Eye Tracking in 360: Methods, Challenges, and Opportunities

Instructors:
Eakta Jain University of Florida
Olivier Le Meur, Univ. of Rennes, CNRS, IRISA
Introduction: Olivier Le Meur

- PhD, University Of Nantes (Fr)
- HDR, French post-doctoral degree, University of Rennes 1
- Associate Professor, University of Rennes 1
- Team leader: PERCEPT / IRISA
- More than 10 years at Technicolor R&D
- Research Interests:
  - Computational modelling of visual attention
  - Image processing (quality, inpainting, HDR...)

Sponsors:

http://www-percept.irisa.fr/
## Eye Tracking in 360

<table>
<thead>
<tr>
<th>Topic</th>
<th>Learning Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 3: Methods to generate saliency maps from eye tracking data</td>
<td>1. Explain why 2D saliency map methods need to be generalized for omnidirectional viewing</td>
</tr>
<tr>
<td>[Le Meur]</td>
<td>2. Discuss the pros and cons of the selected methods</td>
</tr>
<tr>
<td></td>
<td>3. Compare the performance of different methods using standard metrics</td>
</tr>
<tr>
<td></td>
<td>4. Computational saliency models for 360° images</td>
</tr>
</tbody>
</table>
Generating saliency maps from eye-tracking data

• How to generate saliency map for 2D images? [Le Meur & Baccino, 2013]
  • Fixation map and Gaussian convolution

• How to generate saliency map for 360° images? [John et al., 2018]
  • Equirectangular image and Gaussian convolution
  • Equirectangular image and modified Gaussian convolution
  • Viewport and Gaussian convolution
  • Cubemaps and Gaussian convolution
  • Kent distribution, normal distribution defined on a sphere

Generating saliency maps from eye-tracking data – the 2D case

- **Discrete fixation map** $f^i$ for the $i^{th}$ observer:

  $$f^i(x) = \sum_{k=1}^{M} \delta(x - x_k)$$

  Where $M$ is the number of fixations and $x_k = (x_k, y_k)$ is the $k^{th}$ fixation.

- **Continuous saliency map $S$:**

  $$S(x) = \left( \frac{1}{N} \sum_{i=1}^{N} f^i(x) \right) \ast G_\sigma(x)$$

  Where $N$ is the number of observers, $G_\sigma$ the Gaussian kernel and $\sigma$ the number of pixels per degree of visual angle (nppd).
Generating saliency maps from eye-tracking data – the 2D case

Discrete fixation map to continuous saliency map
Generating saliency maps from eye-tracking data – the 2D case

• What is the nppd value?

\[ \theta_H = 2 \times \arctan \left( \frac{H}{2d} \right) \]

\[ \theta_W = 2 \times \arctan \left( \frac{W}{2d} \right) \]

The nppd is equal to the screen resolution divided by the subtended angle.

Assuming that

• the screen resolution is equal to the image resolution
• display full-screen
• the human fovea is mostly symmetric and round structure…
Generating saliency maps from eye-tracking data – the 2D case

From left to right, $\sigma = \{19,38,76\}$
Generating saliency maps from eye-tracking data

• How to generate saliency map for 2D images? [Le Meur & Baccino, 2013]
  • Fixation map and Gaussian convolution

• How to generate saliency map for 360° images? [John et al., 2018]
  • Equirectangular image and Gaussian convolution
  • Equirectangular image and modified Gaussian convolution
  • Viewport and Gaussian convolution
  • Cubemaps and Gaussian convolution
  • Kent distribution, normal distribution defined on a sphere

Generating saliency maps from eye-tracking data in VR

The question is now

does current saliency map computation method generalize to 360° images?

[John et al., 2018]

Generating saliency maps from eye-tracking data in VR

- 2D isotropic Gaussian kernel on the equirectangular image [Sitzmann et al., 2018]:

Horizontal angle (longitude) $\theta \in [-\pi, \pi]$, Vertical angle (latitude) $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$

Sphere and equirectangular representations under equiangular sampling.

