Frame Selection Strategies for Real-Time Structure-from-Motion from an Aerial Platform

Andrew R. Buck, Jack D. Akers,
Derek T. Anderson, James M. Keller
Dept. of Electrical Engineering and Computer Science
University of Missouri
Columbia, MO, USA

Raub Camaioni, Matthew Deardorff,
Robert H. Luke III
US Army DEVCOM C5ISR Center
Fort Belvoir, VA, USA

***Abstract*—Determining depth from a single camera in motion is a challenging problem that has numerous applications, including autonomous navigation of an unmanned aerial vehicle (UAV). Using traditional computer vision techniques such as structure-from-motion (SfM), a depth estimate can be generated using two image pairs from the video stream. The choice of image pairs directly impacts the quality of reconstruction, which is based largely on the camera extrinsics and image features. In this article, we discuss frame selection algorithms to select appropriate image pairs to process for depth estimation. Frames are stored in a rolling buffer, and several measures are computed on potential frame pairs based on camera extrinsics. We use a customized SfM algorithm, EpiDepth, which has been designed for handling sequences of aerial imagery with embedded GPS and camera pose metadata. We demonstrate our technique on a simulated dataset created using Unreal Engine and AirSim.**

***Keywords*—** 3D reconstruction, structure-from-motion, monocular depth estimation, frame selection

I. Introduction

Mapping a 3D environment with a UAV is a common task that relies heavily on computer vision techniques. Depending on the specific application, there may be varying degrees of sensor precision and autonomy. One challenging case involves the use of a remotely controlled UAV with reasonably accurate pose information, but only a single camera. In this situation, 3D information (depth) must be inferred using structure-from-motion (SfM) on sequential frame pairs, and the camera poses are independently determined by the remote operator. The problem then becomes how to select appropriate frames from the video sequence to perform the best SfM reconstruction. This problem is separate (but related) to the general task of 3D scene reconstruction, since we assume that the UAV flight pattern is fixed or that the frame selection algorithm has no control over it.

The motivation for this work arises from our initial experiments with reconstructing depth from actual UAV flight recordings, where camera poses could be derived from GPS/IMU/Magnetometer metadata, and any pair of frames could be used as a stereo pair for SfM. We determined that the relative poses of the frames was one of the most important factors in determining reconstruction quality. A simple frame selection strategy was ignoring many potential good pairs, and often selecting sub-optimal pairs. The frame selection algorithm could therefore be improved and draw upon known properties of the epipolar geometry.

To study this problem, we utilize a simulation environment using Unreal Engine and AirSim [1]. This allows us to generate data in a controlled setting, where we have access to exact pose information and ground truth depth. The results of our study can be easily transferred back to real-world data, since the algorithms depend only on having access to the camera pose information. The lack of sensor noise in our simulated dataset provides ideal conditions for an experiment to study the impact of different methods. Herein, we generate a synthetic dataset to use for the development of three different frame selection strategies and quantitatively compare the results of each.

For each method, we replay the recorded frames and select (in sequence) frame pairs to use for SfM. Each selected frame pair has known camera pose parameters and can therefore be used as input to the EpiDepth algorithm [2] to estimate a dense cloud of 3D points. An overview is shown in Fig. 1. The images are warped according to the epipolar geometry of the two camera views, and the amount of warping can vary depending on the relative poses. Some poses are known to be more effective at estimating depth (e.g. moving while looking nadir or strafing), while others tend to be less effective (e.g. looking straight ahead while moving). Our frame selection strategies use the pose information of the camera views as the primary method of ranking and choosing frame pairs.

We present three different frame selection methods: a naïve approach using a simple strategy, a heuristic-based approach that computes an expected quality based on the camera extrinsics and uses a rolling frame buffer, and a data-driven approach that uses the synthetic dataset to learn which pose configurations work best. The remainder of this paper discusses the details of this dataset, the frame selection methods, and our experimental analysis. We conclude with a discussion on the limitations of this study and present some ideas for future work.

II. Related Work

The problem of 3D mapping from a UAV has been widely studied [3], with most techniques using some form of SfM [4]. The mapping task is often the primary focus, such that equipment and mission parameters are chosen to optimize the final output after offline processing. For instance, high-precision GPS/IMU/magnetometer units can be used with a regular grid flight pattern to reduce the uncertainty in camera poses and produce a uniformly sampled 3D map. However, in some cases real-time operation is required, and the flight trajectory may be unknown ahead of time. This requires a different approach that can make the best use of what data is available.

The EpiDepth algorithm [2] is designed to robustly handle the wide variety of camera poses that can occur in real-time UAV flight and produce 3D depth estimates from image pairs. Using known camera

