SBGames
2017 Curitiba PR

Universidade Federal da Bahia

Hard Shadow Anti-Aliasing for Spot Lights in a Game Engine

Márcio C. F. Macedo (marciocfmacedo@gmail.com)

Almir V. S. Teixeira (a.vinicius.12@gmail.com)

Antonio L. Apolinário Jr. (apolinario@dcc.ufba.br)

Karl A. Aguero (karl@dcc.ufba.br)

PGCOMP (UFBA –BRAZIL)

Agenda

- Introduction;
- Shadow Mapping;
- Revectorization-based Shadow Mapping;
- Results and Discussion;
- Conclusion and Future Works.

Introduction

Context

- Shadows are important in games because:
  - they improve the visual perception of the player;

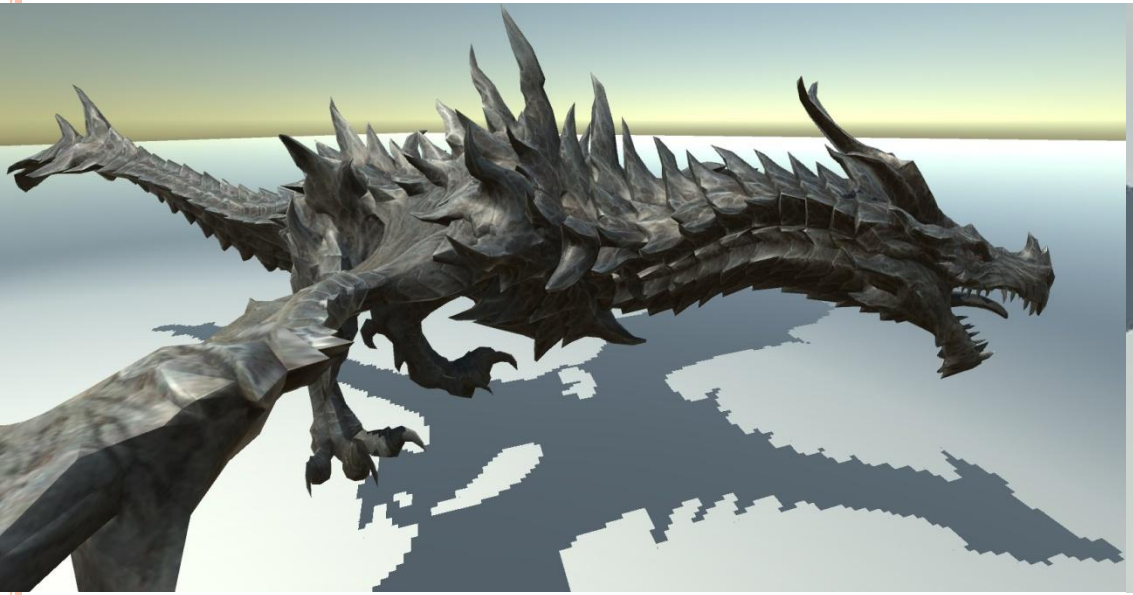
No Shadow

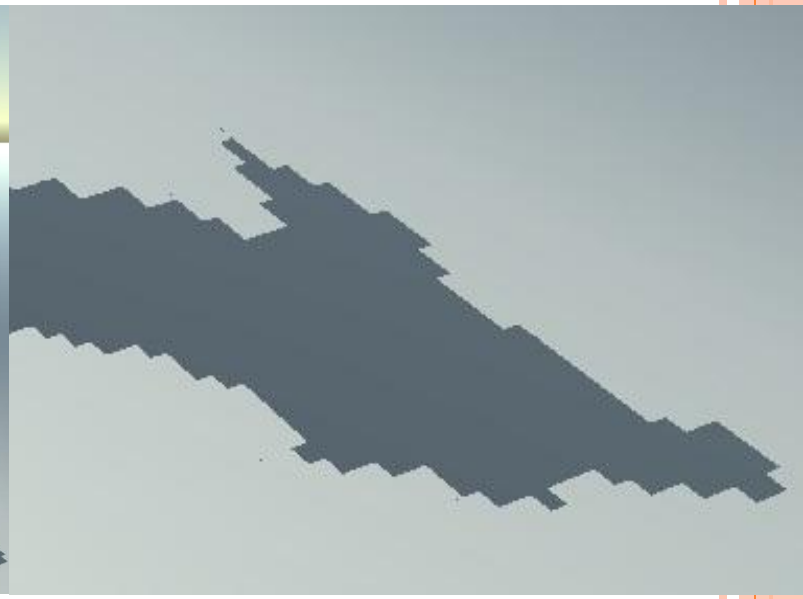
Shadows

Dynamic Shadows on Mass Effect 3. Available in https://www.geforce.com/whats-new/guides/mass-effect-3-tweak-guide#6

