GT Rendu 2020

Toward a noise perception model for photorealistic image synthesis

Jérôme Buisine (PhD Student)
Supervisors: Christophe Renaud and Samuel Delepoulle
Team: IMAP (Images et Apprentissage)

July 7, 2020

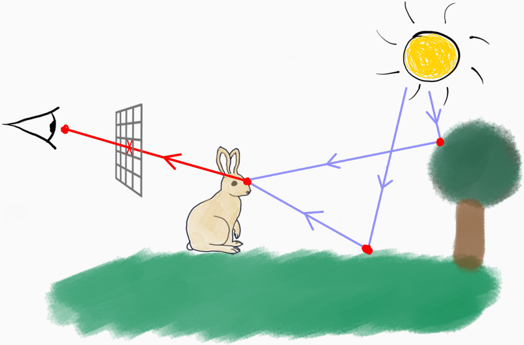
* ANR support : project ANR-17-CE38-0009
Univ. Littoral Côte d'Opale, LISIC, F-62100 Calais, France

Agenda

1. Context
2. Noise
3. Perception
4. Relative & current works
5. Conclusion

Context

Context



Photorealistic image synthesis

- Global illumination rendering
- Monte Carlo

Context: noise in photorealistic image

(a) After 1 sample

(b) After 20 samples

(c) After 10000 samples

Context: ways to tackle the noise problem in rendering

How to improve the rendered image ?

Context: ways to tackle the noise problem in rendering

How to improve the rendered image ?

- by improving the path-tracing strategies
  1. **Integrator:** Bidirectional path-tracing (*Lafortune and Willems 1993*), Metropolis light transport (*Veach and Guibas 1997*)
  2. **Path-guiding:** adaptive variance reduction (*Vorba et al. 2019*)

Context: ways to tackle the noise problem in rendering

How to improve the rendered image ?

- by improving the path-tracing strategies
  1. **Integrator:** Bidirectional path-tracing (*Lafortune and Willems 1993*), Metropolis light transport (*Veach and Guibas 1997*)
  2. **Path-guiding:** adaptive variance reduction (*Vorba et al. 2019*)
- by providing adaptive sampling

Context: ways to tackle the noise problem in rendering

How to improve the rendered image ?

- by improving the path-tracing strategies
  1. **Integrator:** Bidirectional path-tracing (*Lafortune and Willems 1993*), Metropolis light transport (*Veach and Guibas 1997*)
  2. **Path-guiding:** adaptive variance reduction (*Vorba et al. 2019*)
- by providing adaptive sampling
- by filtering as a post-process

How to improve the rendered image ?

- by improving the path-tracing strategies
  1. **Integrator:** Bidirectional path-tracing (*Lafortune and Willems 1993*), Metropolis light transport (*Veach and Guibas 1997*)
  2. **Path-guiding:** adaptive variance reduction (*Vorba et al. 2019*)
- by providing adaptive sampling
- by filtering as a post-process
- using hybrid methods based on the 2 last approaches (progressive rendering)

How to improve the rendered image ?

- by improving the path-tracing strategies
  1. **Integrator:** Bidirectional path-tracing (*Lafortune and Willems 1993*), Metropolis light transport (*Veach and Guibas 1997*)
  2. **Path-guiding:** adaptive variance reduction (*Vorba et al. 2019*)
- by providing adaptive sampling
- by filtering as a post-process
- using hybrid methods based on the 2 last approaches (progressive rendering)

How can humans perceived the photorealistic rendering generated noise ?

Noise

Noise: overview

- Capture $\rightarrow$ a lot of noise perception models
  - **Full-reference:** SSIM (Carnec, Le Callet, and Barba 2003)
  - **No-reference:** BRISQUE (Mittal, Moorthy, and Bovik 2012)
  - ...

- Capture $\rightarrow$ a lot of noise perception models
  - **Full-reference:** SSIM (Carnec, Le Callet, and Barba 2003)
  - **No-reference:** BRISQUE (Mittal, Moorthy, and Bovik 2012)
  - ...

Problem

Noise perception capture models cannot be used for photorealistic image synthesis

Noise: overview

- Capture $\rightarrow$ a lot of noise perception models
  - **Full-reference:** SSIM (Carnec, Le Callet, and Barba 2003)
  - **No-reference:** BRISQUE (Mittal, Moorthy, and Bovik 2012)
  - ...

Problem

Noise perception capture models cannot be used for photorealistic image synthesis

- Rendering

Noise: overview

- Capture $\rightarrow$ a lot of noise perception models
  - **Full-reference:** SSIM (Carnec, Le Callet, and Barba 2003)
  - **No-reference:** BRISQUE (Mittal, Moorthy, and Bovik 2012)
  - ...

Problem

Noise perception capture models cannot be used for photorealistic image synthesis

- Rendering $\rightarrow$ a **lack** of noise perception models

Noise: overview

- Capture $\rightarrow$ a lot of noise perception models
  - **Full-reference:** SSIM (Carnec, Le Callet, and Barba 2003)
  - **No-reference:** BRISQUE (Mittal, Moorthy, and Bovik 2012)
  - ...

