Master SIF - REP (4/20)
Pixel Organisation and Representation

Thomas Maugey
thomas.maugey@inria.fr

UNIVERSITÉ DE RENNES 1

Inria informatics mathematics

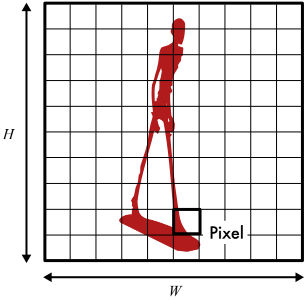
Fall 2020

Table of Contents

Table of Contents

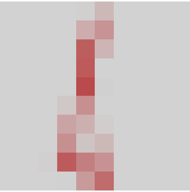
Pixels

First pixels in 1965, when the "digital world" moved from a representation of images by lines to an array of *picture elements* called **pixels**.

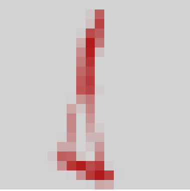


Aspect ratio $= W/H$ $(4/3,\ 16/9, \ldots)$
Resolution $= W \times H$
Density (Pixel per Inch, PPI) $= \frac{\sqrt{W^2+H^2}}{d}$ (where d is the screen diagonal in "inches")

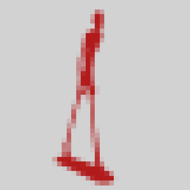
Image resolution: $W \times H$



10×10



20×20



40×40

80×80

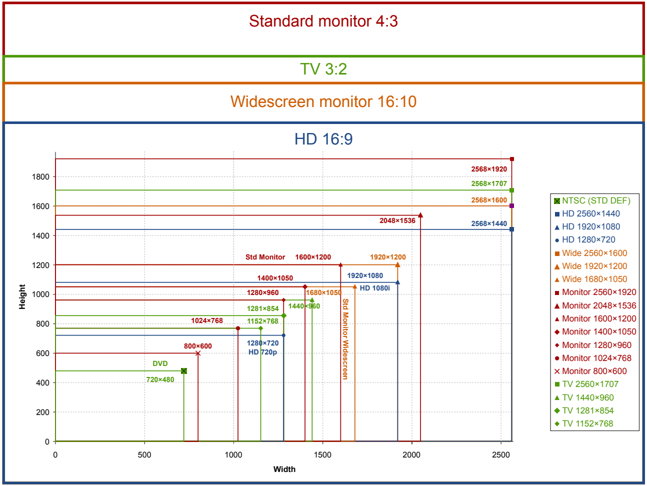
160×160

320×320

640×640

1280×1280

Standard AR and Resolutions



3840×2160 (4K UHDTV), 4096×2160 (4K Cinema), 7680×4320 (8K UHDTV), 15360×8640 (16K Cinema), 61440×34560 (64K Cinema)

[Wikipedia]

Table of Contents

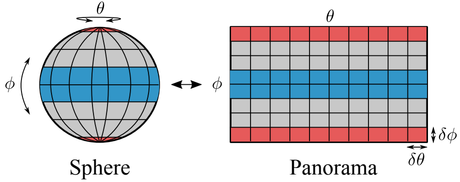
Is matrix organisation always meaningful?

Example of omnidirectional capture

How to represent accurately this image?

Equirectangular representation

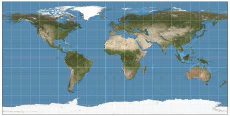
Equirectangular or Panorama description



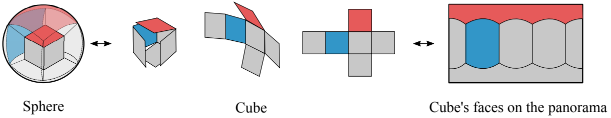
- Most popular
- Suitable for image processing applications

But

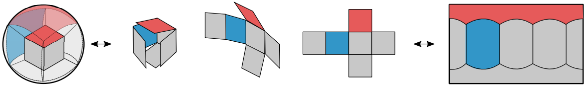
- Radial distortions



Cubemap representation

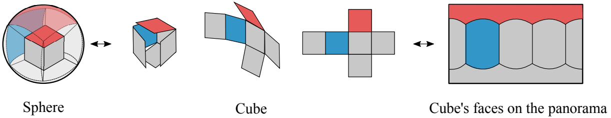


Cubemap representation



Sphere | Cube | Cube's faces on the panorama

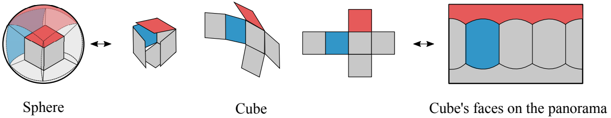
Cubemap representation



- Used by Facebook
- No radial distortion

[Facebook, "Under the hood: building 360 video." https://code.facebook.com/posts/1638767863078802/under-the-hood-building-360-video/]

