EpiDepth: A Real-Time Monocular Dense-Depth Estimation Pipeline Using Generic Image Rectification

Raub Camaioni*[a], Robert H. Luke III[a], Andrew Buck[b], Derek T. Anderson[b]
[a]U.S. Army DEVCOM C5ISR Center, Fort Belvoir, VA, USA
[b]Electrical Engineering & Computer Science Dept, University of Missouri, Columbia MO, USA

ABSTRACT

Because of weight, power, and cost constraints, most unmanned aircraft systems (UAS) contain monocular camera systems. Real-time structure from motion (SfM) algorithms are required for monocular UAS systems to sense and autonomously navigate 3D environments. The SfM algorithm must be designed to work near real-time and handle the wide variety of possible extrinsic parameters produced by UAS image pairs. Common rigid epipolar rectification techniques (homography-based rectification) do not accurately model epipolar geometries with epipoles close to or contained within the image bounds. Common UAS movement types, translation along lens axis, tight radial turns, circular patterns with GPS locked camera focus, can all produce epipolar geometries with epipoles inside the camera frame. Using a generalized epipolar rectification technique, all extrinsic UAS movement types can be handled, and optimized image block matching techniques can be used to produce disparity/depth maps.

The majority of UASs contain GPS/IMU/magnetometer modules. These modules provide absolute camera extrinsic parameters for every image. The essential and fundamental matrices between image pairs can be calculated from these absolute extrinsics. SfM is performed, depth images are produced, and the camera space pixel values can be projected as point clouds in three-dimensional space. These point clouds provide scene understanding that can be used for autonomous reasoning systems.

Keywords: Structure from motion, stereo vision, unmanned aircraft systems, epipolar rectification

1. INTRODUCTION

Autonomous unmanned aircraft systems (UAS) require robust sensing capabilities to navigate complex environments. One possible sensing capability is image-based depth estimation generated from temporally separated monocular images. UASs have restrictive power and weight constraints limiting most systems to monocular camera payloads. Effective monocular dense depth estimation techniques allow small and inexpensive UASs to navigate autonomously. Autonomous navigation enables more complex craft maneuverability and relieves operator burden.

Using image pairs collected temporally, depth estimation can be performed on monocular image streams. Image pairs generated from UAS monocular image streams can have a wide range of extrinsic parameters. Some UASs perform digital image stabilization, which can also change the intrinsic parameters of a camera between image pairs. Translation of the UAS in the direction of the camera view vector, v_c, results in complicated image rectification for depth calculation. Traditional image pair rectification methods using homography-based rectification are unable to rectify most complex image pairs generated by UASs. To generate depth information from the wide range of possible image pairs, a generic rectification method able to handle all possible intrinsic and extrinsic parameters is required.

Epipolar geometry is the mapping relationship of 3D and 2D points between camera pairs. Using epipolar constraints, derived from epipolar geometry, point correspondence search between images can be limited to single line scans. Limiting point correspondence search space drastically improves disparity calculation efficiency and accuracy.

There is a fair amount of research in the areas of structure from motion [1, 2], Multiview stereo [3], or deep learning methods [4, 5]. This paper implements the generic rectification method outlined in [6] and then adds details needed for real-time implementation on a UAS. Definition of terminology used in this paper is shown in Table 1.

*raub.j.camaioni.civ@army.mil

Table 1. Definition of terminology used throughout this paper. Figure 1 shows these variables graphically.

TERMINOLOGY	DESCRIPTION
Camera Location o	The location of the camera in three-dimensional space.
Baseline Vector v_b	The translation vector from camera 1 location to camera 2 location.
Camera Plane P_c	The plane perpendicular to the cameras view vector onto which all image information is projected.
Epipolar Plane P_e	Any plane that contains the baseline vector.
Epipolar Line l_e	An intersection of a given epipolar plane with a camera plane.
Epipolar Point (Epipole) p_e	The intersection of the baseline vector with a camera plane.
Epipolar Sample Ray r_s	A ray vector originating at an image's epipole, pointing along an epipolar line of that image.
Matching Epipolar Lines l_1, l_2	A pair of epipolar lines created by an epipolar plane's intersection with camera 1 plane and camera 2 plane.

2. EPIPOLAR GEOMETRY MATHEMATIC BACKGROUND

2.1 Intrinsic and Extrinsic Parameter Background

This paper will not go into the generation of intrinsic and extrinsic parameters. The algorithm assumes that intrinsic and extrinsic parameters are provided, and that lens distortion has been removed from imagery. In our testing, simulated imagery was generated with Airsim [7] and the Unreal Engine [8]. In our real-world experiments, video feeds with imbedded Key-Length-Value (KLV) data were used to determine the extrinsic and intrinsic parameters. The onboard GPS, IMU, and magnetometer sensors are sufficient for accurate image rectification and depth generation.

2.2 Essential and Fundamental Matrix

Epipolar geometry is mathematically encapsulated in the essential and fundamental matrices. The Essential and Fundamental matrices describe the mapping of 3D points and pixels respectively between two cameras. The essential matrix, E, can be created from the relative rotation matrix, R, and skew-symmetric matrix of the baseline vector, $[v_b]_\times$, between cameras using Equation 1,

$$E = R[v_b]_\times, \qquad (1)$$

where $[v_b]_\times$ is the matrix representation of the cross product,

$$[v_b]_\times = \begin{bmatrix} 0 & -v_{b_z} & v_{b_y} \\ v_{b_z} & 0 & -v_{b_x} \\ -v_{b_y} & v_{b_x} & 0 \end{bmatrix}. \qquad (2)$$