Problem

Noise perception capture models cannot be used for photorealistic image synthesis

- Rendering $\rightarrow$ a **lack** of noise perception models

Objective

Build a noise perception model for computer graphics

Perception

Perception: definition

Just-Noticeable Difference (JND)

Noise can be viewed as a perceptible difference into image

20 samples

1000 samples

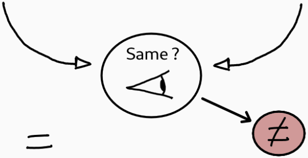
Perception: definition

Just-Noticeable Difference (JND)

Noise can be viewed as a perceptible difference into image

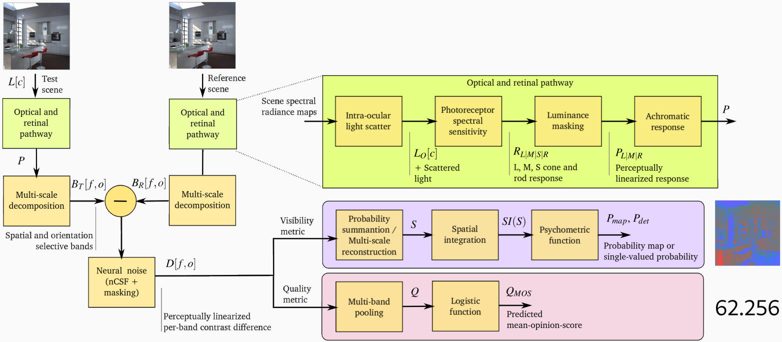
20 samples

1000 samples



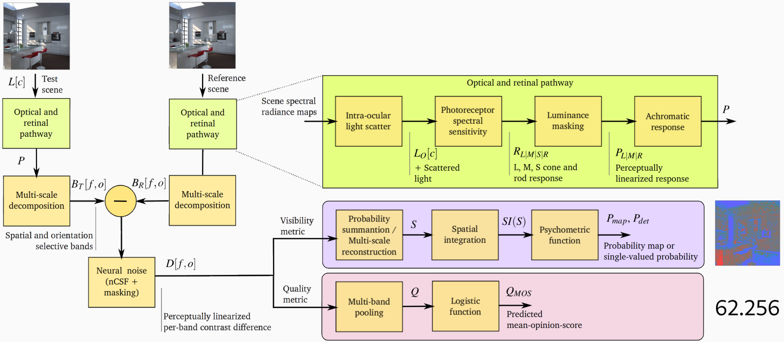
Perception: Visual Difference Predictor

HDR-VDP: a calibrated method for objective quality prediction (*Narwaria et al. 2015*)



Perception: Visual Difference Predictor

HDR-VDP: a calibrated method for objective quality prediction (*Narwaria et al. 2015*)



Problem

- complex model, with a lot of parameters (room luminance, screen luminance...).
- model which requires **reference** which is not available in computer graphics.

Relative & current works

1. How to build a such model ?
2. Previous & current team works
3. Deep Learning approaches

1. **How to build a such model ?**

2. Previous & current team works

3. Deep Learning approaches

Dataset creation: need of human data

Problem of photorealistic image synthesis rendering

- No-reference context during rendering
- No human perceptual reference data

Dataset creation: need of human data

Problem of photorealistic image synthesis rendering

- No-reference context during rendering
- No human perceptual reference data

A solution

Collect human subjective perceptual threshold during rendering as ground truth

Problem of photorealistic image synthesis rendering

- No-reference context during rendering
- No human perceptual reference data

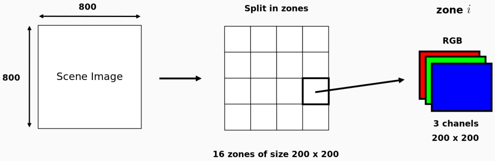
A solution

Collect human subjective perceptual threshold during rendering as ground truth

Build a model

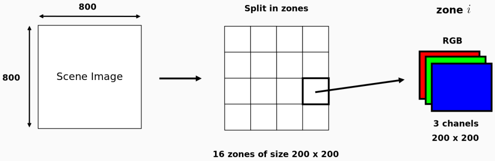
Use these perceptual thresholds into a perceptual noise model

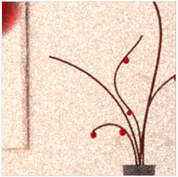
Our way of getting perceptual subjective thresholds



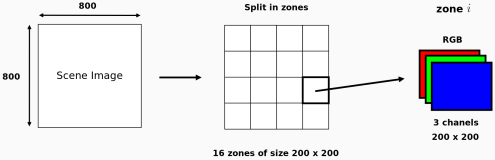
Dataset creation: collect human subjective threshold

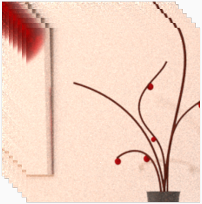
Our way of getting perceptual subjective thresholds





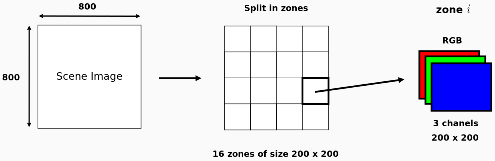
Our way of getting perceptual subjective thresholds