Context

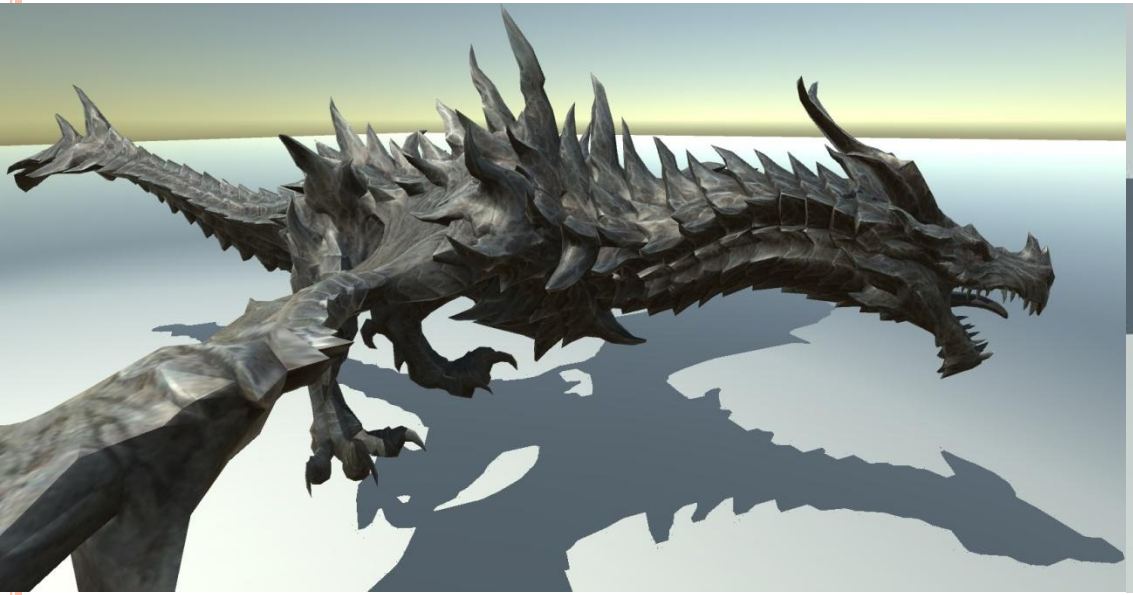
- Shadow Mapping [Williams1978]:
  - Default shadow technique for game engines;
  - Real-time performance;
  - Aliasing artifacts;

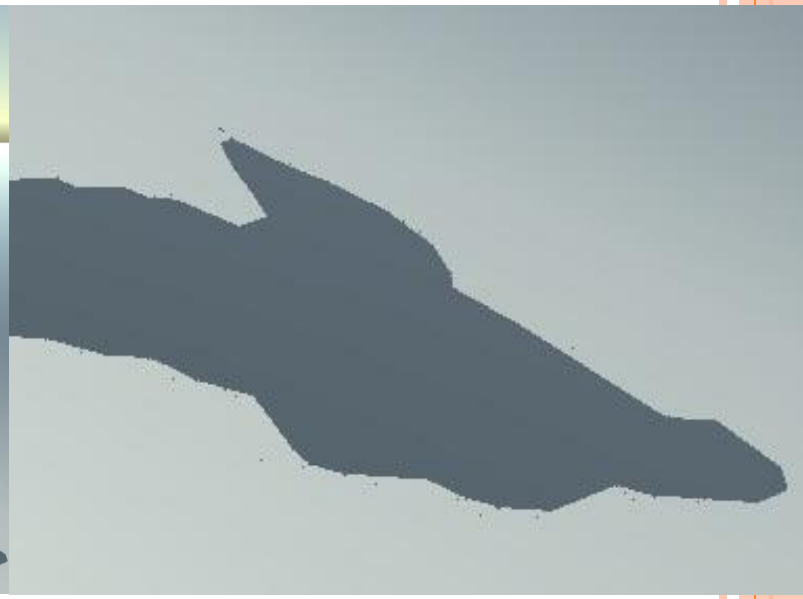




Context

- Revectorization-based Shadow Mapping [Macedo2016]:
  - Shadow anti-aliasing;
  - High visual quality;
  - Minimal overhead;

