could be generated that is closer to the expected frame than the previous frame of the video.

Next, the PSNR values are shown in Figure 2. It supports the conclusions from the MSE comparison. The higher the value, the less noise is present in the method. It can be seen that the compensated image (PSNR) provides higher values than the naive approach (PSNR0). It is also interesting to note that two graphs show similar trends, for instance they have

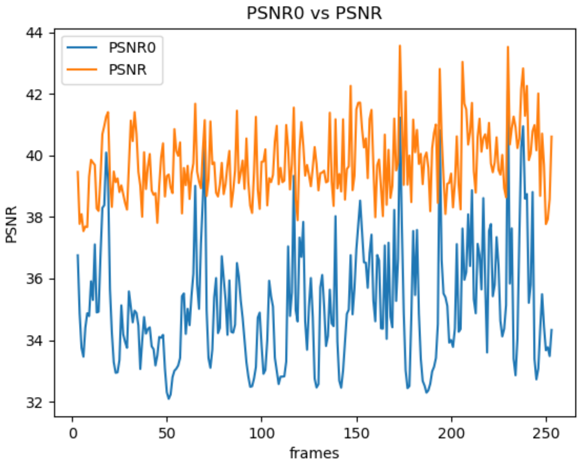


Fig. 2: PersonCOnvergence: PSNR0: previous frame and current frame, PSNR: compensated frame and current frame

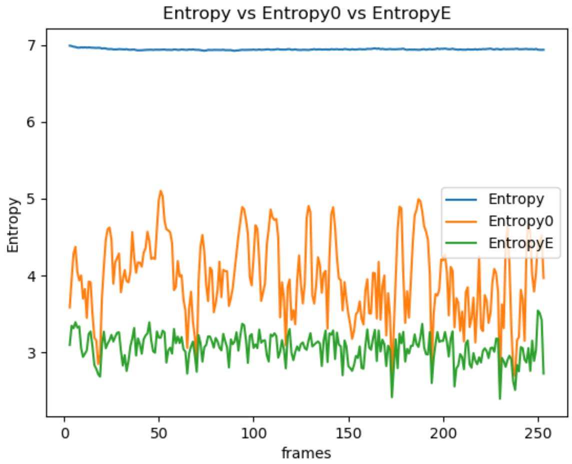


Fig. 3: PersonCOnvergence: Entropy: current frame, Entropy0: previous frame and current frame, EntropyE: compensated frame and current frame

their peaks in the same places. These peaks represent frames where probably there was a sudden movement which was harder to compensate and neither methods could handle them perfectly. Although some of the peaks of the compensated image's graph seem proportionally smaller than the naive approach's which means it could handle those frames better.

Lastly, the entropy values of different images are compared. First the entropy of the current frame is calculated (Entropy) which serves as a baseline so the error values can be compared to this. For the next part, an error image is defined as the difference between two frames and later offset with the middle value of the value range of a given pixel. This results in an image that is gray where is no error and a lower or higher value where there is some difference. For evaluation, the entropy of these error images will be calculated, where lower the value, the less error is present. This is because of the definition of the entropy, if there is no error, the error image is homogeneous containing zero information. Two error images are calculated, first between the previous frame and the current frame (Entropy0) and then between the compensated frame and current frame (EntropyE). As it can be seen in Figure 3, the EntropyE is lower than the Entropy0 meaning the compensated image results in a lower error. Both of them are lower than the baseline Entropy. Although neither of them reaches the ideal zero entropy they both decrease the error significantly below the original image's entropy.

