The Crowdflow Pipeline: Filtering Crowdsourced Responses to create High Quality Optical Flow Annotations

Johan Ospina
Princeton Univeristy
johanos@princeton.edu

Abstract

Datasets for training Deep Learning Models to solve the Optical flow task have been for the most part created in synthetic rendering environments. While these methods are able to mass produce accurate training data for learning algorithms, they are limited by the quality of the rendering algorithm used and the variation of the library of 3D models. Our contribution is a crowdsourced sparse optical flow dataset, which contains point correspondences for a wide variety of real life scenes.

1. Acknowledgements

We would like to acknowledge Lahav Lipson, Henry Tang, Zeyu Ma, and Jia Deng. Without their contributions this work would not have been possible.

2. Introduction

Optical Flow is the task of estimating the per pixel 2D motion from one frame to another. It is still an open problem which is being actively researched. Currently, the best performing methods use Deep Learning architectures [17, 10] that perform much better than classical approaches such as [13]. In order to compare these various methods, quantitative benchmarks have been released [9, 4, 1, 14] to measure various metrics. In spite of their biases, these datasets have proven useful for training and evaluating state of the art methods. To close the gap further on Optical Flow evaluation, we propose the Crowdflow pipeline that will be used to collect high quality, crowdsourced, optical flow annotations from real life scenes.

In this paper we will describe the Crowdflow pruning pipeline. First, we describe how the sequences were collected from YouTube. Next, we show how we generated the points to be annotated by workers. Finally, we explain how to filter worker proposals to a final, empirically supported, low error annotation. All together this will allow us to create an optical flow dataset with low eucledian error matches.

3. Related Work

Historically, the earliest dataset for Optical Flow was introduced by Barron et al,[2] in 1994. It consisted of a mixture of simple real sequences whose optical flow was collected in a controlled way and computer generated sequences of simple 2D functions. As optical flow methods evolved, the need for more robust evaluation metrics led to the creation of the Middlebury dataset [1]. This dataset was a step up from the previous one since it provided more sequences from both real life and synthetic scenes. This dataset was made before the deep learning era began, therefore, the number of sequences is still small. There is 8 different scenes with public ground truth optical flow fieds defined for 2 out of 8 frames in each sequence. As we entered the Deep Learning era following the groundbreaking performance of AlexNet [11], the need for bigger datasets became apparent.

This need was fulfilled by MPI Sintel which created a pipeline to create dense optical flow fields using synthetic rendering environments. This dataset was able to make close to 600 optical flow fields for a open source movie. Around the same time the KITTI benchmark was released for self driving car applications [9]. However, in KITTI, the focus was more on Visual Odometry applications. This dataset included 389 50% dense Optical Flow fields, where on average one had access to 50% of the flow vectors in an image.

While these datasets were big for the time, they were still not as massive as the ImageNet benchmark for object classification [8]. The biggest dataset for optical flow that is the closest comparison to ImageNet to our knowledge so far is the Flying Things dataset. This Dataset consists of over 22,872 optical image pairs with optical flow fields. This daset was created in similar fashion to MPI Sintel but used randomized models and trajectories.

All of these datasets are limited by the medium used to generate the data [4]. In the case of synthetic data, these methods are limited by the availability of 3D meshes and the quality the of the rendering algorithms used. Moreover,

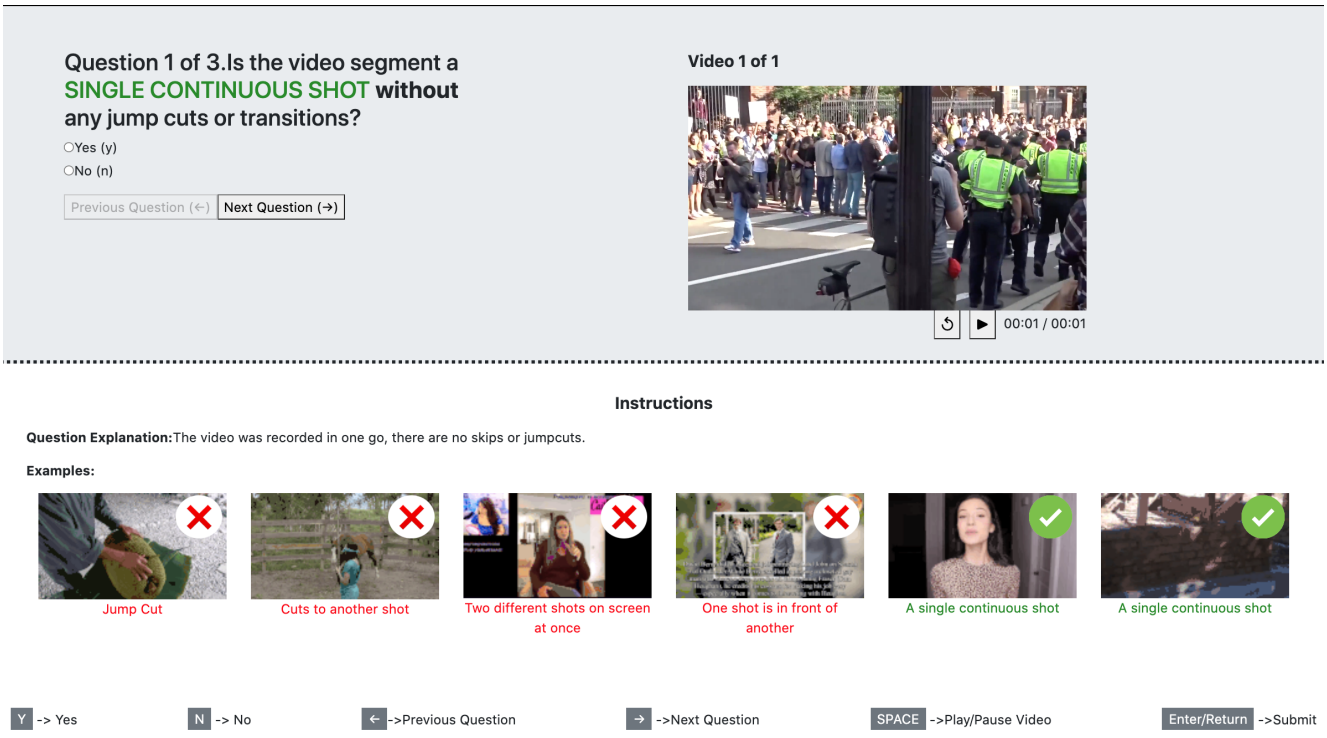

Figure 1: Final Video UI shown to AMT Workers

the methods used to collect real data use tricks or expensive processing setups to generate good flow between images [2, 9]. While the data collected is good, the process of collecting data at a scale comparable to the Flying Things dataset will become unfeasible if one wants to capture different types of relationships and environments.

In order to capture more complicated real life relationships as well as provide a better metric for the quality of current Optical Flow method, we propose the Crowdflow pipeline, a pruning method to be used in creating a real life dataset that will contain sparse optical flow matches for real life squences.

4. Motivation

To motivate Crowdflow, we analyzed potential biases in the popular Optical Flow datasets mentioned previously. The closest current works we are aware of are Middlebury and KITTI both of which were collected in controlled environments. For example, in the Middlebury dataset, the authors used flourescent textures on the items in the scene in order to match the optical flow vectors as they moved the scene around. Additionally, in the case of the KITTI dataset, the Optical Flow data was collected using a laser scanning setup on top of a vehicle.

For our analysis we sampled the angle and magnitude of 600 random flow vectors sampled from each of the various sequences in each dataset. This sampling operation showed that Middlebury and KITTI have bias embedded within them as can be seen in Appendix figures 10, 11, 14, 15. Specifically, this sampling operation showed that Middlebury is biased towards small displacements while KITTI has a more evenly distributed spread past the 0-1 pixel region. Additionally, both of these datasets are heavily biased towards left and right turns (90/270 degree angle flow vectors). This is unsurprising given the fact that Middlebury was collected by moving physical scenes around [1] and KITTI was collected in a medium where the predominant movement directions are either straight ahead or left and right turns [9]. As the push for autonomous driving continues, using these datasets as a validation tool for pretrained deep learning models might give the illusion of good performance if failure mode examples pertaining to the above biases are not present.

In this same vein, when looking at Synthetic Datsets like MPI Sintel or Flying Things there is a more even distribution of angles and magnitudes. However, it is important to note that these datasets have taken architectural steps to both collect and package their data in a simple way. For example both in MPI Sintel and Flying Things, the authors actively avoided reflective and transparent materials to maintain a 1:1 structure between their flow vectors [9, 19]. Fi-

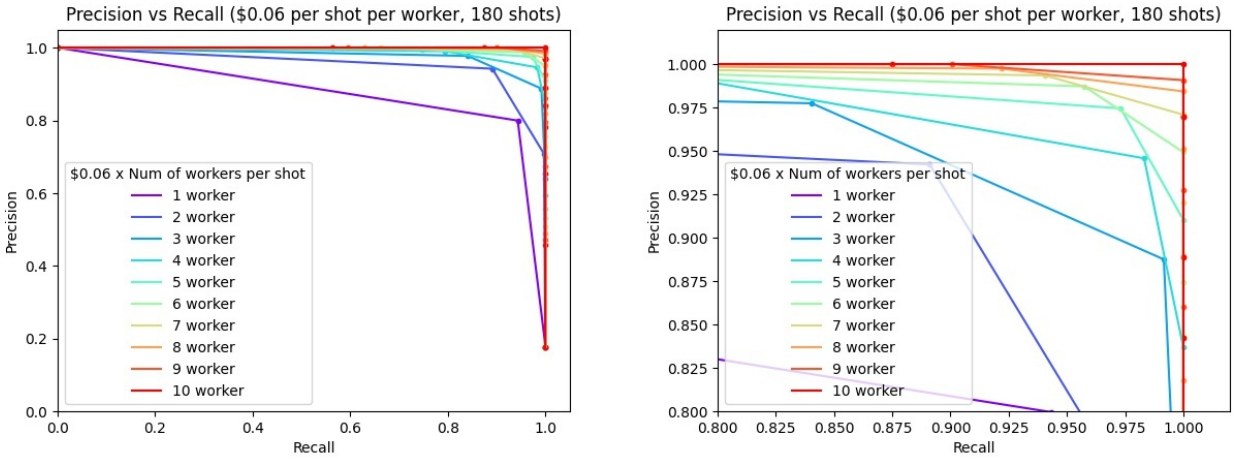

Figure 2: Precision and Recall Curve for Video Classification Results and it's zoomed in version.

nally, a defining limitation in a synthetic environment is the number models available to the creator, it is an intractable problem to generate imagery that would capure the variation of objects found in the real world. Crowdflow dataset will not be aimed to compete with these easily mass producable Optical Flow datasets, but rather it will attempt to fill in the gaps that these methods cannot account for. The open problems in creating the Crowdflow dataset is figuring out where the scenes will come form and what method to use to fitler worker responses down to a good, accurate match.