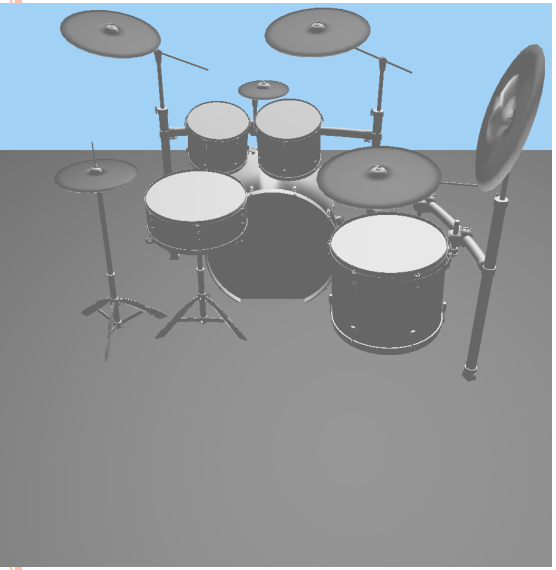




Shadow Mapping

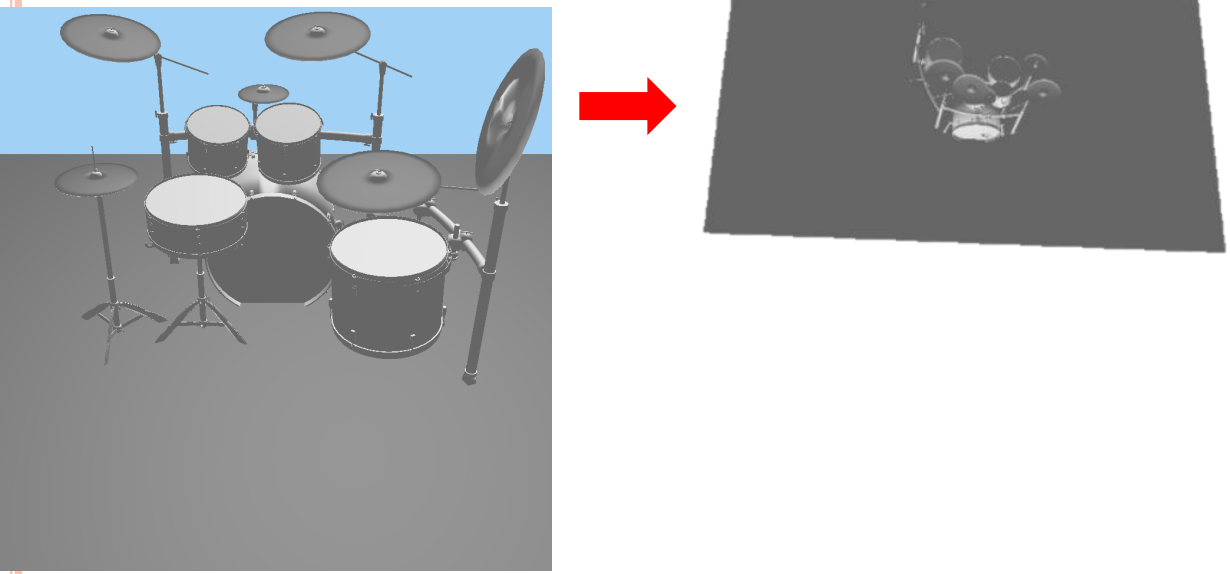
Shadow Mapping

- Step 1 – Load the scene;



