Tanner Bergstrom

1w25cs8u–3

Computer Vision and Pattern Analysis

2025/08/04

Object Distance Estimation in an Image from a Reference Photo

The goal of this project is to find the distance of an object from a camera in an image. A camera projects a snippet of a 3D environment onto a 2D image, eliminating the z-axis, or depth, in the process. The major hurdle in reaching the goal of this project is approximating depth in an image after it has been eliminated.

For approximation, we need context. This context comes from another image. This other image, or reference image, captures a key reference object. The importance of the reference image is that the real-world distance between the camera and the reference object is known. With this reference object, we can find the depth of this object in other images.

The following analysis presents an approach using SIFT descriptor detection, FLANN-based descriptor matching, and image homography to map a reference object to its location within another image and find the distance of the object in that image from the camera through an angular size comparison and focal length conversion. This is realized through a toy program, though the idea is for use as a mobile app. This approach will be referred to as object distance estimation from a reference photo, or ODERP. The only purpose of naming this approach is for internal reference within the scope of this analysis.

Alternative Approach

Apple's *Measure* iOS app was released in 2018. Apple describes this app as using Augmented Reality to measure the real world with the use of an Apple product's camera. To measure within the app, the user must point their camera at one end of the measurement, set

their starting point, then point their camera at the other end and set the end point to complete the measurement. The result is displayed in the AR environment within the phone screen.

This app is built on the ARKit 2 platform. ARKit utilizes device motion tracking, world tracking, and scene understanding to operate an AR environment ("ARKit"). Though the exact methods by which Apple determines real-world measurements in the AR space are unknown, what we do know is that the data from the AR environment is how measurements are determined. Device movement is specifically important as when a user opens the app, they are prompted to "Move iPhone to start," and if the phone is moved too quickly, the user is prompted to "slow down." iPhone Pro models in the range from 12 to 16 utilize LiDAR scanners to approximate depth as well. Apple's approach utilizes computer vision techniques as well as device positioning to create this AR environment from which to measure, whereas ODERP solely utilizes computer vision techniques. Other unique features of ODERP in comparison to *Measure* include monocular distance estimation and single-image object training.

Details of ODERP

OpenCV is an open-source computer vision library that provides most of the tools I used for this program. Importantly, its SIFT, FLANN, and homography functionality and documentation aided this project ("Feature Matching + Homography to find Objects"). SIFT, or Scale-Invariant Feature Transform, is a feature extractor developed by David G. Lowe in 2004 (Lowe). SIFT is an important algorithm for this project as it provides distinct descriptors that can be matched across multiple images.

A good reference object is made up of many descriptors. These descriptors are invariant to scale and rotation, which is necessary for this project as the object being matched across images varies in size within the image and angle towards the camera. For this

program, to utilize the SIFT functionality in OpenCV, SIFT is abstracted into two functions. The first function creates a SIFT object. The second returns the keypoints and descriptors, requiring only an image as a parameter. The function works by first finding keypoints, then computing the descriptors. We then match returned descriptors from both images using FLANN.

FLANN, or Fast Library for Approximate Nearest Neighbor, is a C++ library containing many nearest neighbor algorithms. In OpenCV, a FLANN-based matcher is provided. It is unclear how this FLANN-based matcher is comparing descriptors, potentially by Euclidean distance, a Hellinger kernel, or some histogram-based metric. ("Feature Matching with FLANN")

From the FLANN-based matcher, we use a k-th nearest neighbor match, which returns a list of the 2 nearest neighbors, or 2 best matches, for each descriptor. From here, we want to throw out any matches that are not distinct enough. For this, we use the Lowe's Ratio Test. Each neighbor is assigned a distance, an evaluation of how close to a perfect match it is. If the ratio between the nearest distance and the second-nearest distance is greater than a certain threshold, in this case, 0.7, then the match is thrown out. Then, we use a minimum match count to determine whether or not we perform homography. In other words, we need a minimum number of matches to determine whether an object mapping is valid.