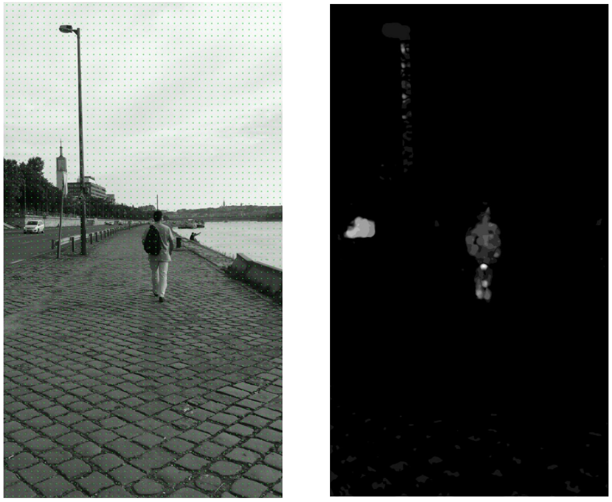
3) Global Motion Estimation: As described in Section II-B the global motion estimate is calculated based on the previously determined optical flow for each pixel. Since there is almost no camera movement or any dynamic background there is no definitive global motion. This can be seen in Figure 4a as the vectors representing the flow have zero magnitude. By calculating the error between this global motion estimation and the original flow vectors the residual error can be acquired. The residual error will represent the objects that move independently from the global motion. This can be seen in Figure 4b. As shown in the figure several blobs can be seen representing several moving objects. In the middle is the main person walking away which can be seen clearly by its silhouette. On the left there is a car represented by a whiter color, meaning the error is greater. This is because the car is moving faster than the other objects. Between the two we can see part of the lamp pole. Although this is most likely not moving it it shown like that. This is probably due to the fact that there is some minimal global motion which does not affect most of the image so it was not calculated, but here the dark pole is against the light background, so any small movement will result in a great change in the pixels intensity values, so the slightest difference shows as movement. It is similar to the rocks on the ground. Here the ground is close to homogeneous, so slight movement does not result in different intensity values. But at the part which is close to the camera it is easier to make out the differences between the tiles, also since they are closer to the camera the small camera movements show stronger on these parts. As a result, some of the tiles show up as movement too.

B. LongJump video

This video shows a person performing a long jump.

1) Video analysis: The first problem is the low resolution of the video, since this means there is less information available. The second problem is the sudden movement of the camera in the middle of the video. The camera is panning, trying to follow the jumper, but he is running towards the camera, until gets so close that it occupies the whole frame then start getting farther from the camera.

2) Optical flow evaluation: The evaluation follows the same properties as in the previous video. First the Mean Squared Error is presented in Figure 5. As mentioned before, there is a sudden movement in the middle of the video where even with human eyes it is hard to follow the movement.



(a) Flow (b) Residual Error

Fig. 4: PersonCOnvergence: Global Motion Estimate

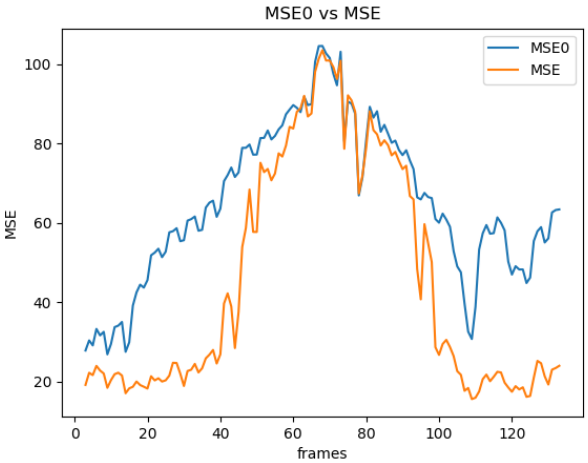


Fig. 5: LongJump: MSE0: previous frame and current frame, MSE: compensated frame and current frame

Because of this, the optical flow calculations (MSE) fails at this point. This can be seen in the figure, where around frame 70 it has the same error as the naive difference (MSE0), which is twice as much as the maximum error in the previous video. The pan movement of the camera and its speed can be clearly seen in the figure, as it starts to get faster at around frame 40 and speeds up, the error keeps increasing, then when it slows down it decreases again at around frame 80.

The same behaviour can be seen in Figure 6. too, only here the smaller the value the higher the error. This figure confirms the same conclusions, that at frame 70 and around it, the optical flow calculation (PSNR) fails, and it is no better than the naive difference (PSNR0). In the slower parts of the video, in the beginning and in the end, it still performs better as the figure shows.