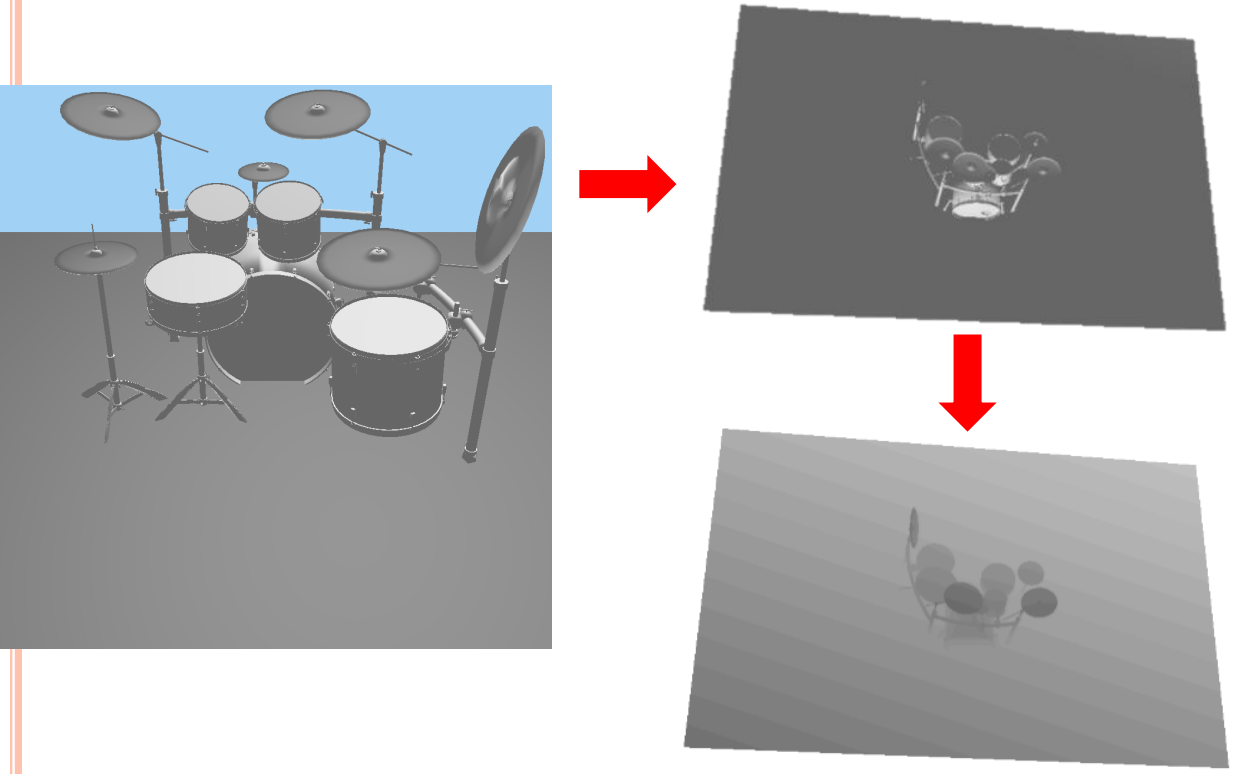
Shadow Mapping

- Step 2 – Render the scene from light source's viewpoint;



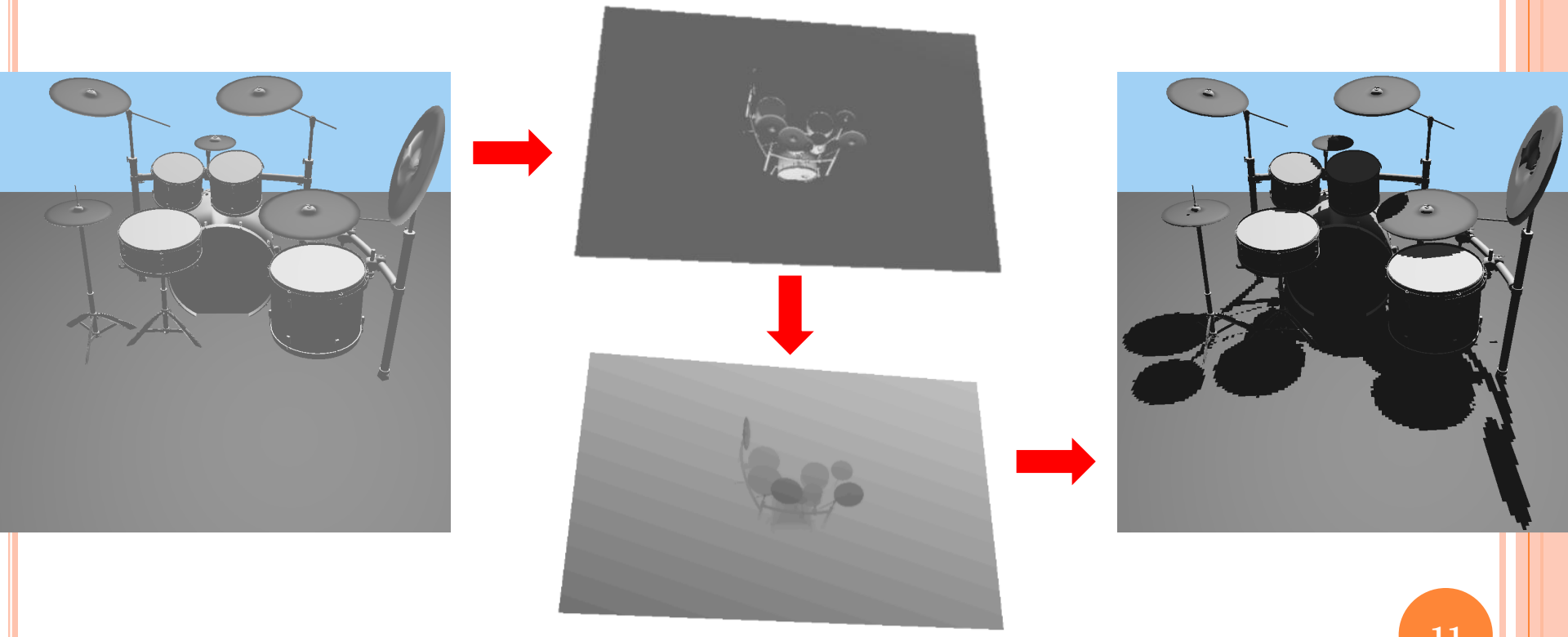
Shadow Mapping

- Step 3 – Generate a shadow map;