Red and blue regions correspond to the pole and equator regions, respectively.

Generating saliency maps from eye-tracking data in VR

• 2D isotropic Gaussian kernel on the equirectangular image [Sitzmann et al., 2018]:

Generating saliency maps from eye-tracking data in VR

- 2D isotropic Gaussian kernel on the equirectangular image [Sitzmann et al., 2018]:

1. Convert the normalized gaze tracker coordinates to longitude $\theta$ and latitude $\phi$ in the 360° panorama

2. Convolve the fixation maps with a Gaussian with a standard deviation of 1° of visual angle

Generating saliency maps from eye-tracking data in VR

• 2D isotropic Gaussian kernel on the equirectangular image [Sitzmann et al., 2018]:

• $F=$fixation map and $\sigma$ the standard deviation of the Gaussian function

```
1: procedure ISOTROPIC(F,\sigma)
2: $G_\sigma \leftarrow 1d\_gaussian(\sigma, kernel\_size)$
3: $K \leftarrow G_\sigma \cdot G_\sigma^T$
4: $S \leftarrow F \odot K$
```

From [John et al., 2018]

Generating saliency maps from eye-tracking data in VR

• 2D isotropic Gaussian kernel on the equirectangular image [Sitzmann et al., 2018]:

  • This technique is fast and simple ⇒ straightforward extension of 2D

  • It does not account for distortions that are present in panorama images.
  • The equiangular sampling is not uniform.
  • Although much more smaller, the poles (red) are described the same number of pixel than the equator (blue).


O. Le Meur & E. Jain - IEEE VR 2019
Generating saliency maps from eye-tracking data in VR

• Modified 2D isotropic Gaussian kernel on the equirectangular image [Upenik & Ebrahimi, 2017; De Abreu et al., 2017]:

  • To account for distortions near the poles ⇒ Modified 2D Gaussian Kernel $K$:

$$K = G_{\sigma_y} \times G_{\sigma_x}^T$$

Where, $G_{\sigma_n}$ is a column vector representing a 1D Gaussian with standard deviation $\sigma_n$.

The main idea is to adapt $\sigma_x$ according to the vertical angle (e.g. elevation).

Generating saliency maps from eye-tracking data in VR

• Modified 2D isotropic Gaussian kernel on the equirectangular image [Upenik & Ebrahimi, 2017]:

\[ K = G_{\sigma_y} \times G^T_{\sigma_x} \]

With a scale factor to stretch horizontally the Gaussian kernel

\[ \sigma_x = \frac{1}{\cos(\phi)} \times \sigma_y \]

Generating saliency maps from eye-tracking data in VR

- Modified 2D isotropic Gaussian kernel on the equirectangular image [Upenik & Ebrahimi, 2017]:

  \[ K = G_{\sigma_y} \times G_{\sigma_x}^T \]

  With a scale factor to stretch horizontally the Gaussian kernel

  \[ \sigma_x = \frac{1}{\cos(\phi)} \times \sigma_y \]

  \[ \phi = 0^\circ, \Rightarrow \sigma_x = \sigma_y, \text{ 2D isotropic Gaussian Kernel} \]

  \[ \phi = 30^\circ, \Rightarrow \sigma_x = 1.2 \times \sigma_y, \text{ 2D anisotropic Gaussian Kernel} \]

  \[ \phi = 60^\circ, \Rightarrow \sigma_x = 2.0 \times \sigma_y, \text{ 2D anisotropic Gaussian Kernel} \]

  \[ \phi = 90^\circ, \Rightarrow \sigma_x \rightarrow +\infty, \text{ 2D anisotropic Gaussian Kernel} \]


