Euclidean Distance Transform Shadow Mapping

Márcio C. F. Macedo (UFBA – Brazil)
Antônio L. Apolinário Jr. (UFBA – Brazil)

PGCOMP (UFBA – Brazil)

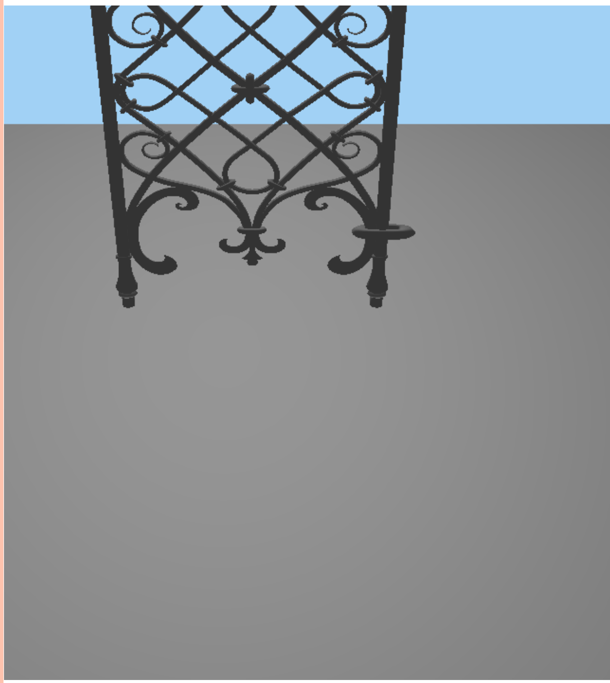
Agenda

- Introduction;
- Euclidean Distance Transform Shadow Mapping;
- Results and Discussion;
- Conclusion and Future Work;