Cubemap representation



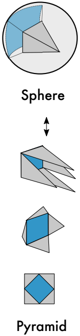
- Used by Facebook
- No radial distortion

But

- Loose some connexion informations

[Facebook, "Under the hood: building 360 video." https://code.facebook.com/posts/1638767863078802/under-the-hood-building-360-video/]

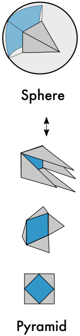
Pyramidal representation



Storage of several pyramidal representations corresponding to different directions on the server

[https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/]

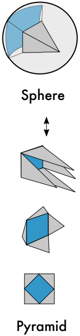
Pyramidal representation

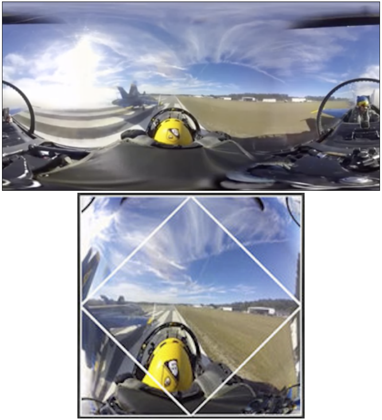


Storage of several pyramidal representations corresponding to different directions on the server

[https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/]

Pyramidal representation

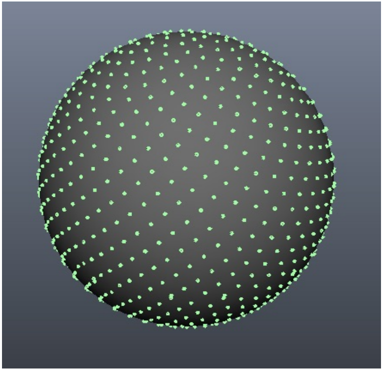




Storage of several pyramidal representations corresponding to different directions on the server

[https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/]