O. Le Meur & E. Jain - IEEE VR 2019
Generating saliency maps from eye-tracking data in VR

• Modified 2D isotropic Gaussian kernel on the equirectangular image [Upenik & Ebrahimi, 2017]:

```plaintext
1: procedure MODIFIEDGAUSSIAN(F, σ)  From [John et al., 2018]
2:   S ← zeros(num_rows, num_cols)
3:   for r = 1 to num_rows do  ▷ Parallelizable loop
4:       φ ← π|r/num_rows − 0.5|
5:       G_{σ_y} ← 1d_gaussian(σ, kernel_size)
6:       G_{σ_x} ← 1d_gaussian(σ/|cosφ|, kernel_size)
7:       K ← G_{σ_y} ⋅ G_{σ_x}^T
8:       S_{row} ← F ⊗ K  ▷ Optimized to only output row r
9:       S(r) = S_{row}
```

Generating saliency maps from eye-tracking data in VR

- Modified 2D isotropic Gaussian kernel on the equirectangular image [Upenik & Ebrahimi, 2017]:

Generating saliency maps from eye-tracking data in VR

• Modified 2D isotropic Gaussian kernel on the equirectangular image [Upenik & Ebrahimi, 2017]:

  • This technique is simple to implement.

  • Time-consuming, since it requires many 2D convolutions (i.e. one per row)

Generating saliency maps from eye-tracking data in VR

• Viewport method [Rai et al., 2017]:

  • Viewport or Field of View (FoV) is a fraction of the omnidirectional image displayed on the 2D screen:
    • Head rotation + field of view \((fovx, fovy)\)

  • For each fixation:
    1. a viewport is created and convolved with a Gaussian kernel.
    2. viewport is projected onto an equirectangular image.
    3. aggregation for equirectangular images.


Chakareski et al., V. (2018, May). Viewport-Driven Rate-Distortion Optimized 360º Video Streaming. In 2018 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.
Generating saliency maps from eye-tracking data in VR

• Viewport method [Rai et al., 2017]:

```
1: procedure VIEWPORT(FIXATIONS, ROTATIONS, \sigma_x, \sigma_y)
2:     S \leftarrow \text{zeros}(\text{num\_rows}, \text{num\_cols})
3:     for i = 1 to M \text{ fixations} do
4:         \text{viewport} \leftarrow \text{zeros}(\text{viewport\_size})
5:         x, y \leftarrow \text{fixation}_i
6:         R \leftarrow \text{rotation}_i
7:         \text{viewport}(y, x) = 1
8:         G_{\sigma_y} \leftarrow 1d\_\text{gaussian}(\sigma_y, \text{kernel\_size})
9:         G_{\sigma_x} \leftarrow 1d\_\text{gaussian}(\sigma_x, \text{kernel\_size})
10:        K \leftarrow G_{\sigma_y} \cdot G_{\sigma_x}^T
11:        S_{\text{viewport}} \leftarrow \text{viewport} \odot K
12:        S_{\text{equirect}} \leftarrow \text{vp2sphere}(S_{\text{viewport}}, R, f_{\text{ov}_x}, f_{\text{ov}_y})
13:        S += S_{\text{equirect}}
```

From [John et al., 2018]

Generating saliency maps from eye-tracking data in VR

• Viewport method [Rai et al., 2017]:

\[ \theta \in [-\pi, \pi] \]

Head orientation: \( \theta = 90^\circ, \phi = 2^\circ \)

Generating saliency maps from eye-tracking data in VR

- Viewport method [Rai et al., 2017]:

Head orientation: \( \theta = 90.1^\circ, \phi = 1.7^\circ \)

Generating saliency maps from eye-tracking data in VR

• Viewport method [Rai et al., 2017]:
  
  • Time-consuming methods (viewport convolution and projection, isotropic or anisotropic Gaussian kernel)
  
  • Does not scale well, depends on the number of fixations to process
  
  • Require to know the HMD’s horizontal and vertical field of view

Generating saliency maps from eye-tracking data in VR

- Cube map projection [Facebook, Maugey et al., 2017]:

  - Projecting the spherical image onto cube faces
    - Cube maps’ pixels are well-distributed — each face is equally important (no more equator or poles)
    - Radial distortion are less noticeable

---


Generating saliency maps from eye-tracking data in VR

• Cube map projection [Facebook, Maugey et al., 2017]:


Radial distortion are less but still noticeable

Artificial borders are created between faces
Double cube projection
Generating saliency maps from eye-tracking data in VR

• Cube map projection [Facebook, Maugey et al., 2017]:

1: procedure CUBEMAP($F$, $\sigma$)
2: $F_{rot} \leftarrow \text{rotatesphereXYZ}(F, \frac{\pi}{4}, 0, \frac{\pi}{4})$
3: $G_{\sigma} \leftarrow \text{1d_gaussian}($$\sigma$, kernel_size$)$
4: $K \leftarrow G_{\sigma} \cdot G_{\sigma}^T$
5: $\text{Cube}_1 \leftarrow \text{equirect2cube}(F) \odot K$
6: $\text{Cube}_2 \leftarrow \text{equirect2cube}(F_{rot}) \odot K$
7: $S_1 \leftarrow \text{cube2equirect}(\text{Cube}_1)$
8: $S_{\text{rotatesphereXYZ}}(\text{cube2equirect}(\text{Cube}_2), -\frac{\pi}{4}, 0, -\frac{\pi}{4})$
9: $S \leftarrow W_1 \cdot S_1 + W_2 \cdot S_2$

From [John et al., 2018]


Generating saliency maps from eye-tracking data in VR

• Cube map projection [Facebook, Maugey et al., 2017]:
  
  • It can be extended to use more than two orientations for a better approximation.
  
  • Transforming high resolution images between formats is time consuming.
  
  • It does not completely remove border discontinuities.

Generating saliency maps from eye-tracking data in VR

• However, natural method to generate 360° saliency map is to process the fixation data directly on the sphere:

  • 2D Gaussian function cannot be used on the sphere

  • Kent distribution is the extension of the Gaussian distribution on the sphere. The isotropic form of the pdf is:

    $$f(x, k, \gamma) = \frac{k}{4\pi \times \sinh(k)} \exp(k \times (\gamma \cdot x)),$$

    $$x, \gamma \in R^3$$
Generating saliency maps from eye-tracking data in VR

- Isotropic form of Kent distribution:
  \[ f(x, k, \gamma) = \frac{k}{4\pi \times \sinh(k)} \exp(k \times \gamma \times x), \]
  \[ x, \gamma \in \mathbb{R}^3 \]

- \( \gamma \) the mean direction of the distribution, around which points are normally distributed.

- \( k (>0) \) represents the concentration of the probability density function.

Three distributions, red is the most compact

Generating saliency maps from eye-tracking data in VR

• Isotropic form of Kent distribution:

```
1: procedure KENT(F, κ)
2:     S ← zeros(num_rows, num_cols)
3:     for r = 1 to num_rows do
4:         φ ← π |r / num_rows - 0.5|
5:         θ ← 0
6:         c = num_cols / 2
7:         \vec{γ} ← sph2cart(θ, φ, 1)
8:         K ← f(N(c, r, kernel_size), κ, \vec{γ})
9:     Normalize K
10:    S_{row} ← F \otimes K
11:    S(r) = S_{row}
```

From [John et al., 2018]

> Parallelizable loop
> θ is constant
> c is constant
> Kernel weights sum to 1
> Optimized to only output row r
Generating saliency maps from eye-tracking data in VR

- Isotropic form of Kent distribution:
  - Due to unequal sampling of the sphere in equirectangular images, the kernel must be recomputed for each row.
  - Required high precision floating point arithmetic, more than 64bits…
  - Very time consuming.
Generating saliency maps from eye-tracking data in VR

• The ground truth saliency maps are computed with Kent method

Which of the four methods best approximate Kent method?