5. Gathering Videos

The first challenge in creating Crowdflow was where to find video frames that would be useful for optical flow calculations. To solve this problem, we decided to crowdsource sequences from workers on Amazon Mechanical Turk (AMT). The input to this first crowdsourcing task came from using a list of the most popular 6000 english nouns created by [16]. From this list, we used the youtube-dl program [3] to create an initial library of YouTube videos by saving the 10 most popular Creative Commons videos which were 4 minutes or less. Using YouTube allowed us to tap into a vast collection of already existing scenes. This collection of YouTube videos was then processed into random 1 second long clips to help with throughput in our crowdsourcing step. Once we filtered the videos to this point, we had a library of videos that would be good for feeind into optical flow algorithsm

figure 18 in the appedix shows the distribution of the video dimensions for the current number of outstanding videos as of writing this document. As it can be seen, most of the videos are in the 16:9 aspect ratio family.

5.1. Filtering Videos

The next step in our pipeline was to formulate a crowdsourcing task to filter video collection down to videos that had the following qualities.

- Continous shots with no jump cuts.
- Real Life shots
- Active movement

We chose the aforementioned properties as they would allow us to sample random consecutive frames from a video without worrying that the annotation task would be illdefined. Since we wanted to capture qualities of real life scenes, removing 3D CGI and animations would allow us to fit this niche well.

Finally, by asking if there is active movement we are able to avoid potential annotation tasks that were too easy. In order to validate this crowdsourcing task we manually annotated 180 videos where 32 were good and 148 were bad before sending them to workers on AMT. Our results show that this method is effective, we were able to acheive (95 %) precision when using more than 3/4 workers to classify each video as can be seen in figure 2.

6. Creating Annotation Tasks and Filtering For Accuracy

As results from AMT workers came in, we created annotation tasks by sampling 2 consecutive random frames from each video. We used the frame rate (fps) of the video to ensure a frame spacing s of at least 200 milliseconds between frames. Ensuring this spacing is just a simple equation

$$s = \frac{fps}{1000} \times 200$$

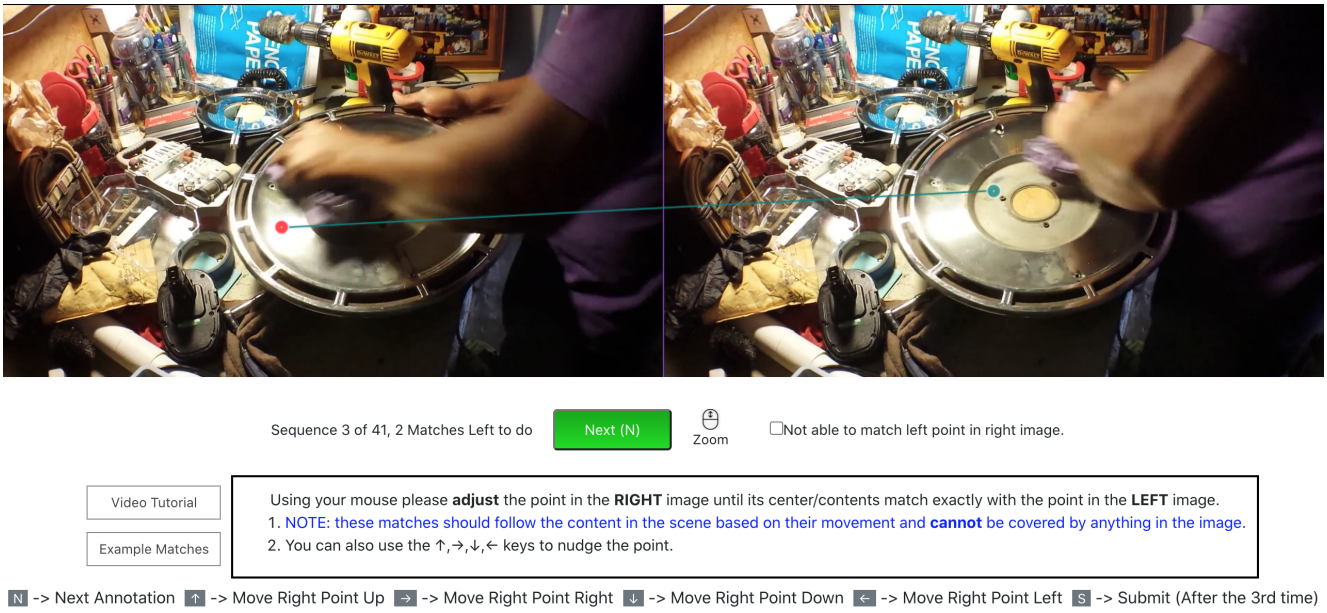

Figure 3: Final Annotation UI Shown to AMT Workers

Let T be the total number of frames in a video, then all one needs to so is sample the video in the range $[0, s-1]$ to get the first frame. Then sample the video in the range $[s, T-1]$. In the event $T < s$. Then we just choose frame 0 and $T-1$.

With crowdsourcing tasks, in order to mitigate errors and get good results. It is good to cross reference the result of all workers for one response [5]. This is also true for this work. Before mobilizing to collect data on real videos from our workers, we needed to also verify how good annotations from human annotators could be. As such we collected empirical data on worst case conditions. Using MPI Sintel we were able to measure the annotation error from its ground truth correspondence between frames. Therefore we manually chose sequences that had a high chance to contain subjectively hard to match points (see figure 19 in the Appendix). It is important to note that this is only possible in MPI Sintel when the point trajectories are not occluded so we took an iterative randomized approach to generating these points. Essentially, we checked if a random point was occluded from one frame to the next. We would then follow this point's trajectory for the next 2 frames and return the last known unoccluded correspondence between the sequence of frames. This was done to vary the baseline between sequences somewhat.

Based on the empirical results obtained from this experiments as well as others that will be mentioned later, we developed the following pruning method to filter worker annotations into high quality optical flow annotations. This pipeline can be broken into two parts, Intraworker pruning and Interworker pruning.

6.1. Intraworker pruning

One can think of Intraworker pruning as looking at how well the worker agrees with their own annotation. Equivalenty, we can think of them as a dart thrower we are betting on. One is less likely to bet on this dart thrower if they cannot hit the bullseye consistently.

Formally we can define this in the scope of our annotation task by asking the worker to do the following. Let A denote an initial point in the left frame of an annotation task, then let B, C denote two different worker attempted matches in the right frame. Finally let $\rightarrow$ denote the act of matching a point between 2 frames. Using this definition figure 4 gives a good visual description of the task given to workers. Externally, the worker is just told to match the points between a left and right images. Internally, we change the order of the frames and initial left point based on their response from the previous stage.

By laying out the task like this, we were able to look at two metrics that make up our intraworker pruning scheme. The first is the Forward Back Error (FB error) between the worker's match of point $A \rightarrow B$ and $B \rightarrow A$. We can judge the worker's ability to match points in the specific image by measuring their FB error. Since the task they are given is to find a correspondence for point A in the left image. Their proposed annotation B should be close to where the match exists. By asking the worker to match point B in the original left frame we are actually seeing if the worker is able to successfully match points in the scene they are

Figure 5: Example of worker results, their observations from B when compared to point the initial point A(cyan) is allows us to compare their Forward Back error (FB error) as well as their intraworker consistency with themselves

working on.

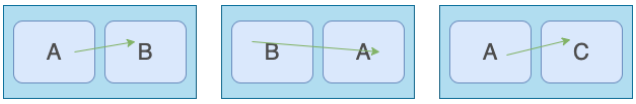

Figure 4: Example of task given to workers, This allows us to compare their Forward Back error (FB error) as well as their intraworker consistency with themselves

In our intial experiments, we found that just measuring FB error was, by itself, not enough to determine if we should accept a worker's annotation. This is due to the fact that sometimes a worker is not correct where the point B show lie but considers that position as the true correspondance for the initial left point. To mitigate this we added a final step where we ask the workers to perform the matching task $A \rightarrow C$. This extra datapoint will allows us to make a decision on whether to accept the workers B and C points for the next step in the pipeline.

With observations B and C at our disposal we can then evaluate how well a worker matched the point by computing the average distance from the centroid of the two observations. This *self consistency* metric should be zero if the worker reliably placed B and C around the same region and scale equal to the distance between the two points if they did not. We found that pruning using these two metrics allowed us to get candidate matches to further process in our pruning pipeline. Figure 6 shows a visual of this stage.

However, we found that these candidate matches were too unreliable if they only came from a single worker. This is due to the fact that there is human error to deal with when one uses crowdsourcing methods. To compensate for this, we found that getting multiple annotation proposals from multiple workers allows us to analyze the interworker statistics in order to prune observations until we arrive at a low error annotation.

6.2. Interworker Pruning

We can formally state the interworker pruning step. Let $\mathcal{P}_w$ be the set of all points collected from one worker. It is straight forward to see that $0 \leq |\mathcal{P}_w| \leq 2$ for all n workers. Therefore the total number of possible worker generated annotation proposals, $\mathcal{P}_{\text{total}}$, is bounded by $0 \leq \mathcal{P}_{\text{total}} \leq 2n$. The Interworker Pruning Step applies 2 analytical techniques on $\mathcal{P}_{\text{total}}$ in order to extract an annotation proposal with less than 2 pixel error on average based on empirical results.