Homography is the perspective transformation of one image onto another. In this project, the reference image is cropped to the outline of the reference object, causing only the reference object to be mapped to the other image. Through homography, we obtain the transformation for projection of the reference object to the image.

To accomplish this, we need the coordinates of the matched descriptors in both the reference image and the destination image. OpenCV's findHomography() function returns the perspective transformation matrix from the reference coordinates to the destination

coordinates. Then, we obtain the coordinates of the reference object within the image using cv.perspectiveTransform(), which takes the coordinates of the cropped reference image and the perspective transformation matrix as input. Once we have the coordinates of the object in the destination image, we can compare the object in both images. What we compare is the angular size of the objects in their respective settings, or images.

Angular size, or apparent size, is a measure of how much visual space an object takes up. From an observation point to an object, imagine two lines, one to the top of the object and one to the bottom of the object. The angle formed between the two lines at the point is the angular size. An object that is closer to the point has a greater angular size than an object that is further away.

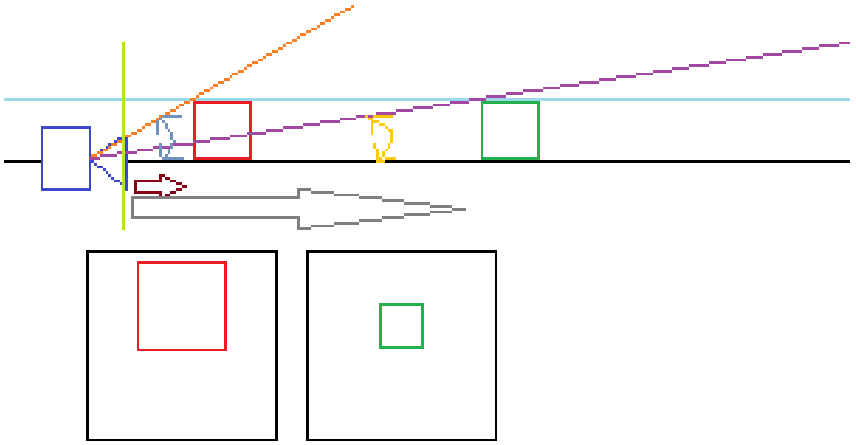

Figure 1. Angular Size and Image Representation

Figure 1 demonstrates angular size and perspective projection. The camera, purple, views an orange cube and a green cube. The orange cube is closer, and as such, the angle and visual space taken up in an image is greater than the green cube. We can find this angle through the tangent. This is the object height, h, divided by the distance between the object and the camera, d, or $\text{ø} = \tan = \text{opp/adj} = h/d$.

What we care about, however, is the distance. So, we get $d = h/\text{ø}$. Because we are comparing the same objects, the height is a constant. From here, we have two formulas, $d1$ =

h/ø1 and *d*2 = *h*/ø2. We know the distance of *d*1 and want to find the distance of *d*2. We know that *d*2 is a scale, *s,* of *d*1, and as such, *d*2 = *d*1*s. Then, *s* = *d*2/*d*1 = ø1/ø2. Because angular size is just a measure of how much visual space an object takes up, we can replace ø with the pixel height, *p*, of the objects in their images. Thus, *d*2 = *d*1**p*1/*p*2.

This program allows for varying focal lengths, or camera zoom. Considering the instrument used for testing this program, an iPhone 11, there are 3 zoom settings: 0.5x, 1x, and 2x. If the reference image was taken with 1x zoom, the zoom of the destination image is entered in comparison, i.e., 0.5x, 1x, or 2x. Finding the distance of an object in an image with zoom applied is simple. Assuming focal length is the only factor that changes, from the same distance, an object's apparent size scales at the same rate as the zoom. For example, the apparent size will double at 2x zoom. This is because as the focal length scales, the angle of view scales inversely. The angle of view is the angle that the image plane captures.