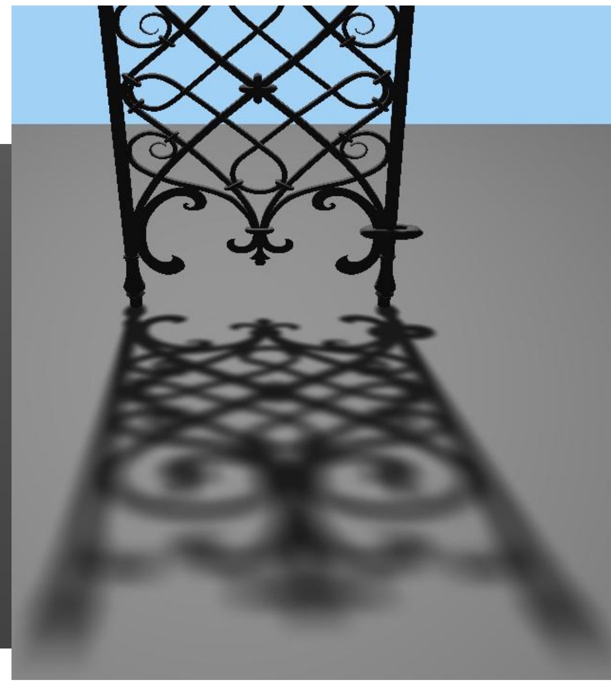
Introduction

Context

No Shadow



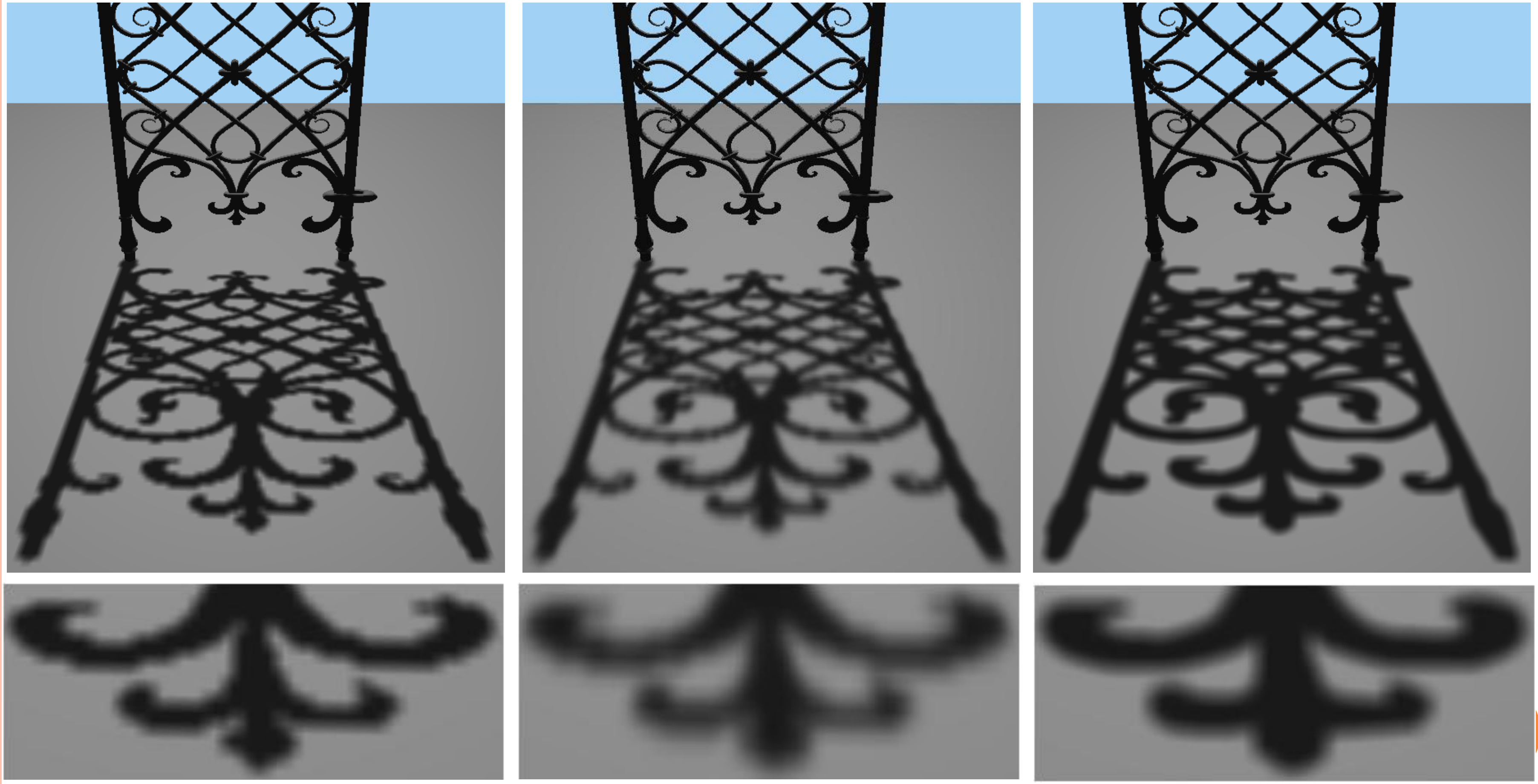
Accurate Shadows



Hard Shadows

Current Scenario

PCF [Reeves1987] | VSM [Donnelly2006] | EDTSM



[Reeves1987] – W. T. Reeves, D. H. Salesin, and R. L. Cook. "Rendering Antialiased Shadows with Depth Maps". Proceedings of the SIGGRAPH, 1987.

[Donnelly2006] – W. Donnelly and A. Lauritzen. "Variance Shadow Maps". Proceedings of the I3D, 2006.

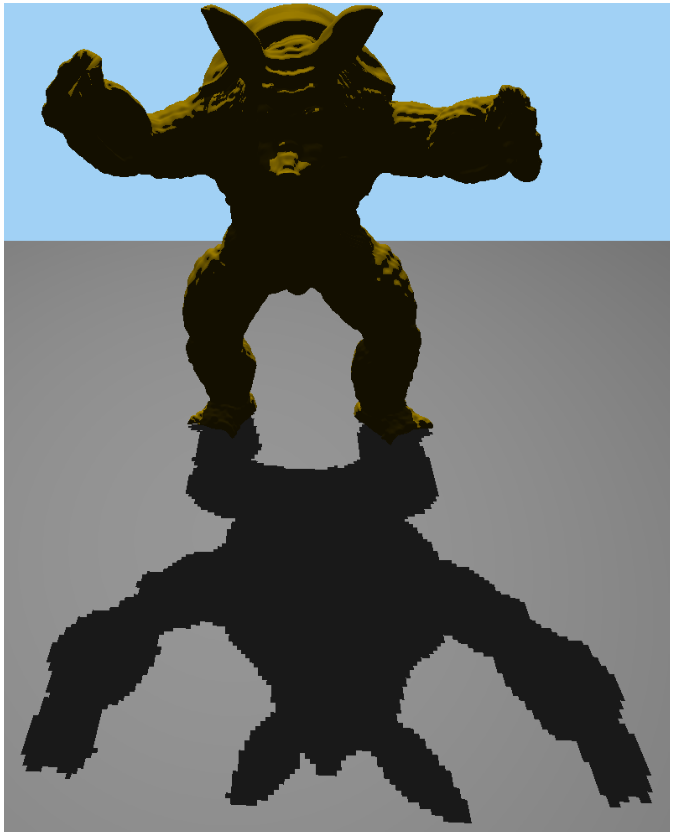
Euclidean Distance Transform Shadow Mapping

ALGORITHM

- Step 1 - Shadow Map Rendering:

Algorithm

- Step 2 - Shadow Mapping [Williams1978]:



[Williams1978] – L. Williams. "Casting Curved Shadows on Curved Surfaces". Proceedings of the SIGGRAPH, 1978.