Uniform sampling



- Equidistant point
- Connectivity preserved

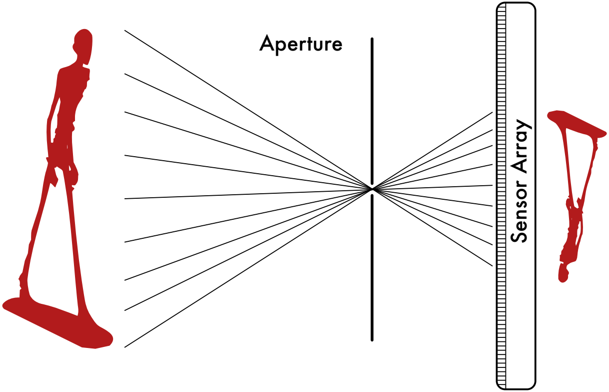
But

- Not a 2D image anymore

Table of Contents

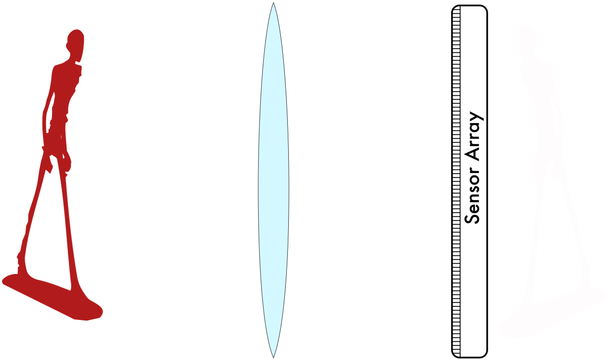
Lens

Aperture does not capture enough light



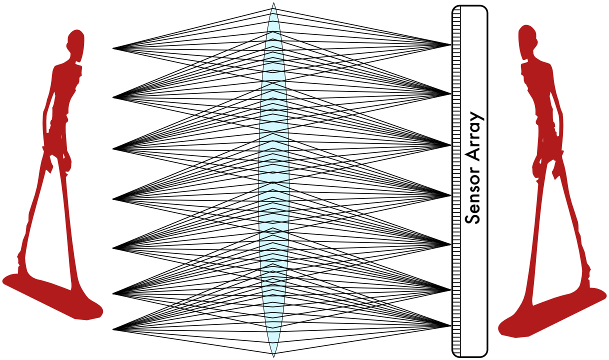
Lens

Aperture is then replaced by a lens



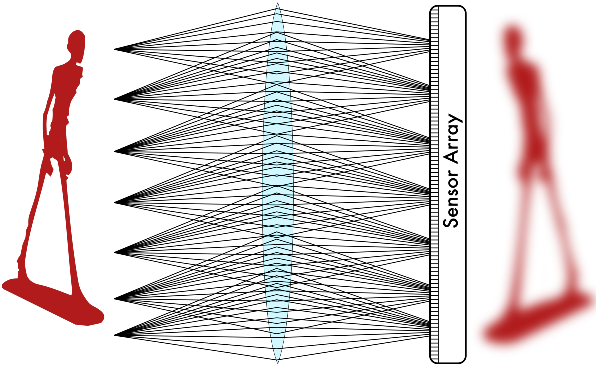
Lens

The lens deviates the light rays



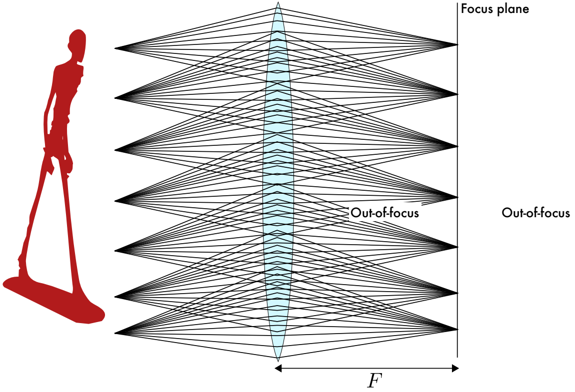
Lens

The rays do not converge to one point, blur appears



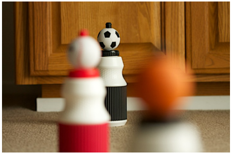
Focus

The focus plane position depends on the object distance.



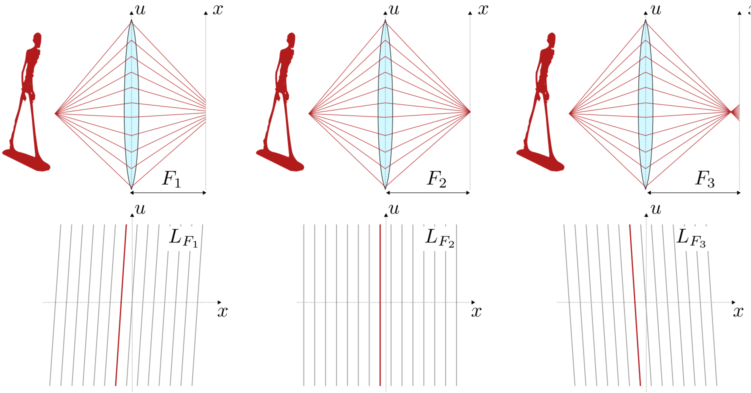
All the objects whose focus plane is placed at the sensor plane will be **in-focus**, all the other ones are **out-of-focus**

Examples

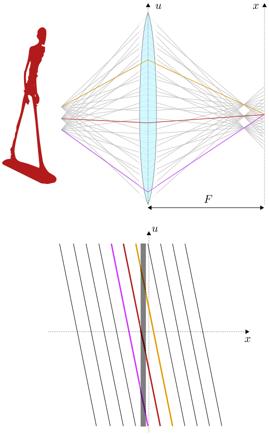


Light field

The light field is parametrized with the two plane coordinates (u, v) and (x, y). It is called $L_F(x, y, u, v)$ and depends on F.



Light field sensing

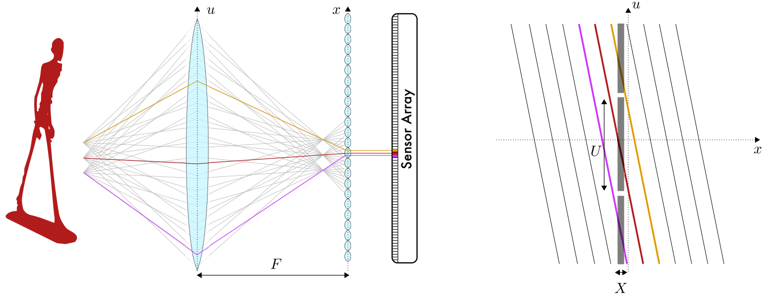


At the same (x, y), their might be several $L_F(x, y, u, v)$ coming from different (u, v).