The first technique we propose is an analysis of the average distance from the centroid of all points $p_i \in P_{\text{total}}$. The reason this metric is important is that it captures the disagreement between workers at a high level. If this value is higher than a threshold we have chosen empirically, we can claim that the workers are in agreement. Just using this metric is too restrictive, however, as the lower this threshold value is the more aligned the points would have to be which impacts the percentage of points that we can collect. In order to capture more points, we relaxed this initial threshold value and added a second step to prune using the Geometric Median that would find the 2 closest workers if 3 workers were not in agreement.

6.2.1 Geometric Median

The main intuition behind using the geometric median can be expressed in the one dimensional case. In this mode of operation one can think of the geometric median as staying near the most popular answers. Formally, the geometric median is defined as the point y that minimizes the following

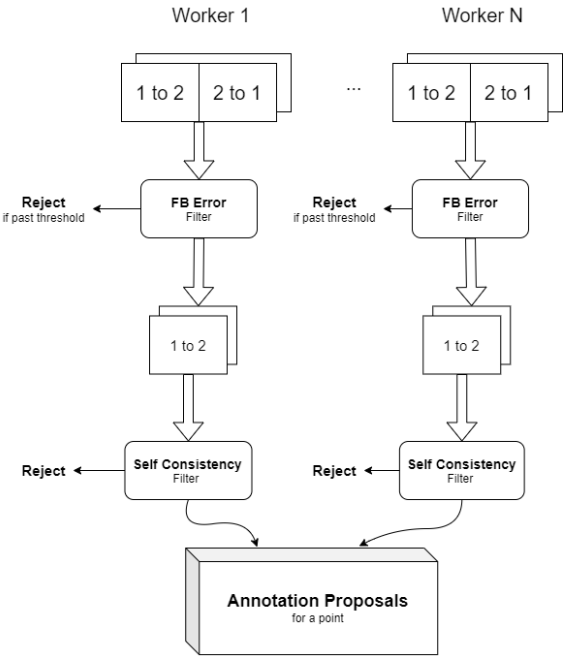

Figure 6: Block diagram of the intraworker stage of the pruning pipeline

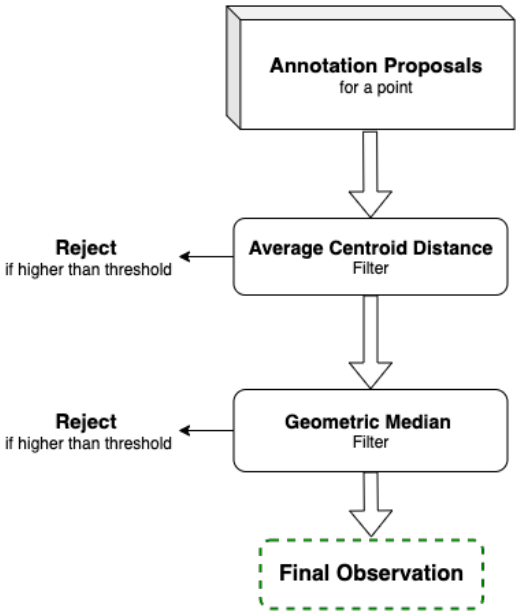

Figure 7: Block diagram of the interworker stage of the pruning pipeline

optimization problem.

$$\arg\min_{y \in \mathbb{R}^n} \sum_{i=1}^{m} \|x_i - y\|_2$$

where y is a point in $\mathbb{R}^n$. This expression is minimizes the sum of distances rather than the sum of squared distances (in the case for the centroid). Additionally a property of the geometric median is that it is less susceptible to outliers when compared with the centroid [7]. While simple to state, the actual computation of the geometric median is more complex as there is no closed form expression. Various methods exist [6, 12] and in this work we used the algorithm proposed by [18].

6.2.2 Worker Distance Metric

As stated previously we found that if 3 or more workers are not in agreement falling back to the 2 closest workers can still yield good results. We define the distance, $d(w_i, w_j)$, between two workers, i and j, as the following:

$$d(w_i, w_j) = \frac{1}{mn} \sum_{k=1}^{m} \sum_{l=1}^{n} \|p_i^k - p_j^l\|_2$$

Where p_i^k is the k^{th} observation for a point by worker i. This is intuitively the average distance between all pairs of points from one worker to another. This quantity is tractable for our purposes because m and n will never be higher than 4 by design and therefore $O(1)$. Additionally, it is a valid distance function since $d(w_i, w_i) = 0$ and $d(w_i, w_j) = d(w_j, w_i)$.

6.3. Putting It All Together

When working with a crowdsourcing system, it is important to realize that the workers want to be correct in order to get their payment. Realizing this means that if the annotation is not confusing to a human annotator, then the points $\mathcal{P}_{\text{total}}$ should all lie closer to the correct match.

This intuition yields the Interworker pruning step. First, as mentioned prior, we compute an initial average distance metric to look at only the annotation proposals that fall below this user-set threshold. We then compute the geometric median and analyze the average distance of all observations to that point. If this value is within an acceptable threshold we consider the geometric median as the final annotation and if it is not we find the 2 closest workers and recompute these metrics on the points collected from those 2 workers.

7. Empirical Results

We now will share the hyperparameters used as well as the empirical results for the pruning method to be used

in creating the crowdflow dataset. As mentioned in previous sections, we first created a set of 100 randomized points from subjectively harder sequences in the MPI Sintel dataset. As such the sequences were ones with various fog effects or plain textured backgrounds (see the ambush/mountain sequences in the MPI Sintel dataset for examples).

To ensure that these points had possible annotations we used the occlusion maps and optical flow maps provided by the authors of MPI Sintel in order to generate a random point and follow its trajectory for the next 2 frames [19]. This was done to ensure more variation in the baselines of the points. If a point was occluded before its end point in the second image, then we would take its match in the first image. We created a pool of these points and then visually sorted them into difficulty buckets based on percieved difficulty.

We then took 50 points from the medium and hard buckets to arrive at 100 points to show AMT workers. Figure 19 in the Appendix shows some of these hard points. We then released tasks to AMT and collected 5 worker reponses per point. For data that uses less than 5 workers, we randomize the workers used per point and report the average value for each metric out of 7 runs.

The empirical results gathered in this section are aimed to answer 2 questions, the first is how accurate can crowdsourced matches be and the second is how many worker responses one actually needs to arrive at high quality responses.

Average Geometric Median Distance	**Yield**	**Average Error**	**Error Std Deviation**
1.0	4%	0.90	0.58
2.0	10%	1.46	1.36
3.0	17%	1.61	1.35
4.0	17%	2.72	3.69
5.0	17%	2.26	2.65
6.0	20%	2.43	2.53
7.0	18%	2.48	2.80

Table 1: Worker Results with only **2** workers with FB error threshold of **6** pixels, and initial average distance threshold of less than **8** pixels. From left to right it shows the yield (percentage of points accepted), average error of the accepted points, and the standard deviation of the ground truth errors for each accepted point.

We first begin with an ablation study to justify the inclusion of the initial *Average Centroid Distance* filtering step in the Interworker pruning step. Tables 1 and 2 show that while there is a negligible impact to the yield (points kept) metric. Rejecting annotation proposals by the average centroid distance empirically allows for lower error metrics.

Average Geometric Median Distance	**Yield**	**Average Error**	**Error Std Deviation**
1.0	4%	0.79	0.3
2.0	11%	1.56	1.69
3.0	14%	1.64	1.54
4.0	17%	2.58	3.41
5.0	18%	2.49	2.85
6.0	22%	2.45	2.96
7.0	20%	2.80	3.36

Table 2: Worker Results with only **2** workers with FB error threshold of **6** pixels, and no initial average distance thresholding step.

Since the geometric median is less sensitive to outliers and we assume that workers will want to do well, we can implicitly ignore, to some degree, problematic annotation proposals that do not agree with the popular responses. From here on the FB error threshold should be assumed to be 6 pixels and the initial distance metric threshold should be assumed to be 8 pixels.

7.1. Falling Back to 2 Workers

During our experiments we also noted that if we require n workers to all pass intraworker pruning, we end up will lower yields with no major decrease to our error metrics. A simple rudimentary calulation can explain why requiring all workers to pass the Intraworker Pruning step means we get less points. Let γ be the probability that a worker passes the Intraworker Pruning step. Then, since these quantities are i.i.d, the probability that all n workers pass intraworker pruning is given by:

$$P[\text{all pass}] = \gamma^n$$

We know that $\gamma < 1$ since we definitely know that γ is not equal to 1. that means this probability will trend to zero as n grows. A byproduct of the above argument is that the probability that at least two workers pass Intraworker pruning is higher that all n workers passing. By adding this fallback component to the pruning pipeline, we are able to significantly increase our yield at the expense of slightly increasing some error metrics when all n worker annotations do not pass Intraworker pruning or Interworker pruning. Tables 3 and 4 show this empirical study when looking at just 3 workers.

Finally, Table 5 shows what happens when we apply the fallback approach to all 5 worker responses we collected for all 100 points. We show that on these harder sequences, 5 workers are able to achieve a 50% yield with less than 2 pixel average ground truth error and tight standard deviation metrics.

Average Geometric Median Distance	Yield	Average Error	Error Std Deviation
1.0	4%	0.95	0.36
2.0	15%	1.58	1.72
3.0	20%	2.04	1.61
4.0	23%	2.43	2.78
5.0	25%	2.41	2.77
6.0	24%	2.74	2.88
7.0	28%	4.15	9.81

Table 3: Worker Results with **3** workers all passing the Intraworker Pruning Step and FB error threshold of **6** pixels, and initial average distance threshold of **8** pixels.

Average Geometric Median Distance	Yield	Average Error	Error Std Deviation
1.0	13%	1.07	1.46
2.0	25%	1.67	2.29
3.0	29%	1.88	2.22
4.0	32%	2.40	3.87
5.0	37%	2.69	3.85
6.0	39%	2.76	3.81
7.0	40%	4.48	12.61

Table 4: Worker Results with **3** workers passing the Intraworker Pruning Step + falling back to the closest **2** workers if they do not pass the Interworker Pipeline Pruning Step with the Geometric Median the first time.

