

# Lucas-Kanade Optical Flow Machine Learning Implementations

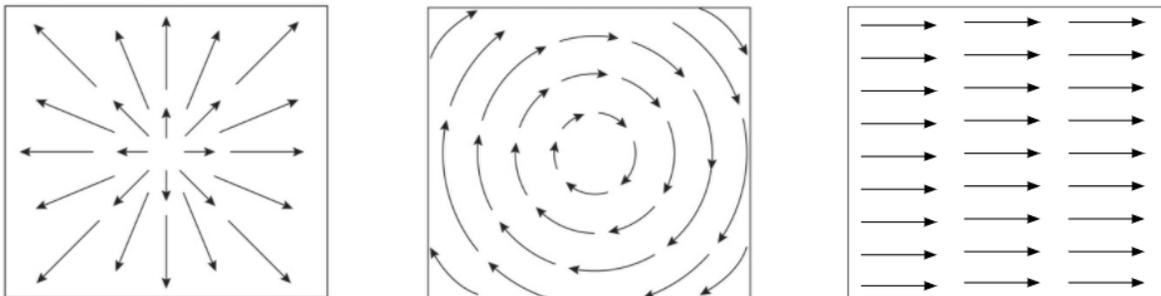
Vedant Gaur<sup>1</sup>

<sup>1</sup>Aragon High School

## ABSTRACT

Optical flow is an effective measurement to gauge motion in a scene, which allows for the computation of pixel-by-pixel motion in a frame pair. This paper aims to address the ambiguity with determining how to gain optical flow results for a given sequence. Due to varying speeds and nuances of a sequence, where it's set, how fast it's moving, a different amount of blur radius, i.e., the extent to which the image is blurred, may have to be applied to gain realistic flow maps. Furthermore, this paper touches on the many variables that can impact the efficacy of the flow outputted by an optical flow algorithm. Thus, we aim to determine whether the composition of results obtained through different blur values provides for more ground-truth flow outputs.

## 1. Introduction



**Figure 1.** A diagram of the primary types of motion that are described through optical flow visualizations (from left to right: forward-backward, rotational, translational).

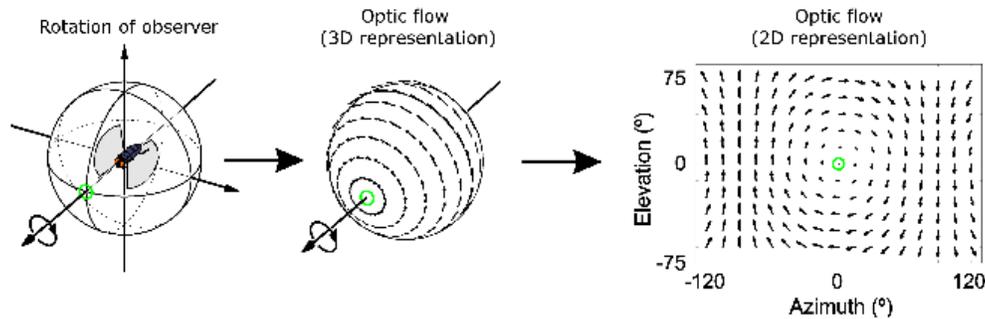
A necessity in being able to understand, and consequently compute optical flow, is a foundational understanding of how perspective works, as well as how perspective shifts impact motion in a scene. Computation on sequences can provide for a plethora of information as to what is happening in a scene without even seeing it. Optical flow has been one of these methods that provides insights into the movement in a sequence, what things are moving, and to what extent are they moving. Optical flow is similarly useful in determining specific objects in a sequence, like edge, or object detection. A simple camera shake can be enough to highlight each of the objects in a scene and provide information in that way. However, traditional methods of computing optical flow are fairly unilateral and don't account for nuances in sequences such as the speed of objects. The motivation of this paper was to look into the usage of blurs and see whether implementations could bolster optical flow measurements.

The computation of the movement that occurs in a sequence of events refers to the process of determining the optical flow in a frame. The algorithm determines patterns of surface movement in the given frame that depends on the perspective of the viewer of the scene, whether that be a shift in the perspective or a shift in the position of surfaces themselves [1, 2, 3, 23, 27]. As the object or perspective shifts, flow vectors are computed to replicate movement on

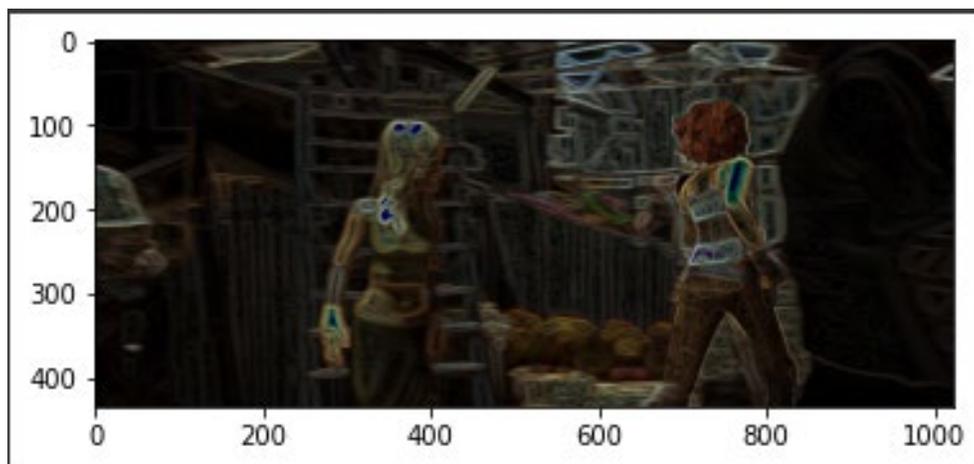
screen. An optical flow estimation is provided through the sequencing of various points in the scene to determine holistic surface movement. When implementing optical flow, there are three main types of movement that can be accurately described. How far or close an object is getting from the viewer, the direction of rotation, and the x-y translation. Depending on the output flow field, one can determine the motion of the object [1, 2, 23].

### 1.1. Optical Flow

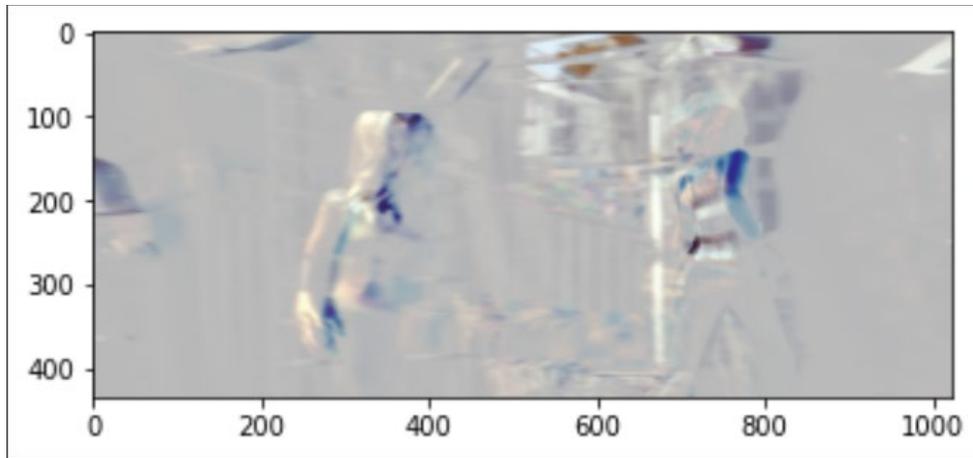
This paper primarily uses optical flow algorithms to calculate flow vectors at each frame in each sequence. Optical flow is the determination of movement through the gamma values, or brightness levels, at each pixel in the frame [1, 3, 23, 27]. An important distinction when considering optical flow is that the optical flow field outputted is the apparent motion in the scene. Because optical flow relies on the motion described by the viewer of the scene, these flow values may not always directly coincide with true motion. If the perspective of the viewer is shifting while objects in the scene remain motionless, the flow field that is outputted will still display motion in the surfaces as the perspective of the person shifts. However, this is useful for determining relative movement. The flow vectors will differ based on the distance from the viewer, allowing optical flow to be used for determining how close or far an object or surface is. Optical flow provides the 3-dimensional motion as it is projected onto an image plane [1, 27].