The entropy values also support the previous conclusions. It can be seen in Figure 7. that in the middle of the image

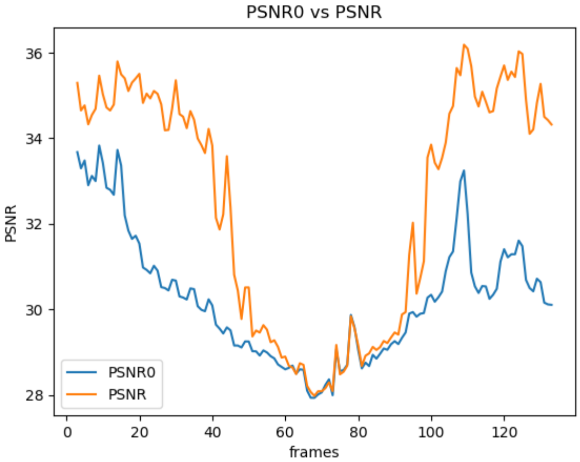


Fig. 6: LongJump: PSNR0: previous frame and current frame, PSNR: compensated frame and current frame

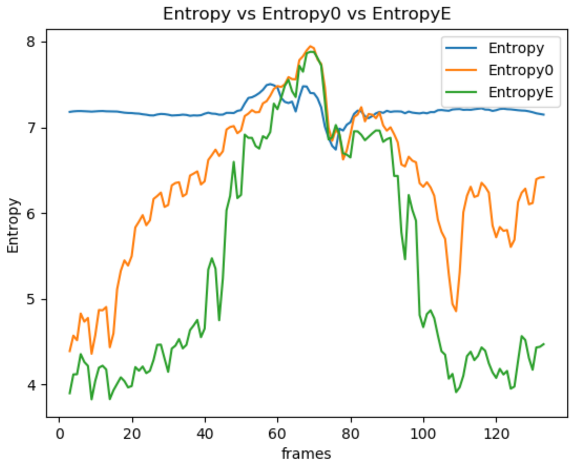


Fig. 7: LongJump: Entropy: current frame, Entropy0: previous frame and current frame, EntropyE: compensated frame and current frame

the entropy of the error based on optical flow (EntropyE) is even higher than the baseline (Entropy) and the same as the naive difference error (Entropy0). Although the optical flow performs slightly better it still fails in the middle of the image. The fact that the values surpass even the baseline means there is more error than valuable information in the original frame.

For the calculations until now, the default value of 3 was used for deltaT, meaning the gap between the two frames used for calculations. Because of the fast movement, lowering this value to 1 seemed promising, the results can be seen in Figure 8. Compared to Figure 5 it can be seen that the interval where the optical flow calculations performs poorly decreased, the decrease can only be seen in the immediate vicinity of frame 70. Also, the peak value of the error also slightly decreased from around 105 to 90.

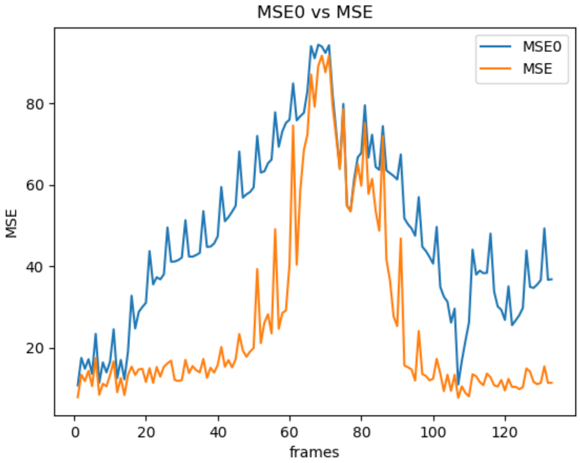


Fig. 8: LongJump, smaller delaT: MSE0: previous frame and current frame, MSE: compensated frame and current frame

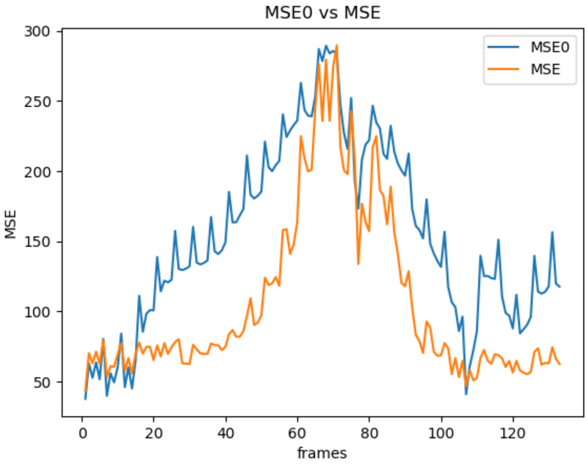


Fig. 9: LongJump, dense RLOF method: MSE0: previous frame and current frame, MSE: compensated frame and current frame

