Accelerated Video Depth Estimation

YIJUN YUAN, University of Waterloo

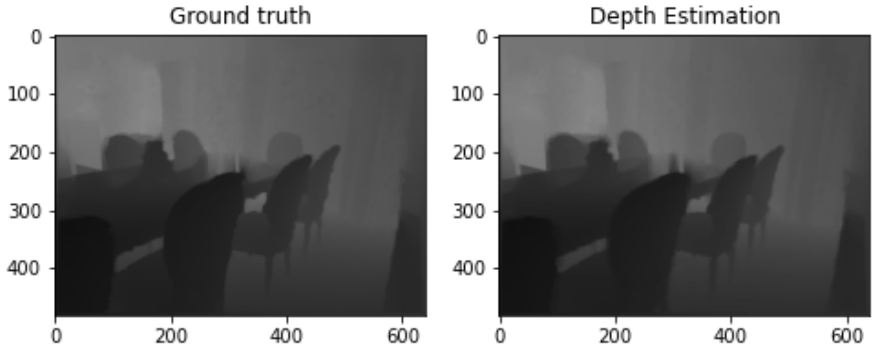


Fig. 1. Ground Truth v.s. Estimation: By using the accelerated video depth estimation, the middle frame's depth information of five consecutive frames could be accurately predicted relying on the depth information the first and the last frame.

Many video editing applications require depth information, and many depth estimation algorithms were proposed in the last couple of years. These methods could provide consistent and accurate video depth estimation. However, the processing time for these video depth estimations is extremely long, especially for these learning-based methods. In this paper, I attempt to propose a method to speed up the video depth estimation. So, more users and applications could take advantage of video depth information. My method relies on optical flows to determine the keyframes and only these frames require accurate depth information. The depth information for all the other frames could be estimated with the information of keyframes. My method is expected to speed up regular video processing speed by 50%.

CCS Concepts: • **Computing methodologies** → **Computer vision**.

Additional Key Words and Phrases: Video Depth Estimation

ACM Reference Format:
Yijun Yuan. 2020. Accelerated Video Depth Estimation. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Many depth estimation techniques have been proposed in the last decade, and researchers managed to achieve increasingly high accuracy with the help of machine learning. However, the processing

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

time also started to increase. Starting from the classic structure-from-motion method, which is very fast and relying on a sequential processing pipeline with an iterative reconstruction component [13], to deep learning method with multiple levels of networks [9]. Recently, another paper with stunning results is published [11], and a convolutional neural network is used to improve the video depth consistency. Their approach significantly improves the performance with the sacrifice of 10 seconds of processing time to compute one frame.

The time efficiency problem is very common when learning-based approaches are used, and there are not many methods to speed up the time required for machine learning processing, aside from using more robust hardware and better frameworks. Inspired by Kellnhoger's research [7], I found that when the displacements between two continuous frames are very small. The difference among their depth estimation should also be very limited or almost negligible. The method proposed in this paper aims to reduce the number of frames required accurate depth information from these slow depth estimation methods and relies on a much faster method to compute estimation for the rest of the frames. This method is expected to have a much better performance on videos with slow motion to videos shooting high-speed moving objects.

In this paper, I attempt to propose a method to accelerate the video depth estimation and analyze the effectiveness and trade-offs of the methods. By using my methods, a certain portion of the video frame could be processed using a much faster method. The next section will discuss the related work in this field. Section III presents the method, and section IV provides the experiment results. I will discuss the limitations and conclude my research in the last two sections.

2 RELATED WORK

Many researchers have proposed researches to estimate the video depth information. These methods are very different from the traditional image depth estimation method, such as the structure-from-motion [13], as continuous frames could provide more information than one image. However, estimating video depth could also be relatively difficult as the depth information could become inconsistent across different frames [11]. This section will focus on the related work using learning-based approaches, as they are the current state-of-art approaches.