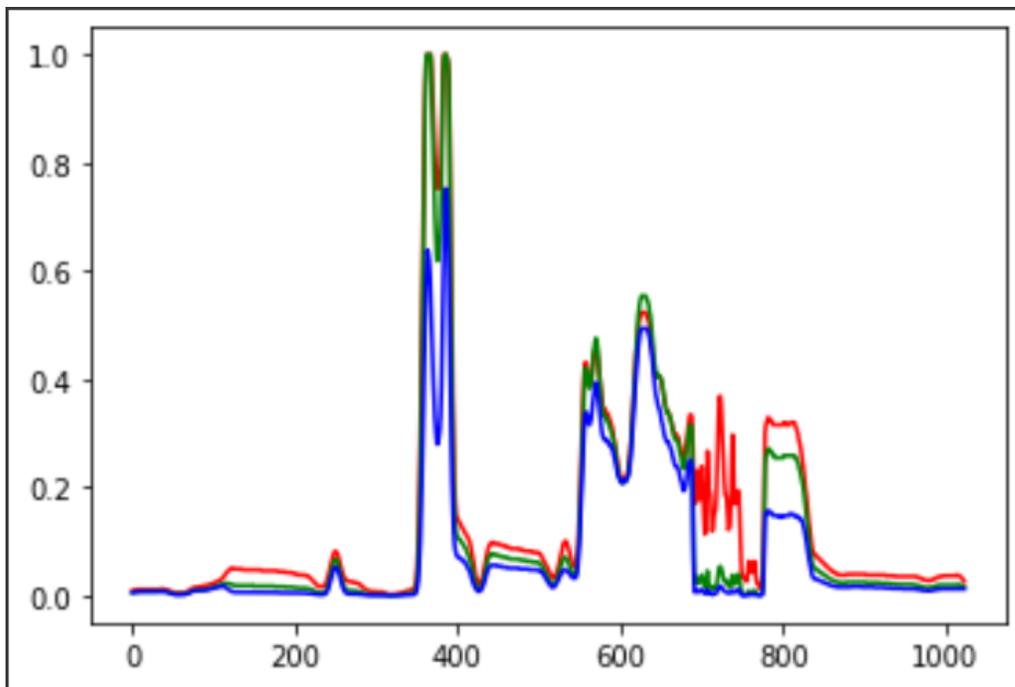
**Figure 2.** The diagram represents the outputted flow field if the user itself is viewing a stagnant scene while rotating. The magnitude and direction of the vectors represents the perception of motion of surfaces as well as the main point of reference, in this case, the center of the scene [28].



**Figure 3.** Gamma corrected sRGB color space visualization that represents luminance of a frame in the Sintel dataset.



**Figure 4.** Visualization of the change in luminance between a frame pair in the Sintel Dataset.



**Figure 5.** Chart representing the luminance of a single slice of a frame at each color channel. The higher spikes represent higher brightness at that pixel.

### 1.1.1. Explanation

Optical Flow relies on the relative movement of surfaces in a sequence, computed through the aforementioned luminance values at each pixel. Consequently, this means that, when computing optical flow, it must be assumed that the brightness is constant in the scene. If such is not true, optical flow will not succeed [2, 23]. Optical flow computation, in popular algorithms like the Lucas-Kanade method is done with two primary calculations: the spatial and temporal derivatives. Each pixel is given the location  $(x, y, t)$ , where  $x$  and  $y$  represents the x and y-position s, respectively, and  $t$  represents the time, or frame of the image. The intensity, or brightness of the pixel can similarly be modeled with  $I(x, y, t)$ . As the pixel moves between frames, the intensity can be equated to the change in both x, y, and t positions:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

when expanded via a Taylor series expansion:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t$$

meaning that:

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

similarly, the equation can also be stated as:

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0$$

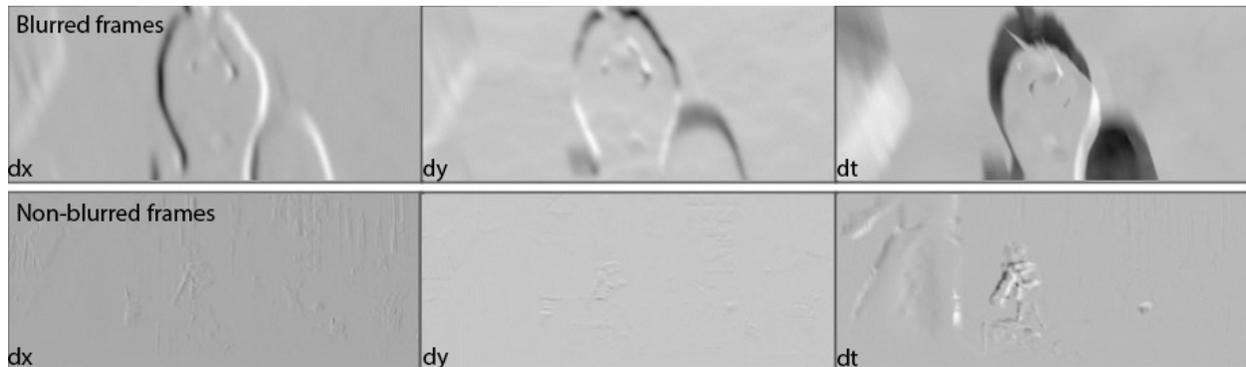
Where  $V_x$  and  $V_y$  are the components of the flow vector of the pixel  $(x, y, t)$ . If we call the flow components  $u$  and  $v$  respectively we can derive the equation:

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

or similarly:

$$\nabla I \cdot \vec{V} = -I_t$$

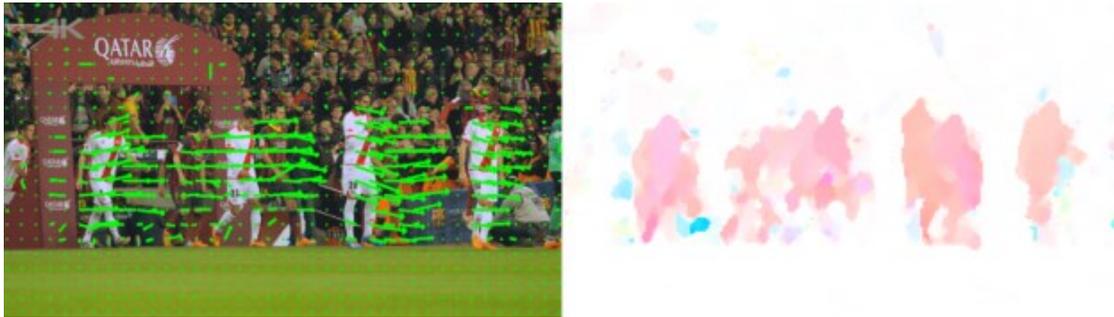
Where  $\vec{V}$  is our unknown. The issue with traditional optical flow stems from this presence of two unknowns, a phenomenon known as the aperture problem which presents ambiguity in the motion of an object in a sequence [23]. To work around this, we can implement various methods to compute optical flow. In the case of this paper, the Lucas-Kanade method is implemented, which makes some assumptions when determining flow vectors.



**Figure 6.** The images above refer to a series of luminance derivatives computed on a pair of images in a sequence. The top three images refer to a sequence which has been blurred then had the derivatives extracted, while the bottom three are computations of the original sequence, without any blur.

For both, the first image represents the horizontal spatial derivative, the second represents the vertical spatial direction, and the third represents the temporal derivative. As is evidenced by comparing the two, blurring can majorly impact the amount of motion that is discernable to both a viewer as well as the algorithm. Because the motion is less precise as a result of the blurring, it is much easier to make out the amount of movement in the first set of images than the second. We can also analyze specific frame pairs for specific, directional movement. In the second set of images, because there is more visible in the visualization for the horizontal derivative than the vertical one, the assumption can be made that in the pair of frames, there was more horizontal motion than there was vertical. This analysis is very important in being able to understand the sequence and how objects move in it.

### 1.1.2. Application



**Figure 7.** An optical flow field visualization of soccer players walking. The field describes the motion of each player, and the individual vectors correlate to the motion of each of the pixels in the scene.