The variables, K_1, and K_2, represent the 3x3 camera projection matrices for the camera at time steps 1 and 2. In the case of monocular image streams, we assume the intrinsic camera parameters are constant,

$$K_1 = K_2. \qquad (3)$$

The fundamental matrix can be generated from the essential and camera matrices using Equation 4,

$$F = K_2^{-T} E K_1^{-1}. \quad (4)$$

2.3 Epipolar Geometry

The Fundamental matrix can be used to generate the epipoles and matching epipolar lines in each image. The fundamental matrix can be decomposed with singular value decomposition. This will provide the location of the epipole in each image. The epipoles are the point intersection of the baseline vector and the camera planes. All epipolar lines contain the epipole from the associated image. The epipoles represent the left and right null spaces of the Fundamental matrix. Given any pixel point in one image, excluding the epipole, the epipolar line containing the point in the second image can be found. Multiplying the homogeneous 3x1 pixel vector in the first image by the 3x3 fundamental matrix will return the function for a line in the second image. Alternatively, matching epipolar lines can be found by projecting a single point with an assumed depth. The projected point and epipole can be used to generate the epipolar line.

Matching epipolar lines represent the intersection of any plane containing the baseline vector and the camera planes. As seen in Figure 1, the plane P_{e_1} intersection with the camera planes P_c and $P_{c'}$ generates the epipolar line pair l_1 and $l_{1'}$. The epipolar plane represents a slice of 3D space. This 3D slice is what projects back onto each camera plane. Creating the constraint that any 3D point in the slice must exist on the projected lines in each image. This constraint drastically reduces the search space when looking for point correspondence between images.

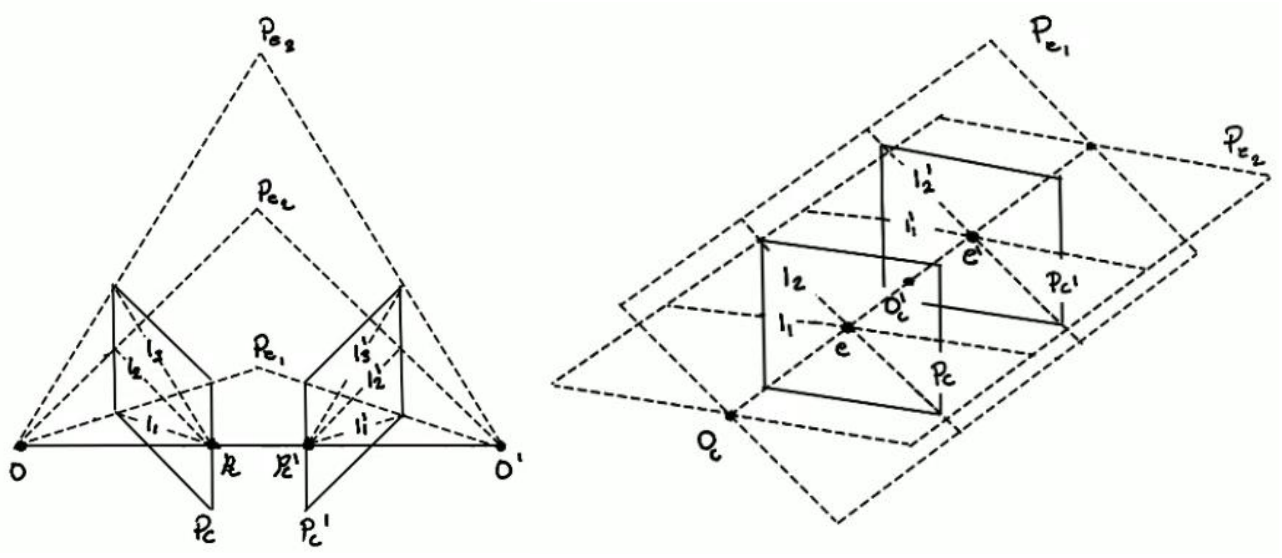


Figure 1. Two cases showing the variables associated with epipolar image calculation. a) A pair of temporally separated images where the UAV moves to the right then turns left. b) A pair of temporally separated images where the UAV is moving in a forward motion.

3. EPIPOLAR IMAGE WARPING

3.1 Epipolar Warping Implementation

Conceptually, by rotating the epipolar plane around the translation vector, the set of all matching epipolar lines can be generated from the intersection of the plane P_e with P_c and $P_{c'}$. With smaller rotations of the epipolar plane, the density of epipolar lines in image space is increase. An optimal strategy for selecting epipolar lines is described in section 3.2.

Matching rotation magnitude with image pixel resolution, an optimal sampling rate that maintains pixel information can be found. A rectified and finite image can be produced by stacking the epipolar line pixel samples as rows. Each epipolar line represents a single row in the warped image. The columns of the warped image are all sampled from the same radial distance from the epipole. Pixel samples along an epipolar line represent the rows of the warped image. Figure 2 illustrates how epipolar lines in the original image map to the warped image. Samples outside of the camera boundary are transferred as black pixel samples. The relative offset between samples outside of the image bounds must be maintained to preserve row alignment in the warped image.

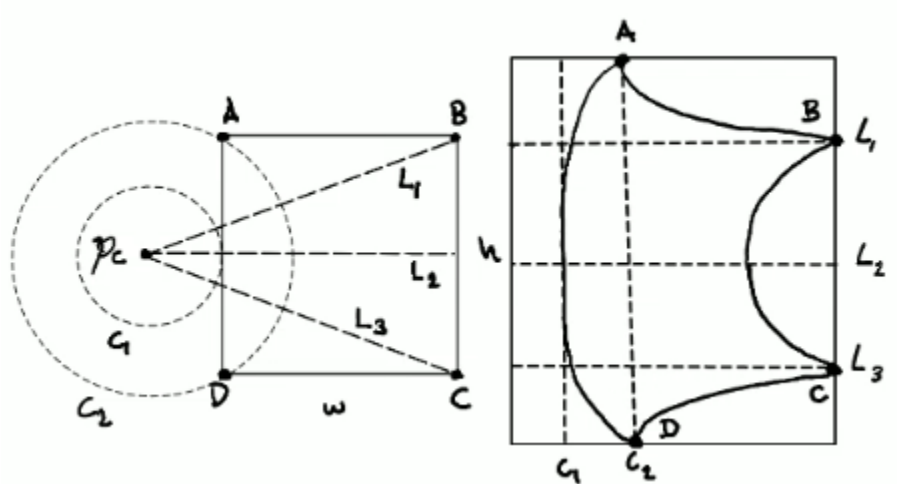


Figure 2. Mapping between original image space and warped image space.