Average Geometric Median Distance	Yield	Average Error	Error Std Deviation
1.0	29%	1.36	1.46
2.0	47%	1.81	2.29
3.0	51%	1.89	2.22
4.0	57%	2.68	3.87
5.0	62%	2.75	3.85
6.0	64%	2.78	3.81
7.0	65%	5.47	21.29

Table 5: Worker Results with 5 workers + fallback using Geometric Median

8. Geometric Median or Geometric Mean

A natural question to ask is whether the Geometric Median is more useful than the Geometric Mean. Here we provide an ablation study aimed to answering this question. Tables 4, 6, 5, 8 show the impact of choosing to use the average distance to Geometric Median over the average distance to the Geometric Mean (a.k.a the Centroid). Recall that the main intuition of using the Geometric Median was that workers want to be right in order to receive their payment. Based on this empirical study, the error metrics are lower when using the Geometric Median instead of the Centroid because more workers are clustering their answers near the right location in the image.

Average Centroid Distance	Yield	Average Error	Error Std Deviation
1.0	15%	2.24	3.57
2.0	23%	2.25	3.54
3.0	29%	2.03	2.41
4.0	33%	2.38	3.47
5.0	36%	2.62	3.52
6.0	39%	2.61	3.18
7.0	41%	3.08	3.88

Table 6: Worker Results with **3** workers + fallback using the **Centroid** for the Interworker Pruning step.

9. Feature Rich Sequences

It is important to note that the results presented in the previous section are for points that were manualy chosen to exemplify worst case conditions for workers. The point of these experiments were to gain an upper bound on the quality of worker matches on the generated points. In the real world there will be many different types of scenes, some easier than others. As such we consider these experiments as a upper bound of how well a worker can match a point following our pruning pipeline. Naturally, we proceeded to create a variant of this initial experiment that used easier sequences.

These easier sequences were made up 39 manually chosen points that were subjectively deemed to be easier to match for a human worker. This means that the points were around feature rich regions with matches that were simple and direct. Our justification for this experiment was to give an lower bound on worker performance in the same way the previous experiment gave an upper bound.

What we found was that workers can indeed do better on these feature rich sequences. When given the same treatment as above the errors and their standard deviation is much lower than their harder point counterparts. We proceed by doing an ablation study that varies the FB error threshold on these harder and easier sequences. As shown in table 7, as we vary the FB threshold, our yield goes down as well as our errors and standard deviations. It is important to note that the easier sequences have higher yield and lower errors for all configurations.

9.1. Actual Data Collection Method

Based on these experiments we argue that for actual data collection one only needs 2 to 3 workers per point. The argument here is that our yield with 3 workers is going to

Experiment Type	Points	Average Error	Error Std Deviation	FB Error
Easy	72%	1.20	0.85	6
Hard	29%	1.93	2.15	6

Experiment Type	Points	Average Error	Error Std Deviation	FB Error
Easy	62%	1.11	0.7	4
Hard	20%	1.49	1.52	4

Experiment Type	Points	Average Error	Error Std Deviation	FB Error
Easy	64%	1.13	0.77	3
Hard	20%	1.54	1.59	3

Experiment Type	Points	Average Error	Error Std Deviation	FB Error
Easy	48%	1.06	0.70	2
Hard	13%	1.09	0.80	2

Table 7: Ablation study varying FB error for results reported for **3** workers showing a comparison between easy and hard points varying the FB error threshold. Note that for the Easy sequences, these numbers stayed the same past 3 workers. Therefore results with 5 workers are not included to save space.

Average Centroid Distance	Yield	Average Error	Error Std Deviation
1.0	42%	2.27	4.01
2.0	50%	2.21	3.75
3.0	53%	2.54	3.95
4.0	56%	2.65	3.84
5.0	62%	3.00	3.85
6.0	64%	3.02	3.81
7.0	64%	3.03	4.02

Table 8: Worker Results with 5 workers + fallback using Centroid

be in the range $[29\%, 72\%]$ and with 5 workers it will be in the range $[51\%, 72\%]$ Consider the following scenario, if we naively pick the best possible yield in these ranges and are only able to collect 1000 points from workers, then this collection could be made up of 3 worker responses per point (334 potential points) or 5 worker responses (200 potential points). It is then important to note that $0.72 \times 334 \approx 240$ points and $0.72 \times 200 \approx 144$, in this case when the yield is similar for 3 or 5 workers. It is better to go with 3 as one can get more unique points. Based on the previous sections, we claim that more unique workers per point is only a benefit when the points are hard to match. While one can throw more workers at the pipeline to get higher yields, it might be better to just release more points and keep the number of workers the same.

10. Adjusting Left Point Experiments

We also explored how accurate workers could be when they were allowed to vary the initial left point in a matching experiment. In order to quantify what this meant we modified our UI so that the worker could move their left point around a radius of 0, 20, and 50 pixels. We then released 50 points to be annotated by AMT workers and collected 5 annotation proposal from each worker using a longer verison of the task shown here 4. Once the worker annotated the first image pair, we then locked in that left position as the initial left point A. Figure 8 show's the adjustable UI that was created to measure worker performance for this experiment.

What we found was that if workers are able to adjust the left point somewhat, it makes them statistically more likely to give single annotations that are low error. Figure 21 in the Appendix, shows the top 200 (80%) results when sorting from lower error to higher error for various square lengths. When the radius is set to 0 pixels, it becomes the aforementioned task used for the previous discussion. Ultimately, this approach was not pursued in order to avoid biasing the dataset and other complications in data collections that would slow down data collection.

11. UX Lessons Learned

Working with crowdsourcing platforms proved to be a non-trivial task during the course of this project. In this section we will touch upon some lessons we learned along the way while building the Crowdflow Pipeline and its user interfaces (shown in 3)

11.1. Train Workers

Initially, we found that the quality of our responses subpar. We realized that due to the nature of the crowdsourcing platform, workers see hundreds of other tasks a day. This means keeping a list of trained workers for one's tasks is a good idea. AMT has a way to give workers special qualifications that we took full advantage of.

First we created 2 separate tutorial tasks for workers to complete, one for the video filtering task, and another for the annotation task. These tutorial tasks served to train the workers and add them to a final workers list if they performed well. Both of our tutorial tasks keep track of how many times it needs to correct a worker while guiding them through the tutorial. We then use this metric to decide to

Figure 8: Left Adjustable UI for a radius of 50 pixels

grant a special qualification to a worker. Once we implemented this system, the quality of our results increased to what we reported in previous sections. As an aside, this also helps the worker as they have a source of steady work for the duration of data collection.

11.2. Pay workers fairly

When working with crowdsourcing platforms, it is important to rememeber that there are people who are working at the other end of the screen. Some of them are passively doing crowdsourcing work while others dedicate themselves full time. As such, they will choose to work on tasks that pay them well [15]. A good rule of thumb is to pay workers the equivalent to minimum wage for the time they spent.

11.3. Give Workers Time

Rejections impact AMT workers in negative ways, as such they will do their best to avoid them. What we found was that if a worker is not given adequate time, they will choose to report that no match is possible for our annotation experiments rather than risk underperforming. Therefore to ease that burden, we timed ourselves doing some annotation tasks and set the time to be twice that amount. At the end of the day, the workers will want to work quickly in order to make more money so putting more time on the clock for them will only help to ease stress that they might not have enough time to submit a good response.

12. Conclusion And Next Steps

We have presented the Crowdflow annotation pruning pipeline. The aim of this pipeline is to create sparse optical flow measurements from real life scenes. We collected Creative Commons YouTube videos that we trimmed and used crowdsourcing to filter to arrive at a collection of videos that would be good for Optical Flow estimation tasks. We then generated a crowdsourcing task that chooses an initial point and takes image pairs from each video to ask AMT workers to annotate the correspondence. We then propose a pruning method and show empirical results that support show that we can arrive at accurate, 1-2 pixel average error annotations. We also show the standard deviation of the ground errors from measurable datasets to show that our annotations will be accurate.

Currently, we are still in the data collection phase. To date, we have collected 3000 points. Initial magnitude and angle statistics can be found in 9.

References

[1] Simon Baker, Daniel Scharstein, J.P. Lewis, Stefan Roth, Michael Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92:1–31, 01 2007. 1, 2

[2] John Barron, David Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 02 1994. 1, 2

[3] Daniel Bolton. youtube-dl. https://github.com/ytdl-org/youtube-dl, 2012. 3

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 1

[5] Jenny J Chen, Natala J Menezes, Adam D Bradley, and T North. Opportunities for crowdsourcing research on amazon mechanical turk. *Interfaces*, 5(3):1, 2011. 4

[6] Michael B. Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 9–21, New York, NY, USA, 2016. Association for Computing Machinery. 6

[7] Keya Rani Das and A.H.M. Rahmatullah Imon. Geometric median and its application in the identification of multiple outliers. *Journal of Applied Statistics*, 41(4):817–831, 2014. 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, page 3354–3361, USA, 2012. IEEE Computer Society. 1, 2

[10] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation, 2021. 1

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 1

[12] Jyh-Han Lin and Jeffrey Scott Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245–249, 1992. 6

[13] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, page 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. 1

[14] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134. 1

[15] Gabriele Paolacci, Jesse Chandler, and Panos Ipeirotis. Running experiments using amazon mechanical turk. *Judgment and Decision Making*, 5:411–419, 08 2010. 10

[16] Desi Quintans. The great noun list. http://www.desiquintans.com/nounlist, 2015. 3

[17] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. 1

[18] Yehuda Vardi and Cun-Hui Zhang. The multivariate l1-median and associated data depth. *Proceedings of the National Academy of Sciences of the United States of America*, 97:1423–6, 03 2000. 6

[19] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black. Lessons and insights from creating a synthetic optical flow benchmark. In A. Fusiello et al. (Eds.), editor, *ECCV Workshop on Unsolved Problems in Optical Flow and Stereo Estimation*, Part II, LNCS 7584, pages 168–177. Springer-Verlag, Oct. 2012. 2, 7