2.1 Supervised Depth Estimation

The supervised depth estimation methods usually require accurate depth information. For fully supervised models, ground truth depth information is used during training. It can be very challenging to acquire the video input with matching depth information, but it is achievable when the cameras and sensors are set-up properly. An encoder-decoder setup is very commonly used when the ground truth is available [1] [17] [9]. Ibraheem developed a straightforward encoder-decoder architecture with skip connections to estimate depth map from a single RGB image [1]. Haokui proposed a modified encoder-decoder network by adding a multi-scale feature fusion module to generate the feature mapping and using convolutional long short-term memory on the feature mapping to estimate the depth information [17].

Due to the difficulty of acquiring the ground truth depth information, some network uses weakly supervised training data, like estimated depth information from structure-from-motion pipeline [8], or other depth information. Nan and his team built their deep learning framework using a supervised learning-based network on accurate sparse depth reconstruction by Stereo DSO [16]. Chaoyang and his team built their network to take two consecutive frames and their optical flow as input, and relied on computed disparity map used for supervision [15].

2.2 Self-supervised Depth Estimation

Without the ground truth depth information, researchers found that information such as stereo pairs or monocular information could be used to train the model.

The self-supervised stereo model requires two frames with small object displacement to predict the pixel disparities. Godard and his team produced accurate results by adding a left-right depth consistency evaluation in their network [4]. Luo and her team used the spatial loss and disparity loss between two consecutive frames to train the network in order to achieve geometric depth consistency across all depth estimation [11].

The self-supervised monocular model requires consecutive frames in the monocular videos as training data. In addition to estimating the depth information, this model relies on the re-projection of consecutive frames to estimate the location of the camera and the objects and reduces the errors among this estimation with multiple frames. Godard and his team also used this method and achieved accurate results by minimizing the pixel level re-projection loss [3].

2.3 Time efficiency

Most of those learning-based methods are not very time efficient and require powerful GPUs for their calculations. [4] requires 25 hours to train the network and 35 microseconds to evaluate a 512 x 256 image on a Titan X GPU. [9] requires around 1.5 seconds with pose estimation on a workstation with GTX 1080 GPU. [11] requires 10 seconds to compute the depth information for one frame. Rene and his team used a motion segmentation approach to predict the depth and they reached an execution time of 1 minute per frame [12].

3 METHOD

The methodology proposed in this paper contains two section. The first section contains the pre-processing, which includes determining keyframes that require accurate depth information and acquiring the depth information for keyframes. The second section computes the depth from the rest of the frames based on the keyframes' depth estimation.

3.1 Pre-Processing

In this process, keyframes are determined using Motion Estimation with Optical Flow technique. In order to determine keyframes that could provide the best performance, the magnitude of optical flow is tracked, and the skip rate could also be controlled by carefully selecting a threshold during this process.

Keyframe: if a frame is considered to be keyframe, then its depth estimation should be calculated accurately using a different method. Keyframes' depth information will be used as guidelines to estimate the depth of the other frames.

3.1.1 Calculating the optical flow. Optical flow is the motion of objects between consecutive frames, caused by the relative movement between the object and camera. The image intensity I could be measured as a function of space (x, y) and time (t). Taking one image and moving the pixels by a certain amount would create the second image. From two consecutive frames, the displacements of pixels are very small, and the intensity of the second image could be represented as $I(x + dx, y + dy, t + dt)$. There are multiple methods to measure the optical flow, Sparse Optical Flow [6] [10], Dense Optical Flow, and using learning-based methods like FlowNet[5]. As the purpose of this research is to improve time efficiency, the sparse optical flow method is chosen as the main method and the deep learning method is considered as an alternative method if the sparse optical flow results are under expectation.

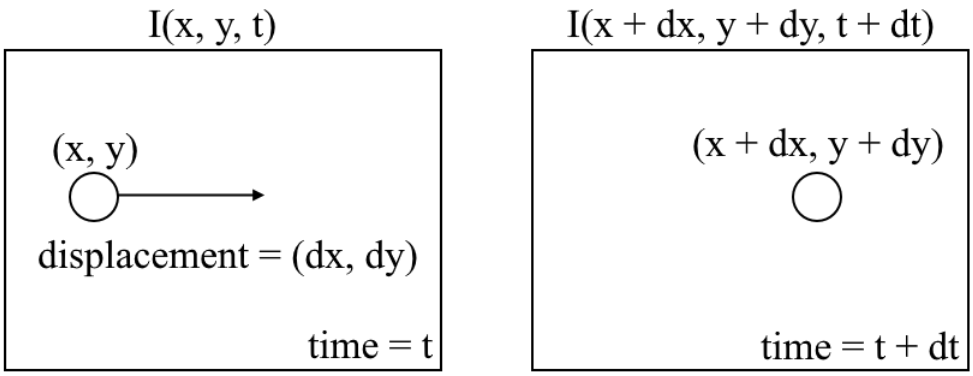


Fig. 2. Optical Flow [2]

