Convolutions on Spherical Images

Marc Eder and Jan-Michael Frahm
Department of Computer Science, University of North Carolina at Chapel Hill
2019 SUMO Workshop: 360° Indoor Scene Understanding and Modeling
CVPR 2019, Long Beach, CA
June 17, 2019
Image representation matters!

Simply resampling the image to a different representation significantly improves accuracy for predictions tasks with convolutional neural networks.
Why does image representation matter?

Gauss’s Theorema Egregium:

*Gaussian curvature of a surface is invariant under local isometry*

Far reaching implications, but particularly relevant to cartography: All *planar projections of a sphere have distortions*
All 360° image representations are distorted

**Cubemap**

*Gnomonic (rectilinear) projection*
- Popular graphics format
- Projects a sphere onto the faces of an inscribing cube
- Distorts most severely in corners of faces

**Equirectangular image**

*Equirectangular projection*
- Simple transformation from sphere to projection
- Indexes image grid with spherical coordinates
- Distorts most severely near poles
So what?

Why do we care about spherical distortion when using CNNs?
Distortion and convolution

1D Discrete Convolution

\[(f * g)[n] = \sum_{m=-\left\lfloor \frac{K}{2} \right\rfloor}^{\left\lfloor \frac{K}{2} \right\rfloor} f[m]g[n - m]\]

Separating the sampling operation from the weighted summation

\[= \sum_{m=-\left\lfloor \frac{K}{2} \right\rfloor}^{\left\lfloor \frac{K}{2} \right\rfloor} f[m] \left( \sum_{l=-\infty}^{\infty} g[l] \delta[l - n + m] \right)\]
Distortion and convolution

\[(f * g)[n] = \sum_{m=-\left\lfloor \frac{K}{2} \right\rfloor}^{\left\lfloor \frac{K}{2} \right\rfloor} f[m] \left( \sum_{l=-\infty}^{\infty} g[l] \delta[l - n + m] \right)\]

Sampling represented by the Dirac delta function

Dirac delta function:
\[\delta[x] = \begin{cases} 
1 & x = 0 \\
0 & \text{o.w.}
\end{cases}\]

Alternatively:
(in continuous form)
\[\delta(x) = \lim_{\sigma \to 0} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\]

(area = 1)
Distortion and convolution

\[(f * g)[n] = \sum_{m=-\lfloor \frac{K}{2} \rfloor}^{\lfloor \frac{K}{2} \rfloor} f[m] \left( \sum_{l=-\infty}^{\infty} g[l] \delta[l - n + m] \right)\]

Adds unexpected scaling bias

**Key observation:** Translational equivariance implicitly assumes all sampled data contribute equal information

*Spherical distortion violates this assumption*

*E.g. Pixel redundancy at poles in equirectangular image*
How can we fix this?

Let’s look at what cartographers do...
The imperfect map
Analyzing spherical distortion

**Equidistant**
Preserves distances between points
(Equirectangular)

**Conformal**
Preserves local angles
(Mercator)

**Equal Area**
Preserves relative sizes of objects
(Gall-Peters)
Analyzing spherical distortion

**Tissot’s Indicatrix:** An infinitely small circle on the Earth (A) appears as an ellipse on a typical map (B)

Recall modeling convolution’s sampling function as the limit of a Gaussian as $\sigma \rightarrow 0$

Analyzing spherical distortion

Equidistant
Preserves distances between points
(Equirectangular)

Conformal
Preserves local angles
(Mercator)

Equal Area
Preserves relative sizes of objects
(Gall-Peters)
Back to spherical images

Let's take another look at those two common spherical image formats...
Distortion in 360° image representations

Cubemap
*Gnomonic (rectilinear) projection*
- Popular graphics format
- Projects a sphere onto the faces of an inscribing cube
  - **Distorts most severely in corners of faces**

Equirectangular image
*Equirectangular projection*
- Simple transformation from sphere to projection
- Indexes image grid with spherical coordinates
  - **Distorts most severely near poles**
Quick summary of spherical distortion

1. Mathematically impossible to remove

2. Disrupts *translational equivariance* critical to CNN function

3. Spreads and deforms content (information) in images
Two solutions

Accumulate deformed content

Pros:
- Works with any image representation

Cons:
- Very inefficient (possibly >100’s of pixels per sample)
- GPU implementation difficult

Example accumulation kernel

Use a compromise projection

Pros:
- Efficient sampling (just a single pixel)
- Can use standard grid convolution with limited modifications to implementation

Cons:
- Some distortion remains

Planar approximation to sphere
ISEA and the icosphere

Our compromise projection: Icosahedral Snyder equal area (ISEA) projection [3]

Projects image onto surface of icosphere, a recursively subdivided regular icosahedron

One of least distorted compromise projections [2]
ISEA and the icososphere
Evaluation

Semantic segmentation improves 12.6% simply due to change of image representation
Semantic segmentation

Train a network with each representation using SUMO dataset [5]

Simple encoder-decoder
## Results

Evaluate mIOU on 15 most frequent semantic classes

<table>
<thead>
<tr>
<th>Representation</th>
<th>Floor</th>
<th>Ceiling</th>
<th>Wall</th>
<th>Door</th>
<th>Cabinet</th>
<th>Rug</th>
<th>Window</th>
<th>Curtain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equirectangular (Gnom. Kernel) [1, 3]</td>
<td>0.9315</td>
<td>0.9710</td>
<td>0.8597</td>
<td>0.6466</td>
<td>0.6376</td>
<td>0.7284</td>
<td>0.7012</td>
<td>0.4703</td>
</tr>
<tr>
<td>Icosphere (ours)</td>
<td>0.9352</td>
<td>0.9703</td>
<td>0.8797</td>
<td>0.6890</td>
<td>0.7037</td>
<td>0.6970</td>
<td>0.7562</td>
<td>0.5744</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Representation</th>
<th>Sofa</th>
<th>Partition</th>
<th>Bed</th>
<th>Chair</th>
<th>Table</th>
<th>Shelving</th>
<th>Chandelier</th>
<th>All Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equirectangular (Gnom. Kernel) [1, 3]</td>
<td>0.7114</td>
<td>0.4172</td>
<td>0.7133</td>
<td>0.4219</td>
<td>0.4587</td>
<td>0.3278</td>
<td>0.4491</td>
<td>0.5904</td>
</tr>
<tr>
<td>Icosphere (ours)</td>
<td>0.7374</td>
<td>0.4683</td>
<td>0.7776</td>
<td>0.4375</td>
<td>0.5018</td>
<td>0.3733</td>
<td>0.4472</td>
<td>0.6639</td>
</tr>
</tbody>
</table>

**ISEA projection gives a 12.6% improvement over state-of-the-art methods that use equirectangular images!**
Other applications and future work

**Not limited to CNNs**

Normalized correlation metrics suffer from same issues with spherical images (e.g. stereo depth)

Image filtering uses convolution too -- 360° panos are a growing social media commodity (e.g. Instagram filters)

Need to build large-scale *realistic* spherical image dataset
Thank you!

Any questions?

For more conversation, come to our poster today or contact Marc Eder at meder@cs.unc.edu.
References

Images:

- Equirectangular Earth image, used with permission from http://planetpixelemporium.com/earth8081.html
- Map projection comic, slide 10, from https://xkcd.com/977/ (creative commons license)
- SUMO dataset images [5]

Citations:

[5] Tchapmi, Lyne and Daniel Huber. The sumo challenge