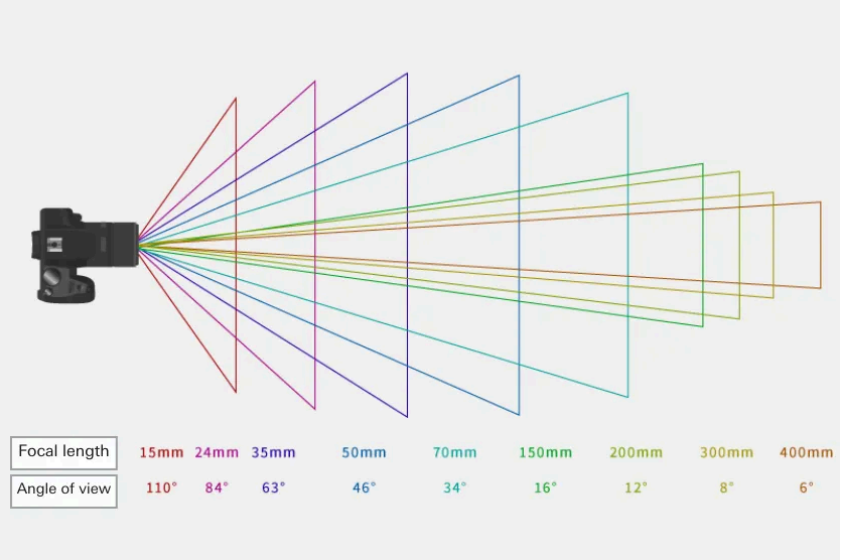

Figure 2. (“What is Angle of View? Learn How to Choose Which Lens to Use”)

Constraints

Because of how varied camera designs are, to ensure consistency, all photos should be captured on the same device. For example, 1x zoom might have distinct meanings across

different devices, such as varying focal length and sensor width. It is also important that all photos are the same resolution, seeing as pixel height is the measurement used for comparison. If an image is shrunk, an object within it will be considered further away than it is. Because SIFT is the keypoint detector used, it is important for reference objects to have many keypoints that SIFT determines to be distinct. This means for best results, reference objects should be planar, relatively large, and have many corners with high visual contrast.

Planar objects are more consistently matched because keypoints are less often lost when the object is rotated than non-planar objects. The larger the object, the better. As objects are pictured further from the camera, they take up less visual space, meaning fewer pixels describing them, thus less detail. Larger objects take up more visual space, so at further distances, details are not as easily lost. Lastly, keypoints in SIFT must be corners, local extrema, so a reference object should have many distinct corners to be matched. Round details and straight lines do not make good features.

Figure 3. Bad Reference Objects for Object Matching with SIFT

Figure 4. Good Reference Objects for Object Matching with SIFT

Demonstration

The first step in using the program is to enter the distance between the camera and the reference object. Then, from a file picker, the reference image is chosen, and then the destination image. Then, the user enters the zoom magnification used on the reference image. Then, the user crops the reference object within the reference image. The program will then display how the object was mapped onto the destination image, and the estimated distance is given.