13. Appendix

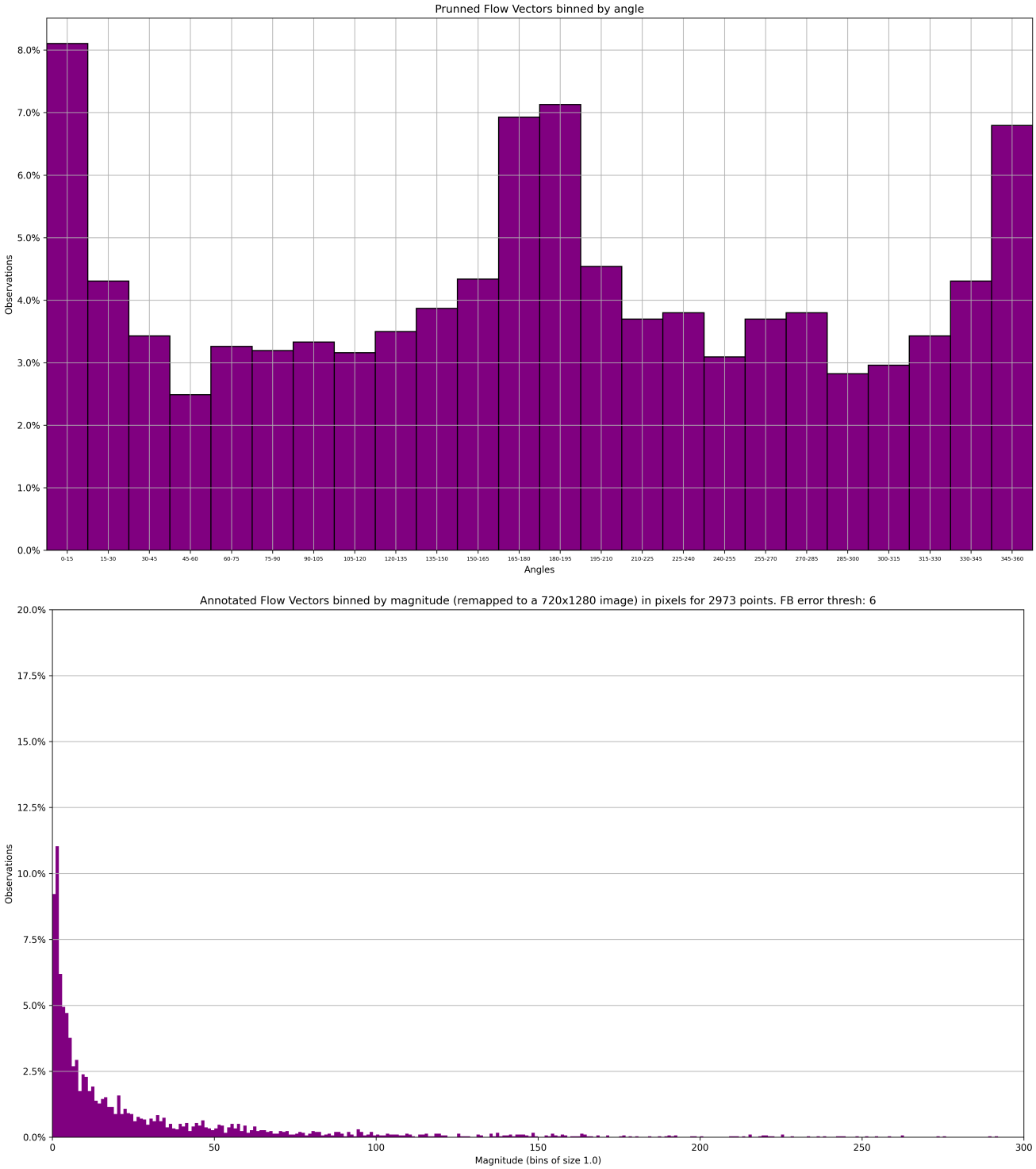

Figure 9: Current Crowd Flow Annotation magnitudes and angles

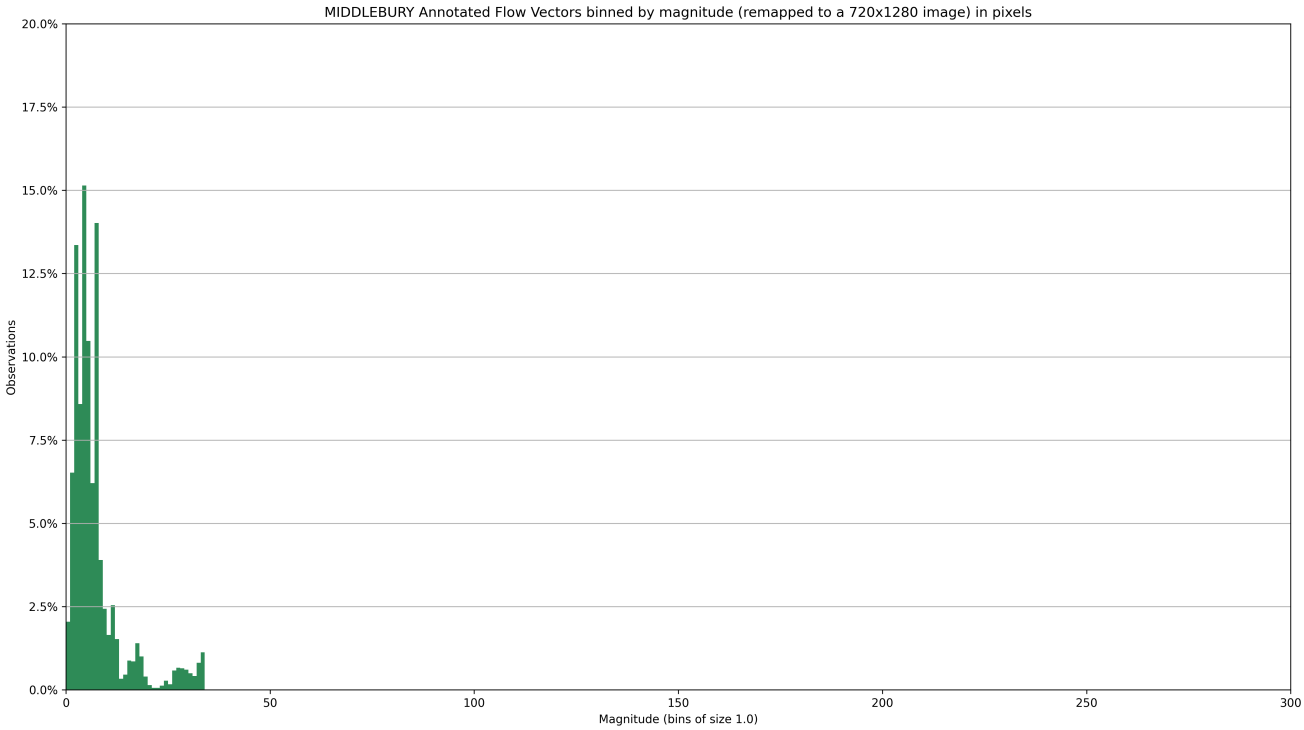

Figure 10: Visualization of the Sampled Magnitudes from Middlebury

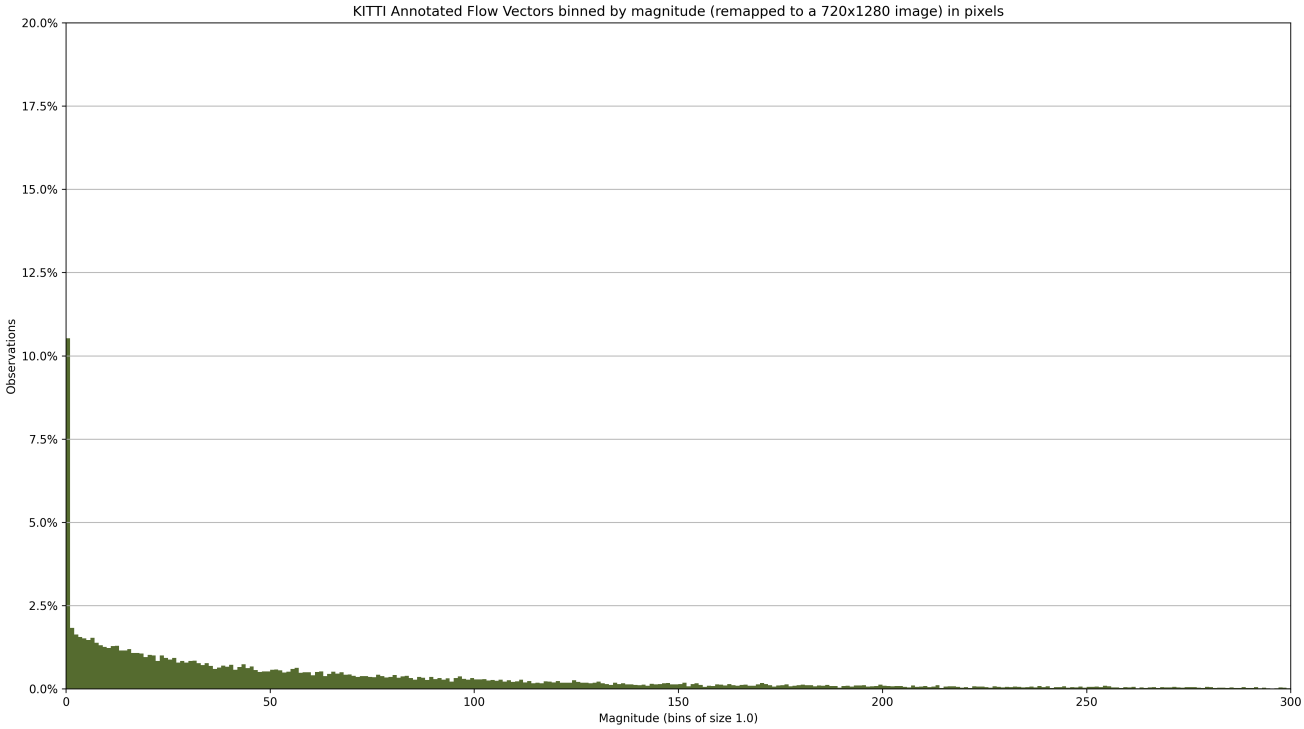

Figure 11: Visualization of the Sampled Magnitudes from KITTI