Shadow Mapping

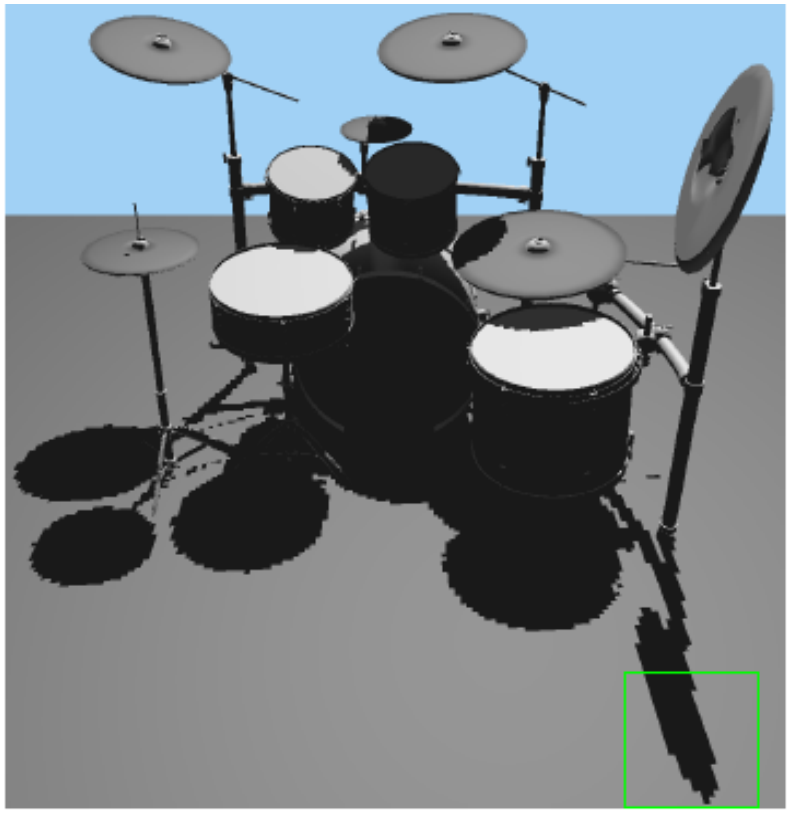
- Step 4 – Compute the real-time shadows;

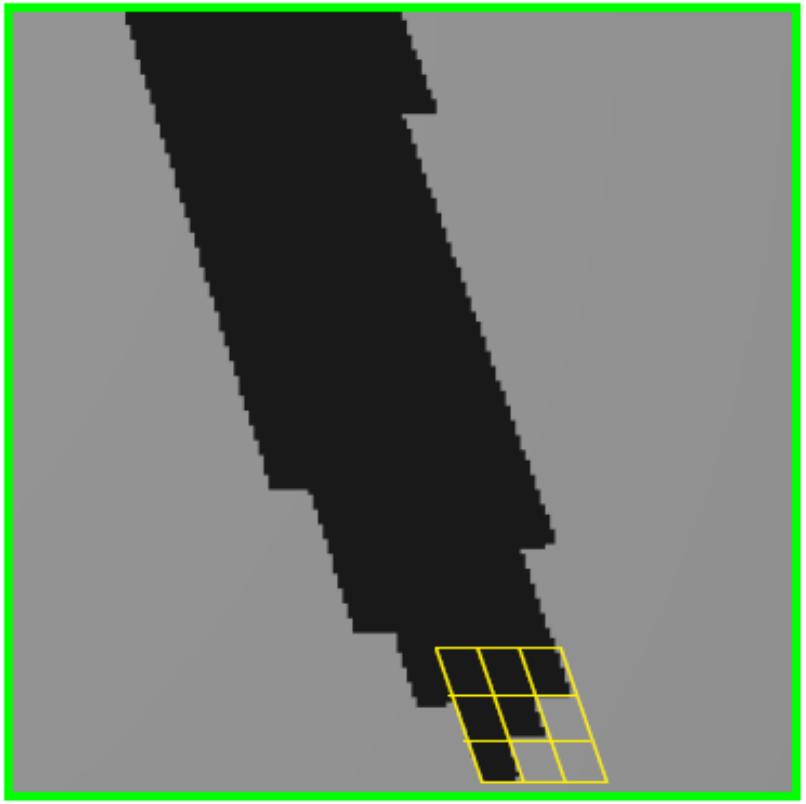


Revectorization-based Shadow Mapping

Revectorization-based Shadow Mapping

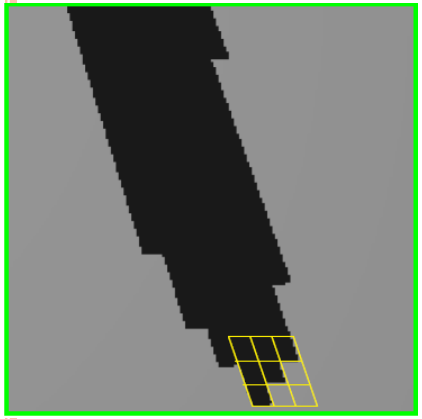
- This technique aims to solve the aliasing problem generated by shadow mapping;