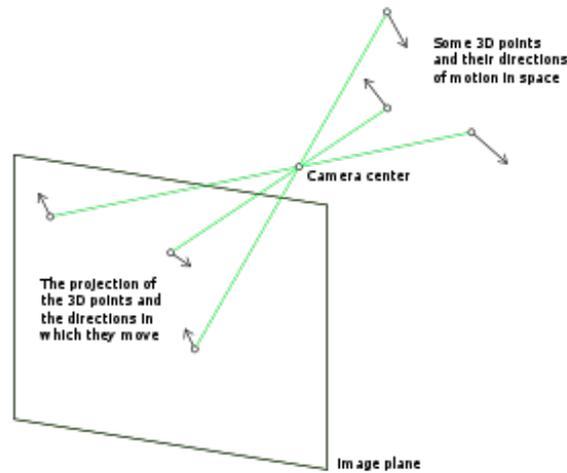
Optical flow has presented itself as a fundamental concept in a variety of fields. At a very base level, optical flow can be used to reflect movements of a computer mouse onto a screen. By using a very low quality, very high frame-rate camera, optical flow can be run on the input stream and determine the motion of the mouse on its surface. This motion can then correspond to mouse movements on the screen, along with assumptions about rigidity, lighting, and object segmentation.

Optical flow can be used for more large-scale tasks such as both object detection, as well as object velocity detection. By running optical flow on a sequence, because the entirety of the surface of an object moves in unison, the individual objects can be segmented and then extracted. If one knows the dimensions of a scene as well as a camera's relative distance from the scene, optical flow measurements can be converted into 3-dimensional movement, and the speed of an object can be calculated. This can be useful for applications such as monitoring traffic on a street, where optical flow can determine the speed of vehicles [25,27].

A perhaps more involved application, optical flow can be applied to provide “missing” frames in a video sequence. Optical flow algorithms can produce interpolated frames between the given frames in the sequence of a finite frame-rate camera. By providing these intermediate frames, the number of frames of a high-speed camera can be emulated without having to use the intensive hardware that is conventionally required to obtain similar results. This can be used to smooth out choppy videos, or to simulate slow-motion videos of these high-frame rate cameras. Optical flow can be used for the stabilization of videos by looking for the “dominant flow” in the sequence. By focusing on this flow and eliminating any unnecessary flow that results from shaking, the video sequence can be made much more stable [25, 27].

## 1.2. Motion Fields

Motion fields are the 2D representation of an object with a free range of three dimensions in the real world. The motion of the point is reflected based on the movement in the real world. Motion fields can effectively replicate both translation and transformation onto the 2D plane, although not exactly a true representation of actual events due to dimensional constraints. Both vertical and horizontal translation are quite easily discernable as the size of the object stays constant and the motion field can easily reflect such movement. All vectors that relate to the movement of the surface will maintain both direction and magnitude as all of the pixels that represent the surface essentially move with it. The other case would be the transformation of the object relative to the focal point (i.e., the object getting bigger or smaller based on the perspective of the camera). Rotational motion similarly outputs both vectors tangent to the edges of the surface, as well as internal vectors with slightly less degrees of magnitude in order to distinguish the motion from simply translational movement [14, 17, 20].



**Figure 8.** A representation of a motion field in which points  $(x_1, y_1, z_1)$  in 3D are projected to corresponding  $(x_2, y_2)$  coordinates on a 2D plane.

The motion field can be defined through the equation:

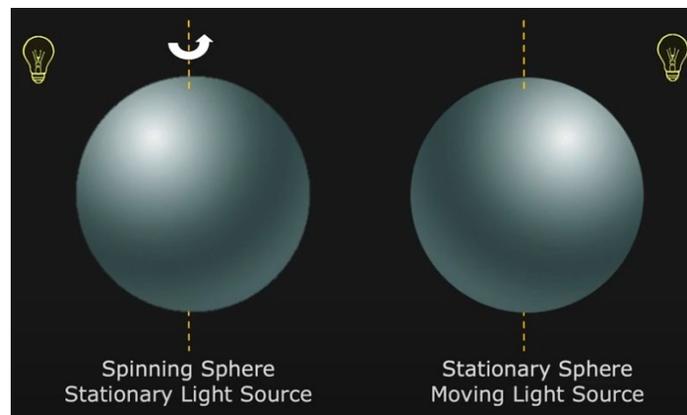
$$\mathbf{v} = f \frac{Z\mathbf{V} - V_z\mathbf{P}}{Z^2}$$

given that:

$$\mathbf{V} = -\mathbf{T} - \boldsymbol{\omega} \times \mathbf{P}$$

Where  $\mathbf{P}$  represents a point in the scene being described,  $Z$  is the distance from the camera to the point,  $\mathbf{V}$  is the relative motion of the point,  $\mathbf{T}$  is the translational motion, and  $\boldsymbol{\omega}$  represents the angular velocity [ ].

Optical flow and motion fields both represent movement in a 2D plane, and while the apparent extracted motion (in this case optical flow) can coincide directly with true motion (i.e., motion field representations), there are several use cases that present themselves in which optical flow and motion field results fail to be the same. While optical flow and motion fields both yield similar intentions and results, the difference lies in true versus the perceived motion [14, 17].

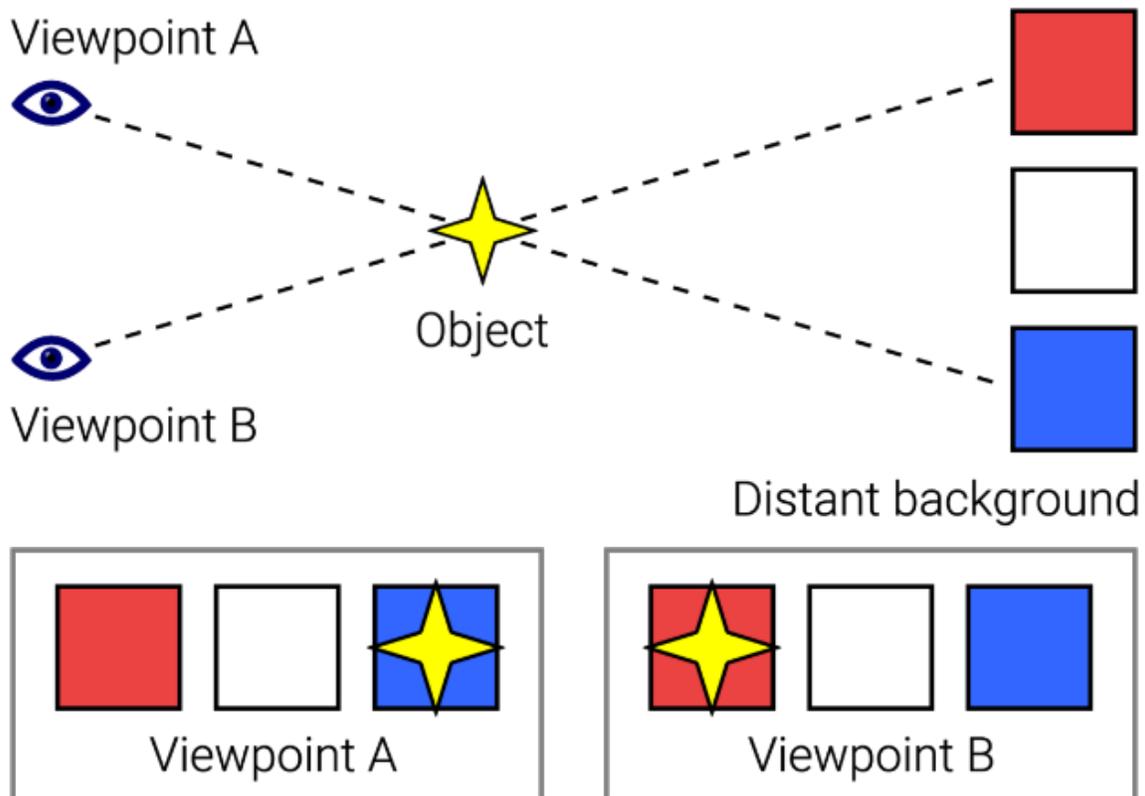


**Figure 9.** In this example, the optical flow results and motion field results are not the same. In the first case, although the ball is moving, since such movement is not evident through the appearance of the ball, there is a zero optical flow field. In the second instance, while there is no true motion of the ball, since the light source moves and changes its appearance, optical flow produces flow vectors.