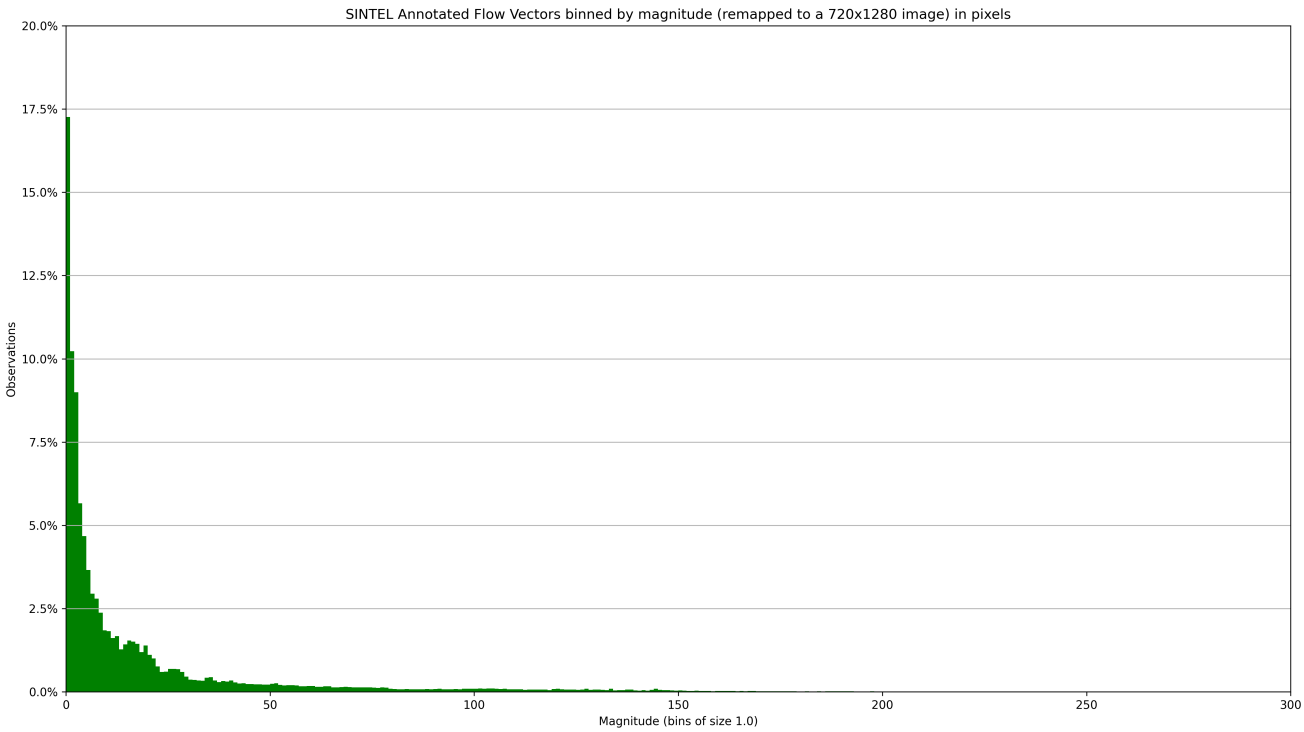

Figure 12: Visualization of the Sampled Magnitudes from SINTEL

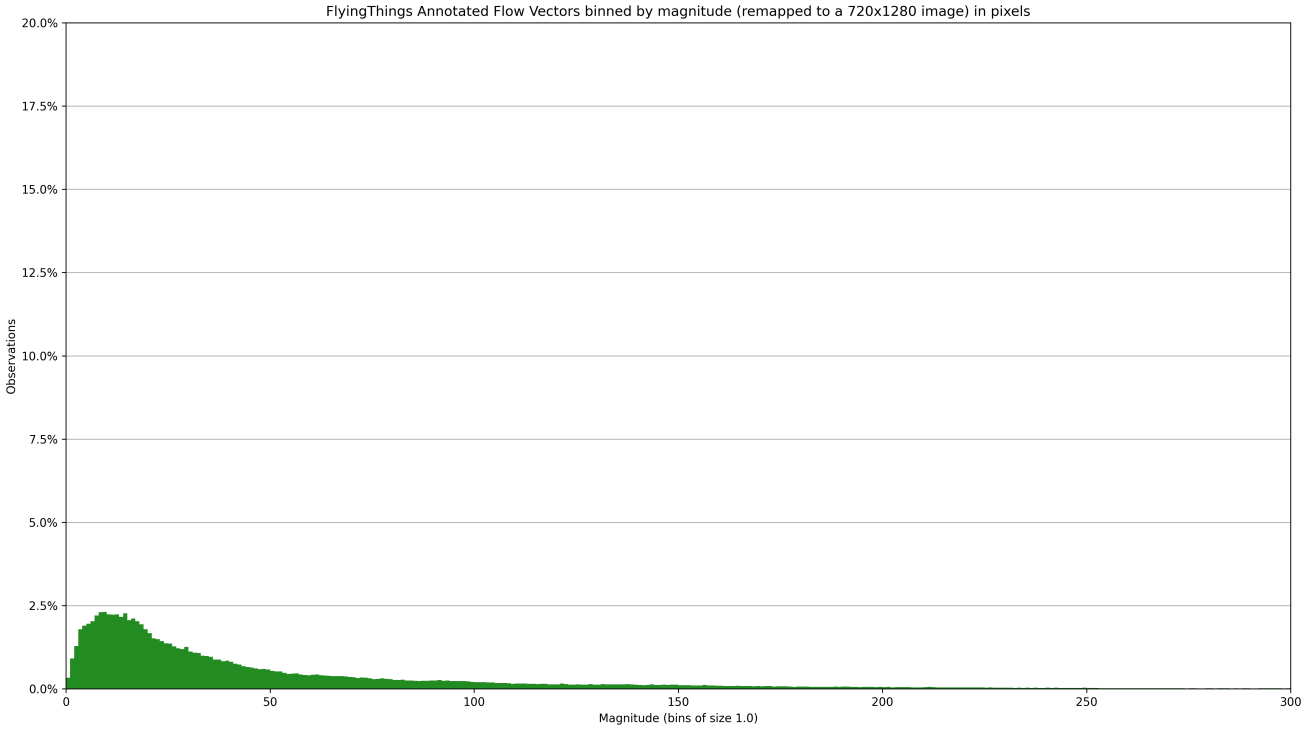

Figure 13: Visualization of the Sampled Magnitudes from FlyingThings

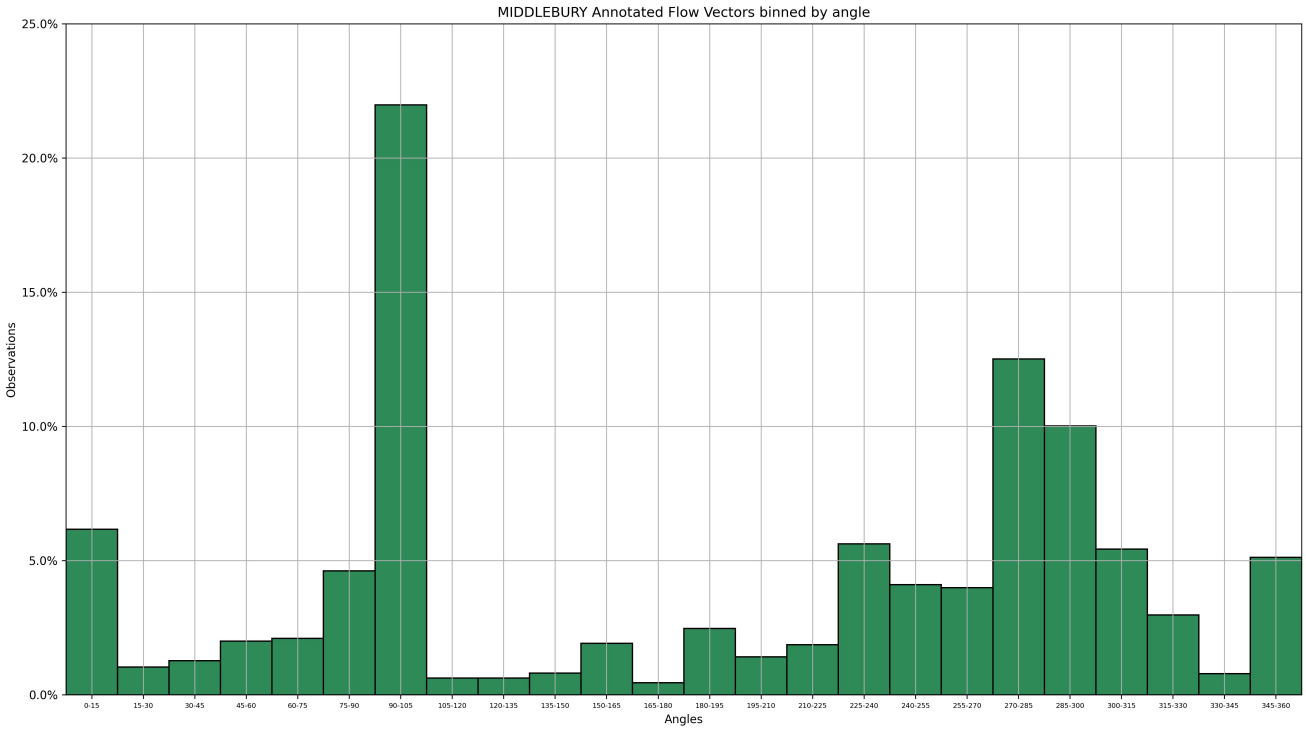

Figure 14: Visualization of the Sampled Angles from Middlebury