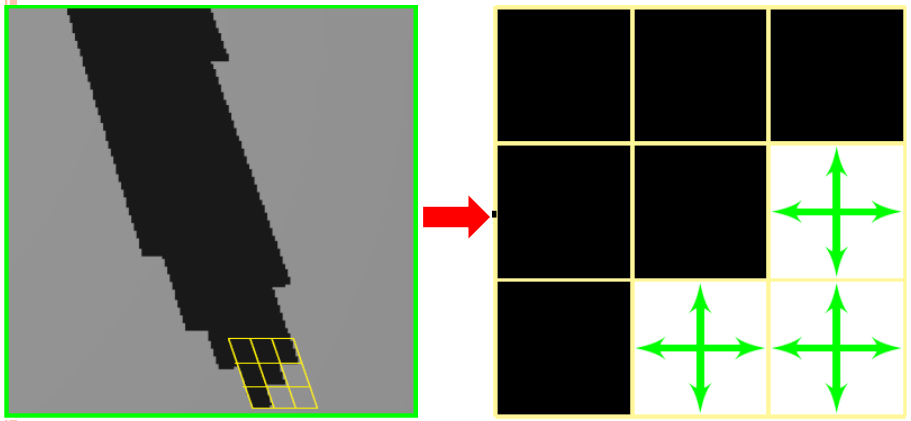
Revectorization-based Shadow Mapping

- Step 1 – Render the aliased shadow;



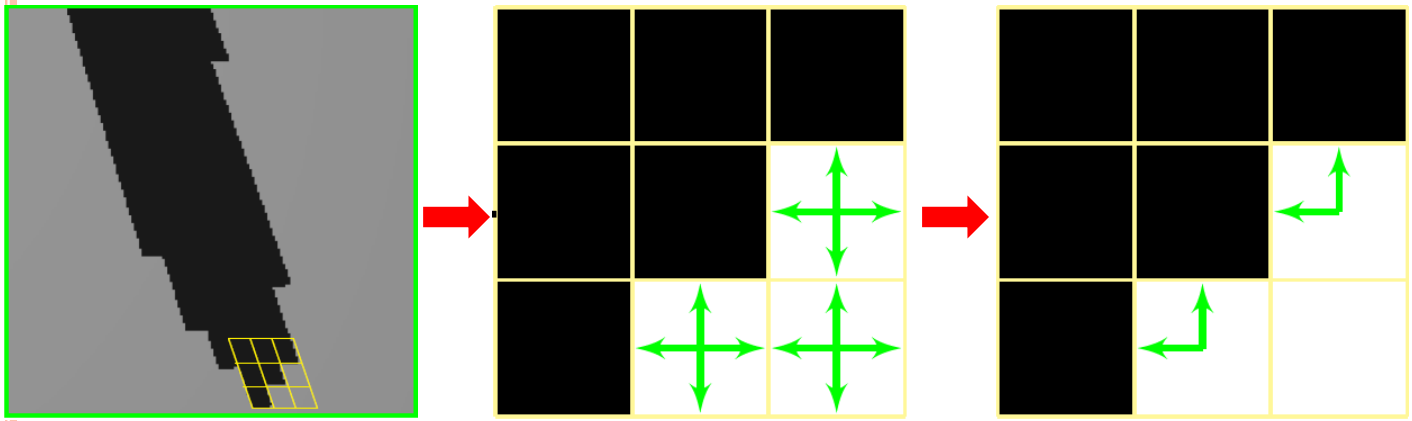
Revectorization-based Shadow Mapping

- Step 2 – Evaluate visibility of neighbourhood;