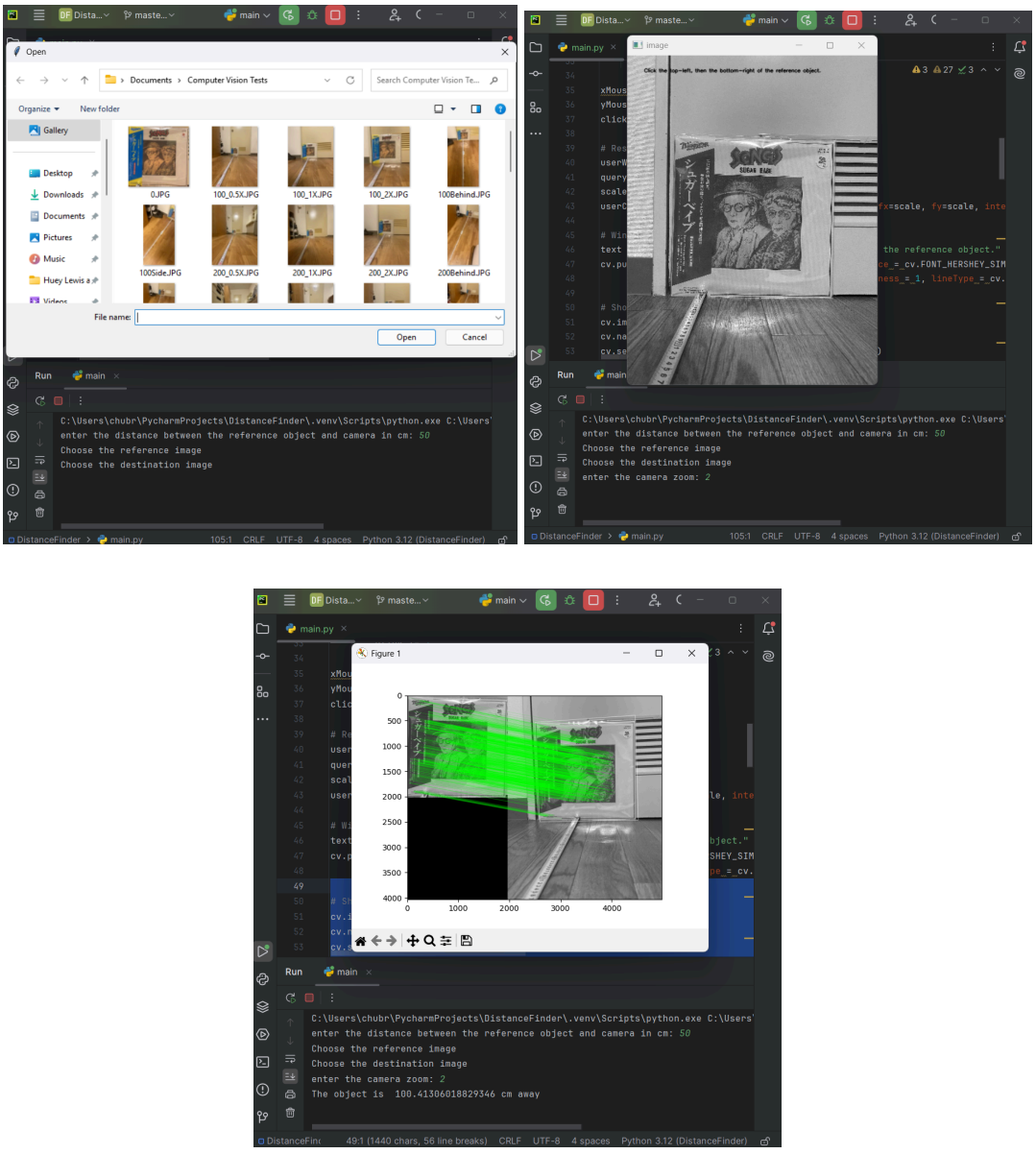

Figure 5. Demonstration of ODERP

Testing

Testing was conducted by photographing the same object at various distances, under various conditions, and with various focal zooms. The same reference image was used across all tests. The reference object is an album vinyl cover and is pictured 50 cm from the camera. The photographing instrument used was an iPhone 11 Pro, capturing images at 0.5x, 1x, and 2x focal zoom. Across all tests, rerunning the program under the same conditions results in seemingly the same output. There could be varied results because of non-deterministic nearest-neighbor algorithms; however, in practice, no changes are observed between iterations.

For the first test, the reference object was photographed at 100, 200, and 300cm. The results are as follows:

	100cm	200cm	300cm
0.5x zoom	97.9593873	186.1927152	288.3251905
1x zoom	98.12012911	191.1097288	294.0536261
2x zoom	100.7018924	201.8078804	302.2650242

Figure 6. ODERP Test One 1x Zoom at 200cm

The most significant difference from the expected distance we see is 0.5x zoom at 200cm, falling about 13.81 cm short. This is also the least accurate measurement, with an error of

-6.9%. The closest approximation we see is 2x zoom at 100 cm, estimating 0.7 cm too far. This is also the most accurate measurement, with an error of only +0.76%.