### 1.3. Parallax

Parallax refers to the perceived motion of objects in a scene as perspective shifts. A common phenomenon in space sciences, a shift in viewer position leads to apparent motion of objects in a scene. It can also be determined that the closer an object is, the faster it will move as the position changes. Since closer objects are larger as a result of depth perception, it is easier for them to cross a viewer's entire field of motion than a farther object which would appear smaller and consequently take longer to move across the field of view [20]. This idea can be implemented in optical flow result extrapolation to determine close and far objects.

Parallax is extremely important regarding optical flow as it shifts the definition of movement in a scene slightly. If a camera were to be pointed at a scene and then shaken around, the sequences would show that the objects moved from the frame of reference of the camera, when in reality, it was the camera that moved. This is both useful and hindering in the calculation process. While it can be useful to determine where objects are in a scene, the optical flow measurements may not truly reflect whether an object is moving or not.

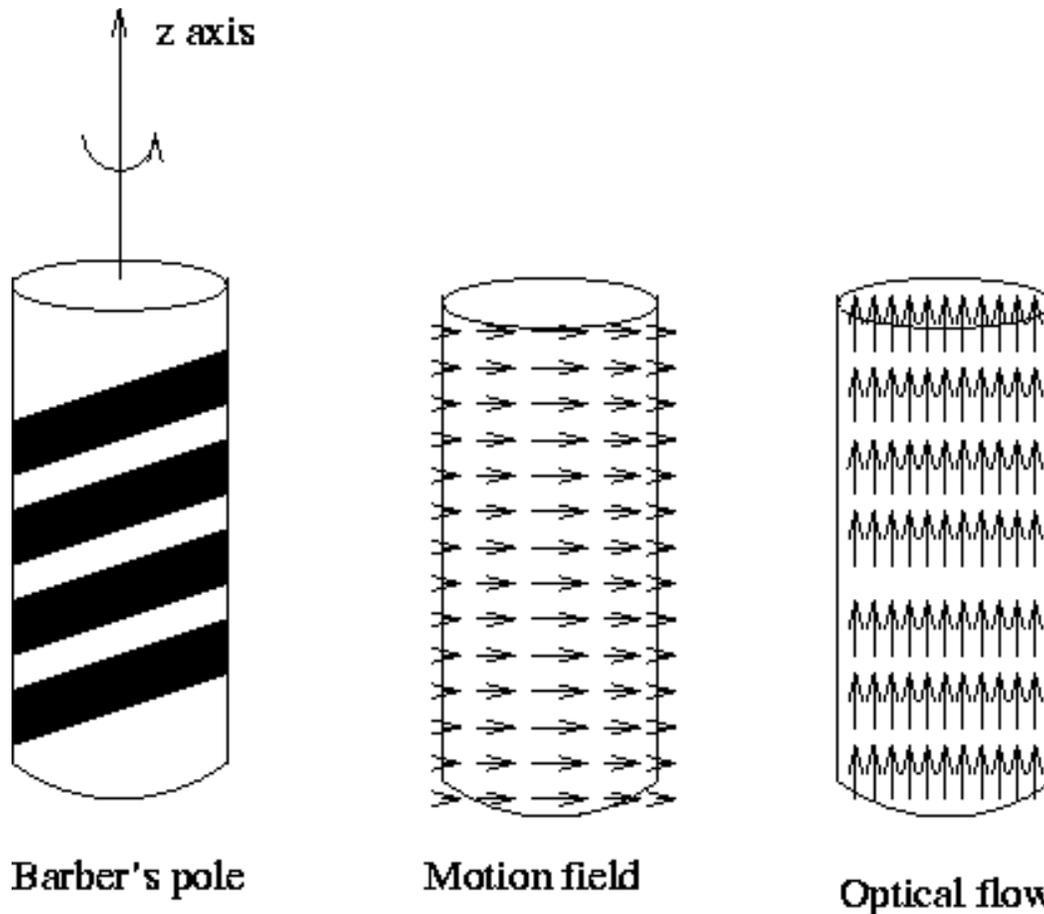


**Figure 10.** As the viewer moves from Viewpoint A to Viewpoint B, the displayed image shifts. At point A, the yellow star appears to be in front of the blue square in the background, yet changes to the red square when the perspective shifts to position B.

### 1.4. The Aperture Problem

The aperture problem presents the idea that object motion is not always completely discernable without being able to see the whole scene that the object is moving in [33]. An example of this is enclosed rectangles with the ends covered up. Because the viewer is not able to see the entirety of the rectangles, thus the entire scene, true motion is not able to

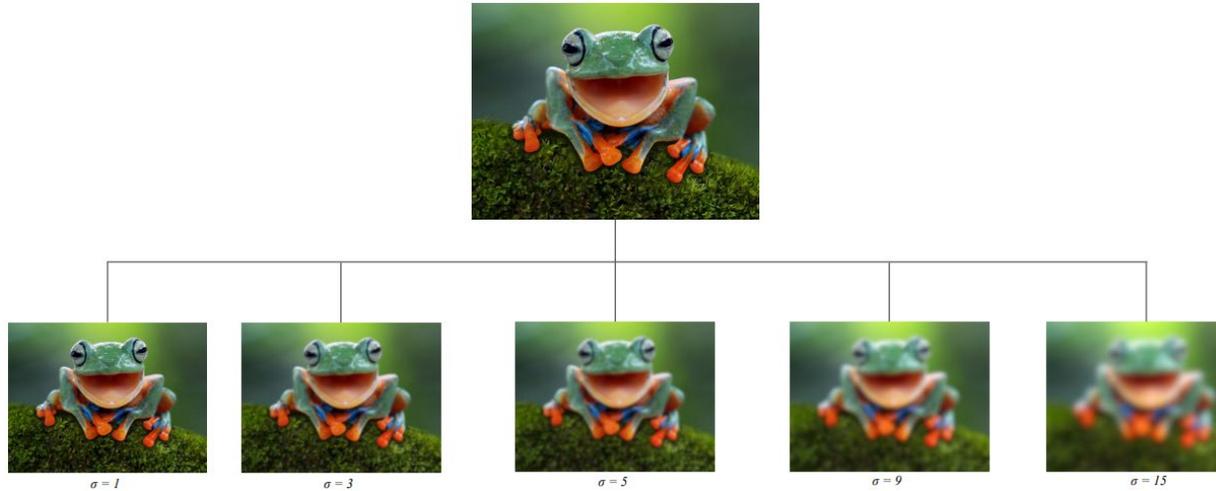
be extrapolated. A real-life example of this phenomenon is barber poles which appear to have a vertical up-and-down motion, while true motion is horizontal, derived from rotation.



**Figure 11.** A barber pole rotates along the z-axis and creates the illusion that the stripes are moving up to a bystander. This is reflected in the optical flow field which describes the motion as upward due to the luminance values being perceived to have moved in such a matter. In truth, the pole is rotated and thus the motion field is horizontal [33].

### 1.5. MPI-Sintel Dataset

The MPI-Sintel dataset is a dataset created to evaluate optical flow algorithms. The dataset provides naturalistic video sequences that are challenging for current methods and is designed to encourage research on long-range motion, motion blur, multi-frame analysis, non-rigid motion. Traditional evaluation methods were largely one-dimensional and were easy to perform well in. The Sintel dataset addresses this issue through sequences that include a wide range of textures and motions. The dataset was used to evaluate regression models and determine error in this paper [21]. The dataset is particularly useful as well because it is completely synthetic, therefore the true optical flow measurements are accessible for developers and researchers. The dataset consists of sequences in the open-source film Sintel, which is completely 3D animated.