3.2 Epipolar Line Sampling Method

There is an infinite set of matching epipolar lines that can be generated. However, there is a subset of epipolar lines that maintains the original image information with minimal number of epipolar line samples. In this paper we optimize the sampling frequency for the first image, and the matching epipolar lines in the second image are left unoptimized. In practice, not optimizing the sampling of the second image does not noticeably degrade performance or efficiency. As well, epipolar lines that are within the field of view of the first image, but out of the field of view in the second image are left black for the second warped image.

Algorithmically, all epipolar sampling calculations are done using epipolar rays and ray cast intersection methods. Epipolar rays represent one side of an epipolar line split by the epipole. This adds complexity in determining the matching epipolar ray in the second image but reduces the search space to half the epipolar line. The reduced search space is only beneficial when the epipole exists inside of the camera frame.

To determine the optimal number of epipolar lines and spacing between each sample, an iterative algorithm is implemented. This paper follows a sampling theory presented in [6]. Optimal line sampling can be generated with the following pseudo code (see Algorithm 1).

ALGORITHM 1

- Determine clockwise image segment path
  - Determine region in Figure 4 containing epipole
  - Determine segment path from region using segment path look-up Table 2
- Set intersection i_n to first point in image segment path
- Set ray $r_n = \overrightarrow{p_e i_n}$

While:

- $r_t = v_c \times \left(v_c \times \frac{r_n}{\|r_n\|}\right)$ // A vector of length 1, perpendicular to r_n
- $p_t = i_n + r_t$
- $r_{n+1} = \overrightarrow{p_e p_t}$
- Determine new intersection i_{n+1} with image segment path by recasting r_{n+1}
- If no intersection exists or total rotation exceeds 360 degrees BREAK
- Store epipolar sampling ray $r_s = \overrightarrow{p_e i_{n+1}}$ to list

Figure 3 is an illustration of the epipolar sampling process with the epipolar point in region 5. Epipolar sampling starts at point "A" and travels a clockwise path intersecting with segments AB, BC, CD, and DA. Figure 4 shows the mapping

from epipolar position to segment path regions. The arrows in figure 4 indicate how points directly on the region boarders should be mapped.

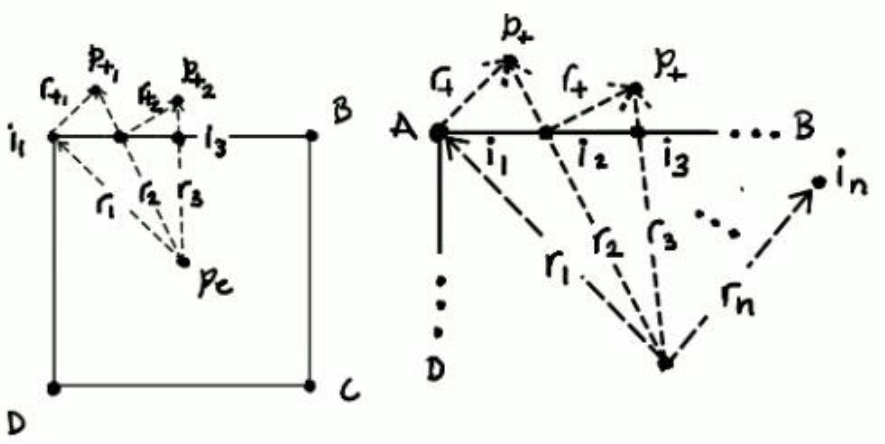


Figure 3. Iterative optimal epipolar ray sampling method.

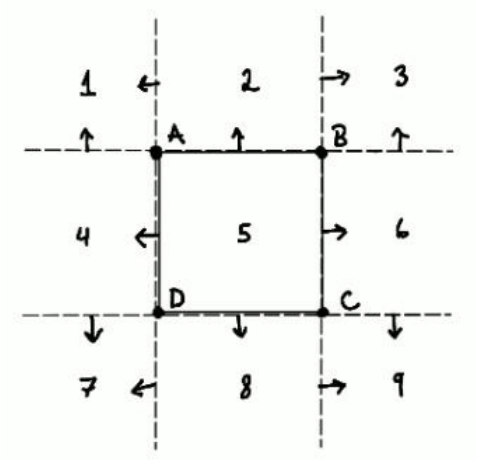


Figure 4. Segment path region definition.

Table 2. Segment path mapping from epipolar point region.

Region	Segment Path
1	B-C-D
2	B-C-D-A
3	C-D-A
4	A-B-C-D
5	A-B-C-D-A
6	C-D-A-B
7	A-B-C
8	D-A-B-C
9	D-A-B

The list of optimal epipolar sampling rays r_s are turned into normalized rays. These rays will be used in the image warping process. In the process of casting rays, ensure normalization does not introduce float underflow. When the camera plane P_c is parallel to the baseline vector v_b, the epipoles will be at infinity. When an epipole is extremely far from the camera origin, normalized rays may not accurately represent epipolar geometry when stored as float32/64 data types.

The epipolar sampling rays in the second image can be found through the following pseudo code (Algorithm 2). It is assumed that image pairs have significant FOV overlap.