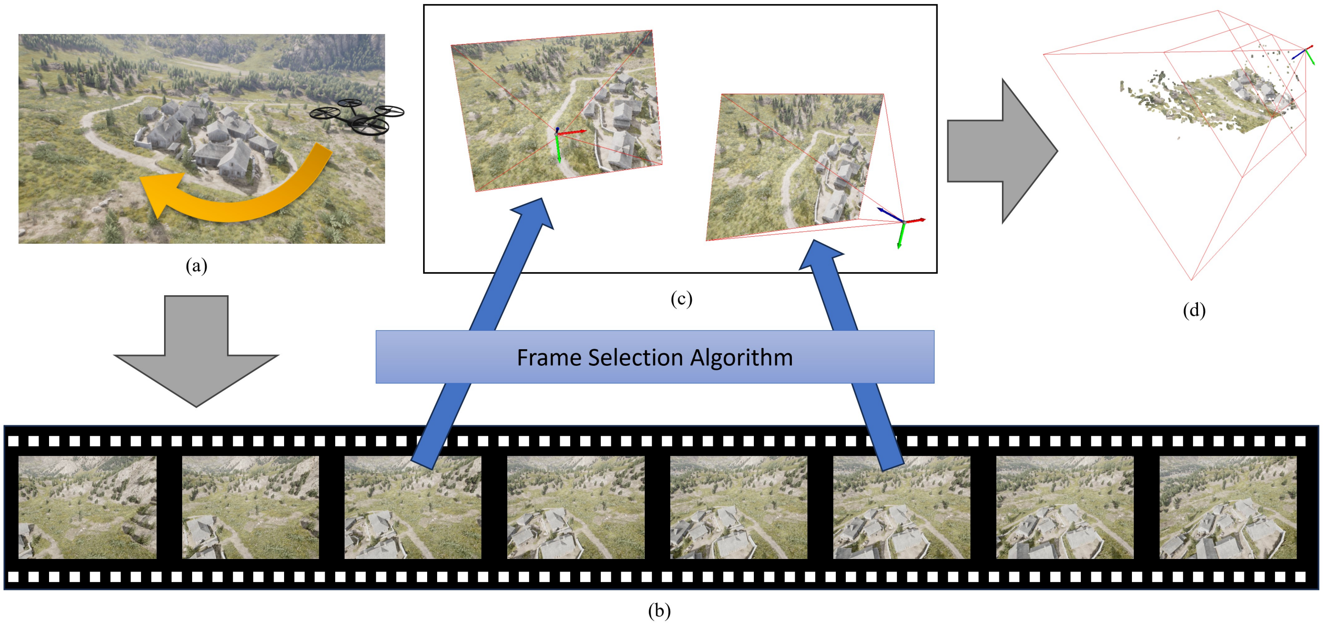


Fig. 1. Overview of the frame selection process. A UAV flies through an environment (a), generating a sequence of image frames (b). Our frame selection algorithm has access to the history of all past frames and must select pairs of frames to use for stereo reconstruction. Each frame pair (c) has known pose parameters and can be used to generate a 3D point cloud (d).

extrinsic parameters, the algorithm takes two frames and performs epipolar warping to align the image features such that optimized block matching can be performed to estimate disparity. The result is a 3D point cloud that is typically aggregated in a world-space map, such as the hierarchical voxel method of UFOMap [5]. Note that EpiDepth does not solve the localization problem addressed by SLAM, but can still produce high-quality maps with accurate sensors.

III. Background

A. Epipolar Warping Effects

The process of rectifying a pair of images into a common image plane can result in significant warping. The EpiDepth algorithm uses the extrinsic camera parameters to transform each image such that all matching epipolar lines appear as horizontal rows in the warped images. Depending on the relative configuration of the two cameras, the amount of warping can vary. For example, forward movement while looking nadir or a sideways strafing motion tends to result in very little warping from the original images. In contrast, forward movement while looking straight ahead is very challenging. Generally, less warping is required if the epipoles are not contained within the images.

Some examples of epipolar warping effects are shown in Fig. 2. Here, we see that the first row with Nadir movement results in almost no warping, although the images have been rotated 90°. The next two rows have similar relative poses: a vertical climb and forward motion, both with the camera pointing downward at 45°. In both of these cases, the warping is noticeable, but modest. The last row shows the dreaded straight ahead movement. Here, the epipoles are in the center of the images (indicated by the red and blue crosshairs), and the image warping is significant.