For the second test, the conditions were the same as the first test; however, obstructions were placed in front of the object. These obstructions acted as uneven terrain for taking a ground-level picture of the object. The results are as follows:

	100cm	200cm	300cm
0.5x zoom	94.80469823	189.0924811	Fail
1x zoom	98.43891263	197.8622079	299.4870901
2x zoom	100.6358504	201.8941641	288.2526875

Figure 7. ODERP Test Two 1x Zoom at 200cm.

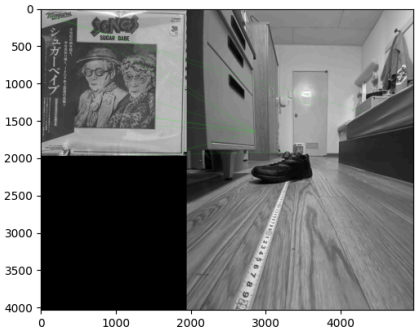

Figure 8. ODERP Test Two 0.5x Zoom at 300cm. Failure to map.

0.5x zoom at 300 cm failed to map correctly, leading to a completely incorrect measurement. The most significant difference from the expected distance we see is 2x zoom at 300 cm, falling about 11.75 cm short. The least accurate measurement we see is 0.5x zoom at 200 cm with an error of -5.5%. The closest approximation we see is 1x zoom at 300 cm, falling about 0.51 cm too short. It is also the most accurate measure, with an error of -0.17%.

For the third test, the object is placed in a new context, slightly askew from the camera, without obstruction. Because of the limitations of the test environment, only 100 and 200cm could be measured. The results are as follows:

	100cm	200cm
0.5x zoom	91.66664481	186.8218064
1x zoom	94.48593855	195.7671404
2x zoom	99.83258843	197.2521544

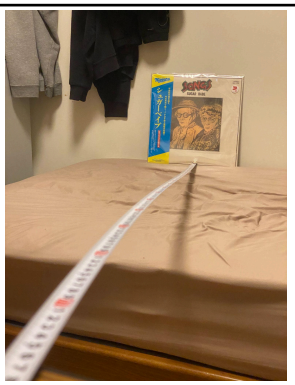

Figure 9. ODERP Test Three 2x zoom at 200 cm.

The most significant difference from the expected distance we see is 0.5x zoom at 200 cm, falling about 13.18 cm short. The least accurate measurement we see is 0.5x zoom at 100 cm with an error of -8.3%. The closest approximation we see is 1x zoom at 100 cm, falling about 0.17 cm too short. It is also the most accurate measure, with an error of -0.17%.

It is difficult to come to any firm conclusions from this data, as it is too limited. The conditions under which the data were collected also gave room to significant error. For example, to capture the reference image, the base of the iPhone was placed 50 cm from the reference object. However, it could be that the iPhone camera, at the top of the device, was tilted closer or further than 50 cm to the reference object, potentially skewing all of the data. Similarly, all other photos were taken under the same conditions. The result is that the data are approximations of approximations.

What can be inferred, though, is that the 2x zoom estimated distance will be greater than the 1x zoom estimated distance, which is greater than the 0.5x zoom estimated distance for most given distances. This implies that there are more factors involved in the resulting apparent size of objects than focal length, at least for the camera model used in these tests.

Comparison to Apple AR *Measure*

Here, we compare the results of ODERP to the output of the Apple *Measure* app. It is important to note that this version of *Measure* does not use LiDAR scanning. Testing was conducted at the same intervals and under the same conditions as the tests for ODERP. Distances were measured in two ways. First, from behind one end of the measurement and in line with both ends of the measurement, the point was set in the AR environment. Then, without moving and only tilting the camera to the other end of the measurement, the other point was set. This is referred to as a behind measurement. Second, standing to the side of both ends of the measurement, one end of the measurement was set, then panning over to the other end of the measurement, the other side was set. This is referred to as a side measurement.

For the first test, the conditions were the same as the first test for ODERP. The results are as follows:

	100cm	200cm	300cm
Behind	97	173	305
Side	99	203	302

Figure 10. iOS Measure Test One 100cm from the side.