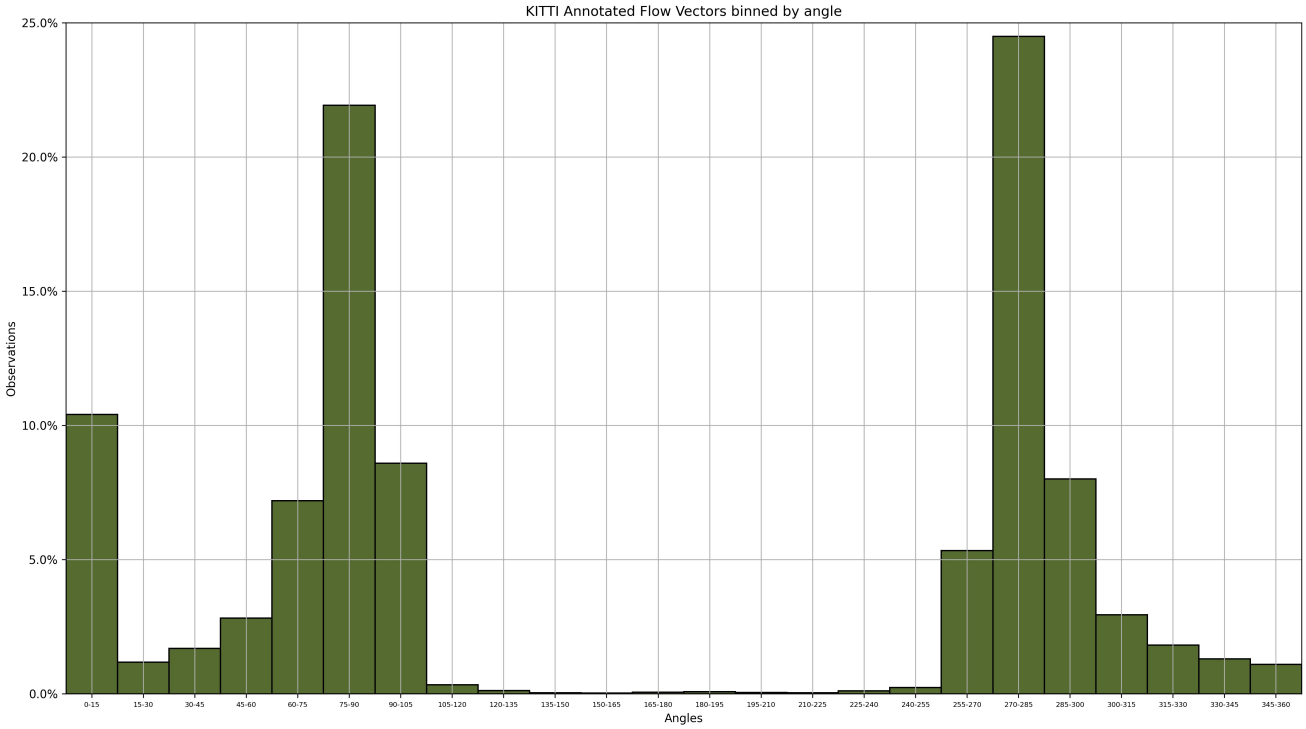

Figure 15: Visualization of the Sampled Angles from KITTI

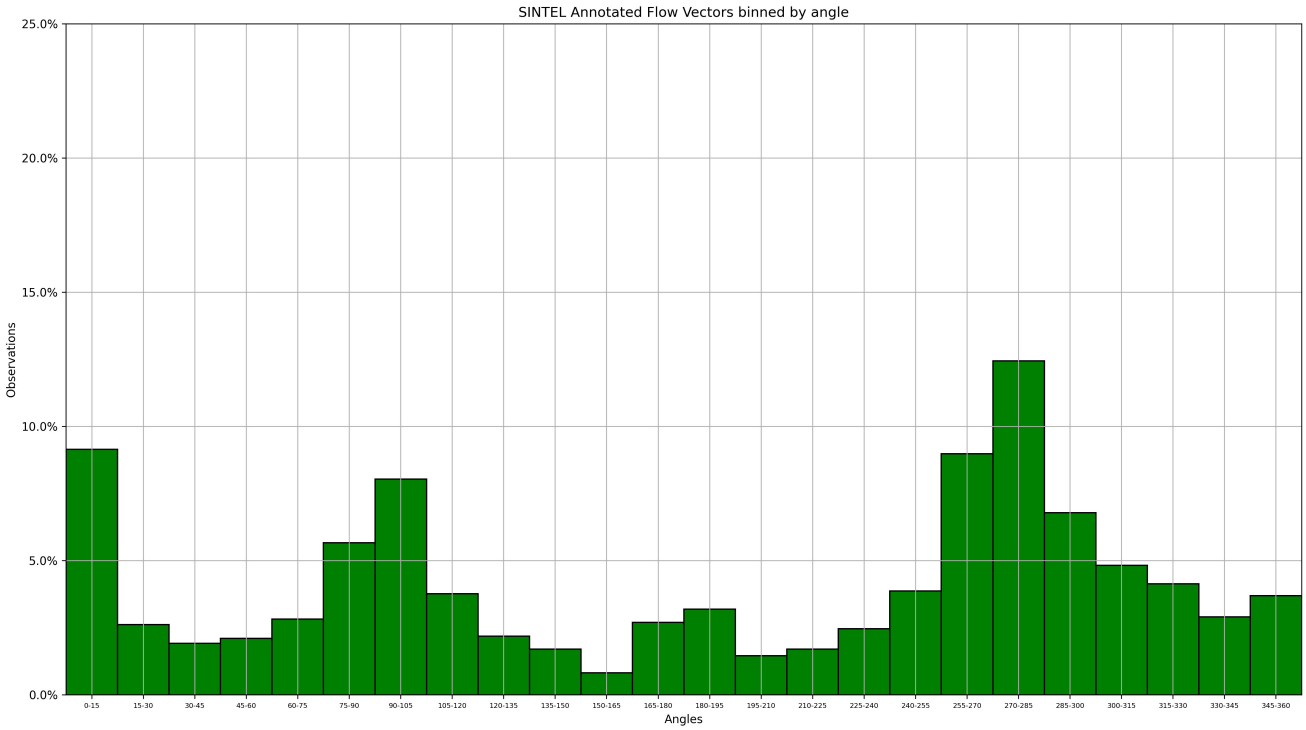

Figure 16: Visualization of the Sampled Angles from SINTEL

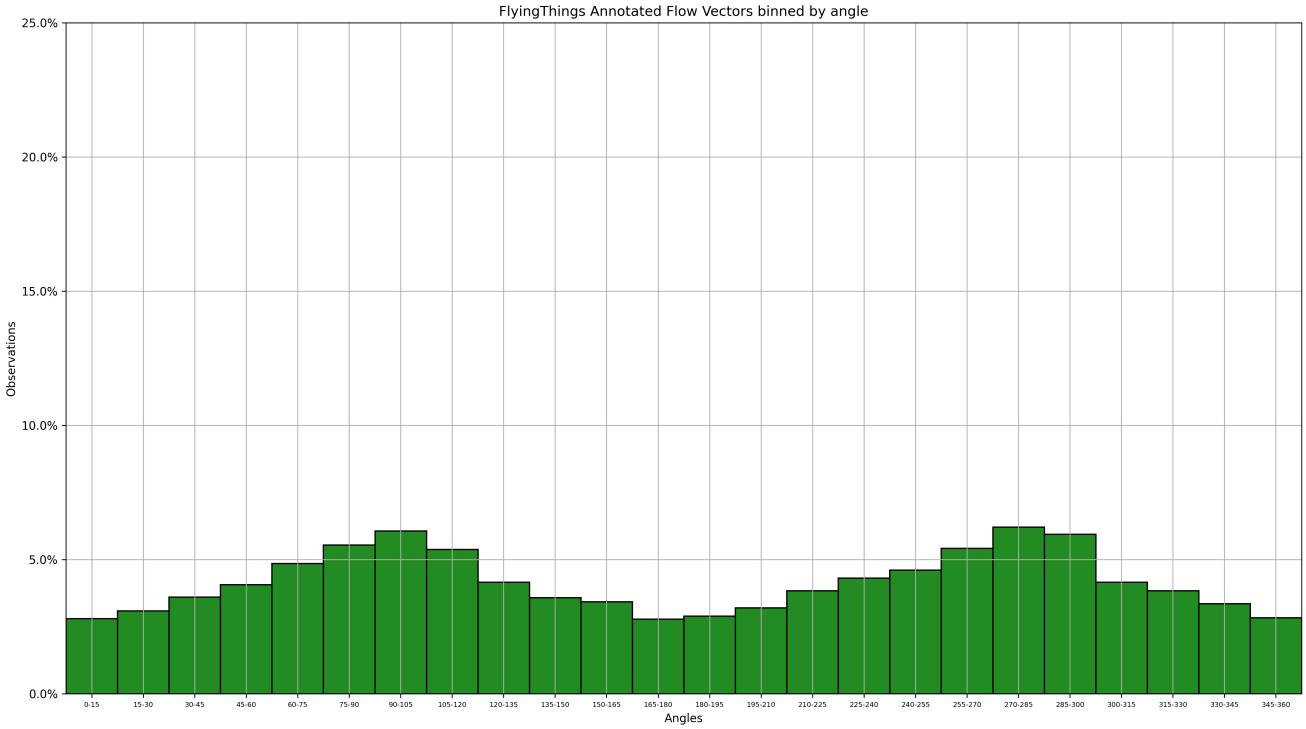

Figure 17: Visualization of the Sampled Angles from FlyingThings

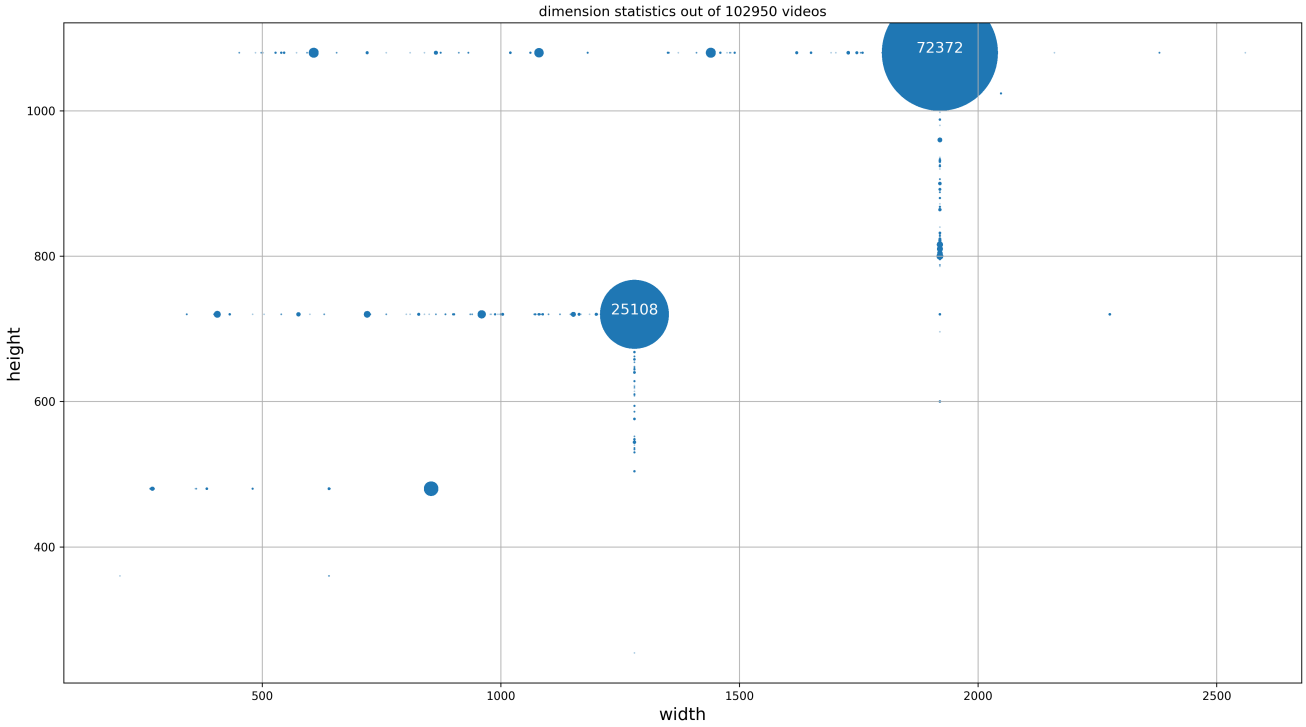

