**Goal**
Segment dynamic objects given a single space-variantly blurred image of a 3D scene captured using a hand-held camera.

**Challenges**
- Single image
- Camera/object motion \(\Rightarrow\) motion blur
- 3D scene \(\Rightarrow\) defocus blur
- General camera motion/3D scene \(\Rightarrow\) space-varying blur
- Depth-motion ambiguity

**Our approach**
- Train a CNN to predict the composite kernel \(h_0\) at each pixel
- Composite kernel is convolution of defocus \(h_o\) and motion \(h_m\) kernels
- Use defocus cue to recover the depth map
- Use motion kernels to segregate the dynamic objects at each depth layer
- Joint model for defocus and motion helps resolve depth-motion ambiguity.

**Scene segmentation**
Layer with maximum area in depth map = Reference depth layer \(d_0\)

**Segmenting moving objects in the reference depth layer \(d_0\)**
Blur on dynamic object pixel \# Blur on pixel affected only by camera motion

\[
g = \sum_{k=0}^{m} \omega_k h_o(i, k)
\]

**Blurr compatibility test [1]**
- Select two pixels with motion kernels \(h_{o1}\) and \(h_{o2}\)
- Find \(n_{12} = (d_0 - h_{o1}(i, k) - h_{o2}(i, j)) / \delta\), where \(\delta = 1, 2, 3\)
- Calculate \(n_{12} = n_{12} \cap n_{22}\)
- Regenerate \(h_{o1}\) and \(h_{o2}\) using \(n_{12}\)
- The two pixels are NOT "blur compatible" if \(h_{o1}\) and \(h_{o2}\) have positive entries at locations other than those in \(h_{o1}\) and \(h_{o2}\)

**Segmenting moving objects at other depths \(d_k\)**
Depth map and motion experienced by reference layer are known \(\Rightarrow\)
Kernel at a pixel lying on any other depth layer can be determined

- Compute relative depth \(d_k = d_0 / d_k\) from \(d_{o1} \div d_{o2} = \frac{1}{d_{o1}} \div \frac{1}{d_{o2}}\)
- Where \(\delta = \text{aperture radius}\)
- Estimate \(\omega_k\) using the method in [2]
- Calculate \(H_{o1}(h_{o1}, h_{o2}, H_{o2})\), where \(H_{o1} = \frac{1}{d_{o1}} \div \frac{1}{d_{o2}}\)
- The motion kernel \(h_{o2}\) at any other depth \(d_k\) can be estimated from equation (1) with \((d_0, k)\) replaced by \((d_k, j)\), where \((d_k, j)\) is obtained by applying \(H_{o1}\) on \((d_0, j)\)
- Let \(h_{o2}\) = motion kernel predicted by our CNN.
- Cross-correlation \((h_{o1}, h_{o2})\) < threshold \(\Rightarrow\) dynamic pixel

**Results**
- GT
- [1]
- [2]
- D^3M

**Kernel classification using CNN**

**References**