Since this method fails in the middle of the video, here other optical flow calculation methods were also explored. Because of the fast movement a deltaT of 1 is used from this point. First the Dense RLOF method was used, the resulting MSE values can be seen in Figure 9. As the figure shows, it performs similarly to the Farneback method. By using the feature based optical flow calculation it is faster, but in accuracy it fails at the same point as the previous method. It can also be seen that the MSE values are significantly higher than before. This is because this method uses color frames compared to grayscale as previously, so the error values have to be roughly multiplied by three for the three color channels. But the error trends are the same.

Next, the Dual TV L1 method was tried. The results are shown in Figure 10. As it can be seen, it performs significantly better than previous methods. It performs better than the naive difference across the whole video, even in the middle part of the video, while the performance degenerates, it does not fail completely. Since this method performs the best, it will be used for the global motion estimation.

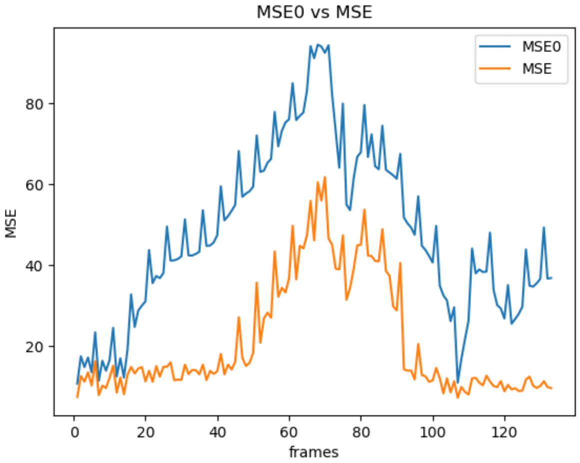


Fig. 10: LongJump, Dual TV method: MSE0: previous frame and current frame, MSE: compensated frame and current frame

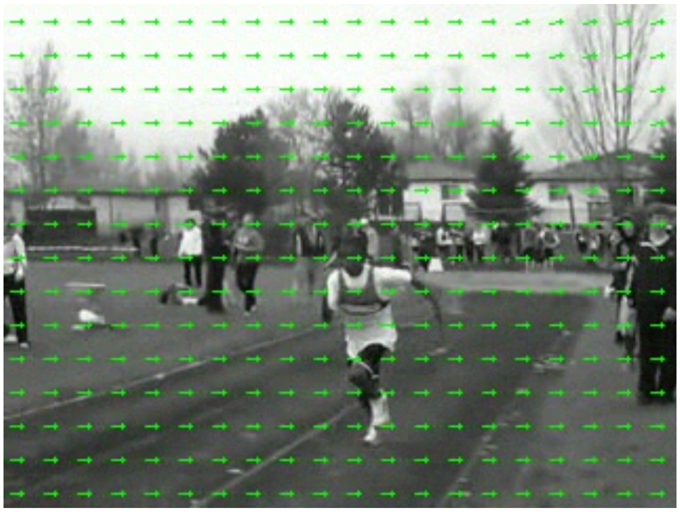


Fig. 11: LongJump GME flow

3) Global motion estimation evaluation: For evaluating the global motion estimation, the same steps were used as the previous video. In this case there is a strong camera movement which can be seen in the global motion estimation as well, as shown in Figure 11. Since the camera is moving to the left the objects in the frame appear to move to the right, which is why the global motion is pointing to the right.

At this point the residual error image seen in Figure 12. is as expected, showing the running person clearly in the middle. There are some other detected blobs which correspond to some moving people in the background which is also correctly classified. The other smaller ones are most likely due to noise.