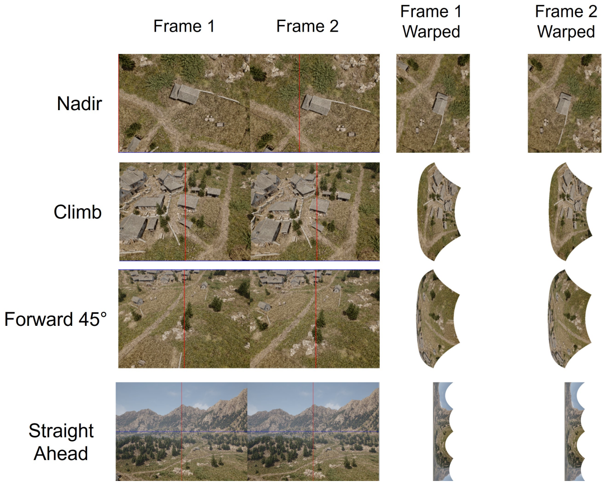
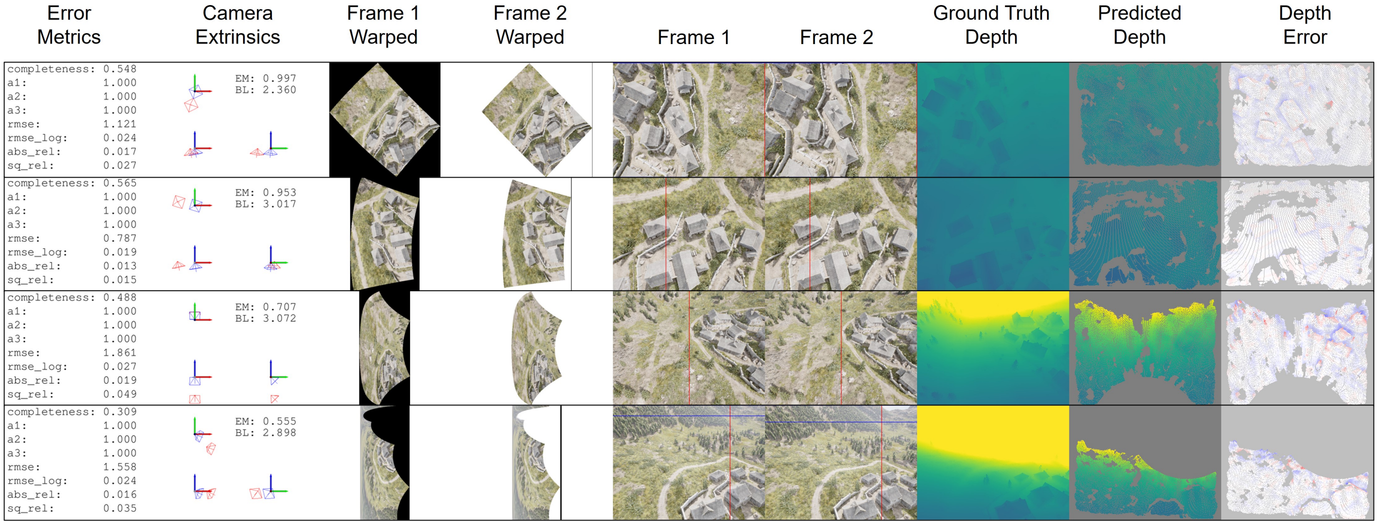


Fig. 2. Epipolar warping effects.

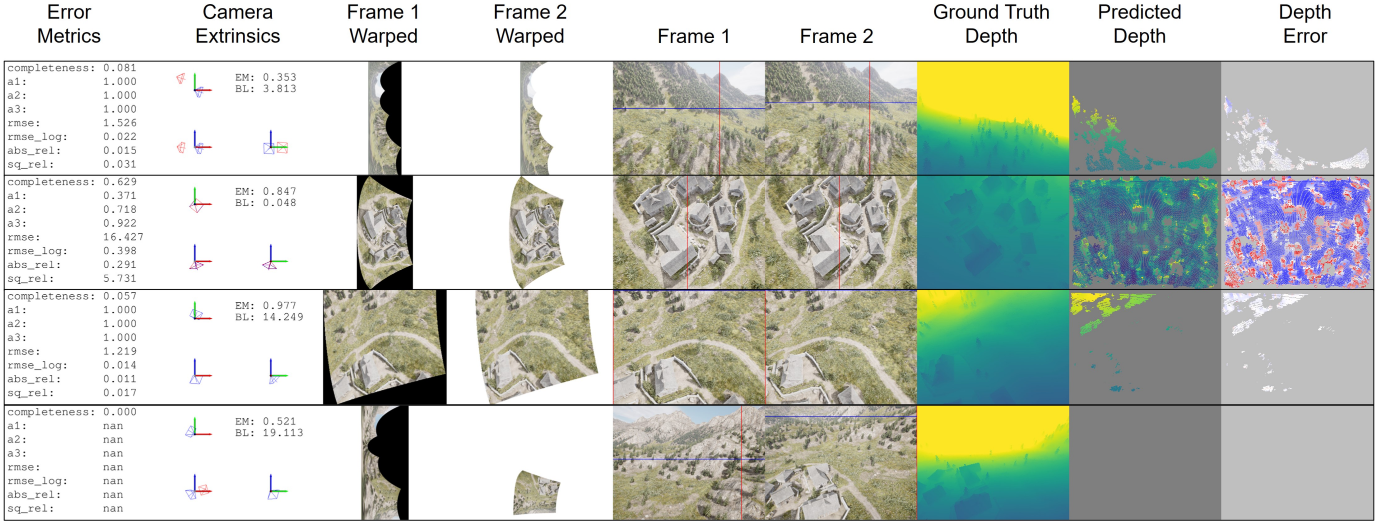
B. Extrinsic Quality Metric

A heuristic method was presented in [6] to represent the expected amount of warping that would occur by running EpiDepth on any two camera poses. The method uses only the extrinsic parameters of the cameras and does not rely on any image content. The metric is a function of the angles between two look vectors A and B as well as the baseline vector D. The intent is that in order to score highly (close to 1), the angle $\angle AB$ should be small, and the angles $\angle AD$ and $\angle BD$ should be close to 90°. Look vectors that stray too far from this ideal score lower, with a minimum score of zero.



```
completeness: 0.548
a1:           1.000
a2:           1.000
a3:           1.000
rmse:         1.121
rmse_log:     0.024
abs_rel:      0.017
sq_rel:       0.027
EM: 0.997
BL: 2.360

completeness: 0.565
a1:           1.000
a2:           1.000
a3:           1.000
rmse:         0.787
rmse_log:     0.019
abs_rel:      0.013
sq_rel:       0.015
EM: 0.953
BL: 3.017

completeness: 0.488
a1:           1.000
a2:           1.000
a3:           1.000
rmse:         1.861
rmse_log:     0.027
abs_rel:      0.019
sq_rel:       0.049
EM: 0.707
BL: 3.072

completeness: 0.309
a1:           1.000
a2:           1.000
a3:           1.000
rmse:         1.558
rmse_log:     0.024
abs_rel:      0.016
sq_rel:       0.035
EM: 0.555
BL: 2.898
```