The value of the light field on the sensor plane at position (x, y) is equal to

$$E_F^{\text{im}}(x, y) = \frac{1}{F^2} \int_u \int_v L_F(x, y, u, v) dv du$$

Plenoptic camera



An array of micro-lenses enables to discriminate the ray directions

$$E_F^{\mathrm{lf}}(x, y, i_u, i_v) = \frac{1}{F^2} \int_{u \in U_{i_u}} \int_{v \in V_{i_v}} L_F(x, y, u, v) dv du$$

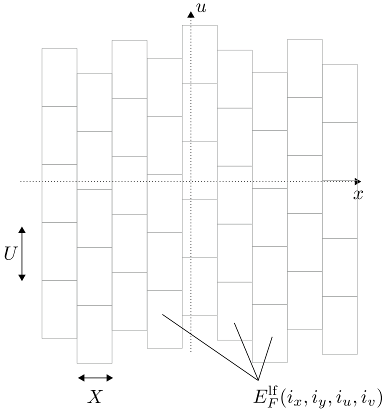
In practice, (x, y) are discretized as well

$$E_F^{\mathrm{lf}}(i_x, i_y, i_u, i_v) = \frac{1}{F^2} \int_{x \in X_{i_x}} \int_{y \in Y_{i_y}} \int_{u \in U_{i_u}} \int_{v \in V_{i_v}} L_F(x, y, u, v) dv du dy dx$$

Often, the indexes are removed and the recorded light field is denoted by $E_F^{\mathrm{lf}}(x, y, u, v)$, where (x, y, u, v) become indices.

LF capture

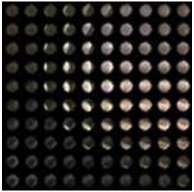
The light field recorded with a distance F looks like:

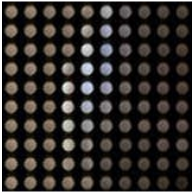


Raw Light field

The pixel array $E_F(i_x, i_y, i_u, i_v)$ can be represented as it is recorded

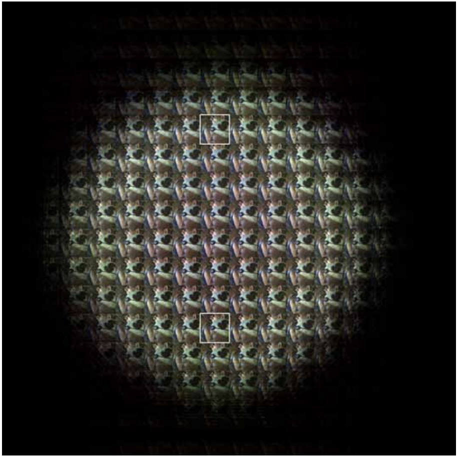
$(i_x, i_y), (i_u, i_v)$





Sub-aperture images

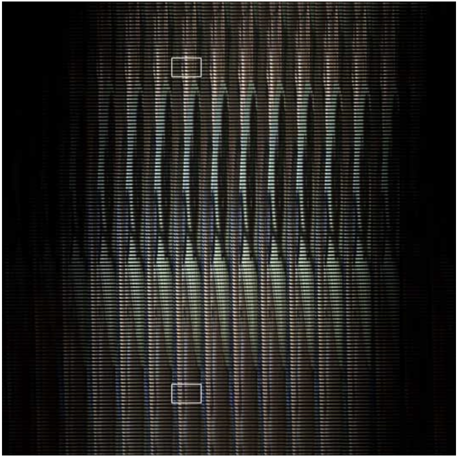
The pixel array $E_F^{\mathrm{lf}}(i_x, i_y, i_u, i_v)$ can be represented by orientations



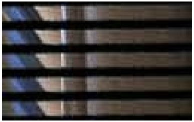
$(i_u, i_v), (i_x, i_y)$

Epipolar images

The pixel array $E_F^{\mathrm{lf}}(i_x, i_y, i_u, i_v)$ can be represented row-by-row



$(i_x, i_u), (i_y, i_v)$



Digital refocusing

The digital refocusing consists in synthesizing images $E^{\mathrm{im}}_{F'}$ with the desired focus F' from a single light field record E^{lf}_{F}.