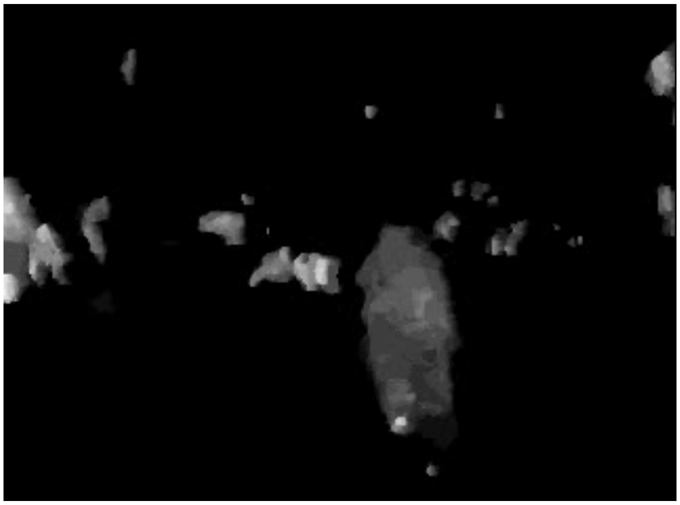


Fig. 12: LongJump GME error

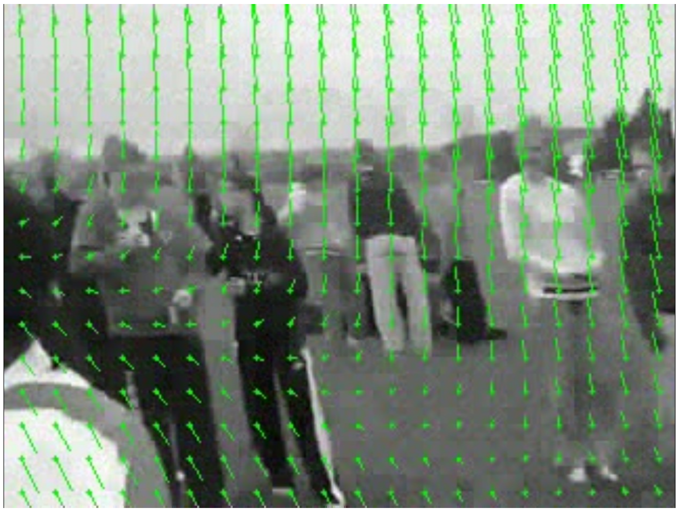


Fig. 13: LongJump GME flow around frame 70

The results are worse in the middle of the video, around frame 70. As seen in Figure 13 there is no clear global motion, also the majority of flow vectors are pointing downwards which is not correct, the camera is still moving to the left. Due to this there is high residual error generated from this.

The error can be seen in Figure 14. There is little usable information in this image. There seem to be slightly higher error where people are standing, but hard to make out. Similarly, the parts of the frame where only the far distant scene can be seen has lower error which is correct, but there is no finer, usable detail in this image.

IV. CONCLUSION

Successfully explored how optical flow calculation works and applied it to global motion estimation. Managed to achieve acceptable results with this method. In conclusion, the goal of the lab was completed. For future work, other optical flow calculation methods can be tried, and their parameters can be fine-tuned for even better results.

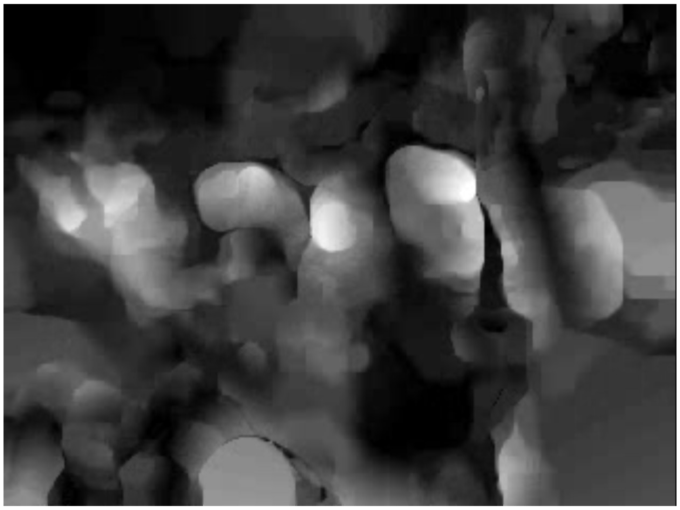


Fig. 14: LongJump GME error around frame 70

REFERENCES

[1] Farnebäck, Gunnar. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In: Image analysis. 2749. 363-370. 10.1007/3-540-45103-X_50.

[2] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In Pattern Recognition, pages 214–223. Springer, 2007.

[3] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation, 5, 2001.