To calculate the sparse optical flow between two frames, the first step is to identify the features to track in the first frame. Shi-Tomasi Corner Detector[1] [6] was used to identify important features. It is a modified version of the Harris corner detector and could achieve better results due to the modified score function.

After identifying all tracking features, the Lucas-Kanade[2] [10] method is used to measure the optical flow. The Lucas-Kanade method assumes that the displacement of the image contents between two nearby frames is small and approximately consistent within the selected window, determined by the tracking feature found in the last step.

3.1.2 Determine key frames. Keyframes could be determined when the magnitude of the optical flow between two frames could be calculated. However, it is more important to understand the concept of skip rate and control it.

Skip rate: The skip rate is used to track how many frames are skipped, in other words, how many frames are non-keyframes. Because only keyframes require accurate depth estimations which would take a very long time to compute, a higher skip rate could significantly improve the time efficiency. On the other hand, a smaller skip rate would require more keyframes, thus increase the processing time.

Threshold: The threshold ϵ is used to determine if the displacement between two frames is big enough to consider them as keyframes. Choosing a large threshold would allow the more frames to be considered as non-keyframes thus increase the skip rate. However, if the threshold is too big, the overall accuracy could be affected.

Here are the rules to determine the keyframe by calculating the optical flow:

(1) The first frame is always a keyframe
(2) If frame i is a keyframe, then frame $i+1$ is not a keyframe if the optical flow between frame i and $i+1$ is less than the threshold $|F_{i,i+1}| < \epsilon$
(3) If frame i is a keyframe, and frame $i+1$ is not keyframe, then frame $i+2$ is not a keyframe if the optical flow between frame i and $i+2$ is less than the threshold $|F_{i,i+2}| < \epsilon$
(4) If frame i is a keyframe and frames i+1 to i+l-1 are not keyframes, then frame i+l and frame i+l+1 are both keyframes when they satisfy following conditions:

[1] Appendix A: Shi-Tomasi Corner Detector
[2] Appendix B: Lucas-Kanade

The optical flow between frame i and i+l is less than the threshold $|F_{i,i+l}| < \epsilon$
The optical flow between frame i and i+l+1 is greater than the threshold $|F_{i,i+l+1}| > \epsilon$

(5) The last frame is a keyframe

It is important to know that Rule #3 tracks the optical flow between frame i and i+2 because it is necessary to ensure that the displacement between any non-keyframe and the keyframe in front of it is within the threshold. Similarly, the displacement between any non-keyframe and the keyframe behind it should also be bound by certain thresholds.

Lemma 1: if frames i and $i + l$ are the only two keyframes between range i to $i + l$, then

$$\forall k \in (i, i + l), \exists \epsilon', s.t. |F_{k,i+l}| < \epsilon'$$

Proof:

$$\text{By Triangle Inequality Theorem, we have} |F_{k,i+l}| < |F_{k,i}| + |F_{i,i+l}|$$
$$|F_{k,i}| = |F_{i,k}| < \epsilon \text{ and } |F_{i,i+l}| < \epsilon$$
$$\text{we can choose } \epsilon' = 2\epsilon$$
$$\text{so } |F_{k,i+l}| < \epsilon'$$

3.1.3 Acquiring the depth estimation for the keyframes. After determining the keyframes, am accurate depth estimation method should be used to predict their depth information. However, the processing time is significantly reduced due to the skip rate.

