Learning for 3D
Recognition and 3D reasoning

Recognition → Statistical properties of visual world → Reconstruction

Reconstruction → Physics of image formation → Recognition
Recognition and 3D reasoning

- Recognition
- Reconstruction
- Statistical properties of visual world
- Physics of image formation
Pose estimation in 3D

Pose estimation in 3D

• Key idea: know relative lengths of each limb

• Assume *scaled orthographic projection*
  • Valid when variation in depth much smaller than depth

\[
x = \frac{X}{Z} \approx \frac{X}{Z_0}
\]
\[
y = \frac{Y}{Z} \approx \frac{Y}{Z_0}
\]

Pose estimation in 3D

\[
l^2 = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2
\]

\[
(u_1 - u_2) = s(X_1 - X_2)
\]

\[
(v_1 - v_2) = s(Y_1 - Y_2)
\]

\[
dZ = (Z_1 - Z_2)
\]

\[
\Rightarrow dZ = \sqrt{l^2 - ((u_1 - u_2)^2 + (v_1 - v_2)^2)/s^2}
\]
Pose estimation for rigid objects
Pose estimation for rigid objects

Pose estimation for rigid objects

- Azimuth / yaw
- Elevation / pitch
- Cyclo-rotation / roll
Viewpoint-conditioned pose

Viewpoint-conditioned pose

Recognition and 3D reasoning

- Recognition
- Reconstruction

Statistical properties of visual world

Physics of image formation
Disparity estimation
Disparity estimation
Disparity estimation

• Goal:
  • Assign disparity value to each pixel

• Basic idea:
  • Disparity image should be smooth

• Energy minimization
  • $\min E(d)$, where $d$ is disparity image
  • $E(d) = E_{\text{data}}(d) + E_{\text{smoothness}}(d)$

• $E_{\text{data}}(d)$: scores based on NCC (for example)

• $E_{\text{smoothness}}(d) = \sum_{i,j} \rho(d(i, j) - d(i, j + 1)) + \rho(d(i, j) - d(i + 1, j))$
Measuring patch similarity is hard
Measuring patch similarity is hard

- Idea: learn to compute patch similarity?
The KITTI Dataset and Benchmark

Stereo pair from here

Depth from here
The KITTI Dataset and Benchmark

Learning patch similarity for disparity estimation
Learning patch similarity for disparity estimation
Learning patch similarity for disparity estimation

Learning stereo without depth supervision

• Given scenes with $\geq 3$ rectified views
• Use 2 views to produce depth
  • Compute scores for each disparity
  • Match pixel to pixel with best disparity
  • Disparity = $1/\text{depth}$
• Use depth to produce 3rd view
• Use reconstruction error
Plane-sweep stereo

- Stereo till now:
  - Go pixel by pixel
  - For each pixel, compute score for each disparity

- Plane-sweep:
  - Go disparity-by-disparity (or depth-by-depth)
  - For each disparity, compute scores for all pixels
Plane-sweep stereo

- For every possible depth value $d$:
  - Assume every pixel in middle image has the same depth $d$
  - Use $d$ to compute disparity to left and right image
  - Reproject left and right images to center coordinate system using disparity
  - Compute score for all pixels
Plane-sweep stereo

If depth is correct, appearance should match!
Plane-sweep stereo

- Score for pixel only depends on the two patches at that pixel
- Can be computed using *convolutions*
Deep Stereo

Estimating depth from a single image

• Why is this even possible?
Estimating depth from a single image

• Why is this even possible?
Estimating depth from a single image

- Why is this even possible?
Estimating depth from a single image

• Why is this even possible?
Estimating depth from a single image
Estimating depth from a single image

Estimating depth from a single image

- Yet another image-to-image translation
- Again, resolution issues

Metric depth is a bad target
Metric depth is a bad target

• Only relative depths matter
• Only logarithmic scales matter

\[ D(y, y^*) = \frac{1}{n^2} \sum_{i,j} ((\log y_i - \log y_j) - (\log y_i^* - \log y_j^*))^2 \]

Humans perceive surface normals, not just depth, through a combination of various pictorial cues.
Estimating normals from a single image

Estimating normals from a single image
Estimating normals from a single image

Estimating normals from a single image