Fig. 3. Examples of EpiDepth results on frame pairs with good extrinsic camera parameters.



```
completeness: 0.081
a1:           1.000
a2:           1.000
a3:           1.000
rmse:         1.526
rmse_log:     0.022
abs_rel:      0.015
sq_rel:       0.031
EM: 0.353
BL: 3.813

completeness: 0.629
a1:           0.371
a2:           0.718
a3:           0.922
rmse:         16.427
rmse_log:     0.398
abs_rel:      0.291
sq_rel:       5.731
EM: 0.847
BL: 0.048

completeness: 0.057
a1:           1.000
a2:           1.000
a3:           1.000
rmse:         1.219
rmse_log:     0.014
abs_rel:      0.011
sq_rel:       0.017
EM: 0.977
BL: 14.249

completeness: 0.000
a1:           nan
a2:           nan
a3:           nan
rmse:         nan
rmse_log:     nan
abs_rel:      nan
sq_rel:       nan
EM: 0.521
BL: 19.113
```

Fig. 4. Examples of EpiDepth results on frame pairs with bad extrinsic camera parameters.

The effect of this metric is that any potential image pair can be evaluated quickly to assess if EpiDepth is likely to produce a good result. The first condition ensures that images are looking in the same general direction, and the second condition makes sure that the separation is perpendicular to the camera orientation (not moving along the camera axis). Although this is a hand-crafted heuristic, it has been shown to be useful for identifying good image pairs and is an important part of our frame selection algorithms.

IV. Simulated UAV Dataset

For this study, we created a simulated dataset using Unreal Engine and AirSim in the Mountain Village Environment [7]. The simulated UAV was programmed to takeoff and climb up to a fixed altitude, and then move to random waypoints with random look directions (between nadir and horizontal). This produced a wide variety of motion examples that covers most of the possible movement patterns we expect to encounter in practice. We collected about 1000 frames of data, where each frame includes the RGB color image, the ground truth depth image, and the pose (position and orientation) of the camera. By having access to simulated ground truth depth, we can quantitatively evaluate the quality of the EpiDepth reconstruction.

Any pair of images from our dataset could be used as input to EpiDepth. However, in order to produce a good result, the images need to have significant overlap and acceptable camera extrinsics. Typically, they are also captured very close to one another in time. Figures 3 and 4 show several examples of image pairs from the dataset and the corresponding EpiDepth results. In both figures we show each frame pair as a row with the inputs, outputs, and several error metrics.

In each row of these figures, Frame 1 and Frame 2 are the input RGB images that EpiDepth receives. The camera extrinsics are shown as top, front, and side view diagrams of the two camera positions. The EM value is the extrinsic metric heuristic computed for this frame pair, and the BL value is the baseline distance. The positions of the

TABLE I
FRAME SELECTION RESULTS

	Total Pairs	Completeness	RMSE-Log
Naïve	142	0.432 ± 0.226	0.035 ± 0.010
Heuristic-Based	592	0.364 ± 0.190	0.030 ± 0.138
Data-Driven	453	0.322 ± 0.249	0.024 ± 0.027

epipoles are indicated by red and blue crosshairs on the Frame 1 and Frame 2 images. To show the amount of warping, both warped images are also shown. The ground truth depth is displayed for each frame pair, but is not available to EpiDepth and is used only for scoring. The depth color map is constant for all images, with a maximum depth of 100 meters. The EpiDepth prediction is shown next to the ground truth, with undeclared pixels shown as gray. The rightmost image shows the difference between the prediction and ground truth on a red-white-blue color map, where white indicates an accurate prediction, red indicates the prediction was too far, and blue indicates the prediction was too close.

Several error metrics are computed on the comparison of each prediction to the ground truth, and these are listed on the left side of each row. Of these, completeness measures the percentage of declared pixels in the predicted depth image. The a1, a2, and a3 scores are the average inlier rates for the prediction, where a1 measures the percentage of pixels where the predicted depth is within a ratio of 1.25 of the ground truth depth, and a2 and a3 use ratios of 1.25^2 and 1.25^3 respectively. The rmse and rmse_log scores are the root-mean-square errors in Euclidean and log space, and the abs_rel and sq_rel scores are the mean absolute differences and mean squared differences of the predicted depths relative to the ground truth.

Figure 3 shows several good cases of frame pairs chosen for EpiDepth. The top two rows show nadir examples with high EM scores, and depth predictions that are both complete and accurate. The bottom two rows show examples looking out toward the horizon, with lower EM scores and less completeness, but still high accuracy. Figure 4 shows some cases where EpiDepth does poorly. The first row shows a case where the UAV is moving forward and the epipoles are contained within the images. Note that there is no depth prediction near the epipoles. The second row shows a case where the baseline distance is too small, resulting in large prediction errors. The third row shows a case where the baseline is too large for the configured window size, so corresponding features are missed, which results in low completeness. Finally, the last row shows a case where the images do not overlap, yielding no depth prediction at all.