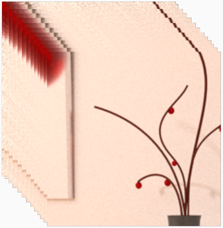




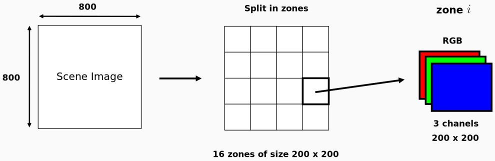
Dataset creation: collect human subjective threshold

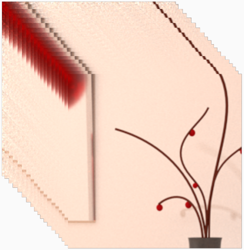
Our way of getting perceptual subjective thresholds





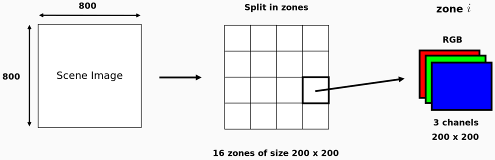
Our way of getting perceptual subjective thresholds

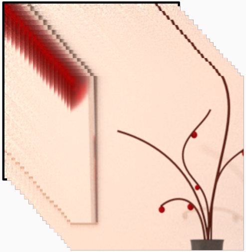


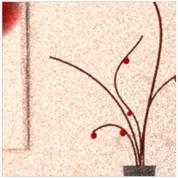


Dataset creation: collect human subjective threshold

Our way of getting perceptual subjective thresholds



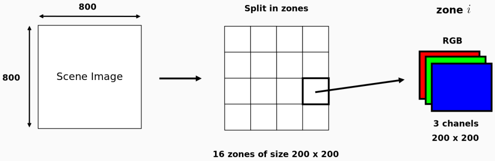


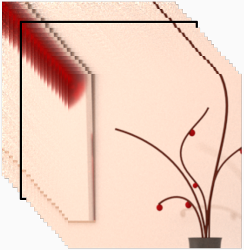


20 samples

3000 samples (reference)

Our way of getting perceptual subjective thresholds



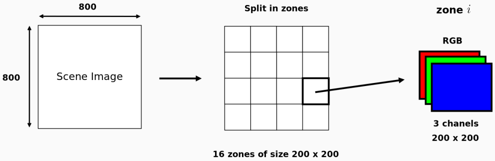


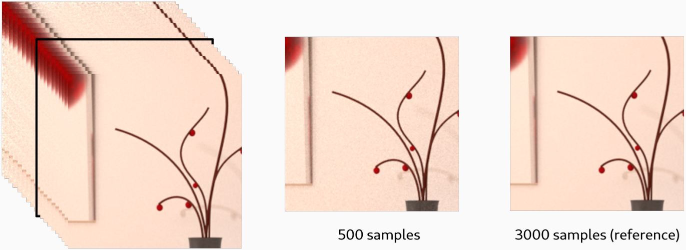
220 samples

3000 samples (reference)

Dataset creation: collect human subjective threshold

Our way of getting perceptual subjective thresholds



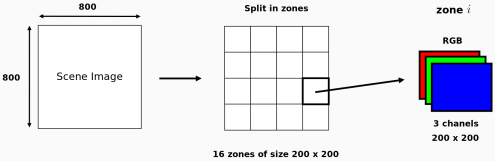


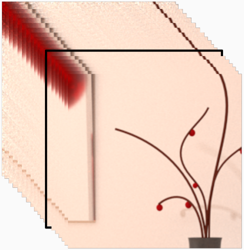
500 samples

3000 samples (reference)

Dataset creation: collect human subjective threshold

Our way of getting perceptual subjective thresholds



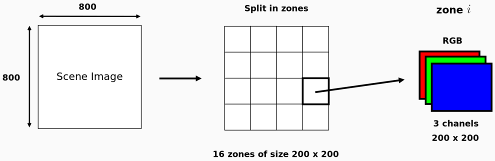


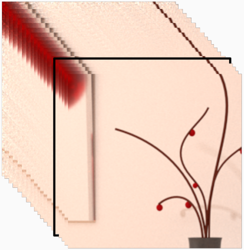
900 samples

3000 samples (reference)

Dataset creation: collect human subjective threshold

Our way of getting perceptual subjective thresholds

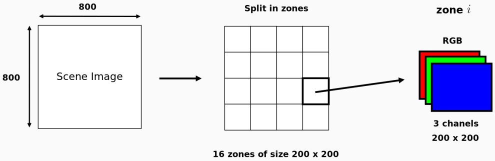


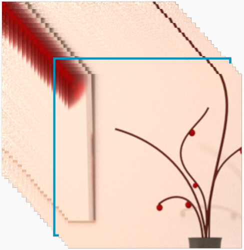


1400 samples

3000 samples (reference)

Our way of getting perceptual subjective thresholds



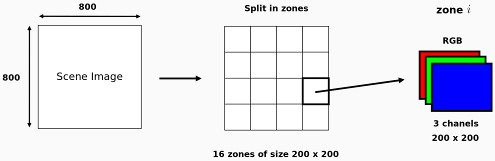


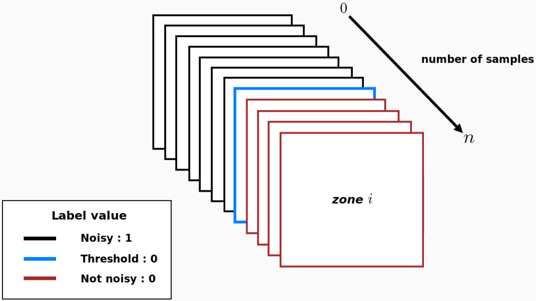
1400 samples

3000 samples (reference)