Equirectangular image and Gaussian convolution
Equirectangular image and modified Gaussian convolution
Viewport and Gaussian convolution
Cubemaps and Gaussian convolution
Generating saliency maps from eye-tracking data in VR

• Similarity metrics between two saliency maps [Le Meur & Baccino, 2013; Bylinskii et al., 2019]
  
  • Root Mean Square Error (RMSE): \( RMSE = \sqrt{\frac{\sum_x (Kent(x) - approx(x))^2}{N}} \)

  • Correlation Coefficient (CC): \( CC = \frac{\text{cov}(Kent, approx)}{\sigma_{Kent} \sigma_{approx}} \)

  • Kullback-Leibler Divergence (KLD): \( KLD = \sum_x approx(x) \times \log\left(\frac{approx(x)}{Kent(x)}\right) \), assuming \( \sum_x approx(x) = \sum_x Kent(x) = 1 \).


Bylinskii et al. (2019). What do different evaluation metrics tell us about saliency models?. IEEE transactions on pattern analysis and machine intelligence, 41(3), 740-757.
Generating saliency maps from eye-tracking data in VR

- Eye tracking dataset:
  - Salient360! Dataset [Rai et al., 2017]
  - More than 40 observers
  - 60 (360°) images (equirectangular images cover up to 18k by 9k resolution)
  - + saliency maps generated using the viewport method (isotropic Gaussian)

Generating saliency maps from eye-tracking data in VR
Generating saliency maps from eye-tracking data in VR
Generating saliency maps from eye-tracking data in VR
Generating saliency maps from eye-tracking data in VR
Generating saliency maps from eye-tracking data in VR
Generating saliency maps from eye-tracking data in VR
Generating saliency maps from eye-tracking data in VR

• Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>[Rai et al., 2017]</td>
<td>[Rai et al., 2017]</td>
<td>[Rai et al., 2017]</td>
<td>[Rai et al., 2017]</td>
</tr>
<tr>
<td>CC (↑)</td>
<td>0.994</td>
<td>0.988</td>
<td>0.984</td>
<td>0.408</td>
</tr>
<tr>
<td>KLD (↓)</td>
<td>0.011</td>
<td>0.097</td>
<td>0.108</td>
<td>0.865</td>
</tr>
<tr>
<td>RMSE (↓)</td>
<td>0.018</td>
<td>0.022</td>
<td>0.024</td>
<td>0.145</td>
</tr>
<tr>
<td>Time (s)</td>
<td>5.16</td>
<td>8.80</td>
<td>280</td>
<td>X</td>
</tr>
</tbody>
</table>
Generating saliency maps from eye-tracking data in VR

• Results

[Image of saliency maps for Kent and Viewport]

Kent

Viewport

[Rai et al., 2017]
Generating saliency maps from eye-tracking data in VR

• Our dataset:
  • Replicated data collection of [Rai et al. 2017]
  • Same equipment:
    • Oculus DK2 with SMI eye tracker (60 Hz)
  • Same set of images (four different starting orientations)
  • 20 observers viewed each image (25s, 6s of a gray screen between images)
  • Natural free viewing
Generating saliency maps from eye-tracking data in VR

• Our dataset:

<table>
<thead>
<tr>
<th>Method</th>
<th>Modified Gaussian</th>
<th>Isotropic Gaussian</th>
<th>Cubemap</th>
<th>Viewport</th>
<th>Viewport</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Upenik et al., 2017]</td>
<td>[Sitzmann et al., 2018]</td>
<td>[Maugey et al., 2017]</td>
<td>[Rai et al., 2017]</td>
<td>Anisotropic Gaussian</td>
</tr>
<tr>
<td>Dataset</td>
<td>[Rai et al., 2017]</td>
<td>[Rai et al., 2017]</td>
<td>[Rai et al., 2017]</td>
<td>[Rai et al., 2017]</td>
<td>Ours</td>
</tr>
<tr>
<td>CC (↑)</td>
<td>0.994</td>
<td>0.988</td>
<td>0.984</td>
<td>0.408</td>
<td>0.930</td>
</tr>
<tr>
<td>KLD (↓)</td>
<td>0.011</td>
<td>0.097</td>
<td>0.108</td>
<td>0.865</td>
<td>0.273</td>
</tr>
<tr>
<td>RMSE (↓)</td>
<td>0.018</td>
<td>0.022</td>
<td>0.024</td>
<td>0.145</td>
<td>0.044</td>
</tr>
<tr>
<td>Time (s)</td>
<td>5.16</td>
<td>8.80</td>
<td>280</td>
<td>X</td>
<td>1031</td>
</tr>
</tbody>
</table>