V. FRAME SELECTION METHODS

In this section, we discuss three different methods for selecting frame pairs to use with EpiDepth. For each method, we assume that the algorithm is provided a stream of image frames with known camera poses. As new images arrive, the algorithms must produce appropriate frame pairs that will be processed by EpiDepth. The first method uses a simple naïve strategy with no frame buffer. The second method uses a frame buffer and the extrinsic quality metric as a heuristic. The third method is a data-driven approach that uses simulation to build a model of expected performance, and then uses this model to predict which frames will be best. We use the aforementioned simulated dataset as a common example to compare the approaches.

A. Naïve Frame Selection Method

In this first frame selection method, the strategy is to keep a current candidate frame and attempt to match it with new incoming frames. The complete algorithm is given in Algorithm 1. The process starts by initializing the first frame as f_0. For each new frame f_t, we compute the baseline distance, d, between it and the candidate frame. If this value is in the acceptable range between d_{min} and d_{max}, we then proceed to compute the rotational distance, r, between the two camera poses, defined here as the angular distance between the look vectors. If r is less than the maximum acceptable value r_{max}, then the frame pair (f_0, f_t) is yielded to EpiDepth for processing, and the algorithm continues by assigning $f_0 \leftarrow f_t$. If at any point the distance between f_0 and f_t becomes larger than d_{max}, the old frame f_0 is dropped and replaced with f_t.

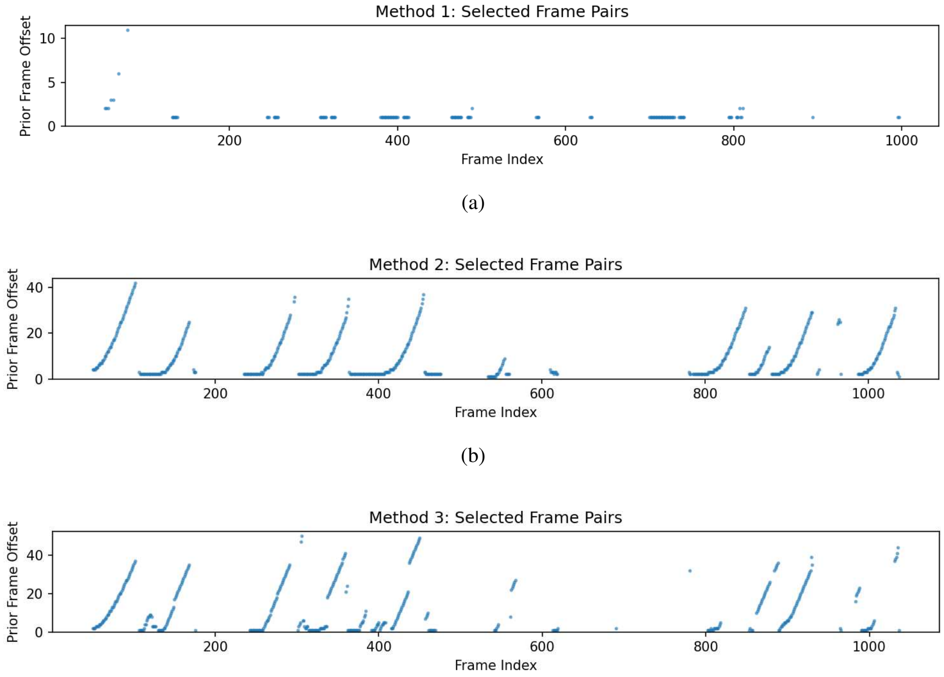
Algorithm 1 Naïve Frame Selection

```
Define $d_{min}$, $d_{max}$, and $r_{max}$
Initialize frame $f_0$
for each new frame $f_t$ do

    // Get baseline distance
    $d \leftarrow \textsc{Distance}(f_0, f_t)$

    // Check if extrinsics are acceptable
    if $d_{min} \leq d \leq d_{max}$ then
        $r \leftarrow \textsc{Rotation}(f_0, f_t)$
        if $r \leq r_{max}$ then
            yield $(f_0, f_t)$
            $f_0 \leftarrow f_t$
        end if
    end if

    // Drop old frame
    if $d > d_{max}$ then
        $f_0 \leftarrow f_t$
    end if

end for
```

The result of running this method on our simulated dataset is shown in Fig. 5a. This scatter plot shows a mark for each selected frame pair, with the horizontal axis showing the original frame index of the most recent frame, and the vertical axis showing how far back the selected match was in the sequence. In total, 142 frame pairs were selected with the majority of the pairs being only one or two frames apart. The outliers near the start are due to the takeoff sequence, which is different from the rest of the flight. Overall, the frame pairs that this method selected were generally good as indicated in Table I, but the total number of pairs produced was low.

B. Heuristic-Based Frame Selection Method

To improve upon the naïve frame selection method, the next approach uses the extrinsic quality metric as a heuristic along with a rolling frame buffer. The complete algorithm is given in Algorithm 2. Each time a new frame f_t arrives, it is added to the buffer, which is searched in reverse to find an acceptable match. For our experiments, we use a buffer size of 50 frames. After updating the buffer with the new frame and removing the oldest frame if the buffer is full, we



(a)

(b)

(c)

Fig. 5. Frame pairs selected by the naïve method (a), the heuristic-based method (b), and the data-driven method (c) on the simulated dataset. The horizontal axis shows the frame index (time), and the vertical axis shows the relative offset (back in time) of frames that were matched. Dots indicate frames that were selected.

begin a loop to check each prior frame f_{t-i} to see if it is good enough. We first check to make sure that the frames are far enough apart to produce a good reconstruction. If so, we then continue to evaluate the extrinsic quality metric for the frame pair. If the quality q is above the threshold q_{min}, the frame pair (f_t, f_{t-i}) is yielded to EpiDepth and the search stops. The effect of this is that each frame is matched with the most recent frame that would be an acceptable match, as defined by the distance and extrinsic quality metric.