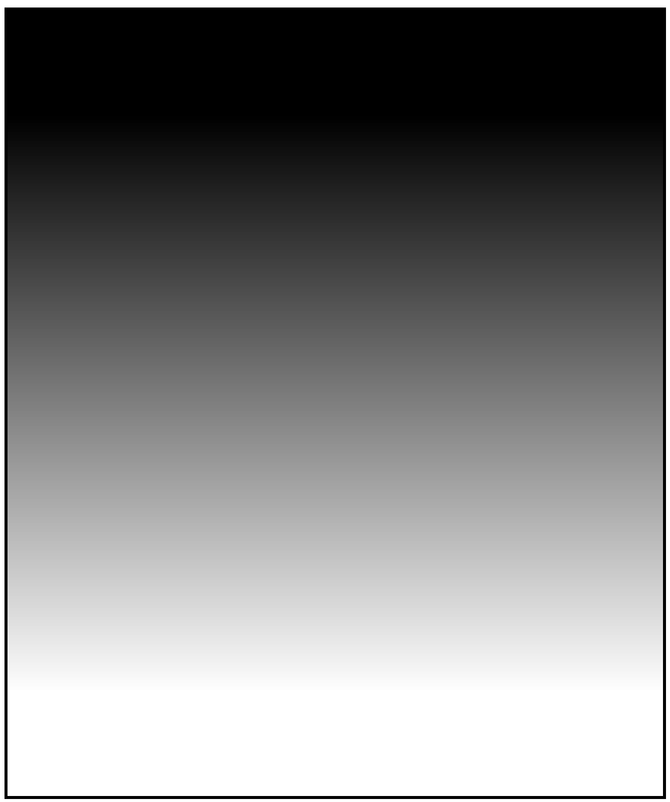
Algorithm

- Step 3 - Shadow Revectorization [Macedo2016]:

[Macedo2016] – M. Macedo, A. Apolinário. “Revectorization-Based Shadow Mapping”. Proceedings of Graphics Interface, 2016.

Algorithm

- Step 4 – EDT Shadowing:

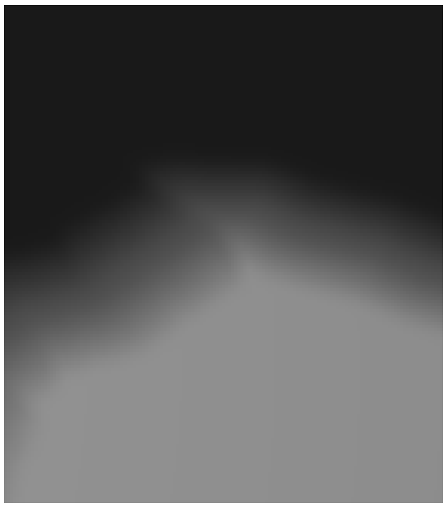


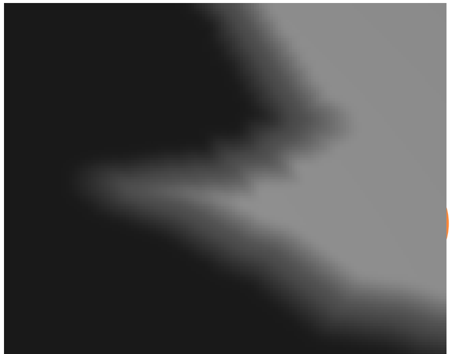
$$D = \frac{1}{2} + \frac{D}{P}$$

$$D = \frac{1}{2} - \frac{D}{P}$$

Algorithm

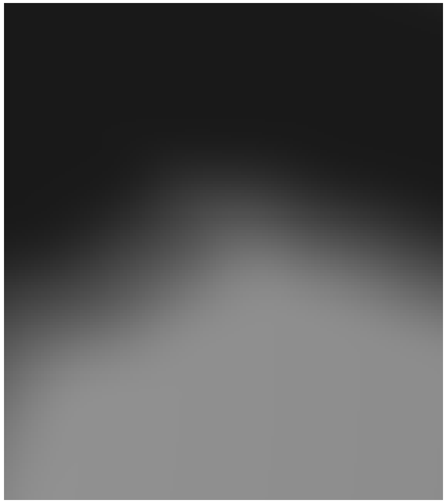
- Step 5 – EDT Filtering:

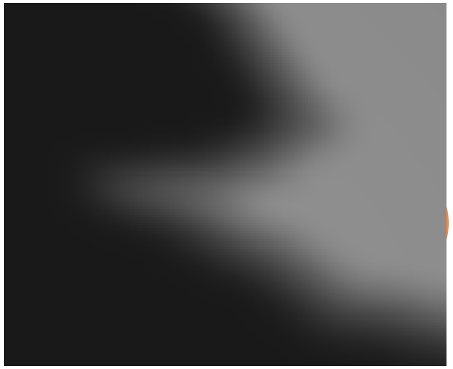




ALGORITHM

- Step 5 – EDT Filtering:





Results and Discussion

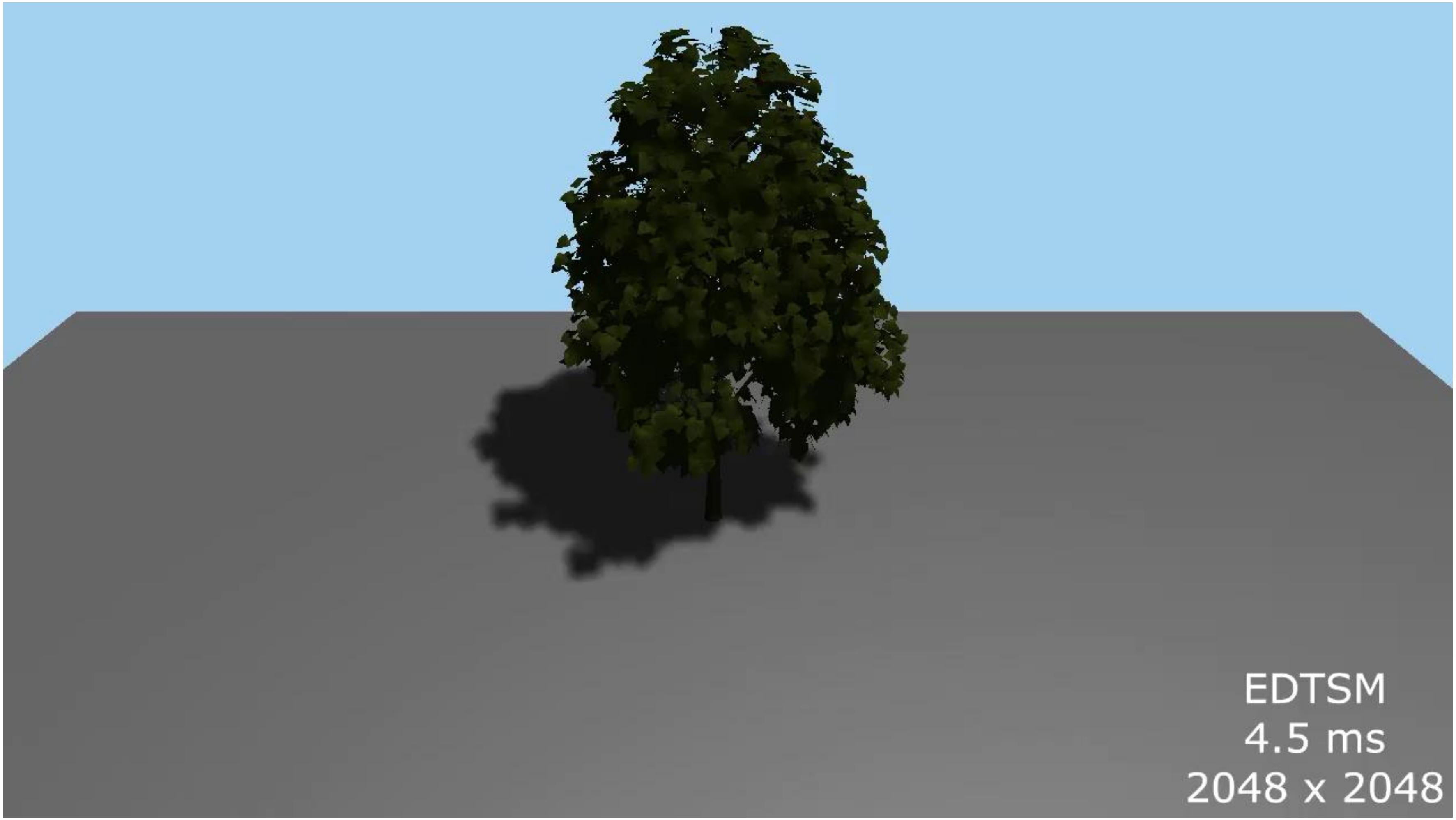
Experimental Setup

- For all tests, we used an Intel® Core™ i7-3770K CPU @3.50Ghz, 8GB RAM, NVIDIA GeForce GTX Titan X;
- EDTSM (our approach) was implemented using OpenGL and GLSL languages;
- To compute the EDT, we have used the PBA algorithm [Cao2010] implemented in CUDA;
- A kernel size of 15 x 15 was used to suppress skeleton and banding artifacts for our technique and related work;