3.2 Computing Depth

Depth information of non-keyframes could be estimated based on the depth information of keyframes. By combining Lemma 1 and the rules in section 3.1.2, it is easy to conclude that the magnitude of the optical flow between any non-keyframes and the keyframes before or after can be bound by a specific threshold, meaning the maximum magnitude of the displacement can be controlled. In this case, any non-keyframes' depth information would be very similar to the keyframes' around them.

For any non-keyframe k, let frame i be the keyframe in front of it and frame $i + l$ be the keyframe after it, D_i and D_{i+l} are the depth estimation for both keyframes with $F_{k,i+l}$ being the optical flow from frame k to $i + l$ and $F_{i,k}$ being the optical flow from frame i to k, the depth information for D_k could be estimated using the following formula:

$$D_k = \frac{|F_{k,i+l}|}{\sum |F|} D_i + \frac{|F_{i,k}|}{\sum |F|} D_{i+l}, \sum |F| = |F_{i,k}| + |F_{k,i+l}|$$

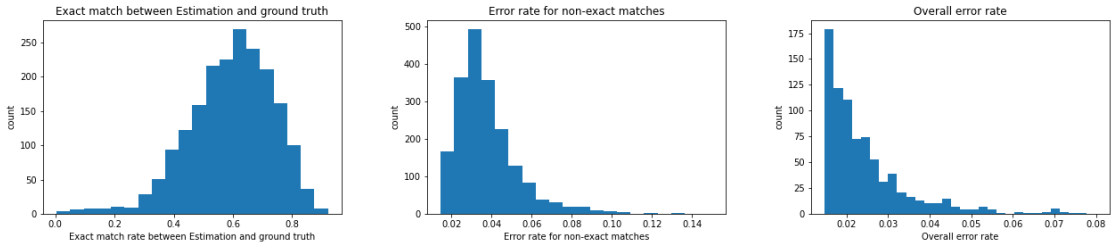
4 EXPERIMENT

4.1 Setup

Due to the limitation on the evaluation dataset, the input must be continuous frames or video with ground truth depth information. Not many datasets satisfy this requirement.

For general efficiency and performance evaluation, I evaluate my method on the NYU Depth V2 datasets [14] to target the sparse optical flow method specifically, as I want to ensure the sparse optical flow could provide accurate results. For the video performance test, I used the video as well as the result from Luo's research [11] to study the trade-offs between different thresholds (skip rate) and accuracy.

The system is written in python, and the processing time of one non-keyframe in our method is around 50 microseconds and around 10 microseconds per frame when factoring in all frames include keyframes. The system is running on i7 4-core CPU; however, the system is only designed to run on one thread without multiple-threading performance boost.



(a) (b) (c)

Fig. 3. Results for the general efficiency and performance evaluation

4.2 General efficiency and performance

NYU Depth V2 dataset [14] is used here with 654 sets of five continuous frames. Since the goal of this test is to evaluate the performance of the sparse optical flow method, keyframes are not determined by the threshold. Instead, the first frame and the last frame are directly considered as the keyframes while the middle three frames are considered as non-keyframes, achieving a skip rate of 60%.

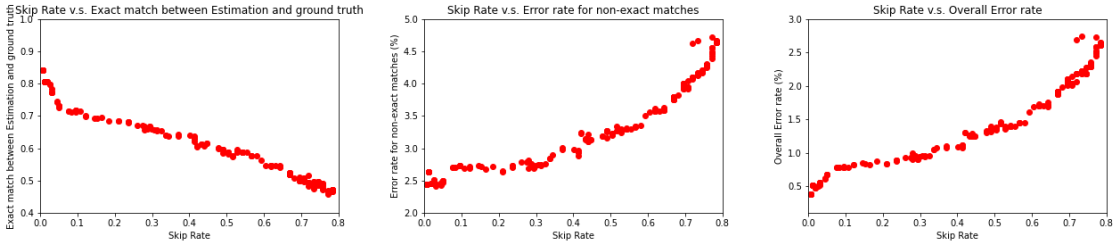
Three evaluation metrics are used in this test: the absolute match rate, the error rate for non-exact matches, and the overall error rate. The absolute match rate measures the number of pixels with an exact match between estimation and the ground truth. The error rate for non-exact matches checks the difference between estimation and the ground truth when the prediction is not accurate. The overall error rate is very similar to the error rate for non-exact matches. However, the overall error rate is measuring the average error on all pixels in one estimation. It is important to keep both errors small because if the overall error rate is generally small, with the error rate for non-exact matches being relatively high and not following a similar distribution, it is not difficult to conclude that the absolute match rate might be too high and some pixels are having extremely large errors.