This method results in a total of 592 frame pairs being selected, which is many more than the naïve approach. The results shown in Table I show that although the completeness was reduced, the average RMSE-Log score was improved. The plot in Fig. 5b shows the selected frame pairs, which now includes pairs that have a much larger frame index offset. These "rising trails" can be explained by the slowing motion of the UAV as it approaches each waypoint. As the distance between successive frames decreases, the algorithm has to search farther back in the buffer to find an acceptable paring. The effect is that the same frame may be used multiple times as an anchor for several frame pairs.

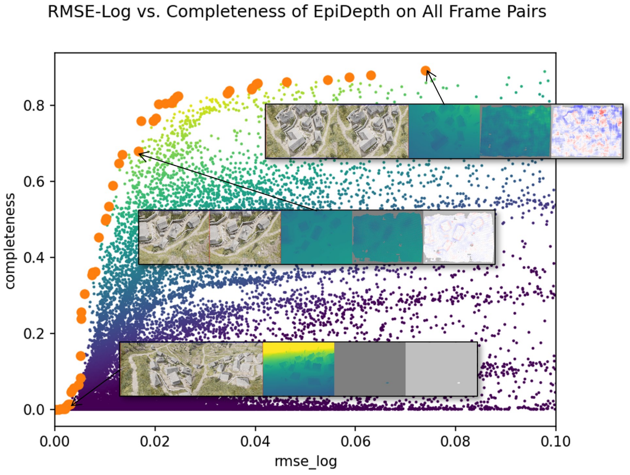


Fig. 6. Scatter plot of the RMSE-Log and completeness scores for all possible frame pairs in the simulated dataset. The Pareto optimal set is shown in orange, along with some example images. The weighted score for each point is indicated by the color mapping, with the highest scoring pairs in bright yellow and the lowest scoring pairs in dark blue.

C. Data-Driven Frame Selection Method

The final frame selection method seeks to improve upon both the naïve and heuristic-based methods by utilizing the ground truth provided by the simulated data set. Since each frame in the dataset has a corresponding ground truth depth image, we can evaluate the accuracy of any possible frame pair and compute several error metrics. Many of these metrics are described in Section IV. Perhaps the most relevant for assessing the overall quality of reconstruction are the completeness scores and the RMSE-Log values. We would like for completeness to be high, indicating that a large part of the image receives a depth prediction. We also want the RMSE-Log value to be low, showing that the EpiDepth prediction is a close match to the ground truth. These two objectives are plotted in Fig. 6, which shows the completeness and RMSE-Log values for all possible frame pairs. The Pareto optimal set is highlighted in orange, spanning frame pairs that have very low error, but also low completeness, to pairs that have high completeness, but also high error. Between these two extremes are frame pairs that result in a good balance of completeness and error. We define a linear weighting of these two features that results

Algorithm 2 Heuristic Frame Selection

```
Define $N$, $d_{min}$, and $q_{min}$
Initialize rolling frame buffer $B$
for each new frame $f_t$ do

    // Update the frame buffer
    $B$.insert($f_t$)

    // Find an acceptable frame to pair with $f_t$
    for $i = 1$ to $N$ do

        // Check if far enough apart
        $d \leftarrow$ DISTANCE($f_{t-i}, f_t$)
        if $d < d_{min}$ then
            continue
        end if

        // Check if EQ is good enough
        $q \leftarrow$ EXTRINSICQUALITY($f_{t-i}, f_t$)
        if $q \geq q_{min}$ then
            yield $(f_0, f_t)$
        else
            // Stop looking for a pair with $f_t$
            break
        end if

    end for

end for
```

in an overall score, indicated by the color mapping. The best frame pairs are plotted in bright yellow, and the worst pairs are plotted in dark blue.

We next train train a neural network to predict the expected overall score of any given frame pair based on the camera extrinsic parameters. The input is formatted as the flattened translation vectors and rotation matrices of the two camera poses. This is fed through two hidden layers of size 64 and 32 respectively, and a single output neuron with ReLU activation functions. We train on 80% of the available data (approximately 50,000 frames) for 1000 epochs and achieve an MSE loss of 0.4515 on the remaining test set. This trained network is then used as part of the real-time data-driven frame selection method.

The overall method for the data-driven approach is given in Algorithm 3. The process is very similar to the heuristic-based method, using a fixed-size frame buffer and scanning through the buffer to try and find a match for each incoming frame. Instead of using the first acceptable frame, this method finds the frame in the buffer that produces the best predicted score as evaluated by the neural network. If this frame meets some minimum threshold and also satisfies the distance and extrinsic metric requirements, the frame pair is yielded to EpiDepth. Otherwise no pair is produced for this incoming frame.

The results of this method on the simulated dataset given in Table I show that 453 frame pairs were selected, which is fewer than the heuristic-based approach, but still many more than the naïve method.