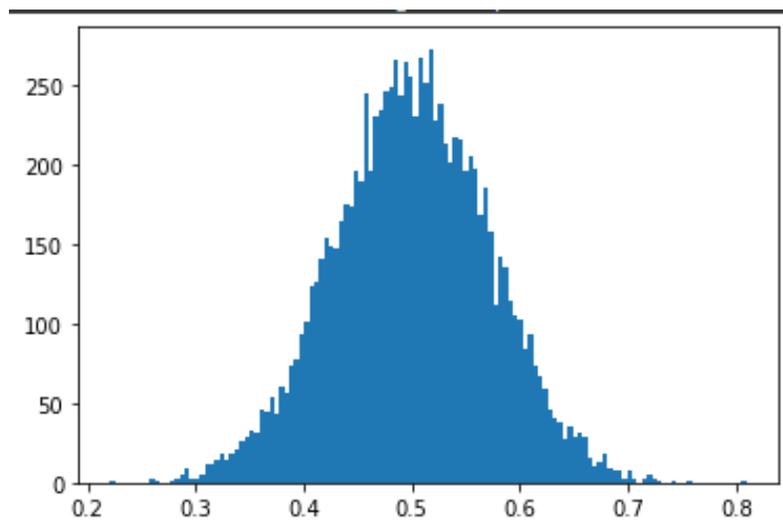
**Figure 12.** Demonstration of different sigma values (left to right: 1, 3, 5, 9, 15) for the blur radius of the gaussian blur function.

## 2. Necessary Tools

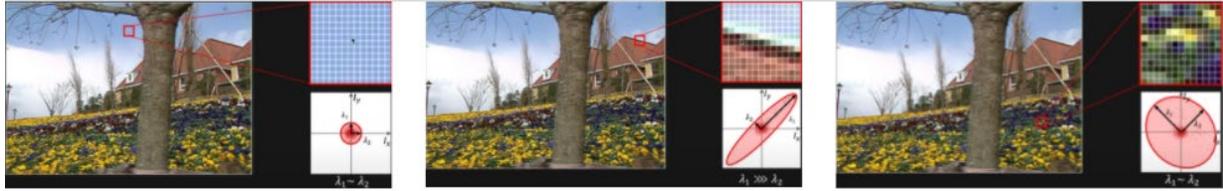
In order to successfully understand how the process of blurring works, there are a few fundamental theorems that are needed to provide the basis of implementation.

### 2.1. Central Limit Theorem

The central limit theorem (CLT) states that for any distribution, the mean of various samples of the set will be distributed following a normal distribution, or more commonly a bell curve. As either the number of samples increases, or the size of the samples themselves increase, the closer the sample means will be to a normal distribution [28]. Via a mathematical representation, if  $x_1, x_2, x_3, \dots, x_n$  is a given set with  $n$  random samples with mean  $\mu$ , variance  $\sigma^2$ , and a sample mean  $\bar{X}_n$ , then  $Z = \lim_{n \rightarrow \infty} \sqrt{n} \left( \frac{\bar{X}_n - \mu}{\sigma} \right)$  is a standard normal distribution.



**Figure 13.** A representation of a normal distribution through plotting random sample means.



**Figure 14.** Varying textures with a computation of Lucas-Kanade. The smoothed and edged patches perform significantly worse than the textured patch.

## 2.2. Gaussian Blur

Gaussian blur is a method that utilizes a Gaussian function to blur, or smooth, an image. A key advantage of applying a gaussian filter to images is to reduce the noise that may be present [4, 8]. This noise can make computation inefficient and produce unclear flow results when passed through an optical flow algorithm. By denoising the frames, we can extrapolate much more ground-truth flow results that corroborate with the motion in the scene. Furthermore, blurring is also useful when down sampling an image as it can avoid the aliasing of the image. The filter runs a weighted kernel (i.e., a convolutional matrix) that multiplies each pixel in the window by the value that corresponds to the pixel position in the kernel [4]. A weighted average is taken of all of the weighted values and applied to the pixel at the center of the kernel. This process is repeated for each pixel and a new matrix is created that outputs these modified values [4, 8]. In one dimension, the Gaussian function can be mathematically defined as:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

and in two dimensions, the function is just the product of two single-dimension functions:

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$x$  being the horizontal distance from the origin,  $y$  being the vertical distance, and  $\sigma$  being the standard deviation of the Gaussian distribution.

## 3. Lucas-Kanade Optical Flow

The Lucas-Kanade method [5] of computing optical flow addresses the issue of an under constrained optical flow by implementing a primary assumption. The algorithm assumes that the neighboring pixels around a single pixel in a scene will have the same optical flow values as each other, albeit a very small set of neighboring pixels [5, 10, 12, 13, 24]. We can call this “patch” of pixels  $W$  and use such to solve for the optical flow. The Lucas-Kanade method also relies on two separate assumptions. One, the base optical flow assumption that the brightness constancy is satisfied, and two, that the motion displayed in the path/sequence is relatively small. The Lucas-Kanade method is an example of sparse optical flow, in which only the flow at select pixels in the frame are calculated [10, 12, 24]. Other optical flow algorithms, which implement a denser method of optical flow in which the flow at all points in the frame, or a majority of pixels, is computed. Sparse optical flow is beneficial in that it provides for quick computations and allows for a much faster generation of training data. Lucas-Kanade can also be used to compute dense optical flow, however.



**Figure 15.** The red square is a representation of a patch  $W$  used for the Lucas-Kanade method.

### 3.1. History

The Lucas-Kanade method was determined by researchers Bruce D. Lucas and Takeo Kanade in 1981. Although almost 40 years have gone by since that point, the algorithm is still widely referenced and subsequently built upon by newer proposals. Dr. Takeo Kanade currently remains a professor at Carnegie Mellon University and continues work in computer vision and its applications [31].

### 3.2. Explanation

According to the Lucas-Kanade method, for all pixels  $(l, k) \in W: I_x(l, k)u + I_y(l, k)v = -I_t(l, k)$  represented through a matrix, assuming the size of  $W$  is  $n \times n$ :

$$\begin{bmatrix} I_x(1, 1) & I_y(1, 1) \\ I_x(1, 2) & I_y(1, 2) \\ \vdots & \vdots \\ I_x(n, n) & I_y(n, n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_t(1, 1) \\ -I_t(1, 2) \\ \vdots \\ -I_t(n, n) \end{bmatrix}$$

giving us a set of  $n \times n$  equations, or simply if:

$$A = \begin{bmatrix} I_x(1, 1) & I_y(1, 1) \\ I_x(1, 2) & I_y(1, 2) \\ \vdots & \vdots \\ I_x(n, n) & I_y(n, n) \end{bmatrix}, V = \begin{bmatrix} u \\ v \end{bmatrix}, B = \begin{bmatrix} -I_t(1, 1) \\ -I_t(1, 2) \\ \vdots \\ -I_t(n, n) \end{bmatrix}$$

the matrix equation can be written as:

$$AV = B$$

to solve the linear system, we can implement the least squares principle, which allows for determination of the best way to fit a regressive curve to data points. The method aims to minimize the square of the residuals:

$$A^T AV = A^T B$$

in matrix form:

$$\begin{bmatrix} \sum_w I_x(l, k)^2 & \sum_w I_x(l, k)I_y(l, k) \\ \sum_w I_x(l, k)I_y(l, k) & \sum_w I_y(l, k)^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum_w I_x(l, k)I_t(l, k) \\ -\sum_w I_y(l, k)I_t(l, k) \end{bmatrix}$$

or simply:

$$V = (A^T A)^{-1} A^T B$$

This estimation only holds when the following arguments are true:

- $A^T A$  must be invertible (i.e.,  $\det(A^T A) \neq 0$ )
- The eigenvalues of  $A^T A$  must follow the conditions:
  - $\lambda_1 > \epsilon$  and  $\lambda_2 > \epsilon$
  - $\lambda_1 > \lambda_2$  but not  $\lambda_1 \gg \epsilon$

### 3.3. Analysis

The reason that the Lucas-Kanade algorithm works well is because all the equations derived at each pixel in the patch are not linearly dependent on each other. This statement works when the algorithm is presented with a patch of distinctive texture that allows for  $I_x$  and  $I_y$  to differ between each pixel [21]. In a smoothly textured patch, the equations at each pixel are the same, meaning that the spatial derivatives will be close to 0, as displayed in *Figure 10*. Both eigenvalues are fairly small and thus it is difficult to reliably compute the optical flow. An edged patch presents a sharp gradient in one direction and a weak gradient in the other. This means that the algorithm may not be able to successfully determine the true motion of the surface, much like the aperture problem. Because of this, edges are also quite unreliable when implementing Lucas-Kanade. A more well-conditioned scenario is when the patch is textured (e.g., *Figure 10*). Texture allows for much more diverse, and larger gradients at each pixel, addressing the fallbacks of the other two scenarios [21].

