FlowNet - Learning Optical Flow with CNNs

Prahal Arora
What is Motion ... 

There are two possible answers.

1. First, physical movement of pixels, therefore motion has to be measured in a physical way.

1. Second, motion is human percept--motion is what we perceive in our brain, something we can sense and communicate
What do you observe?
Same video in slow motion
What is Optical Flow ...

Apparent motion of objects as perceived by an observer (eye or camera)

Measure objects/pixels displacement across 2 consecutive video frames/images
Optical flow
Why is it so important to study optical flow:

- Video compression
- Tracking of objects
- Driving assistance systems
Visualizing optical flow field

- Flow can be visualized using vectors
  ○ visualization quickly becomes unreadable

- Use HSV (Hue Saturation Value) components
  ○ codify the direction using color (Hue)
  ○ codify the magnitude of the movement by color intensity
Visualizing optical flow field

- Stationary object (white)
- Moving object (color = direction)
FlowNet

- Input: 2 images
- Output: Optical Flow Field
- Trained end to end
Background

- **Goal:** Estimate Optical Flow (Motion) from images

- **Related work:**
  - DeepMatching - Interpolate dense flow fields, preserve image boundary [Revaud 2015]
  - DeepFlow - Aggregate features from fine to coarse [Weinzaepfel 2013]
  - EpicFlow - Focus on sparse matching (subset of pixels) [Revaud 2015]
  - FlowNet - Real-time, fast, and correlated features learning [Fischer 2015]
The contracting part of the network extracts a rich feature representation. Two architectures are proposed for the contracting part.
In simpler one, input image are stacked jointly
Alternatively, two images are processed separately, then the features are correlated and processed further.
Correlation layer

- Measuring patch similarity in neighborhood
- Convolution of 2 data, no weights

\[ c(x_1, x_2) = \sum_{o \in [-k,k] \times [-k,k]} \langle f_1(x_1 + o), f_2(x_2 + o) \rangle \]
Optimization trick

- Compute correlation only in neighborhood of size \( D = 2d + 1 \)

- Reduce total computation from \( c \times w^2 \times h^2 \) to \( c \times w \times h \times D^2 \)

Where,

\( D \): Neighborhood size
\( d \): Maximum displacement
\( w \): Width
\( h \): Height
\( c \): Number of channels
The expanding part of the network, produces high resolution flow from the coarser features
It makes use of upconvolution layers (deconvolution and transposed convolution) with concatenation of downsamol data
Post processing: Variational refinement

- small motions (first row) the predicted flow is changed dramatically.

- larger motions (second row), big errors not corrected, but the flow field is smoothed, resulting in lower EPE
Loss and performance metric used

\[ L_{epe} = \frac{1}{N} \sum \sqrt{(U - U')^2 + (V - V')^2} \]

- L: Loss/EPE
- U, V: Ground Truth (horizontal, vertical displacement)
- U', V': Flow Estimate (horizontal, vertical displacement)
- N: Total Pixels in Image
Dataset

- Middlebury (Moving, rigid scenes)
- KITTI (Street view scenes)
- Sintel (Animation movie)
- Flying Chairs (Background (from Flickr) + chair + transformation)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frame pairs</th>
<th>Frames with ground truth</th>
<th>Ground truth density per frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middlebury</td>
<td>72</td>
<td>8</td>
<td>100%</td>
</tr>
<tr>
<td>KITTI</td>
<td>194</td>
<td>194</td>
<td>~50%</td>
</tr>
<tr>
<td>Sintel</td>
<td>1,041</td>
<td>1,041</td>
<td>100%</td>
</tr>
<tr>
<td>Flying Chairs</td>
<td>22,872</td>
<td>22,872</td>
<td>100%</td>
</tr>
</tbody>
</table>
Flying chair dataset

- Random background images from Flickr
- Overlay segmented images of chairs
- Randomly sample affine transformation parameters for the background and the chairs
- Data have little in common with the real world
- Generate arbitrary amounts of samples
## Results

### FlowNet worse

<table>
<thead>
<tr>
<th>Method</th>
<th>Sintel Clean</th>
<th>Sintel Final</th>
<th>KITTI</th>
<th>Middlebury train</th>
<th>Middlebury test</th>
<th>Chairs</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train</td>
<td>test</td>
<td>train</td>
<td>AEE</td>
<td>AAE</td>
<td>test</td>
<td>CPU</td>
</tr>
<tr>
<td>EpicFlow [30]</td>
<td>2.40</td>
<td>4.12</td>
<td>3.70</td>
<td>6.29</td>
<td>3.47</td>
<td>3.8</td>
<td>0.31</td>
</tr>
<tr>
<td>DeepFlow [35]</td>
<td>3.31</td>
<td>5.38</td>
<td>4.56</td>
<td>7.21</td>
<td>4.58</td>
<td>5.8</td>
<td>0.21</td>
</tr>
<tr>
<td>EPPM [3]</td>
<td>-</td>
<td>6.49</td>
<td>-</td>
<td>8.38</td>
<td>-</td>
<td>9.2</td>
<td>-</td>
</tr>
<tr>
<td>LDOF [6]</td>
<td>4.29</td>
<td>7.56</td>
<td>6.42</td>
<td>9.12</td>
<td>13.73</td>
<td>12.4</td>
<td>0.45</td>
</tr>
</tbody>
</table>