ALGORITHM 2
For each epipolar sample r_s in image one: - First image pixel position $p_{xy} = p_e + r_s$ - 3D point p_{xyz} is generated by projecting p_{xy} with assumed depth $10v_b$ - Second image pixel position $p_{xy'}$ is generated by projecting p_{xyz} to second camera - Store matching epipolar sampling ray $r_{s'} = \overrightarrow{p_{e'}\ p_{xy'}}$ to list

These epipolar rays will be used to sample the original image pixels and generate a pair of rectified warped images. The row indices of each image represent matching epipolar lines. The epipolar constraint applies to the rows of the warped image pairs. Corresponding points must exist on the same row in each warped image.

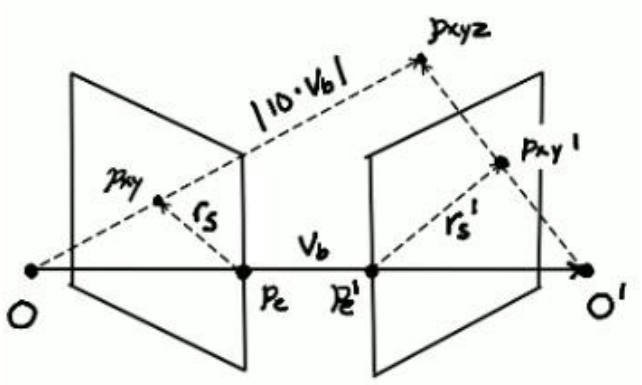


Figure 5. Illustration of creating matching epipolar sampling rays in the second image.

3.3 Pixel Sampling Method

Using the optimal sampling set of epipolar rays, intersections with the image frame segments A-B-C-D-A are found. If the epipole is outside of region 5, each ray will intersect with two lines segments. The first and second intersection indicate entering and leaving the image frame respectively. These intersections with the epipole can be used to create the line segments used for pixel sampling. The epipole and first intersection represent the segment outside of the image border. The first and second intersection represent the sample inside of the image border. When the epipole is contained inside the image border there will only be one intersection for each ray and the segments generated will be inside of the image border.

Pixel sampling begins at the epipole moving outward along the line segment with the step length of one pixel. In cases where the epipole is extremely far from the camera location, iteratively sampling from the epipole in pixel steps is not possible. In this case, the distance from the epipole to the border of the image is calculated for each epipolar ray. Once all epipolar ray lengths to the border of the image have been calculated, the minimum epipolar ray length is removed from each segment. All epipolar lengths are converted to integers before taking the difference to ensure consistent sampling frequency across all epipolar rays. Otherwise, fractional pixel misalignment will occur between the warped image rows. The integer cast of the minimum epipolar ray length is used as the number of black samples taken outside of the image frame. These black samples maintain the alignment of each epipolar line sample. By removing the minimum line segment length outside of the camera frame, the number of black samples is minimized and bounded. The maximum dimensions of the warped image size are defined by the following equations:

$$h_o = pixel\ height\ of\ original\ image$$
$$w_o = pixel\ width\ of\ original\ image$$

$$h_w = pixel\ height\ of\ warped\ image$$
$$w_w = pixel\ width\ of\ warped\ image$$

$$h_w = 2w_o + 2h_o$$
$$w_w = \sqrt{(w_o^2 + h_0^2)}$$

Figure 6. illustrates how the maximum dimensions of the warped image are found. Maximum warped image width is determined from the largest sampling length l_2 after subtracting the minimum black sampling length outside of the image frame. Maximum image height is determined from epipoles in the center of image frame. The number of epipolar samples generated by the iterative ray casting method is bounded by the image frame perimeter when r_t is normalized to pixel size one.

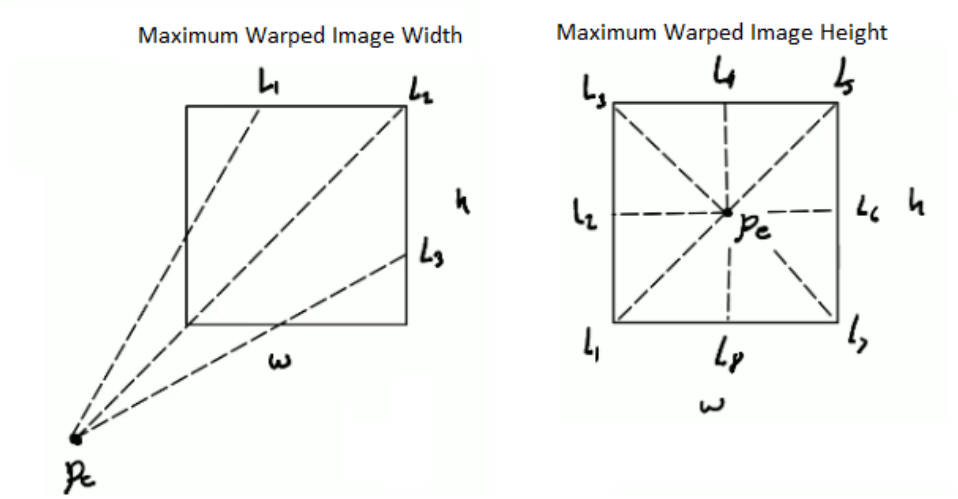


Figure 7. Illustration of derived maximum warped image width and height from original image.

The warped image is generated by sampling the original image along the epipolar rays. Samples will be fractional pixel values. To avoid artifacts from integer rounding, bilinear interpolation of the four nearest pixels is used to generate the new pixel in the warped image.

3.4 Sampling Direction

Depending on the camera extrinsics, one of the warped image pair's horizontal axes can be flipped. Sampling outward from the epipole in each image may result in opposite sampling directions along the line created by the epipolar plane. The 2D camera depictions in Figure 8 illustrates how the camera extrinsics can invert the image. When sampling outward from the epipole in the left camera 1, the features ABC are seen in the order CBA. In the right image the features are seen in the order ABC. When calculating depth, block matching kernels expect the features to be properly aligned. Matching the features ABC to CBA will result in a large error. Figure 9 shows a camera configuration where the epipolar sampling is in the same direction and the orientation of the feature space is consistent between images.