[Cao2010] – T.-T. Cao, K. Tang, A. Mohamed, and T.-S. Tan. "Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU". Proceedings of the I3D, 2010.

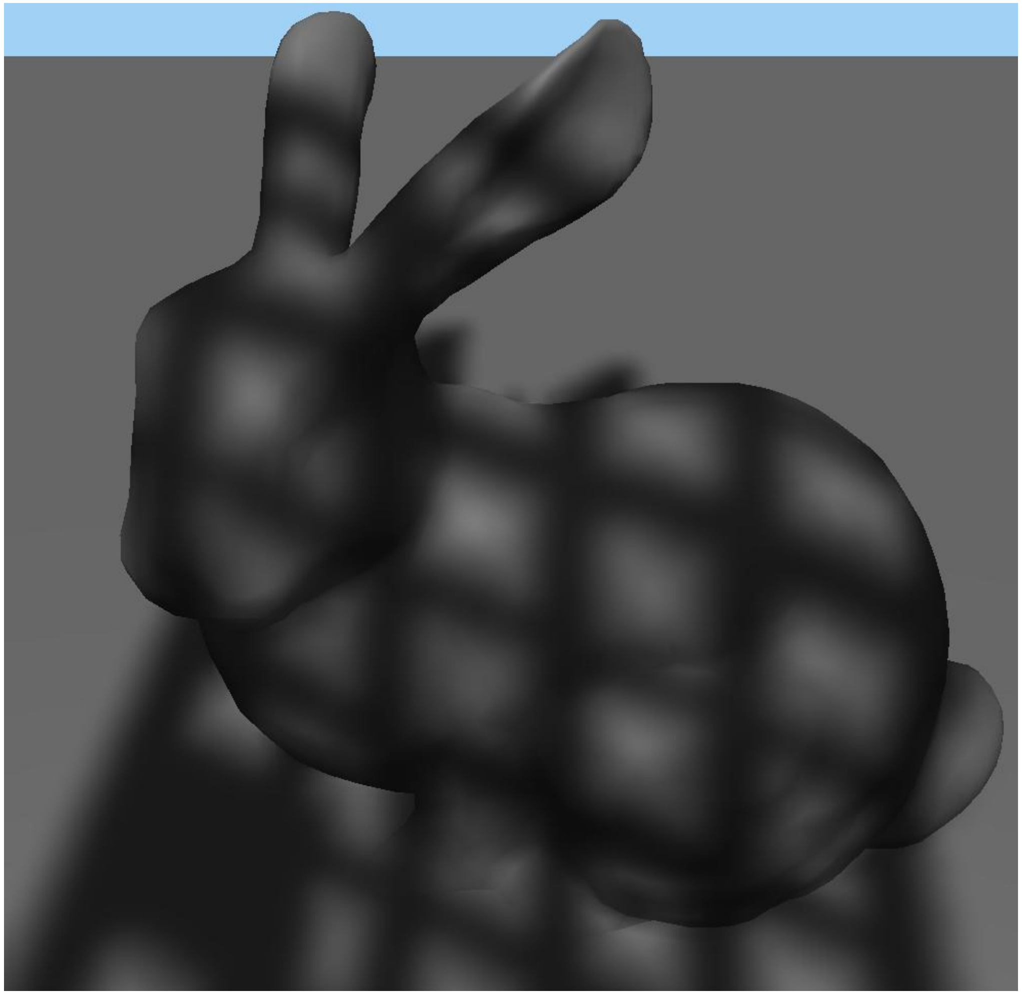
Visual Quality

- Temporal Coherence:



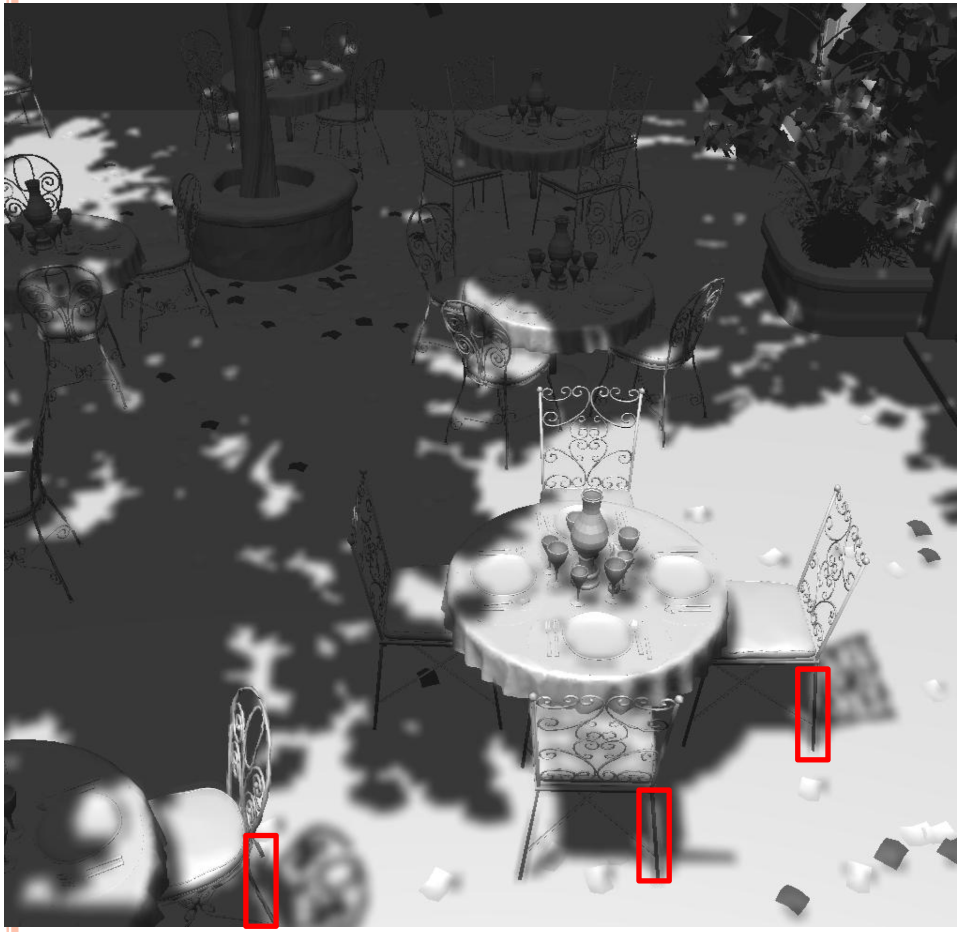
Visual Quality

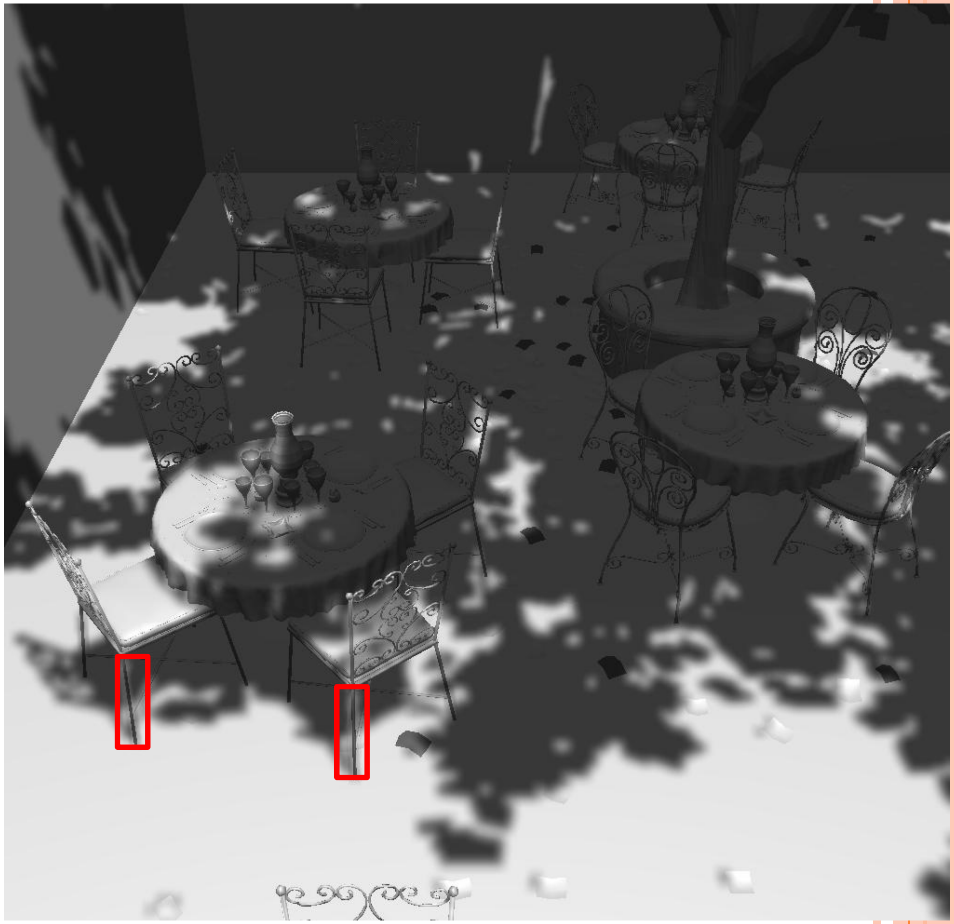
- Non-Planar Receiver:



Visual Quality

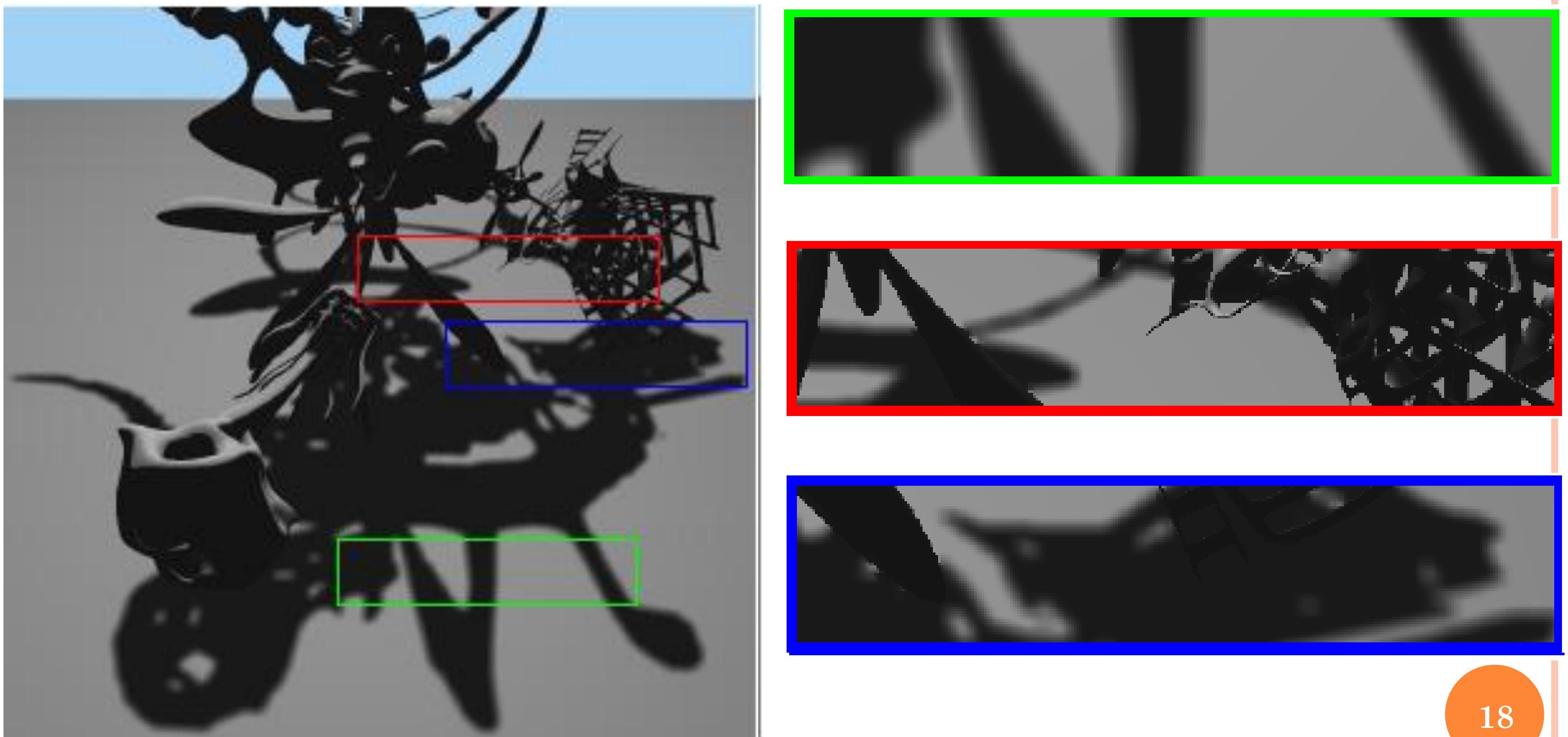
- Complex Scenario:





VISUAL QUALITY

- Related Work:



[illegible]

[Reeves1987] – W. T. Reeves, D. H. Salesin, and R. L. Cook. “Rendering Antialiased Shadows with Depth Maps”.
[illegible]

Rendering Time

- Shadow Map Resolution:

Scene	Method	Shadow Map Resolution 512^2	1024^2	2048^2	4096^2
1	PF	4.1 ms	4.3 ms	4.5 ms	5.5 ms
	PCF	5.0 ms	5.1 ms	5.2 ms	5.6 ms
	EDTSM	6.3 ms	6.4 ms	6.5 ms	7.4 ms
	RPCF	22.2 ms	22.7 ms	23.2 ms	27.7 ms
2	PF	4.7 ms	4.8 ms	5.1 ms	6.3 ms
	PCF	5.3 ms	5.4 ms	5.6 ms	6.4 ms
	EDTSM	6.4 ms	6.5 ms	6.7 ms	7.5 ms
	RPCF	12.9 ms	13.6 ms	15.2 ms	17.8 ms
3	PF	10.5 ms	10.6 ms	10.8 ms	12.0 ms
	PCF	11.3 ms	11.4 ms	11.4 ms	11.7 ms
	EDTSM	12.8 ms	12.9 ms	13.0 ms	13.7 ms
	RPCF	26.2 ms	27.3 ms	27.8 ms	30.0 ms

Rendering Time

- Viewport/Output Resolution:

		Output Resolution		
Scene	**Method**	SD	HD	Full-HD
1	PF	3.2 ms	4.3 ms	4.8 ms
	PCF	3.2 ms	5.1 ms	5.3 ms
	EDTSM	4.1 ms	6.4 ms	8.0 ms
	RPCF	10.6 ms	22.7 ms	25.0 ms
2	PF	3.2 ms	4.7 ms	5.7 ms
	PCF	3.2 ms	5.4 ms	5.8 ms
	EDTSM	4.3 ms	6.4 ms	8.1 ms
	RPCF	7.1 ms	12.9 ms	16.6 ms
3	PF	9.9 ms	10.6 ms	11.3 ms
	PCF	9.8 ms	11.4 ms	11.9 ms
	EDTSM	10.7 ms	12.9 ms	14.7 ms
	RPCF	16.4 ms	27.3 ms	30.3 ms

Rendering Time

- Kernel Size:

Scene	Method	Kernel Size 7 × 7	15 × 15	23 × 23	31 × 31
1	PF	3.9 ms	4.3 ms	4.5 ms	4.7 ms
	PCF	3.4 ms	5.1 ms	7.4 ms	10.2 ms
	EDTSM	5.9 ms	6.4 ms	6.8 ms	7.2 ms
	RPCF	22.2 ms	76.9 ms	142.8 ms	200.0 ms
2	PF	4.5 ms	4.7 ms	5.1 ms	5.3 ms
	PCF	3.5 ms	5.4 ms	7.5 ms	10.5 ms
	EDTSM	6.2 ms	6.4 ms	6.7 ms	7.0 ms
	RPCF	12.9 ms	39.6 ms	89.2 ms	142.8 ms
3	PF	9.8 ms	10.6 ms	10.7 ms	11.1 ms
	PCF	9.8 ms	11.4 ms	13.5 ms	17.0 ms
	EDTSM	12.3 ms	12.9 ms	13.5 ms	14.2 ms
	RPCF	26.2 ms	77.5 ms	166.6 ms	285.7 ms

Conclusion and Future Work

Final Considerations

- Conclusion:
  - Our technique outperforms related work in terms of visual quality, mainly for low-resolution shadow maps;
  - Our technique is more scalable than PCF for high order filter sizes;
  - We believe that our approach is useful for games and other interactive applications;
- Future Work:
  - Minimize shadow overestimation;
  - Speed up the EDT computation;
  - Extend the approach for soft shadows;

ACKNOWLEDGMENTS

- We are grateful to:
  - The authors of [Cao2010] - for sharing the source code for GPU-Based Euclidean Distance Transform computation;
  - NVIDIA Corporation – for providing the NVIDIA GeForce GTX Titan X through the GPU Education Center program;
  - CAPES – for financial support;

[Cao2010] – T.-T. Cao, K. Tang, A. Mohamed, and T.-S. Tan. "Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU". Proceedings of the I3D, 2010.

Thank You!

Márcio C. F. Macedo (marciocfmacedo@gmail.com)
Antônio L. Apolinário Jr. (apolinario@dcc.ufba.br)