Figure 18: Example of a short caption, which should be centered.

Figure 19: Examples of ”Harder” points

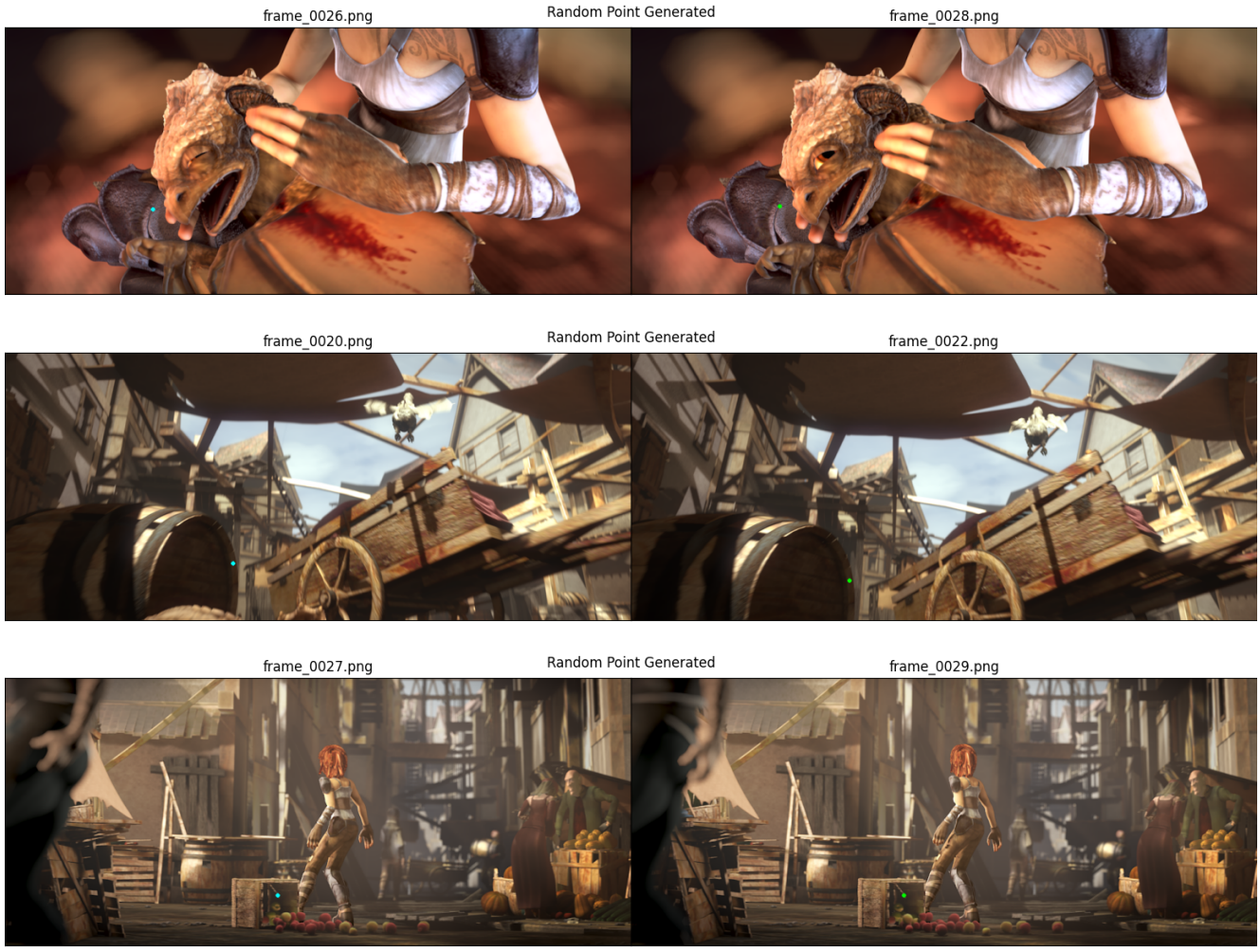

Figure 20: Examples of ”Easier” points

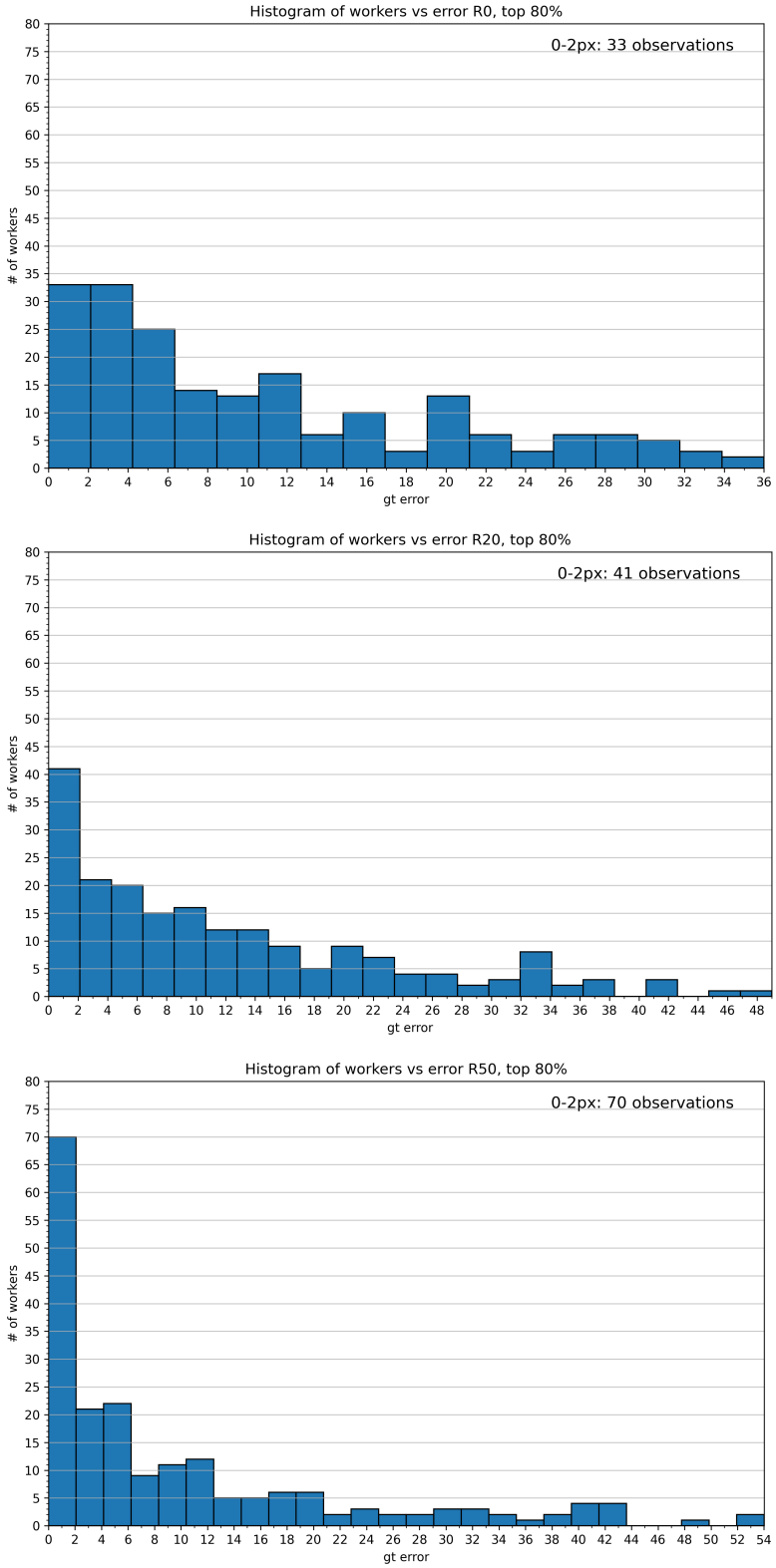

Figure 21: Left Adjustable Results for a radius of 50 pixels for top 80% of results 250 results