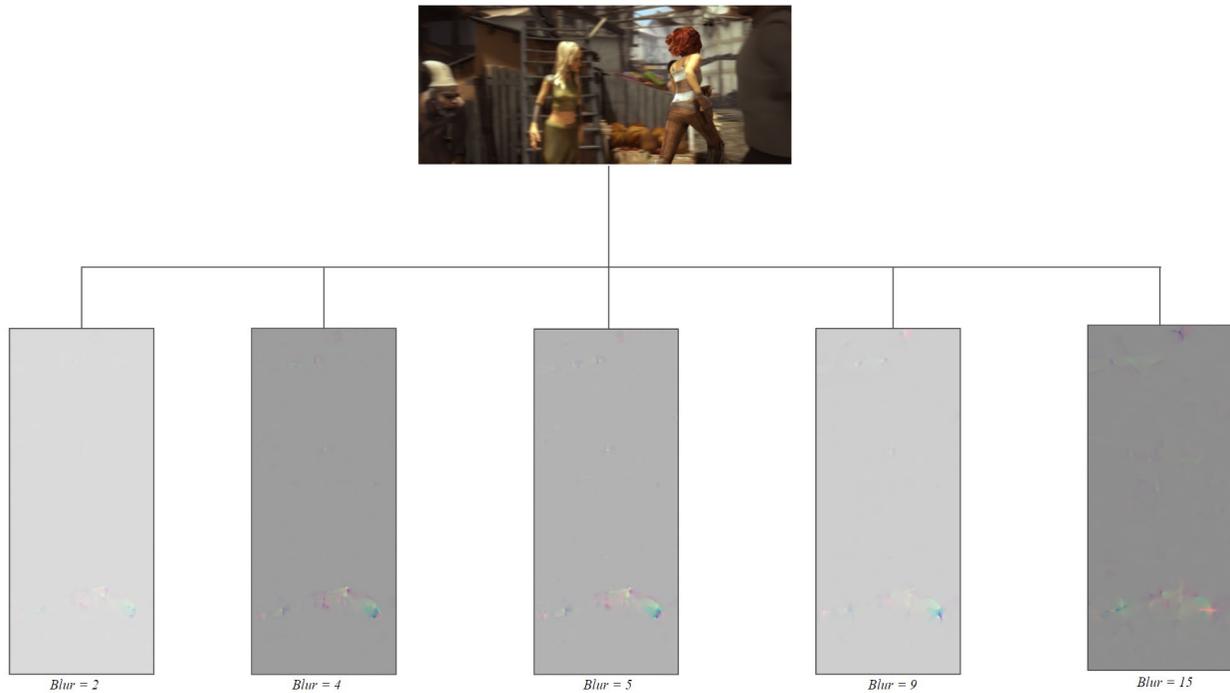
## 4. Machine Learning Integration

The basis of evaluation was determining whether a simple regression model would be able to determine the best blur radius, and thus extrapolate the most accurate flow results. Optical flow is estimated at each sequence in the Sintel dataset, iterating through each frame in every sequence. This process is repeated another four times with the sigma value inputted into the Gaussian blur function being the only modified variable. With this, we can compare the flow of the exact same sequence, with only the blur being changed. After obtaining the flow fields of each frame, the flow fields are extracted to compare the efficacy of the results, as well as to train and consequently evaluate a linear regression model that compounds all the optical flow estimations.

While the graph in (Figure 17) does convey that too much blur is detrimental to optical flow measurements, blurring a sequence can be very useful in calculations. With a sequence that has minimal movement, the optical flow measurements may come out extremely small, or may not be conducive at all. By blurring the sequence, the movements are exaggerated and thus it is easier to compute optical flow measurements.

To obtain a wide variety of data that allows for a much more holistic regression model, we can pick a random pixel from random sequences in the dataset. The various sequences that are provided by the dataset are iterated over and one is randomly chosen. Next, a random frame in the chosen sequenced is picked, and a random pixel from the random frame. The consolidated generated flow and true flow obtained by the Sintel dataset are sent to the regression function that trains on 80% of the data and evaluates on the remaining 20%. The mean squared error (MSE) is computed to determine how much more accurate flow results the model was able to produce through an amassment of varying pixels and blur radii.

#### 4.1. Results



**Figure 16.** A representation of how the blur radius impacts the outputted flow field. In the case of this frame, the blur radius of two performed fairly poorly while the blur radii of four, five, and 15 performed well, with the radius of five outputting the clearest results.

The main goal of the research was to determine the answer to the question of how exactly the blur radii of a gaussian blur impact flow results as well as whether the composition of various blur values would allow for better, more accurate flow results through a regression model. As for blur radii, it was found that on average pixel pairs with a higher variance in r, g, and b values saw that a higher blur value produced better flow results. There were some cases in which the distribution of most optimal blur and variation either did not coincide, or completely contradicted this claim, but a majority maintained such distribution. The regression model on average performed very well compared to original calculations. When taking the average of 50 iterations of the regression function running on 1000 datapoints, the regression model performed **17.23** times better than individual calculations with varying blur radii (determined by calculating the root mean squared error [RMSE] of the original calculations with the true flow from Sintel and comparing those to the RMSE of the output of the compiled regression model and the true flow that corresponded in Sintel).

## 4.2. Conclusions

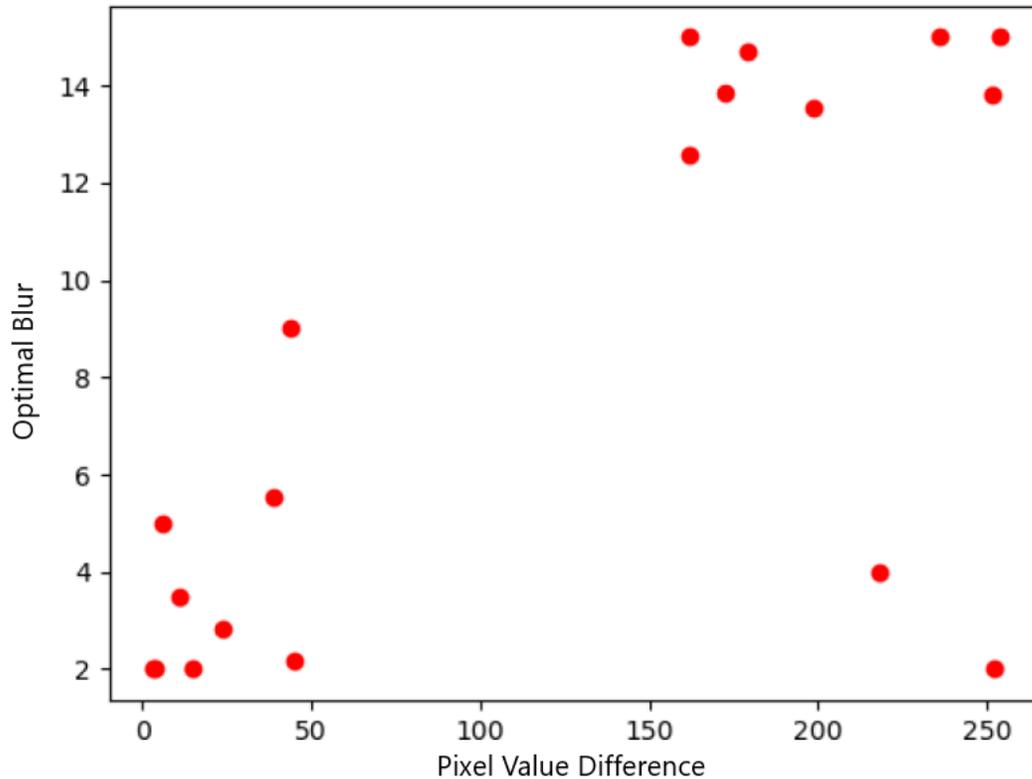
Not surprisingly, the regression model outperformed most any single calculations. Because varying blur radii have varying levels of efficiency, a pseudo-average of these results in the form of a linear model was able to produce more robust results. There were several cases in which the calculated flow did outperform of the regression model. This is most likely because the regression is simply the line of best fit. Although on average it will do better than a single calculated flow due to the convergence of various results, there may be times when a calculation with the most optimal blur radius may outperform, or even severely underperform. A fallback that may have impacted the regression fitting is that some of the calculated flows did not coincide with the true flow in the Sintel dataset. These outliers may have made it so that the fitting of the regression line would be off from the optimal fitting as it would have to consider these skewed datapoints.