Figure 3 presents the results of three evaluation metrics. For all 654 test sets and 1962 individual results, the average exact match rate is around 60%, meaning around 40% of pixels are having an average error of around 3%. The average overall error rate for all test sets is around 1.5%. From the exact match graph, a majority of cases fall around the average rate of 60%, with some cases falling below a 20% match rate. Similar to the over error rate, most cases are within 3% and some cases are having an error rate of around 7%. These are considered to be minor cases and an acceptable result. The results from this test have proven that the sparse optical flow method could provide reliable results as well as time efficiency of 50 microseconds pre non-keyframe.

4.3 Video performance test

The goal of this test is to evaluate the performance of the actual video. The dataset in this test requires both the input video and its depth information. Six seconds of test video with 156 frames from Luo's research [11] is used here. One key goal in this test is to study the trade-offs between the threshold (skip rate) and the absolute match rate, the error rate for non-exact matches, and the overall error rate metrics.

Figure 4 presents the results of trade-offs and three evaluation metrics. A total of 200 different thresholds are tested in this evaluation. According to the graph, there will be an accuracy penalty even when the skip rate is relatively small. With the skip rate falling very close to zero, the exact match will not start to drop from 100%, but around 85%. Similarly, for the error rate for non-exact



(a) (b) (c)

Fig. 4. Results for the video performance test

matches and overall error rate, they are to increase from 2.5% and 0.4% respectively. There is a significant drop in performance when increasing the skip rate from 0% to 5%. After a 5% skip rate, the trade-offs between skip rate and performance starts to improve. 50% skip rate could be considered as a point with reasonable trade-offs between both time efficiency and performance, as the exact match rate only drops around 10% and the overall error rate raises around 1% compared to the result using a 5% skip rate. The maximum threshold tested provides a skip rate of around 80%, and I believe that anything above 75% or 80% would provide inaccurate results. However, using a skip rate of 70% does not significantly affect the performance compared to the result from a 50% skip rate with a 2% overall error rate and a 50% exact match rate.

In this test, I think that the trade-offs between the performance and time efficiency are acceptable, and skipping 50% of the frames or even 70% of the frames would still provide reasonable results.

5 DISCUSSION & LIMITATION

5.1 The impact on very high resolution

In this research, very high-resolution video or images are not tested due to the lack of datasets. The datasets used in this work are the datasets commonly used for a similar type of study, which are lower than 2K resolution. Especially the 2K or 4K resolution videos are available and very popular online recently, the performance of my method could not be verified on these resolutions would be a limitation. However, it is expected to have similar performance as lower resolution contents, and the processing time is also expected to be geometric growth when the resolution increases.

5.2 Potential performance impact on different depth estimation methods

My method could potentially affect the performance of depth estimation methods, and I expect different levels of impact based on the nature of estimation methods. If the depth estimation methods could provide accurate estimation with one frame or a pair of frames [1] [9] [8] [16] [15], then my method is unlikely to affect their performances. However, if the depth estimation methods heavily rely on a continuous input of frames to improve accuracy, then my method could affect the performance. For example, the methods in research [11] [3] [17] use multiple frames to maintains overall depth consistency, and when my method reduces the number of frames, the performance could be affected. In this case, it is recommended to increase the number of keyframes by reducing the thresholds, trading off time efficiency for performance.

6 CONCLUSION

With the accelerated video depth estimation method, the processing time of the video depth estimation could be significantly reduced. Based on the evaluation results, reducing 50%-75% of frames would still provide reasonable accuracy. My method could solve the time efficiency problem caused by several state-of-the-art depth estimation algorithms that require an extremely long time to estimate the video depth information, making more video-based visual effects applications accessible by users who have limited computation powers.