[Anisotropic Gaussian]
[Isotropic Gaussian]
Generating saliency maps from eye-tracking data in VR

• Our conclusion & recommendation (take-home message):

  Saliency map should be computed with the modified 2D isotropic Gaussian kernel on the equirectangular image
Predicting saliency maps in VR, a brief review

• Computational models of visual attention
  
  • From an input image, models aims to compute a saliency map
  
  • A number of models for 2D images
    
    • A new breakthrough with deep architecture
Predicting saliency maps in VR, a brief review

- Computational models of visual attention

  - Few models for 360° images:
    - SalNet 360 [Monroy et al., 2018]
    - GBVS360 and BMS360 [Lebreton & Raake, 2018]
    - SalGAN 360 [Chao et al., 2018]
Predicting saliency maps in VR, a brief review

- Computational models of visual attention
  - SalNet360 [Monroy et al., 2018]

Key idea:
- 6 patches covering the whole sphere are extracted from the 360° image
- Two CNN (CNN 2D) + Refinement (CNN Saliency map + spherical coordinates per pixel)

Predicting saliency maps in VR, a brief review

• Computational models of visual attention
  • SalNet360 [Monroy et al., 2018]

Predicting saliency maps in VR, a brief review

• Computational models of visual attention

• SalNet360 [Monroy et al., 2018]

<table>
<thead>
<tr>
<th>Comparison of the three experimental scenarios.</th>
<th>KL</th>
<th>CC</th>
<th>NSS</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base CNN</td>
<td>1.597</td>
<td>0.416</td>
<td>0.630</td>
<td>0.648</td>
</tr>
<tr>
<td>Above + Patches</td>
<td>0.625</td>
<td>0.474</td>
<td>0.566</td>
<td>0.659</td>
</tr>
<tr>
<td>Above + Spherical coords</td>
<td>0.487*</td>
<td>0.536*</td>
<td>0.757*</td>
<td>0.702*</td>
</tr>
</tbody>
</table>
Predicting saliency maps in VR, a brief review

- Computational models of visual attention
  - GBVS360 [Lebreton & Raake, 2018]

Key ideas:
- Input equirectangular image is projected into several rectilinear images corresponding to different orientations of the user’s viewpoint
- Gabor-based feature extraction
- Feature maps are then back-projected to the equirectangular domain

Predicting saliency maps in VR, a brief review

- Computational models of visual attention
  - GBVS360 [Lebreton & Raake, 2018]