The generated image is equal to

$$E^{\mathrm{im}}_{F'}(x', y') =$$

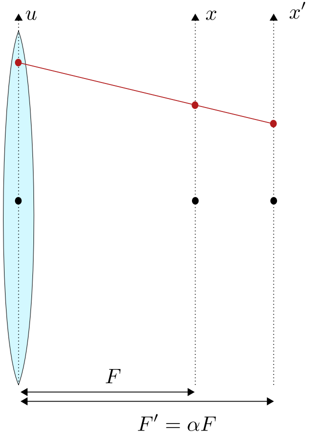
Digital refocusing

The digital refocusing consists in synthesizing images $E_{F'}^{\text{im}}$ with the desired focus F' from a single light field record E_F^{lf}.

The generated image is equal to

$$E_{F'}^{\text{im}}(x', y') = \frac{1}{F'^2} \int_u \int_v L_{F'}(x', y', u, v) du dv dx' dy'$$

Light field's property



With a new distance $F' = \alpha F$

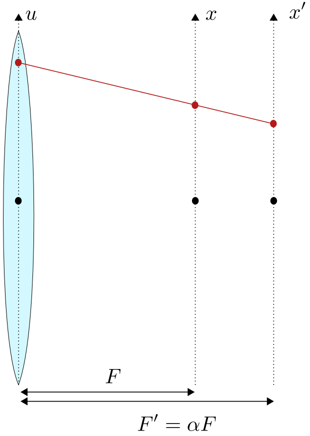
$L_{F'}(x', y', u, v)$ is equal to the value of light field L_F at position:

$$x =$$

$$y =$$

The acquired image is thus:

Light field's property



With a new distance $F' = \alpha F$

$L_{F'}(x', y', u, v)$ is equal to the value of light field L_F at position:

$$x = u(1 - \frac{1}{\alpha}) + \frac{x'}{\alpha}$$

$$y = v(1 - \frac{1}{\alpha}) + \frac{y'}{\alpha}$$

The acquired image is thus:

$$E^{\text{im}}_{\alpha F}(x', y') = \frac{1}{\alpha^2 F^2} \int_u \int_v L_F \left(u(1 - \frac{1}{\alpha}) + \frac{x'}{\alpha}, v(1 - \frac{1}{\alpha}) + \frac{y'}{\alpha}, u, v \right) dvdu$$

Digital refocusing

The generated image is equal to

$$E^{\text{im}}_{\alpha F}(i_{x'}, i_{y'}) = \frac{1}{\alpha^2 F^2} \int_{x' \in X_{i_{x'}}} \int_{y' \in Y_{i_{y'}}} \int_u \int_v L_{F'}(x', y', u, v)\, du\, dv\, dx'\, dy'$$

Using previous relation between L_F and $L_{\alpha F}$:

$$E^{\text{im}}_{\alpha F}(i_{x'}, i_{y'}) = \frac{1}{\alpha^2 F^2} \int\int\int\int L_F\left(u(1-\frac{1}{\alpha}) + \frac{x'}{\alpha}, v(1-\frac{1}{\alpha}) + \frac{y'}{\alpha}, u, v\right) du\, dv\, dx'\, dy'$$

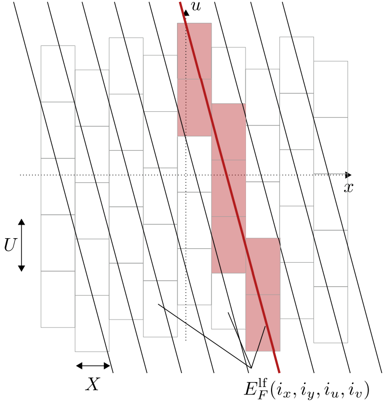
We can write (with $\iota(.)$ equal to the round operation):

$$E^{\text{im}}_{\alpha F}(i_{x'}, i_{y'}) \approx \frac{1}{\alpha^2 F^2} \sum_{i_{x'}} \sum_{i_{y'}} \sum_{i_u} \sum_{i_v} E^{\text{lf}}_F\left(\iota(i_u(1-\frac{1}{\alpha}) + \frac{i_{x'}}{\alpha}), \iota(i_v(1-\frac{1}{\alpha}) + \frac{i_{y'}}{\alpha}), i_u, i_v\right)$$

Digital refocusing

It comes down to summing along the rays of equations

$$\left(\iota(i_u(1-\frac{1}{\alpha})+\frac{i_{x'}}{\alpha}),\iota(i_v(1-\frac{1}{\alpha})+\frac{i_{y'}}{\alpha}),i_u,i_v\right)$$



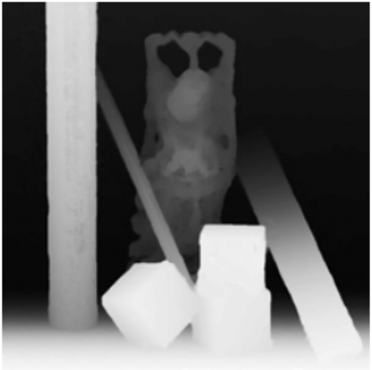
Or simply shifting and summing the sub-aperture images

Examples





Depth Estimation



← Epipolar plane image

[Wanner, S., Goldluecke, B. (2012, June). Globally consistent depth labeling of 4D light fields. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 41-48). IEEE.]