The most significant difference from the expected value was a behind measurement at 200cm, falling 27 cm short. This is also the least accurate measurement with an error of -13.5%. The closest approximation we see is a side measurement at 100cm, falling 1 cm

short. The most accurate measurement was a side measurement at 300cm with an error of +0.67%. The most significant difference between both AR methods was 30cm at 200cm. The smallest difference between methods was 2cm at 100cm.

For the second test, the conditions were the same as in the second test for ODERP; however, the obstructions acted as uneven terrain rather than blocking the view of the object, as the reference object is not important to AR measurement. The results are as follows:

	100cm	200cm	300cm
Behind	79	214	326
Side	94	199	266

Figure 11. iOS Measure Test Two 100cm from behind (note that the endpoints shifted but were initially placed at the appropriate positions.)

The most significant difference from the expected value was a side measurement at 300cm, falling 34 cm short. The least accurate measurement is a behind measurement at 100cm with an error of -21%. The closest approximation we see is a side measurement at 200cm, falling 1 cm short. This is also the most accurate measurement with an error of -0.5%. The most significant difference between AR methods was 60cm at 300cm. The smallest difference between methods was 15cm at 100cm and 200cm.

For the third test, the new environment presents a measurement taken across a ground plane with drastic height differences. The measurement for 100cm takes place midair. This is a significant challenge for the *Measure* app as to set a measuring point, something must be

locked onto. Without the presence of the measuring tape, the *Measure* app would have nothing to lock onto at 100cm. The results are as follows:

	100cm	200cm
Behind	48	93
Side	102	182

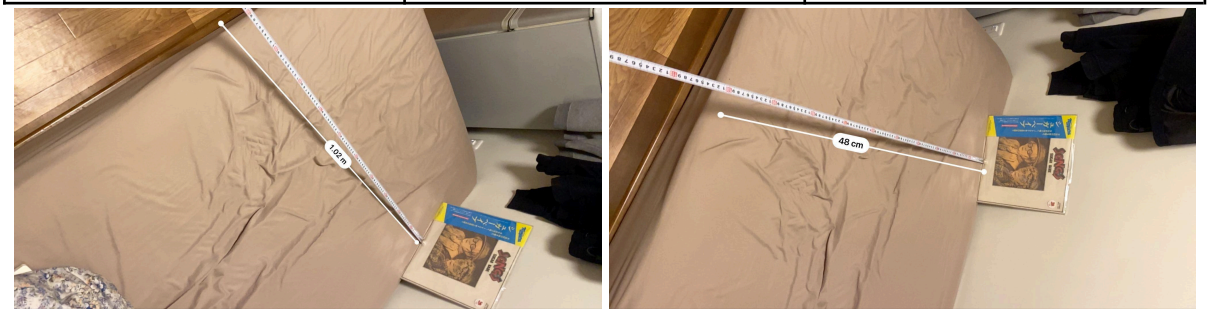

Figure 12. iOS Measure Test Three 100cm from the side. 100cm from behind.

The most significant difference from the expected value was a behind measurement at 200cm, falling 107 cm short. This is also the least accurate measurement with an error of -53.5%. The closest approximation we see is a side measurement at 100cm, 2 cm too long. This is also the most accurate measurement with an error of +2%. The most significant difference between methods was 89cm at 200cm. The smallest difference between methods was 54cm at 100cm.

This data, overall, is limited. Still, we can make some important observations. One, side measurements seem to perform much better than behind measurements. Two ideas for why this might be the case are that side measurements provide more scene data and more phone movement data. And two, measuring across flat surfaces seems to provide more accurate estimations.

In comparison to ODERP, Apple's *Measure* app is as accurate in the best cases. In the worst cases, however, *Measure* had an error of -53.5% whereas ODERP had an error of -8.3%. It is important to consider, ODERP requires an accurate measurement for the reference object and an accurate crop for outlining the reference object. The error can be far

greater given that either of these conditions is inaccurate. It is also important to note that ODERP failed where *Measure* technically never did.