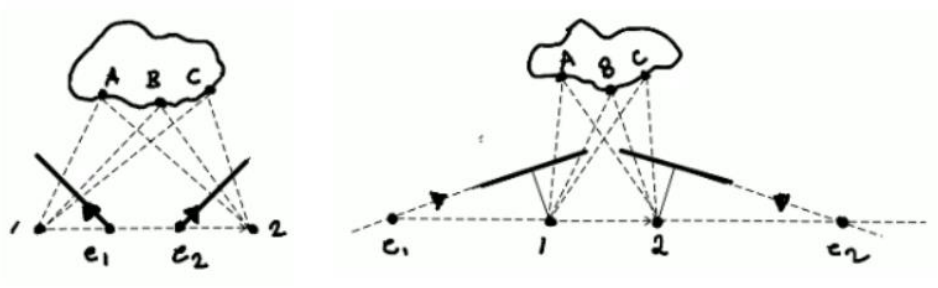


Figure 8. 2D Camera diagrams with arrow indicating epipolar sampling direction. Illustrates camera configurations that can invert image horizontal axis.

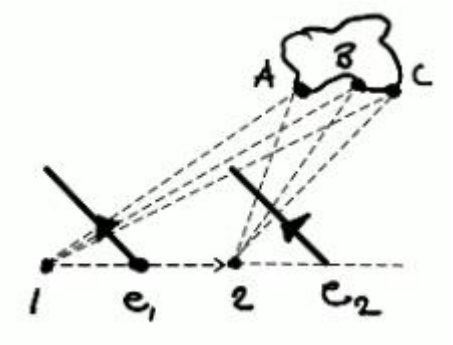


Figure 9. 2D Camera diagram with same epipolar sampling direction. Feature order is preserved.

Depending on the extrinsics of the cameras, it is possible that one warped image's features will be the mirror of the other. One such example is shown in Figure 10. To detect the cases where epipolar sampling direction will flip the horizontal axis of the warped image, the following algorithm (Algorithm 3) is used.

ALGORITHM 3
- Generate the vectors r_s , $r_{s'}$ and $r_t = \overrightarrow{op_e}$, $r_{t'} = \overrightarrow{o'p_{e'}}$ for each image - Calculate the cross products $r_{cross} = r_s \times r_t$ and $r_{cross'} = r_{s'} \times r_{t'}$ - If $sign(r_{cross} \cdot r_{cross'}) < 0$, then the horizontal axis of one of the warped images must be flipped.

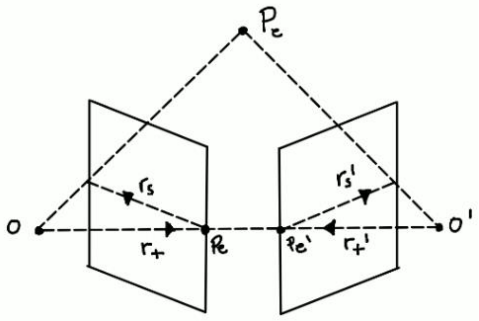


Figure 10. Calculation of vectors used to determine when epipolar sampling inverts the horizontal warped image axis.

Because sampling direction is the same for all epipolar lines in a single image it is only necessary to perform the calculation once after warping.

3.5 Disparity Polarity

Most block matching algorithms are designed with traditional rigidly mounted stereo cameras, translations and rotations are precalculated and constant. Rotations are close to identity and the translation vector is perpendicular to the viewing direction. Some implementations also assume that the translation only contains components along one axis of the image plane, i.e., no diagonal translations. These constraints result in desirable effects on the disparity maps for image pairs. Under these constraints, all disparities will have the same sign. This can drastically reduce the search space and increase point correlation accuracy.

In the general case, depending on the depth of a point in image one, the disparity could be negative or positive. Figure 11 depicts the locations of three points where the disparity is zero between the images. If the depth is increased or decreased the resulting disparity will be negative or positive respectively. Plotting all the points where disparity is zero, the disparity boundary can be formed. The disparity boundary takes the shape of a circle. Disparity of points inside the circle have the same sign. Disparity of points outside the circle will have the opposite sign. The circles size increases as the rotations between the cameras approaches identity. This 2D simplification of an image pair can be extended to a three-dimensional image pair where the circle will instead be an ellipsoid.

The lack of generality of many block matching algorithms puts a subtle restriction on the 3D environments that can be handled given a specific extrinsic pair. Camera configurations that produce both positive and negative disparities have larger search spaces and are more likely to produce poor correspondence. During our practical and simulated testing, rotations between camera pairs is loosely bound around the identity rotation. This causes the disparity boundary to be a significant distance from the camera origins. All points inside the disparity boundary maintain a consistent disparity sign between images. Because of this constraint, highly optimized and publicly available block matching algorithms can be used.