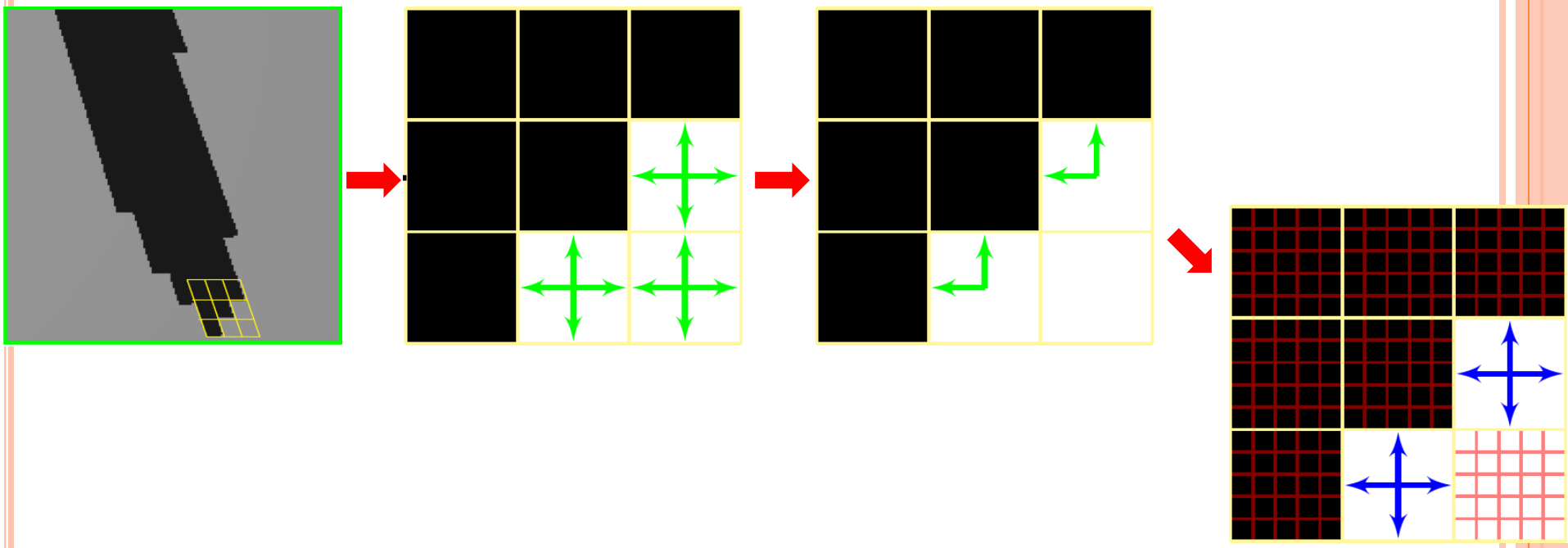
Revectorization-based Shadow Mapping

- Step 3 – Detect aliasing directions;