<table>
<thead>
<tr>
<th>Model</th>
<th>KL</th>
<th>CC</th>
<th>NSS</th>
<th>AUC</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU Munich (3) [36]</td>
<td>0.44892</td>
<td>0.57862</td>
<td>0.80527</td>
<td>0.72598</td>
<td>1</td>
</tr>
<tr>
<td>TU Munich (6) [36]</td>
<td>0.42050</td>
<td>0.61478</td>
<td>0.80697</td>
<td>0.72209</td>
<td>1</td>
</tr>
<tr>
<td>SJTU [38]</td>
<td>0.48060</td>
<td>0.53244</td>
<td>0.91810</td>
<td>0.73478</td>
<td>1</td>
</tr>
<tr>
<td>Wuhan Univ. [35]</td>
<td>0.50825</td>
<td>0.53836</td>
<td>0.93588</td>
<td>0.73630</td>
<td>3.25</td>
</tr>
<tr>
<td>TU Munich (1) [36]</td>
<td>0.50084</td>
<td>0.55356</td>
<td>0.91531</td>
<td>0.74677</td>
<td>3.25</td>
</tr>
<tr>
<td>TU Munich (5) [36]</td>
<td>0.47503</td>
<td>0.55859</td>
<td>0.69606</td>
<td>0.71286</td>
<td>4</td>
</tr>
<tr>
<td>BMS360-Eq</td>
<td>0.59873</td>
<td>0.55365</td>
<td>0.93629</td>
<td>0.73556</td>
<td>4</td>
</tr>
<tr>
<td>BMS360</td>
<td>0.52546</td>
<td>0.47202</td>
<td>0.85887</td>
<td>0.73473</td>
<td>3</td>
</tr>
<tr>
<td>BMS-Eq</td>
<td>0.54289</td>
<td>0.54452</td>
<td>0.87607</td>
<td>0.72247</td>
<td>3</td>
</tr>
<tr>
<td>GBVS360-Eq</td>
<td>0.6980</td>
<td>0.52704</td>
<td>0.85053</td>
<td>0.71402</td>
<td>5.75</td>
</tr>
<tr>
<td>BMS</td>
<td>0.53085</td>
<td>0.44963</td>
<td>0.76229</td>
<td>0.71104</td>
<td>4</td>
</tr>
</tbody>
</table>

Eq = Equatorial prior based on a Gaussian mixture of 3 Gaussians (2 for the pole regions and one for the central region)

Predicting saliency maps in VR, a brief review

- Computational models of visual attention
  - SalGAN 360 [Chao et al., 2018]

Key ideas:
- Global saliency map SalGan
- Local saliency maps by considering 9 rotations (cube maps) and saliency maps computed over cube faces (6x9=54)

Predicting saliency maps in VR, a brief review

• Computational models of visual attention
  • SalGAN 360 [Chao et al., 2018]

<table>
<thead>
<tr>
<th>Method</th>
<th>KL↓</th>
<th>CC↑</th>
<th>NSS↑</th>
<th>AUC↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SalGAN</td>
<td>1.236</td>
<td>0.452</td>
<td>0.810</td>
<td>0.708</td>
</tr>
<tr>
<td>SalGAN&amp;FSM [15]</td>
<td>0.896</td>
<td>0.512</td>
<td>0.910</td>
<td>0.723</td>
</tr>
<tr>
<td>Maugey <em>et al.</em></td>
<td>0.585</td>
<td>0.448</td>
<td>0.506</td>
<td>0.644</td>
</tr>
<tr>
<td>SalNet360[13]</td>
<td>0.458</td>
<td>0.548</td>
<td>0.755</td>
<td>0.701</td>
</tr>
<tr>
<td>GBVS360 [12]</td>
<td>0.698</td>
<td>0.527</td>
<td>0.851</td>
<td>0.714</td>
</tr>
<tr>
<td>SalGAN360</td>
<td>0.431</td>
<td>0.659</td>
<td>0.971</td>
<td>0.746</td>
</tr>
</tbody>
</table>

## Conclusion

<table>
<thead>
<tr>
<th>Topic</th>
<th>Learning Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 3: Methods to generate saliency maps from eye tracking data</td>
<td>1. Explain why 2D saliency map methods need to be generalized for omnidirectional viewing</td>
</tr>
<tr>
<td>[Le Meur]</td>
<td>2. Discuss the pros and cons of the selected methods</td>
</tr>
<tr>
<td></td>
<td>3. Compare the performance of different methods using standard metrics</td>
</tr>
<tr>
<td></td>
<td>4. Computational saliency models for 360° images</td>
</tr>
</tbody>
</table>
THANKS!!