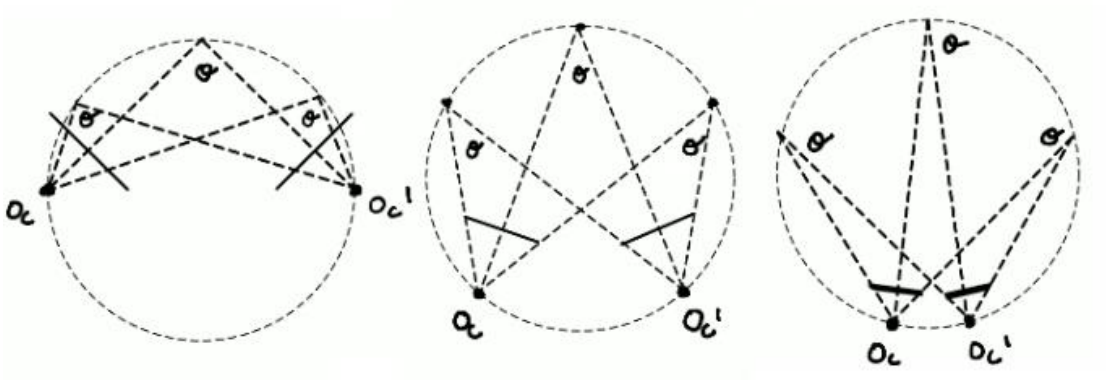


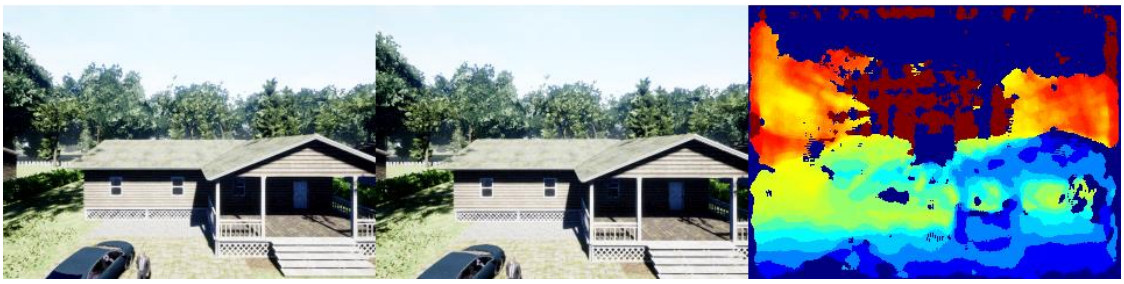
Figure 11. Illustration of zero disparity boundary as rotation approach identity. The inscribed angle theorem can be used to generate the disparity circle for a 2D camera pair. To maintain a constant disparity circle diameter, while decreasing the rotation between the camera pairs, the baseline must decrease. The camera origin points travel along the disparity circles perimeter. Points in space must be at increasingly larger depths to cross the disparity boarder as the rotation nears identity.

3.6 Increased Warping Near the Epipole

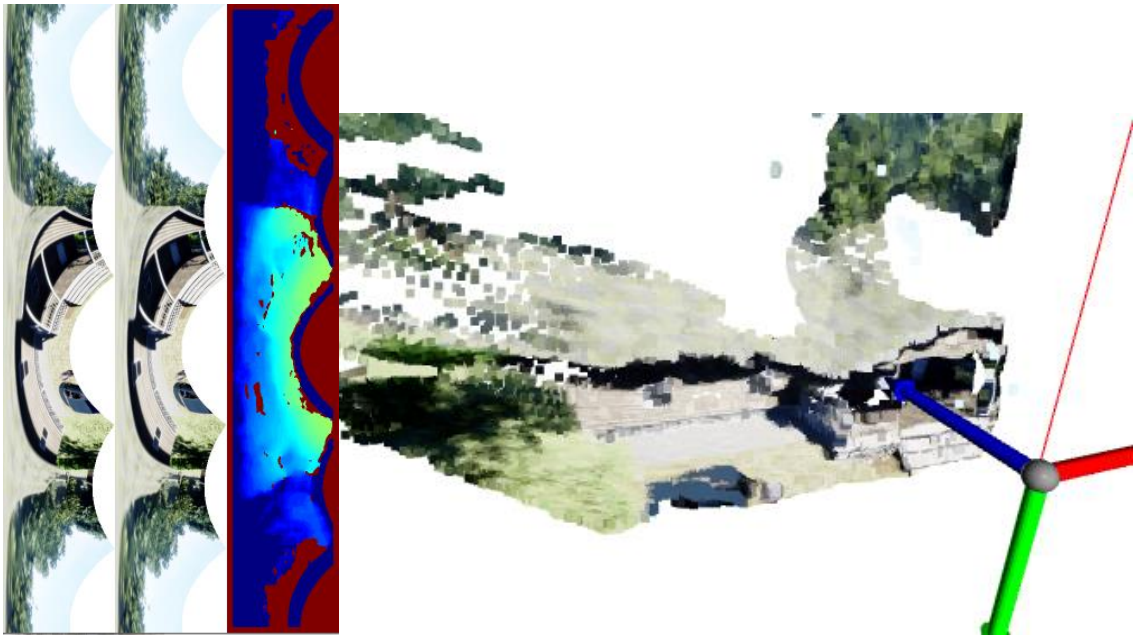
As the epipole approaches or enters the image frame the warped image space becomes significantly distorted. Pixels near the epipole are drastically over-sampled and warped. The columns of the warped image represent pixel samples at the same radial distance from the epipole. Figure 12 shows forward camera motion simulated using Airsim in the Unreal

engine. In this example, the epipole is in the middle of each image. The left side of the warped images, Figure 12 b), show the large distortion that occurs near the epipole.

The correlation between depth and disparity decreases as points approach the epipole. Points on the epipole can map to any 3D point and maintain the same pixel projection in each image. This causes an increase in depth estimation error for points close to the epipole. For this reason, points close to the epipole should be discounted or removed.



a)



b) c)

Figure 12. a) Simulated imagery using AirSim and the Unreal Engine. Forward motion, 0 degrees below horizon. Image order: Image1, Image2, depth. B) Forward motion, 0 degrees below horizon. Image order: warped1, warped2, disparity. C) Projection of Image1 pixel values to 3D points using the calculated depth values

3.7 Warping Look-Up Table

During the process of generating a warped image, a look-up table mapping the pixels from the warped image to the original image is created. Because of the oversampling that occurs near the epipole, the mapping is one-to-many from the original to warped image space. After the depth map is created and unwarped back to the original image space there can be multiple

depth values for a single pixel. To handle the one-to-many mapping, the interpolated pixel that is closest to the integer location of the original pixel location is selected.