The relationship between the speed of the sequence as well as the blur radius also produced fairly clear results. If a linear regression line were to be fit to the data, it would describe an equation with positive slope and y-intercept.

This interpretation of how blur coincides with the computation of optical flow is significant in that it gives an idea about how to tackle various sequences. There is no single type of motion in both the real world, as well as computer-generated sequences or animations. Being able to determine that the more variance in pixel values, and therefore faster motion is computed better after blurred allows us to apply a close-to-optimal blur value in order to produce robust flow results that more directly model the actual motion in the scene.

Pixel	RMSE (Pixel-True)	RMSE (Prediction-True)
1	304.67	55.31
2	59.45	33.82
3	483.65	4.18
4	105.29	30.21
5	14.85	122.55
6	18.61	18.88
7	71.95	15.84
8	26.43	20.53
9	208.73	27.14
10	41.28	14.605
11	59.84	21.11
12	42.38	12.42
13	52.55	30.49
14	29.13	13.27
15	127.43	7.08
16	120.43	31.95
17	163.27	9.19
18	178.02	17.67
19	272.73	5.26
20	26.4	13.54
21	78.005	19.74
22	33.34	107.401
23	122.31	34.96
24	29.41	12.43
25	68.38	17.99
26	110.65	24.75
27	75.87	12.98
28	32.63	71.48
29	3831.11	11.02
30	166.46	20.55
31	359.43	44.57
32	27.05	481.88
33	80.47	29.301
34	453.59	14.57
35	24.88	42.78
36	72.67	7.11
37	234.59	15.43
38	27.96	21.66
39	108.28	13.66
40	175.98	6.91
41	19.102	19.006
42	108.56	7.87
43	47.86	16.04
44	35.46	14.42
45	699.305	8.41
46	80.59	8.36
47	40.45	17.21
48	114.77	61.97
49	30.02	241.904
50	39.24	9.88
AVG	194.71024	38.42574

**Figure 17.** Table of the root mean squared error (RMSE) of both the original flow results as well as the prediction outputs.



**Figure 18.** A scatterplot representation that shows the distribution of the relationship between the total r, g, and b difference between two pixel pairs and their optimal blur. We see that when the difference is lower, the optimal blur is generally closer to 0 while, conversely when it is high, the optimal blur tends to be higher.

## 5. Prospective Implementations

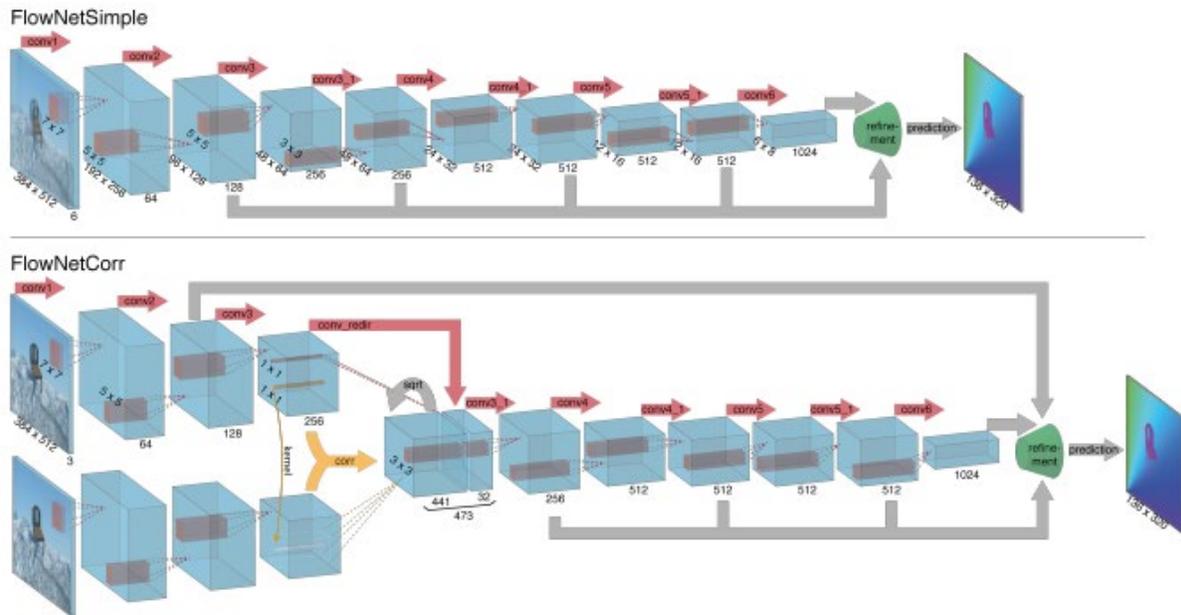
Aside from the variables focused on in this paper, there are many that can be isolated and modified to determine their impact on better flow results such as different forms of regression, different optical flow algorithms, or other sequence manipulations. Furthermore, more robust flow results can be extrapolated by combining results from various methods of computing optical flow. Because each may have its drawbacks, having a wider range of results may allow for higher accuracy in estimation.

### 5.1. Related Algorithms

While Lucas-Kanade is an extremely useful method of computing optical flow, it is not the sole algorithm, or one that produces the most effective results. Because the algorithm computes sparse optical flow, exact flow for each pixel may not be able to be extrapolated. Algorithms like the Horn-Schunck method [9] describe the flow field in a different manner. Horn-Schunck is a method of dense optical flow, rather than the sparse one of Lucas-Kanade. The algorithm assigns a displacement vector to each pixel describing motion, like Lucas-Kanade. However, a key difference in the two algorithms is that Horn-Schunck assumes smoothness over an image [6, 9, 26], rather than the neighboring constraint of Lucas-Kanade (on top of the base optical flow constraint of brightness). The method has a key objective function which is manipulated to gain optical flow results:

$$E(u, v) = \int |I_2(p + w) - I_1(p)|^2 + \lambda(|\nabla u|^2 + |\nabla v|^2) dp$$

## 5.2. FlowNet



**Figure 19.** Diagram of the FlowNet CNN architecture. The top represents the simple CNN, and the bottom represents the CNN with an additional correlation layer [7].

FlowNet is a convolutional neural network (CNN) that computes optical flow in an end-to-end training manner. FlowNet works similarly to normal optical flow computation in that it takes in a frame pair and outputs a flow field. However, the architecture to do this is achieved through deep learning. In the use case of a CNN for optical flow, the neural net essentially has to identify the object or feature that is moving in the first frame and locate that same feature in the next frame [7, 30]. The FlowNet model has two architectures to compute the optical flow of a scene, a standard CNN as well as one with a correlation layer that matches the features between the two input images. FlowNet segments the process into two primary parts: the contracting and refining processes [7, 31]. The contracting part of the model represents the features in the image and matches the information from the inputted image. The refinement section converts each pixel of the inputs into an optical flow field representation, i.e., the optical flow prediction. More specifically, the features maps are scaled to their original resolutions and both convolutional and unpooling layers determine pixel-by-pixel flow predictions [7]. The standard CNN version of FlowNet stacks the two frame inputs together and passes it through the convolutional layers. The model with a correlation layer splits the two frames and performs feature extraction on both. These features are combined and looked at together to then determine the flow through further convolutional layers [7, 30]. This combination is preformed through the correlation layer which matches the feature information from the two frames. The neural network was trained through the Middlebury, KITTI, Sintel, and Flying Chairs dataset. The model performs extremely well on these datasets, namely the Flying Chairs set, although struggles in others [7].

FlowNet is a powerful means of predicting optical flow for a sequence. Its dense flow field predictions are able to generalize very well on datasets like Sintel and generally provide very ground-truth flow results. However, the downside to FlowNet, as with any neural architecture, is that it is very specialized. FlowNet will almost certainly outperform most optical flow algorithms on datasets that it has been trained on e.g., Flying Chairs. However, because of this specialization, it does not perform as well in other scenarios. Algorithms like Lucas-Kanade don't learn, and therefore have a general method in computing optical flow. Although when compared to trained models like FlowNet,

the neural models will perform better in specific scenarios, the general application of more basic algorithms may provide for better flow fields in others.