Dataset creation: collect human subjective threshold

Our way of getting perceptual subjective thresholds





Dataset creation: overview

313	312	274	271
310	301	308	235
248	292	222	240
211	151	139	177

(a) Human thresholds (Mean Opinion Score)

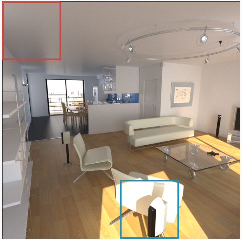
(b) Human reference

(c) After 900 samples

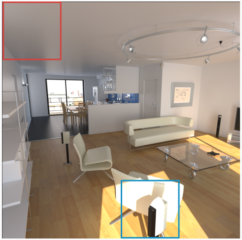
Dataset creation: overview

313	312	274	271
310	301	308	235
248	292	222	240
211	151	139	177

(a) Human thresholds (Mean Opinion Score)



(b) Human reference

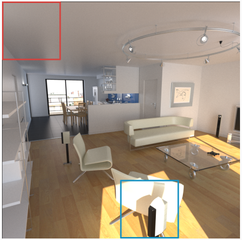


(c) After 900 samples

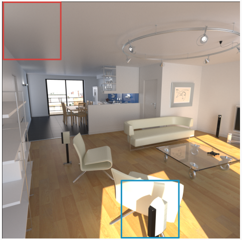
Dataset creation: overview

313	312	274	271
310	301	308	235
248	292	222	240
211	151	139	177

(a) Human thresholds (Mean Opinion Score)



(b) Human reference
SSIM: 0.70 (< 0.95)



(c) After 900 samples
SSIM: 1

Structural Similarity Index (SSIM)

SSIM metric quantifies the visibility of errors between a distorted image and a reference image using a variety of known properties of the human visual system.

Database creation: expected model

Binary classification

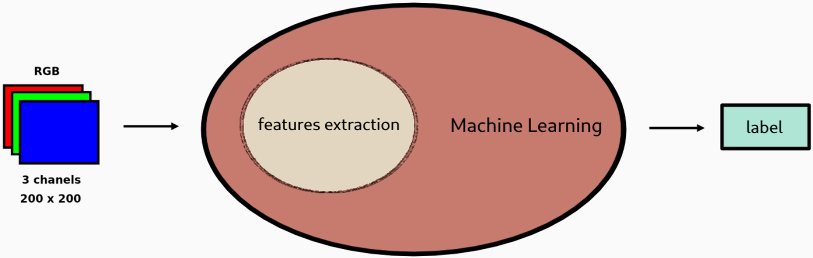
- Model which labels image as **noisy** or **not** (**converged** or **not**)
- Supervised learning

Database creation: expected model

Binary classification

- Model which labels image as **noisy** or **not** (**converged** or **not**)
- Supervised learning

Common pipeline used



Why this kind of model ?

Why this kind of model ?

- **stopping criterion** during rendering based on sub-blocks of rendered image

Why this kind of model ?

- **stopping criterion** during rendering based on sub-blocks of rendered image
- **save** computation time

Why this kind of model ?

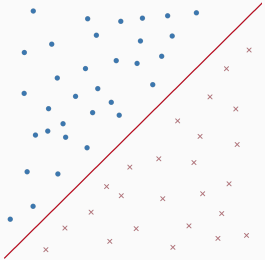
- **stopping criterion** during rendering based on sub-blocks of rendered image
- **save** computation time
- target more complex parts of the scene

1. How to build a such model ?
2. **Previous & current team works**
3. Deep Learning approaches

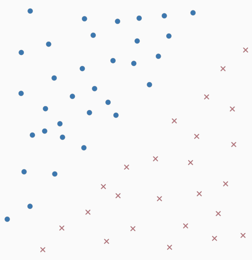
Perception model for synthesis images:

- Image noise detection in global illumination methods based on FRVM (*J. Constantin, A. Bigand, et al. 2015*)
- Perception of noise in global illumination based on inductive learning (*J. Constantin, I. Constantin, et al. 2016*)
- Perception of noise and global illumination: Toward an automatic stopping criterion based on SVM (*N. Takouachet et al. 2017*)

Support Vector Machine (SVM):



(a) Linear classifier model



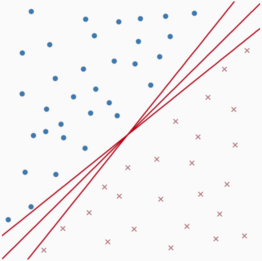
(b) SVM classifier model

Team works: previously

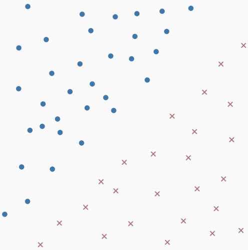
Perception model for synthesis images:

- Image noise detection in global illumination methods based on FRVM (*J. Constantin, A. Bigand, et al. 2015*)
- Perception of noise in global illumination based on inductive learning (*J. Constantin, I. Constantin, et al. 2016*)
- Perception of noise and global illumination: Toward an automatic stopping criterion based on SVM (*N. Takouachet et al. 2017*)