3.8 Creating Disparity Map

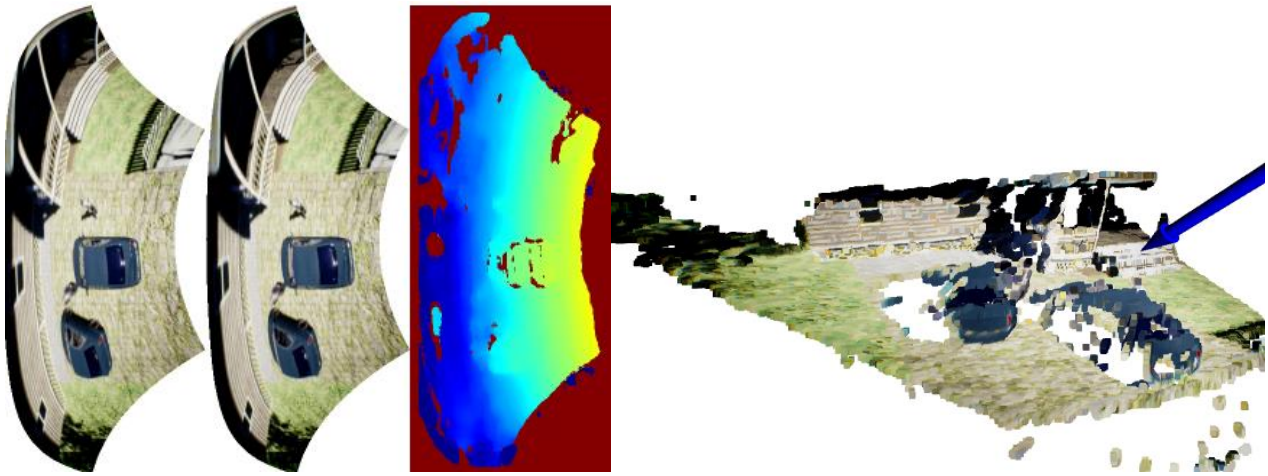
The rows of the warped image pair are rectified. The 2D projection of any 3D point will be contained in the same row of each image. Any block/kernel-based disparity matching algorithm can be used to generate the disparity map. The generated disparity map can be used to determine pixel correspondences. Using pixel correspondences in the warped image and the look-up table, 3D points can be triangulated.

The warped pixel disparity map cannot be mapped back to the original image space because the original image space is not rectified. In the original image, pixel disparities represent movement along the epipolar lines, which can have both x and y pixel components. Instead, the disparity magnitudes and depth values are mapped back to the original image space.

3.9 3D Point Triangulation

Because the camera extrinsics are generic, there are no simplistic methods for generating the depth map from the disparity map. Every point must be individually projected using iterative triangulation solvers. In the generic case, small disparities do not necessarily mean large depth values. The correlation of disparities to depth is largely dependent on the camera configuration. Small disparities can mean close objects and large disparities far objects, which is significantly different from classical stereo vision hardware. The end-to-end EpiDepth process is shown in Figure 13 using simulated data.

a)



b) c)

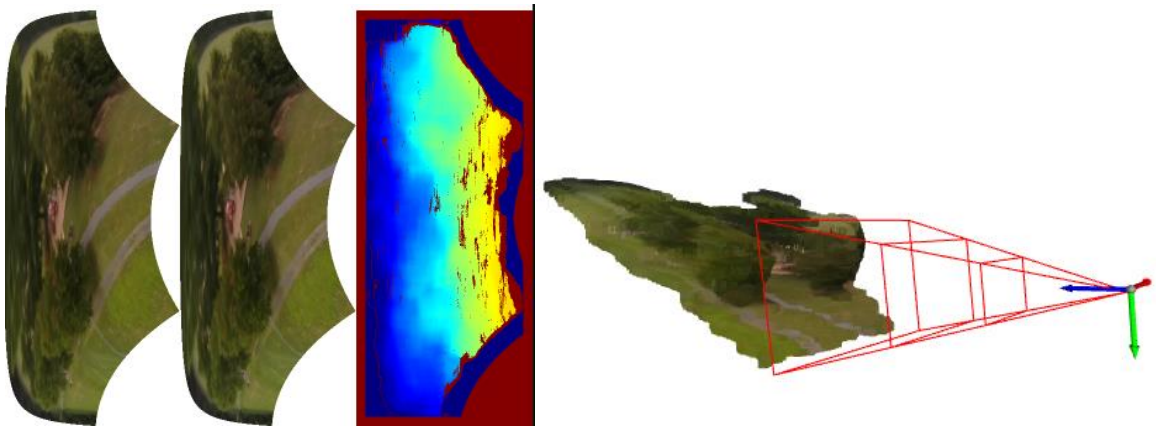
Figure 13. a) Simulated imagery using AirSim and the Unreal Engine. Forward motion, 45 degrees below horizon. Image order: image1, image2, depth. C) Forward motion, 45 degrees below horizon. Image order: warped1, warped2, disparity. C) Projection of Image1 pixel values to 3D points using the calculated depth values

4. EXPERIMENTS

The computer used for all experiments had an Intel i9-8950HK CPU running at 2.9 GHz with 64GB of RAM. A frequency of 2Hz can be achieved on image resolutions of 320x240 using only CPU resources. Processing speed quadratically increases with down sampling. Outdoors, low resolution input images are sufficient for simple navigation tasks.

The following section consists of a series of image pairs generated from a simulated environment using the Black Hornet UAS [9]. Image pairs with a variety of intrinsic and extrinsic parameters were collected and their warped, disparity, depth, and 3D projection images are shown below in Figures 14, 15, and 16. All image pairs are collected temporally from a monocular RGB image stream.