Figure 13. 170cm measured as 348.

Both approaches function very differently and thus, excel in different areas. For example, ODERP can measure in midair, whereas *Measure* cannot. Also, given a measurement with a large obstruction in the middle, *Measure* could measure this from the side, whereas ODERP, without a view of the reference object, could not. Given an ideal environment for *Measure*, a measurement from the side across a flat surface, in these tests, across all distances, the average difference from the expected is 2cm. Given an ideal environment for ODERP, 2x zoom with no obstructions, the average difference is 1.59cm. If we round to the nearest cm, as *Measure* does, this average is 2cm. In terms of accuracy, the claim that either approach is better across the board cannot be made. However, often, ODERP can be more accurate. A major drawback of ODERP in its current form is a lack of user friendliness. In this regard, *Measure* has it surely beat.

Applications and Improvements

As a toy program, ODERP could have applications to situations where many rough estimate measurements are needed. For example, in home construction, an estimator may need the measurements of rooms to estimate drywall, electrical, and flooring costs.

Potentially, this estimator could bring a reference object, set it in one corner of a room, and take pictures of this object from across the room. Then, they could bring these pictures home and run the program for each picture to get their measurements. This niche, however, is fulfilled by other tools such as laser distance measurers, which are much more convenient. For a purpose like this, ODERP would be more practical in use within a mobile app where common construction equipment is already trained, and all the user must do is take a picture of a select construction object.

To improve ODERP, firstly, algorithms like ORB or SURF could be used in place of SIFT, as they offer immense calculation speedup. This would be a great quality-of-life improvement as ODERP is slow to compute. Alternatively, deep learning object recognition could be used instead. The benefit of this is that simple, round objects could be used as reference objects, as descriptor-generating algorithms generate few descriptors for such objects. It is also important to tailor the focal zoom conversion calculation to each device. In its current state, ODERP only considers the focal length in conversion. However, more factors affect the angle of view, such as the sensor size of the camera and the distance between the lens and the image. Focal length is the most significant factor for angle of view, and thus, the current conversion is an approximation. As we can see in testing from the differences between focal zooms, though, an improvement to the conversion could be made. The difficulty is that cameras are varied in design. An iPhone 11 Pro has 3 separate cameras, all of which have differing sensor sizes. The distance between the lens and the image is also negligible, as the cameras are small. A more traditional camera only has 1 sensor size, but the distance between the lens and the image is significant between focal lengths. Overall, ODERP ideally is tailored to devices.

Conclusion

Overall, ODERP offers a monotonic depth estimation approach with the use of computer vision, requiring only one physical measurement for estimation across any number of images that contain the reference object. Given appropriate reference object measurement, imaging conditions, and cropping, ODERP provides seemingly reliable approximations, with the least accurate measurement in this series of tests falling 8.3% short. In comparison, *Measure* requires appropriate device movement, endpoint setting, and scene understanding to ensure appropriate estimation. *Measure* is also more user-friendly compared to ODERP in its current form but cannot measure points not bound to visible space. ODERP has much room for improvement in terms of computational speed, ease of use, and camera zoom calibration, however, the toy example ODERP provides a foundation for a handy application.

Works Cited

"ARKit." *Apple Developer Documentation*, 2025, developer.apple.com/documentation/arkit

"Feature Matching with FLANN," *Open Source Computer Vision*, 17 Jun. 2025, docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html

"Feature Matching + Homography to find Objects," *Open Source Computer Vision*, 3 Aug. 2025, docs.opencv.org/4.x/d1/de0/tutorial_py_feature_homography.html

Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints," *Computer Science Department, University of British Columbia, Vancouver, B.C., Canada*, 22 Jan. 2024.

"Use the Measure App on Your iPhone, iPad, or iPod Touch." *Apple Support*, 12 Mar. 2025, support.apple.com/en-us/102468

"What is Angle of View? Learn How to Choose Which Lens to Use," *Tamron*, 30 Sep. 2024, www.tamron.com/global/consumer/sp/impression/detail/article-what-is-angle-of-view.html