FlowNet could be implemented alongside algorithms like Lucas-Kanade in order to gain more robust, well-rounded flow results. Similar to the implementation in this paper of stitching together various optical flow estimations to produce better flow estimations, a combination of both neural model predictions as well as basic algorithmic predictions may account for disparities in either FlowNet or Lucas-Kanade.

## 6. Acknowledgements

This paper was made possible due to the immense support and guidance of my research mentor, Mason McGill. A lot of the concepts that pertained to the implementation of the research were new to me or at least somewhat unfamiliar and I was able to utilize Mr. McGill as a great resource to successfully undertake such tasks. The Horizon Academic Research Program (HARP) also aided in providing resources for the research to take place.

## 7. Resources

### Data Availability

Website of the MPI-Sintel dataset used for optical flow algorithm evaluation: <http://sintel.is.tue.mpg.de/>  
Download page: <http://sintel.is.tue.mpg.de/downloads>

### Code Availability

GitHub repository maintained by the author of the paper for the code developed as part of this research:  
[https://github.com/vedantgaur/lucas\\_kanade-optical-flow](https://github.com/vedantgaur/lucas_kanade-optical-flow)

## 8. References

- [1] Barron, John L.; Fleet, David J. & Beauchemin, Steven (1994). "Performance of optical flow techniques" (PDF). *International Journal of Computer Vision*. 12: 43–77. CiteSeerX 10.1.1.173.481. doi:10.1007/bf01420984. S2CID 1290100.
- [2] Beauchemin, S. S.; Barron, J. L. (1995). "The computation of optical flow". *ACM Computing Surveys*. ACM New York, USA. 27 (3): 433–466. doi:10.1145/212094.212141. S2CID 1334552.
- [3] Fleet, David J.; Weiss, Yair (2006). "Optical Flow Estimation" (PDF). In Paragios, Nikos; Chen, Yunmei; Faugeras, Olivier D. (eds.). *Handbook of Mathematical Models in Computer Vision*. Springer. pp. 237–257. ISBN 978-0-387-26371-7.
- [4] 3. *The Gaussian Kernel - University of Wisconsin–Madison*.  
<http://pages.stat.wisc.edu/~mchung/teaching/MIA/reading/diffusion.gaussian.kernel.pdf>.
- [5] B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop*, pages 121--130
- [6] B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185–203, 1981.

- [7] Fischer, Philipp, et al. "FlowNet: Learning Optical Flow with Convolutional Networks." *ArXiv.org*, 4 May 2015, <https://arxiv.org/abs/1504.06852>.
- [8] "Gaussian Blur." *Gaussian Blur - an Overview | ScienceDirect Topics*, <https://www.sciencedirect.com/topics/engineering/gaussian-blur>.
- [9] Horn, Berthold K.P.; Schunck, Brian G. (August 1981). "Determining optical flow" (PDF). *Artificial Intelligence*. 17 (1–3): 185–203. doi:10.1016/0004-3702(81)90024-2. hdl:1721.1/6337.
- [10] *Lecture 30: Video Tracking: Lucas-Kanade*. <http://www.cse.psu.edu/~rtc12/CSE486/lecture30.pdf>
- [11] Lin, Chuan-en. "Introduction to Motion Estimation with Optical Flow." *AI & Machine Learning Blog*, AI & Machine Learning Blog, 24 Apr. 2020, <https://nanonets.com/blog/optical-flow/>.
- [12] *Lucas-Kanade in a Nutshell - Fu-Berlin.de*. [http://www.inf.fu-berlin.de/inst/ag-ki/rojas\\_home/documents/tutorials/Lucas-Kanade2.pdf](http://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/documents/tutorials/Lucas-Kanade2.pdf).
- [13] *Lucas-Kanade Optical Flow - Carnegie Mellon School of ...* <http://www.cs.cmu.edu/~16385/s15/lectures/Lecture21.pdf>.
- [14] *Motion and Form Tutorial*, <http://web.mit.edu/persci/demos/Motion&Form/master.html>.
- [15] *Motion and Optical Flow - Electrical Engineering and ...* [https://web.eecs.umich.edu/~jjcorso/t/598F14/files/lecture\\_1015\\_motion.pdf](https://web.eecs.umich.edu/~jjcorso/t/598F14/files/lecture_1015_motion.pdf).
- [16] *Motion and Optical Flow - Homes.cs.washington.edu*. [https://homes.cs.washington.edu/~shapiro/EE596/notes/Optical\\_Flow.pdf](https://homes.cs.washington.edu/~shapiro/EE596/notes/Optical_Flow.pdf).
- [17] *Motion Fields - University of Washington*. <https://grail.cs.washington.edu/projects/motion-fields/motion-fields.pdf>.
- [18] *Motion Fields Are Hardly Ever Ambiguous - People.csail.mit.edu*.
- [19] [https://people.csail.mit.edu/bkph/papers/Motion\\_Fields-OPT.pdf](https://people.csail.mit.edu/bkph/papers/Motion_Fields-OPT.pdf).
- [20] *Moving Camera Lecture 22: Camera Motion*. [http://www.cse.psu.edu/~rtc12/CSE486/lecture22\\_6pp.pdf](http://www.cse.psu.edu/~rtc12/CSE486/lecture22_6pp.pdf).
- [21] *MPI Sintel Dataset*. <http://sintel.is.tue.mpg.de/>.
- [22] *Optic Flow*, <https://isle.hanover.edu/Ch07DepthSize/Ch07Optic%20Flow.html>.
- [23] *Optical Flow Estimation - Department of Computer Science ...* <https://www.cs.toronto.edu/~fleet/research/Papers/flowChapter05.pdf>.
- [24] *Optical Flow Measurement Using Lucas ... - Ijcaonline.org*. <https://research.ijcaonline.org/volume61/number10/pxc3884611.pdf>.

[25] “Optical Flow.” *Optical Flow - an Overview* | *ScienceDirect Topics*,  
<https://www.sciencedirect.com/topics/engineering/optical-flow#:~:text=4.1%20Optical%20Flow-,Optical%20flow%20is%20defined%20as%20the%20apparent%20motion%20of%20individual,projected%20onto%20the%20image%20plane.>

[26] *Optical Flow: Horn-Schunck*. [http://www.cs.cmu.edu/~16385/s17/Slides/14.3\\_OF\\_\\_HornSchunck.pdf](http://www.cs.cmu.edu/~16385/s17/Slides/14.3_OF__HornSchunck.pdf).

[27] *Optical Flow: Techniques and Applications*.  
<https://www.dgp.toronto.edu/~donovan/stabilization/opticalflow.pdf>.

[28] “Optical Flow.” *OpenCV*, [https://docs.opencv.org/3.4/d4/dee/tutorial\\_optical\\_flow.html](https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html).

[29] “The Role of Probability.” *Central Limit Theorem*, [https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704\\_probability/BS704\\_Probability12.html#:~:text=The%20central%20limit%20theorem%20states,will%20be%20approximately%20normally%20distributed.](https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_probability/BS704_Probability12.html#:~:text=The%20central%20limit%20theorem%20states,will%20be%20approximately%20normally%20distributed.)

[30] *State Estimation Using Optical ... - University of Florida*.  
[https://mae.ufl.edu/ricklind/rick\\_pro/of/GNC06\\_OF.pdf](https://mae.ufl.edu/ricklind/rick_pro/of/GNC06_OF.pdf).

[31] Synced. “A Brief Review of FlowNet.” *Medium*, Towards Data Science, 11 Sept. 2017,  
<https://towardsdatascience.com/a-brief-review-of-flownet-dca6bd574de0>.

[32] *Takeo Kanade - The Robotics Institute Carnegie Mellon ...* <https://www.ri.cmu.edu/ri-faculty/takeo-kanade/>.

[33] *University of Alberta Dictionary of Cognitive Science: Aperture Problem*,  
[http://www.bcp.psych.ualberta.ca/~mike/Pearl\\_Street/Dictionary/contents/A/aperture.html](http://www.bcp.psych.ualberta.ca/~mike/Pearl_Street/Dictionary/contents/A/aperture.html).