Structure Tensor

Let $I(\mathbf{p})$ be an image realization at a pixel position $\mathbf{p}$.
Let $\nabla I_x(\mathbf{p})$ and $\nabla I_y(\mathbf{p})$ be the horizontal and vertical gradients respectively.
We define the tensor structure at position $\mathbf{p}$ as

$$\mathbf{J}(\mathbf{p}) = \mathbb{E}_{w,\mathbf{p}}\left([\nabla I_x(\mathbf{r}),\ \nabla I_y(\mathbf{r})]^\top [\nabla I_x(\mathbf{r}),\ \nabla I_y(\mathbf{r})]\right)$$

which gives

$$\mathbf{J}(\mathbf{p}) = \begin{pmatrix} \sum_\mathbf{r} w(\mathbf{r})\nabla I_x(\mathbf{p}-\mathbf{r})^2 & \sum_\mathbf{r} w(\mathbf{r})\nabla I_x(\mathbf{p}-\mathbf{r})\nabla I_y(\mathbf{p}-\mathbf{r}) \\ \sum_\mathbf{r} w(\mathbf{r})\nabla I_x(\mathbf{p}-\mathbf{r})\nabla I_y(\mathbf{p}-\mathbf{r}) & \sum_\mathbf{r} w(\mathbf{r})\nabla I_y(\mathbf{p}-\mathbf{r})^2 \end{pmatrix}$$

if w comes from G_σ, a Gaussian kernel centered around $\mathbf{p}$, we have

$$\mathbf{J}(\mathbf{p}) = \begin{pmatrix} (G_\sigma * \nabla I_x^2)(\mathbf{p}) & (G_\sigma * \nabla I_x \nabla I_y)(\mathbf{p}) \\ (G_\sigma * \nabla I_x \nabla I_y)(\mathbf{p}) & (G_\sigma * \nabla I_y^2)(\mathbf{p}) \end{pmatrix}$$

Structure Tensor

Tensor's structure property:

The orientation $\mathbf{n}$ is the solution of the following equation:

$$\mathbf{J}(\mathbf{p})\mathbf{n} = \lambda\mathbf{n}$$

So the eigenvectors of $\mathbf{J}(\mathbf{p})$ are the major orientation at position $\mathbf{p}$ and their corresponding energy is given by the eigenvalues λ_1 and λ_2 (with $\lambda_1 > \lambda_2$).

The major orientation (λ_1) is given by the first eigenvector

$$\mathbf{n} = \begin{pmatrix} J_{2,2}(\mathbf{p}) - J_{1,1}(\mathbf{p}) \\ 2J_{1,2}(\mathbf{p}) \end{pmatrix}$$

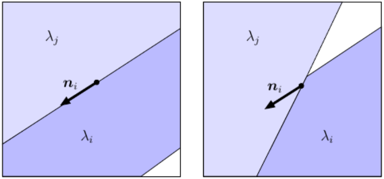
with a level of confidence equal to

$$C = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} = \frac{(J_{2,2}(\mathbf{p}) - J_{1,1}(\mathbf{p}))^2 + 4J_{1,2}^2}{(J_{1,1}(\mathbf{p}) + J_{2,2}(\mathbf{p}))^2}$$

[Bigun, J. (1987). Optimal orientation detection of linear symmetry.]

Depth Estimation

↓



(a) Allowed (b) Forbidden

↓

Table of Contents

References

- C. Grunheit, A. Smolic, and T. Wiegand, "Efficient representation and interactive streaming of high-resolution panoramic views," in International Conference on Image Processing, vol. 3, 2002, pp. 209âĂŞ212.
- M. Yu, H. Lakshman, and B. Girod, "Content adaptive representations of omnidirectional videos for cinematic virtual reality," in 3rd International Workshop on Immersive Media Experiences. ACM, 2015, p. 16.
- Ren, N. G. (2006). Digital light field photography. Ph. D. thesis Stanford University.
- http://clim.inria.fr/