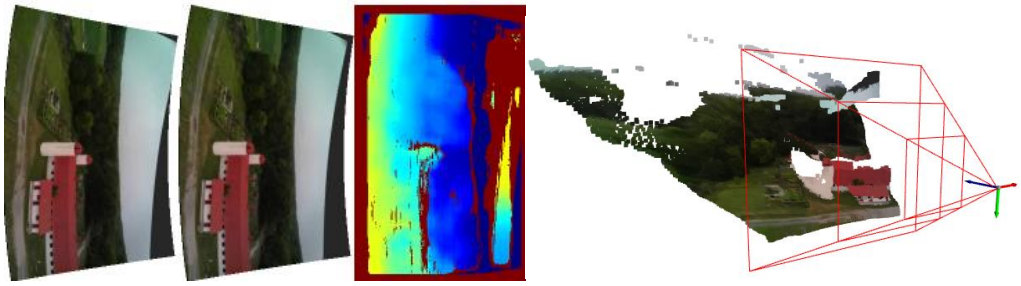
a)



b) c)

Figure 14. a) Forward motion, 45 degrees below horizon. Image order: image1, image2, depth. B) Forward motion, 45 degrees below horizon. Image order: warped1, warped2, disparity. C) Projection of Image1 pixel values to 3D points using the calculated depth values.

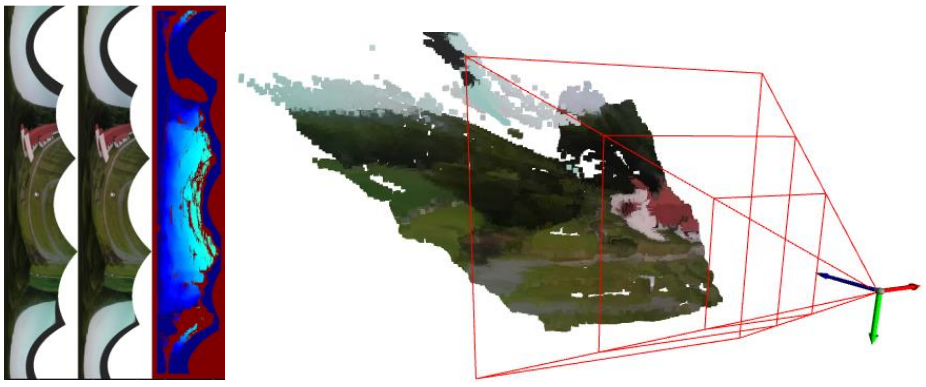
a)



b) c)

Figure 15. a) Altitude climbing motion, 5 degrees below horizon. Image order: image1, image2, depth. B) Altitude climbing motion, 5 degrees below horizon. Image order: warped1, warped2, disparity. C) Projection of Image1 pixel values to 3D points using the calculated depth values.

a)



b) c)

Figure 16. a) Forward motion, 0 degrees below horizon. Image order: image1, image2, depth. B) Forward motion, 0 degrees below horizon. Image order: warped1, warped2, disparity. C) Projection of Image1 pixel values to 3D points using the calculated depth values.

5. SUMMARY AND FUTURE WORK

This paper showed an end-to-end process to calculate depth information from a UAS with a monocular camera. Background was provided for epipolar mathematics, as well as its application to image warping. Matching of warped images was then shown and used to calculate depth values for the generation of dense three-dimensional models of the environment viewed by the UAS. Multiple examples generated from a UAS data collection were provided.

There are significant improvements that can be made to both algorithm efficiency and performance. Faster depth map generation would result in a more agile autonomous system. Most of the processing time is spent in image warping. Algorithmic improvements to warping efficiency would drastically improve processing times. GPU implementation could further improve processing capabilities, with the hope of reaching real-time performance.

A logical extension of this dense depth map generation would be for indoor applications. For accurate indoor navigation task, large image resolutions will be required to resolve small obstacles. This further pushes the necessity of increase processing performance. As well, it is difficult to perform image matching in indoor environments due to large flat colored walls. It may be necessary to calculate image matching using something other than pixel correspondence.

Real drone data uses GPS, IMU, and magnetometer to determine camera extrinsics. These sensors introduce small errors into the EpiDepth process, which result in error in the depth image, or complete failure to the process. Error could be reduced by refining initial extrinsics with image-based, feature matching, motion estimation techniques.

REFERENCES

[1] Stone, K., Keller, J. M., Anderson, D. T., "Moving Beyond Flat Earth: Dense 3D Scene Reconstruction from a Single FL-LWIR Camera," SPIE Defense, Security, and Sensing, (2013)

[2] J. L. Schonberger, J. L., and Frahm J. M., “Structure-from-motion revisited,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016)

[3] Schonberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J. M., “Pixelwise view selection for unstructured multi-view stereo,” in European Conference on Computer Vision (ECCV), (2016)

[4] Zhou, T., Brown, M., Snavely, N., Lowe, D.G.; “Unsupervised Learning of Depth and Ego-Motion from Video”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1851-1858, (2017)

[5] Godard, C., Aodha, O. M., Firman, M., Brostow, G., “Digging Into Self-Supervised Monocular Depth Estimation,” The International Conference on Computer Vision (ICCV), (2019)

[6] Pollefeys, M., Koch, R., and Van. Gool, L., “A simple and efficient rectification method for general motion,” Proc. ICCV, 496–501 (1999).

[7] Shah, S., Dey, D., Lovett, C. and Kapoor, A., “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” Field and Service Robotics, 621–635 (2018)

[8] Epic Games. Unreal Engine. Retrieved from https://www.unrealengine.com, (2022)

[9] Teledyne FLIR. Black Hornet PRS. https://www.flir.com/products/black-hornet-prs/, (2022)