Support Vector Machine (SVM):



(a) Linear classifier model



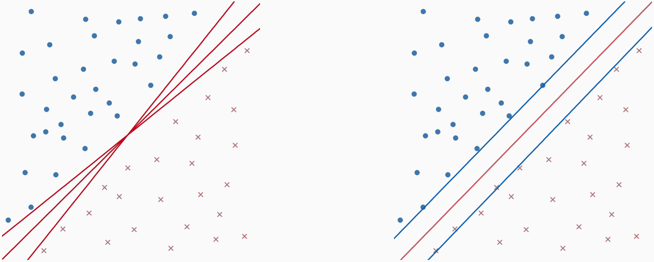
(b) SVM classifier model

Team works: previously

Perception model for synthesis images:

- Image noise detection in global illumination methods based on FRVM (*J. Constantin, A. Bigand, et al. 2015*)
- Perception of noise in global illumination based on inductive learning (*J. Constantin, I. Constantin, et al. 2016*)
- Perception of noise and global illumination: Toward an automatic stopping criterion based on SVM (*N. Takouachet et al. 2017*)

Support Vector Machine (SVM):



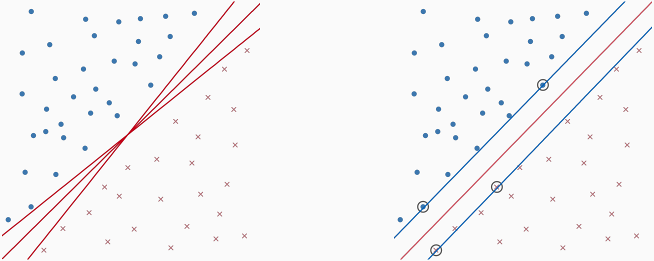
(a) Linear classifier model

(b) SVM classifier model

Perception model for synthesis images:

- Image noise detection in global illumination methods based on FRVM (*J. Constantin, A. Bigand, et al. 2015*)
- Perception of noise in global illumination based on inductive learning (*J. Constantin, I. Constantin, et al. 2016*)
- Perception of noise and global illumination: Toward an automatic stopping criterion based on SVM (*N. Takouachet et al. 2017*)

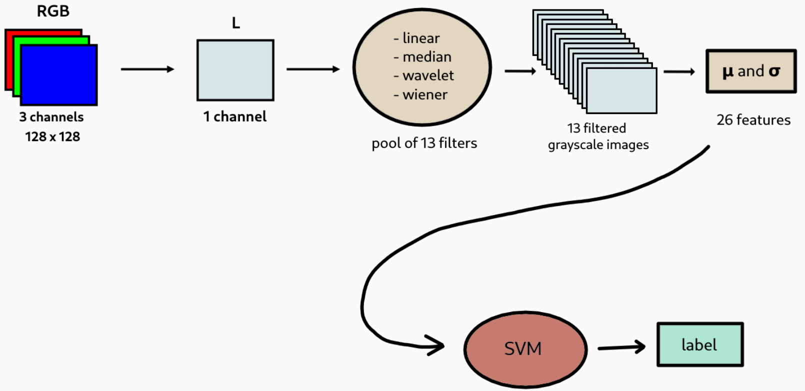
Support Vector Machine (SVM):



(a) Linear classifier model (b) SVM classifier model

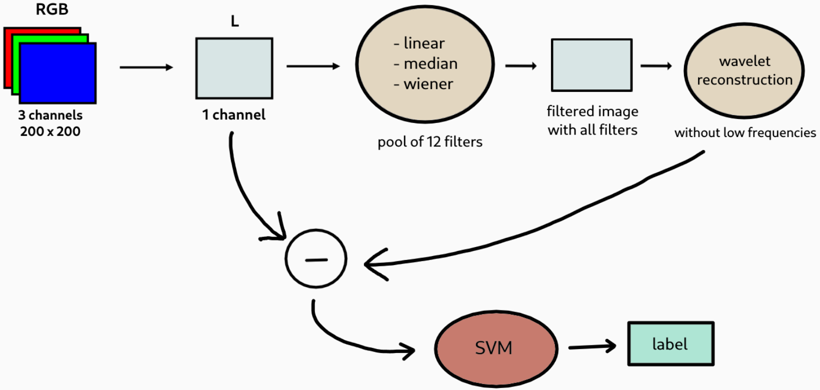
Team works: previously

Image noise detection in global illumination methods based on FRVM (*J. Constantin, A. Bigand, et al. 2015*)



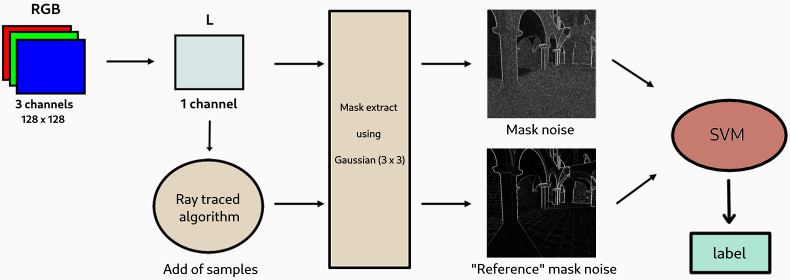
Team works: previously

Perception of noise in global illumination based on inductive learning (*J. Constantin, I. Constantin, et al. 2016*)