REFERENCES

[1] I. Alhashim and Peter Wonka. 2018. High Quality Monocular Depth Estimation via Transfer Learning. *ArXiv* abs/1812.11941 (2018).

[2] Chuan en Lin. 2019. *Introduction to Motion Estimation with Optical Flow.*

[3] C. Godard, Oisin Mac Aodha, and G. Brostow. 2019. Digging Into Self-Supervised Monocular Depth Estimation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 3827–3837.

[4] C. Godard, Oisin Mac Aodha, and Gabriel J. Brostow. 2017. Unsupervised Monocular Depth Estimation with Left-Right Consistency. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 6602–6611.

[5] Eddy Ilg, N. Mayer, Tonmoy Saikia, Margret Keuper, A. Dosovitskiy, and T. Brox. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 1647–1655.

[6] Jianbo Shi and Tomasi. 1994. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* 593–600.

[7] Petr Kellnhofer, Thomas Leimkühler, Tobias Ritschel, Karol Myszkowski, and Hans-Peter Seidel. 2015. What Makes 2D-to-3D Stereo Conversion Perceptually Plausible? 59–66. https://doi.org/10.1145/2804408.2804409

[8] Maria Klodt and A. Vedaldi. 2018. Supervising the New with the Old: Learning SFM from SFM. In *ECCV.*

[9] Chao Liu, Jinwei Gu, Kihwan Kim, S. Narasimhan, and J. Kautz. 2019. Neural RGB®D Sensing: Depth and Uncertainty From a Video Camera. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 10978–10987.

[10] Bruce Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI). *[No source information available]* 81.

[11] Xuan Luo, J. Huang, R. Szeliski, K. Matzen, and Johannes Kopf. 2020. Consistent video depth estimation. *ACM Transactions on Graphics (TOG)* 39 (2020), 71:1 – 71:13.

[12] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. 2016. Dense Monocular Depth Estimation in Complex Dynamic Scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 4058–4066.

[13] J. L. Schönberger and J. Frahm. 2016. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 4104–4113.

[14] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 746–760.

[15] Chaoyang Wang, S. Lucey, Federico Perazzi, and O. Wang. 2019. Web Stereo Video Supervision for Depth Prediction from Dynamic Scenes. *2019 International Conference on 3D Vision (3DV)* (2019), 348–357.

[16] Nan Yang, Rui Wang, J. Stückler, and D. Cremers. 2018. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. *ArXiv* abs/1807.02570 (2018).

[17] Haokui Zhang, Chunhua Shen, Y. Li, Yuanzhouhan Cao, Y. Liu, and Y. Yan. 2019. Exploiting Temporal Consistency for Real-Time Video Depth Estimation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 1725–1734.

A APPENDIX: SHI-TOMASI CORNER DETECTOR

Shi-Tomasi Corner Detector [6] is a modified version of Harris Corner Detector. The original scoring function in Harris Corner Detector was:

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda - 2)^2$$

And Shi-Tomasi proposed:

$$R = min(\lambda_1, \lambda_2)$$

Both methods using the value of R to classify as a corner, edge, or flat. With Shi-Tomasi's modification, the score will only be bigger enough to be considered as a corner when both λ_1 and λ_2 are larger than the threshold, which provides better results in practice.

B APPENDIX: LUCAS-KANADE

Bruce D. Lucas and Takeo Kanade estimated the optical flow using a differential method in 1981 [10]. They assumed that with a slight time increment, the displacement of objects among the two consecutive images is very limited. Furthermore, they also assumed that the displacement of all pixels within a certain window is consistent. From the figure 2, the intensity of the same pixel in both images should be consistent meaning:

$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

By applying Taylor Series Approximation, and let $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, the following equation will be derived:

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

Let $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, and $I_t = \frac{\partial I}{\partial t}$, the above formula could be converted into:

$$I_x u + I_y v = -I_t$$

By definition, pixels with a certain window should have consistent displacement, so $\forall p_i \in P$, we have:

$$I_x(p_i)u + I_y(p_i)v = -I_t(p_i)$$

The value of u, v could be calculated by solving the above equation with multiple pixels following the least squares principle.