### FlowNet better

<table>
<thead>
<tr>
<th>Method</th>
<th>Sintel Clean</th>
<th>Sintel Final</th>
<th>KITTI</th>
<th>Middlebury train</th>
<th>Middlebury test</th>
<th>Chairs</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>train</td>
<td>test</td>
<td>train</td>
<td>AEE</td>
<td>AAE</td>
<td>test</td>
<td>CPU</td>
</tr>
<tr>
<td>FlowNetS</td>
<td>4.50</td>
<td>7.42</td>
<td>5.45</td>
<td>8.43</td>
<td>-</td>
<td>8.26</td>
<td>-</td>
</tr>
<tr>
<td>FlowNetS+v</td>
<td>3.66</td>
<td>6.45</td>
<td>4.76</td>
<td>7.67</td>
<td>-</td>
<td>6.50</td>
<td>-</td>
</tr>
<tr>
<td>FlowNetS+ft</td>
<td>(3.66)</td>
<td>6.96</td>
<td>(4.44)</td>
<td>7.76</td>
<td>7.52</td>
<td>9.1</td>
<td>0.98</td>
</tr>
<tr>
<td>FlowNetS+ft+v</td>
<td>2.97</td>
<td>6.16</td>
<td>(4.07)</td>
<td>7.22</td>
<td>6.07</td>
<td>7.6</td>
<td>0.32</td>
</tr>
<tr>
<td>FlowNetC</td>
<td>4.31</td>
<td>7.28</td>
<td>5.87</td>
<td>8.81</td>
<td>9.35</td>
<td>-</td>
<td>1.15</td>
</tr>
<tr>
<td>FlowNetC+v</td>
<td>3.57</td>
<td>6.27</td>
<td>5.25</td>
<td>8.01</td>
<td>7.45</td>
<td>-</td>
<td>0.34</td>
</tr>
<tr>
<td>FlowNetC+ft</td>
<td>(3.78)</td>
<td>6.85</td>
<td>(5.28)</td>
<td>8.51</td>
<td>8.79</td>
<td>-</td>
<td>0.93</td>
</tr>
<tr>
<td>FlowNetC+ft+v</td>
<td>3.20</td>
<td>6.08</td>
<td>(4.83)</td>
<td>7.88</td>
<td>7.31</td>
<td>-</td>
<td>0.33</td>
</tr>
</tbody>
</table>

+v variational refinement  +ft fine tuning
Outperforms EpicFlow for small objects with large displacement
Results.. (cont)

<table>
<thead>
<tr>
<th>Images</th>
<th>Ground truth</th>
<th>EpicFlow</th>
<th>FlowNetS</th>
<th>FlowNetC</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="groundtruth1.png" alt="Ground truth" /></td>
<td><img src="epicflow1.png" alt="EpicFlow" /></td>
<td><img src="flownets1.png" alt="FlowNetS" /></td>
<td><img src="flownetc1.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="groundtruth2.png" alt="Ground truth" /></td>
<td><img src="epicflow2.png" alt="EpicFlow" /></td>
<td><img src="flownets2.png" alt="FlowNetS" /></td>
<td><img src="flownetc2.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="groundtruth3.png" alt="Ground truth" /></td>
<td><img src="epicflow3.png" alt="EpicFlow" /></td>
<td><img src="flownets3.png" alt="FlowNetS" /></td>
<td><img src="flownetc3.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="groundtruth4.png" alt="Ground truth" /></td>
<td><img src="epicflow4.png" alt="EpicFlow" /></td>
<td><img src="flownets4.png" alt="FlowNetS" /></td>
<td><img src="flownetc4.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="groundtruth5.png" alt="Ground truth" /></td>
<td><img src="epicflow5.png" alt="EpicFlow" /></td>
<td><img src="flownets5.png" alt="FlowNetS" /></td>
<td><img src="flownetc5.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="groundtruth6.png" alt="Ground truth" /></td>
<td><img src="epicflow6.png" alt="EpicFlow" /></td>
<td><img src="flownets6.png" alt="FlowNetS" /></td>
<td><img src="flownetc6.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="groundtruth7.png" alt="Ground truth" /></td>
<td><img src="epicflow7.png" alt="EpicFlow" /></td>
<td><img src="flownets7.png" alt="FlowNetS" /></td>
<td><img src="flownetc7.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="groundtruth8.png" alt="Ground truth" /></td>
<td><img src="epicflow8.png" alt="EpicFlow" /></td>
<td><img src="flownets8.png" alt="FlowNetS" /></td>
<td><img src="flownetc8.png" alt="FlowNetC" /></td>
</tr>
<tr>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="groundtruth9.png" alt="Ground truth" /></td>
<td><img src="epicflow9.png" alt="EpicFlow" /></td>
<td><img src="flownets9.png" alt="FlowNetS" /></td>
<td><img src="flownetc9.png" alt="FlowNetC" /></td>
</tr>
</tbody>
</table>

EPE: Estimated Pixel Error
Thorough comparison with EpicFlow, DeepFlow

Pros

Processes image 5 to 10 frames per second, also very fast

Pros

Smother flow (variational refinement)

Pros

Captures fine details and small objects and also small and big displacements

Pros

No handcrafted methods for aggregation, matching, and interpolation
Pro 6.0

- FlowNet 2.0 is even better!
  - FlowNet 2.0 newer version was just released last year (2017).
  - Crispier, more stacking.
  - “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks”
  - Marginal improvement over FlowNet 2015
The correlation layer is not simple, with k and D as additional hyper parameters

The performance only competitive, could not beat some state of the art methods

Unrealistic images, synthetic data set
Demo
Thank you