Team works: previously

Perception of noise and global illumination: Toward an automatic stopping criterion based on SVM (*N. Takouachet et al. 2017*)

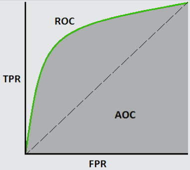


Team works: previously

Model	features	zones	Accuracy Train	Accuracy Test	AUC ROC Train	AUC ROC Test
SVM	(J. Constantin 2015)	12	0.9592	**0.8756**	0.9677	**0.8755**

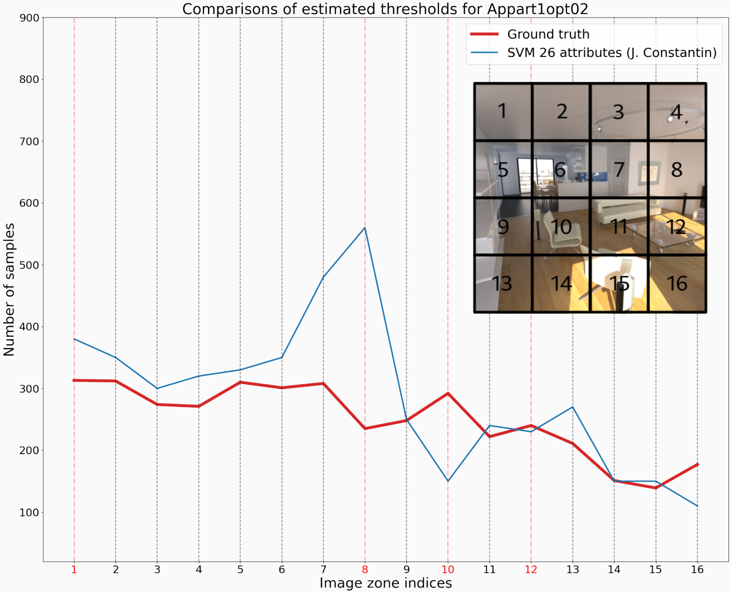
Training parameters

- Use of 4 viewpoints from 3 scenes (same renderer)
- 12 zones used from training / 4 for testing
- ROC is a probability curve and AUC represents degree or measure of separability



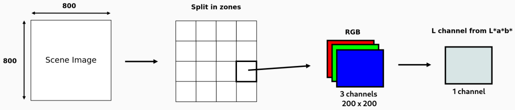
- $TPR = \frac{TP}{TP+FN}$
- $FPR = \frac{FP}{TN+FP}$

Team works: previously

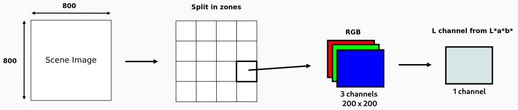


LISIC ANR

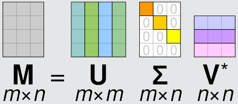
Team works: use of singular values



Team works: use of singular values



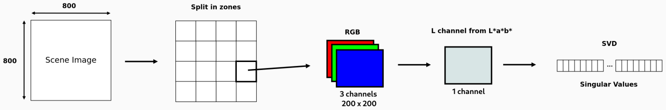
Singular Value Decomposition



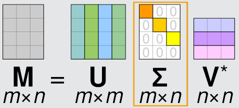
where:

- M is an $m \times n$ real or complex matrix
- U is an $m \times m$ real or complex unitary matrix.
- Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal.
- V is an $n \times n$ real or complex unitary matrix.

Team works: use of singular values



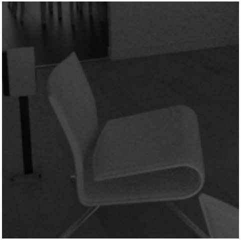
Singular Value decomposition



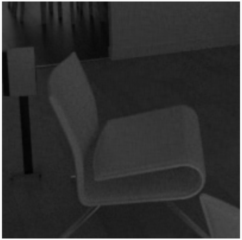
where:

- M is an $m \times n$ real or complex matrix
- U is an $m \times m$ real or complex unitary matrix.
- Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal.
- V is an $n \times n$ real or complex unitary matrix.

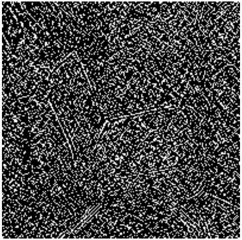
Possibility to decompose image using SVD into structure dependent and non-dependent images (Wang et al. 2013).



