

Depth from a single image

D.A. Forsyth,

University of Illinois at Urbana Champaign

Depth from single image



Data example from NYUV2

What should a depth predictor predict?

- Absolute depth:
 - Might be very difficult to get right
 - compare:
 - zoomed picture of a doll-house room
 - picture of real room
- Relative depth:
 - depth up to per-image scale
 - more likely to get this right, but harder to use

Scale and error

- Nearby objects are more interesting than distant ones
 - you collide with them first
- Want smaller error in small depths than in large depth
 - squared error tends to yield small errors in large predictions
- Relative error = $\text{error}/(\text{true value})$
- Want smaller relative error in small depths, too

What should a depth predictor predict?

- Disparity = (1/depth)

tends to be dictated by the error in large depths. The *inverse depth*, sometimes known as *disparity* (Section 15.10) has a convenient property. Assume the true depth is d , so true disparity is $\delta = 1/d$. If the predictor predicts disparity with some error ϵ , then the corresponding depth is

$$\frac{1}{\delta + \epsilon} \approx d - \epsilon d^2$$

which means that a fixed error in disparity is a bigger error in large depth and a smaller error in small depth.

Depth predictors mostly predict

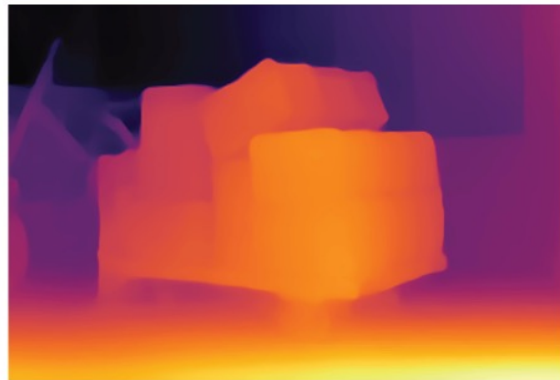
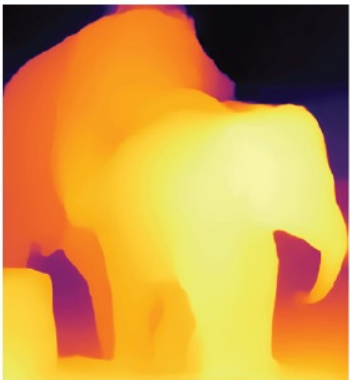
$$a_d \delta + b_d$$

where a_d and b_d are constants, determined *per image*. At training time, these

Losses

Depth prediction losses tend to be a mixture of L_1 and L_2 norms on the prediction

Depth predictors work really well



Evaluating depth predictions

- Absrel

Desirable properties of a depth predictor are: (a) accuracy and (b) good zero-shot behavior. One key metric for accuracy of depth predictors is *AbsRel*. Compute the depth prediction from whatever the predictor produces for an image. You might need to use the ground truth depth map for that image to do this. For example, if it predicts $a_d\delta + b_d$, use the depth map to extract a_d and b_d and then recover predicted depth \hat{z}_{ij} for the i, j 'th pixel. Write z_{ij}^* for the ground truth depth at each pixel. Now evaluate the mean over all pixels and all images of

$$\frac{|\hat{z}_{ij} - z_{ij}^*|}{z_{ij}^*}.$$

- Delta

Another metric, usually called δ , measures the fraction of pixels such that

$$\max\left(\frac{\hat{z}_{ij}}{z_{ij}^*}, \frac{z_{ij}^*}{\hat{z}_{ij}}\right) > 1.25.$$

Rough SOTA

As of writing, depending on the dataset and on the model, for depth predictors you could expect an AbsRel of between 0.06 and 0.25, a δ of between 1.9% and 25%, and a frame rate of between 5 and 90 FPS. For normal predictors, you could

- Generally, faster is less accurate
 - (good sign that payoff makes sense)

Open source predictors

Resource pointers: 20.1 *Some strong open source depth predictors*

These include:

- MiDAS <https://github.com/isl-org/MiDaS>;
- Depth-Anything <https://github.com/LiheYoung/Depth-Anything>;
- Omnidata <https://github.com/EPFL-VILAB/omnidata>.
- https://github.com/baegwangbin/surface_normal_uncertainty

These links also connect to information about available datasets. In fact, the Omnidata link leads to a pipeline that builds datasets, as well as predictors for depth and other surface properties.

Dataset resources

Resource pointers: 20.2 *Some strong datasets used for depth prediction*

These include:

- NYUV2, a dataset of scans of indoor scenes; contains images, depth and segmentation information (https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html).
- ScanNet, a dataset of scans of indoor scenes; contains images, depth and segmentation information (<http://www.scan-net.org>).
- Kitti, a dataset of outdoor scenes mostly to do with cars and driving, with depth and other information (https://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction).

Mostly, if you need normals, you need to estimate them from the depth maps.

Things to think about...

- Limits for single image depth prediction?
 - speed
 - accuracy
 - spatial scale of depth predictions
 - synthetic image training
- (eg depthanything v3)