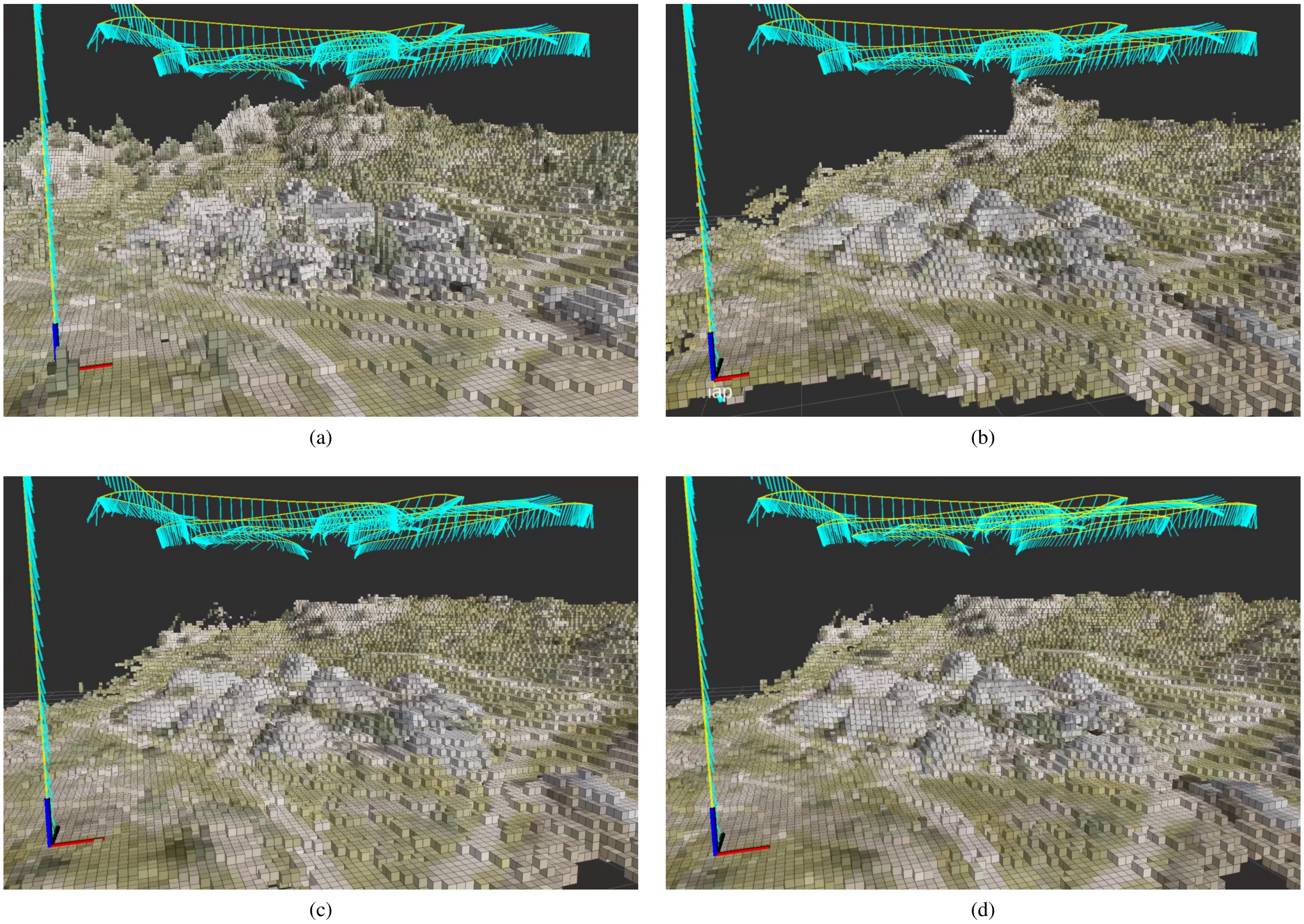
Algorithm 3 Data-Driven Frame Selection

```
Define $N$, $d_{min}$, $q_{min}$, and $p_{min}$
Initialize rolling frame buffer $B$
for each new frame $f_t$ do

    // Update the frame buffer
    $B$.insert($f_t$)

    // Find an acceptable frame to pair with $f_t$
    $f_{best} \leftarrow \varnothing$
    $p_{best} \leftarrow -\infty$
    for $i = 1$ to $N$ do

        // Check if acceptable distance and EQ
        $d \leftarrow$ DISTANCE($f_{t-i}, f_t$)
        $q \leftarrow$ EXTRINSICQUALITY($f_{t-i}, f_t$)
        if $d < d_{min}$ or $q < q_{min}$ then
            continue
        end if

        // Predict the quality of the frame pair
        $p \leftarrow$ PREDICT($f_{t-i}, f_t$)
        if $p > p_{best}$ and $p \geq p_{min}$ then
            $p_{best} \leftarrow p$
            $f_{best} \leftarrow f_{t-i}$
        end if

    end for

    // Yield the best frame pair if one was found
    if $f_{best} \neq \varnothing$ then
        yield $(f_{best}, f_t)$
    end if

end for
```

The average completeness was lower, but the RMSE-Log score was the best of all the methods. Fig. 5c shows a frame pair selection pattern similar to the heuristic-based approach, but slightly sparser. Again, the “rising trail” pattern is caused by the UAV easing into each waypoint and pausing, causing new frames to match farther back in the buffer to an anchor frame that was far enough away to satisfy the minimum distance requirement.