(a) L chanel (500 samples)

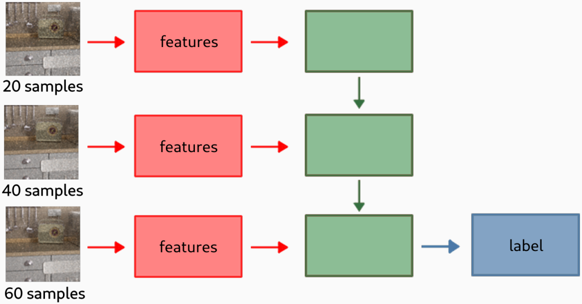


(b) SVD reconstruction (0, 50)



(c) SVD reconstruction (50, 200)

Team works: Recurrent Neural Networks



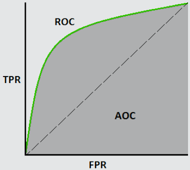
Input and parameters

- Σ singular values from SVD
- Window size of 5

Model	features	zones	Accuracy Train	Accuracy Test	AUC ROC Train	AUC ROC Test
SVM	(J. Constantin 2015)	12	0.9592	0.8756	0.9677	0.8755
RNN	Singular values [0, 200[	12	0.9404	**0.8966**	0.9249	**0.8859**

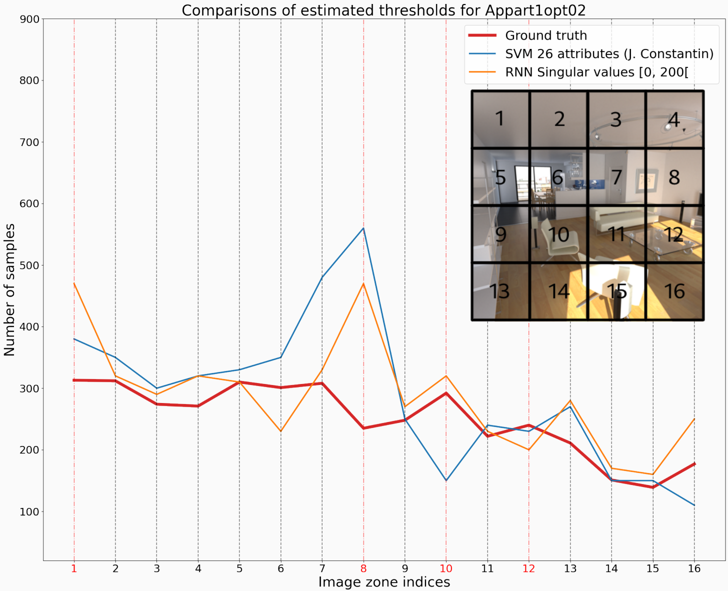
Training parameters

- Use of 4 viewpoints from 3 scenes (same renderer)
- 12 zones used from training / 4 for testing
- ROC is a probability curve and AUC represents degree or measure of separability



- $TPR = \frac{TP}{TP+FN}$
- $FPR = \frac{FP}{TN+FP}$

Team works: Recurrent Neural Networks



Team works: conclusion

Encountered problems:

- difficulty to generalize using dataset
- scene structure gives strong influence for model performance
- need more data to fit well

Relative & current works

1. How to build a such model ?
2. Previous & current team works
3. **Deep Learning approaches**

DL approaches: build of new dataset

Previous dataset

- 9 viewpoints from scenes
- different renderers (maxwell, igloo, cycle...)
- hence, different algorithms

Previous dataset

- 9 viewpoints from scenes
- different renderers (maxwell, igloo, cycle...)
- hence, different algorithms

New dataset

- 40 viewpoints with 10000 images of 1 sample
- only **pbrt-v3** renderer
- use of **path-tracing**
- available soon

DL approach: build of new dataset

Why saving image of 1 sample ?

- generate $\binom{10000}{k}$ images of k samples from pool of 10000 samples

DL approach: build of new dataset

Why saving image of 1 sample ?

- generate $\binom{10000}{k}$ images of k samples from pool of 10000 samples
  $\Rightarrow \binom{10000}{20} \approx 4.3\mathrm{e}61$

Why saving image of 1 sample ?

- generate $\binom{10000}{k}$ images of k samples from pool of 10000 samples $\Rightarrow \binom{10000}{20} \approx 4.3\mathrm{e}61$
- **posterior** study of samples **distribution**

DL approach: build of new dataset

Why saving image of 1 sample ?

- generate $\binom{10000}{k}$ images of k samples from pool of 10000 samples
  $\Rightarrow \binom{10000}{20} \approx 4.3\mathrm{e}61$
- **posterior** study of samples **distribution**
- use of deep learning approach (RNN, GAN, Autoencoder...)

Conclusion

Conclusion

Presented works:

- Singular values vector seems to fit well using RNN
- Lack of data and need of new dataset
- Enable posterior samples study using this new dataset

Perspectives

Improve dataset:

- check convergence of all generated scenes
- use of web experiment (SIN3D) app to collect human thresholds

Perspectives

Improve dataset:

- check convergence of all generated scenes
- use of web experiment (SIN3D) app to collect human thresholds

Use of Deep Learning:

- exploit new dataset with CNN / RNN
  - using preprocessed images
  - features based only
- samples distribution study
- denoising approaches

Thanks for your attention

Resources:

- **Scene files:** `https://gogs.univ-littoral.fr/Prise3D/p3d_pbrt-scenes.git`

References

- Mathieu Carnec, Patrick Le Callet, and Dominique Barba. "An image quality assessment method based on perception of structural information". In: *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*. Vol. 3. IEEE. 2003, pp. III–185.
- Joseph Constantin, Andre A. Bigand, et al. "Image noise detection in global illumination methods based on FRVM". In: *Neurocomputing* 164 (2015), pp. 82–95.
- Joseph Constantin, Ibtissam Constantin, et al. "Perception of noise in global illumination based on inductive learning". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2016, pp. 5021–5028.