VI. QUALITATIVE 3D ANALYSIS

Ultimately, our goal is to generate accurate and high-quality 3D reconstructions of an environment. As the UAV moves and generates a sequence of images, we select the best frames to use for SfM with EpiDepth. Although each frame pair can be evaluated independently, some aspects of the problem are only observed after combining multiple projections into a 3D map. We use UFOMap to aggregate the 3D point clouds into a common hierarchical voxel space using a probabilistic observation model. As more subsequent points are observed within a grid cell, the probability of occupancy increases, whereas observing free space between the camera and a projected point decreases the value. The effect is that by utilizing multiple


(a)

(b)

(c)

(d)

Fig. 7. UFOMap reconstructions of the simulated datset using the different frame selection methods. The path of the drone is shown as a yellow line, with the look vector of each frame drawn in cyan. (a) Ground truth. (b) Naïve method. (c) Heuristic-based method. (d) Data-driven method.

updates to the map, small errors are averaged out, resulting in a more accurate and complete model.

The UFOMap reconstructions from the different frame selection methods on the simulated dataset are shown in Fig. 7. Here, we give a qualitative assessment of the results by comparing each method to the known ground truth. We observe that the ground truth reconstruction (Fig. 7a) is the most detailed, showing sharp edges on buildings and individual trees. The reconstructions from the three frame selection methods (Fig. 7b, Fig. 7c, and Fig. 7d) are all very good, but with slightly less definition than the ground truth. It is difficult to find significant differences between them. Perhaps this is not unexpected, since the 3D projections all come from the same possible set of input images and differ only in which projections are added to the map. Since there is no noise in the simulated camera poses, the reconstructions are all very accurate, although limited by the image features that can be detected with frame matching. A quantitative analysis with additional experiments could reveal more differences between the approaches.

VII. Conclusions and Future Work

Selecting the best frames to use for SfM is an important part of a real-time 3D mapping system. The frame selection methods presented here are suitable for use on a UAV system where pose information is available, but not necessarily controlled by the algorithm. We believe that the data-driven and heuristic approaches using a frame buffer offer better results than the naïve method. These methods generate more frame pairs, which can help average out any errors in the overall reconstruction.

Our experiments with this simulated dataset showcased nearly ideal operating conditions, with very little sensor noise, and very accurate pose information. In real-world situations, these may become significant factors that influence the frame selection algorithms. The random flight pattern exhibited here was used to capture a broad distribution of all possible poses, but a focused study using anticipated movement behaviors could be insightful. For instance, flying in a zig-zag pattern could produce useful stereo pairs while still moving forward towards a goal.

Future efforts will focus on evaluating these frame selection algorithms on real data and introducing sensor noise into the simulated dataset. While real datasets can be difficult to score quantitatively due to the lack of ground truth, simulated datasets can be be used measure the quality of a 3D reconstruction. We have explored several metrics for comparing voxel maps [8] and can decompose the analysis to focus on the reconstruction accuracy of different variables, such as object type and range [9]. Ultimately, these frame selection methods are just one part of a larger system that is subject to many uncertainties, but by improving and understanding each component in detail, we create a more robust and general system that can handle a variety of real-world situations.

References

[1] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[2] R. Camaioni, R. H. Luke, A. Buck, and D. T. Anderson, “EpiDepth: A real-time monocular dense-depth estimation pipeline using generic image rectification,” in *Geospatial Informatics XII*, vol. 12099. SPIE, May 2022, pp. 101–114.

[3] S. Jiang, W. Jiang, and L. Wang, “Unmanned Aerial Vehicle-Based Photogrammetric 3D Mapping: A survey of techniques, applications, and challenges,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 2, pp. 135–171, Jun. 2022.

[4] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, “A survey of structure from motion,” *arXiv:1701.08493 [cs.CV]*, 2017.

[5] D. Duberg and P. Jensfelt, “UFOMap: An Efficient Probabilistic 3D Mapping Framework That Embraces the Unknown,” *arXiv:2003.04749 [cs.RO]*, Mar. 2020.

[6] A. R. Buck, D. T. Anderson, R. Camaioni, J. Akers, R. H. Luke, and J. M. Keller, “Capturing uncertainty in monocular depth estimation: Towards fuzzy voxel maps,” in *2023 IEEE International Conference on Fuzzy Systems (FUZZ)*, 2023, pp. 1–6.

[7] “Mountain Village Environment,” https://www.unrealengine.com/marketplace/en-US/product/mountain-village-environment, (Accessed: 22 Nov. 2022).

[8] J. Akers, A. Buck, R. Camaioni, D. T. Anderson, R. H. L. III, J. M. Keller, M. Deardorff, and B. Alvey, “Simulated gold-standard for quantitative evaluation of monocular vision algorithms,” in *Geospatial Informatics XIII*, K. Palaniappan, G. Seetharaman, and J. D. Harguess, Eds., vol. 12525, International Society for Optics and Photonics. SPIE, 2023, p. 125250A. [Online]. Available: https://doi.org/10.1117/12.2657567

[9] A. Buck, “Simulated data to train and evaluate deep learning-based passive monocular vision algorithms at medium to long ranges,” in *MSS*, 2023.