- Eric P Lafortune and Yves D Willems. "Bi-directional path tracing". In: (1993).
- Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. "No-reference image quality assessment in the spatial domain". In: *IEEE Transactions on image processing* 21.12 (2012), pp. 4695–4708.
- Nawel N. Takouachet et al. "Perception of noise and global illumination: Toward an automatic stopping criterion based on SVM". In: *Computers & Graphics* 69 (2017), pp. 49–58.
- Manish Narwaria et al. "HDR-VDP-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images". In: *Journal of Electronic Imaging* 24.1 (2015), p. 010501.
- Eric Veach and Leonidas J Guibas. "Metropolis light transport". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 1997, pp. 65–76.

Jiřié Vorba et al. "Path guiding in production". In: *ACM SIGGRAPH 2019 Courses*. 2019, pp. 1–77.

Shuigen Wang et al. "A novel SVD-based image quality assessment metric". In: *2013 IEEE International Conference on Image Processing*. IEEE. 2013, pp. 423–426.

Backup: SIN3D calibration

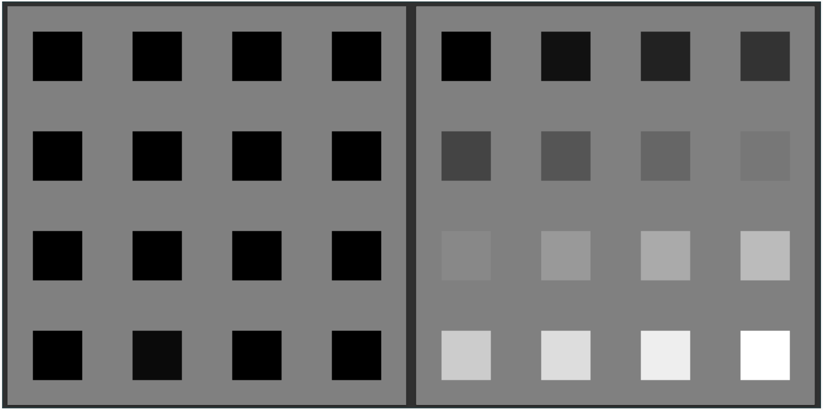


Figure 10: Calibration scene

Backup: SIN3D overview

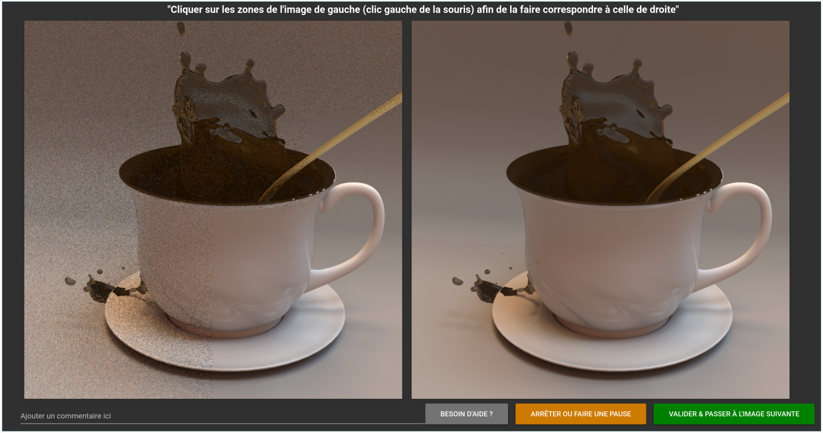
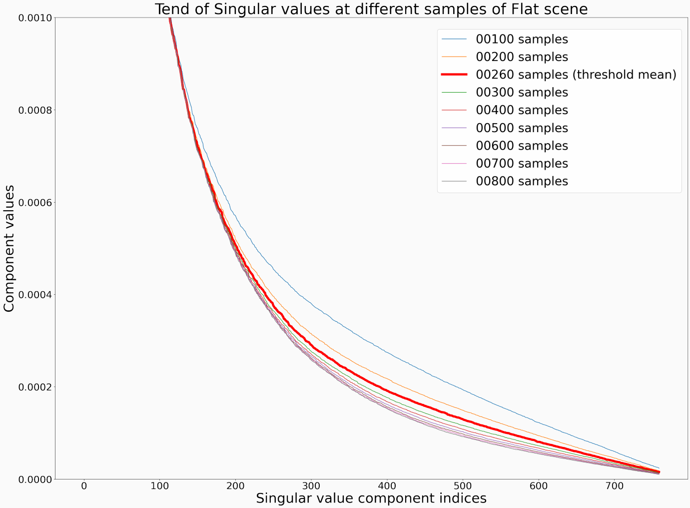
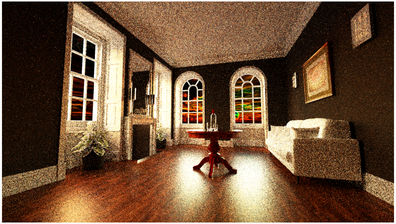


Figure 11: User interface

Backup: use of singular values



Backup: distribution analysis



(a) Variance

(b) Standard deviation

(c) Skewness

(d) Kurtosis

Backup: RAWLS

New dataset:

- Use of new image format: **RAWLS** for *RAW Light Simulation*

Python package:

rawls https://prise-3d.github.io/rawls