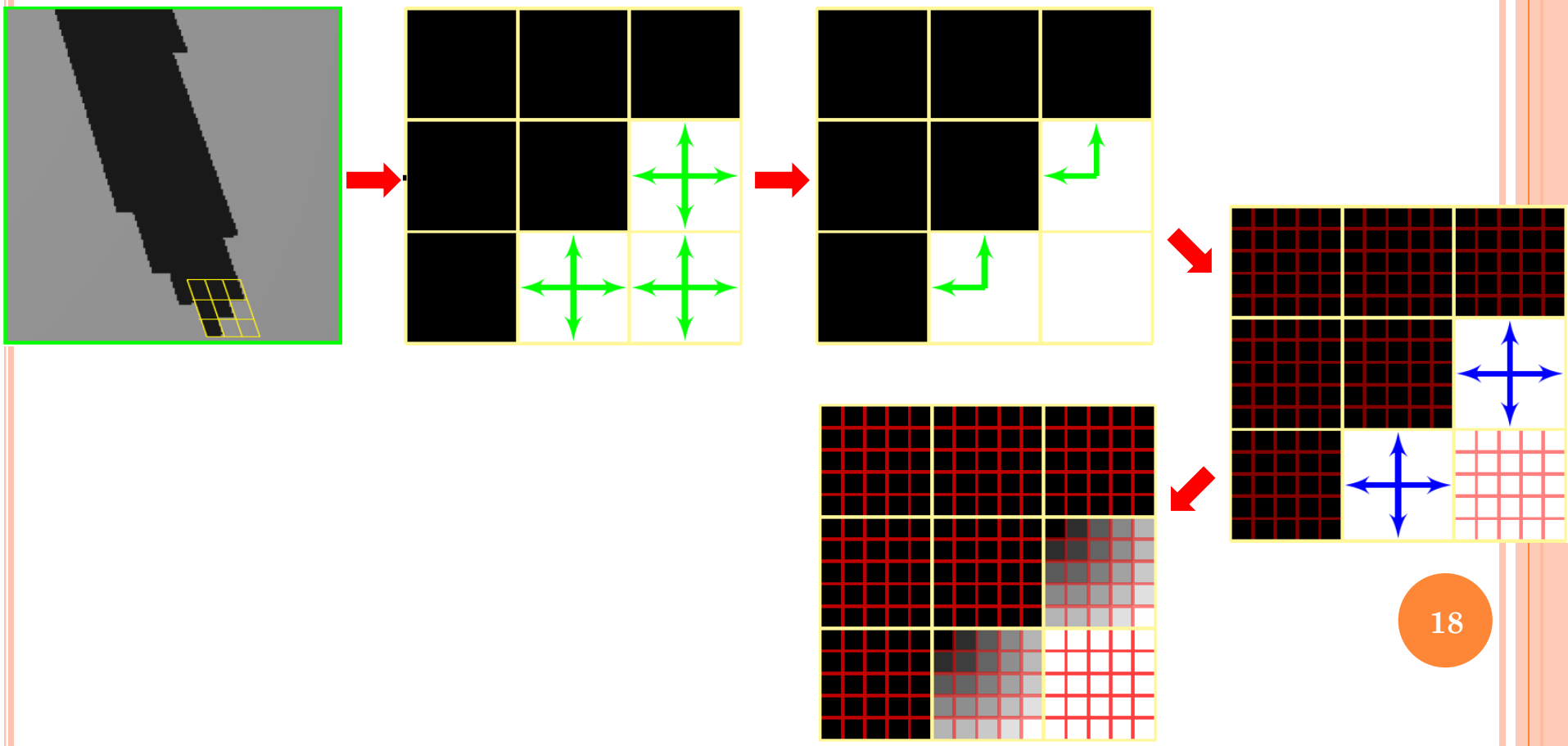
Revectorization-based Shadow Mapping

- Step 4 – Traverse the aliased shadow boundary;



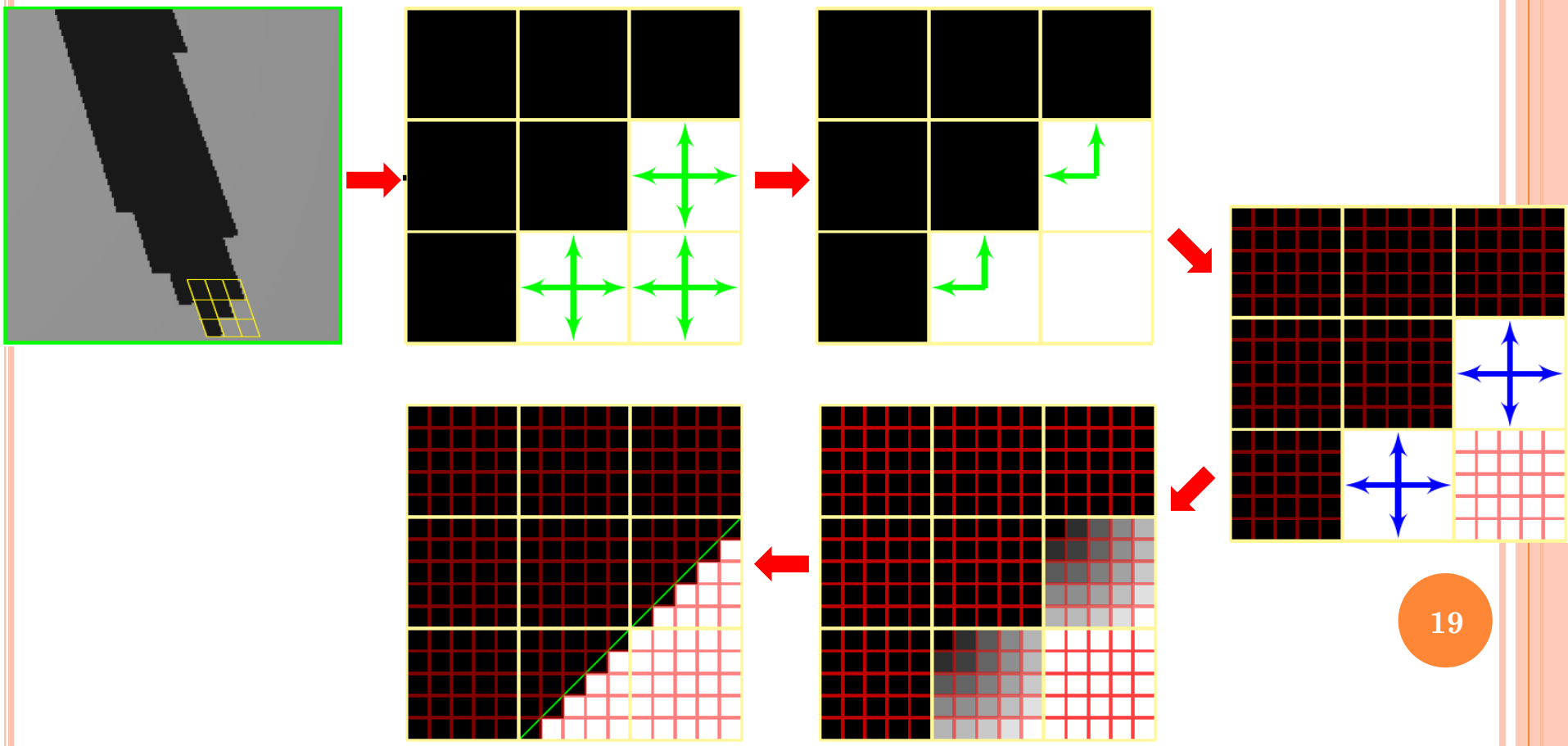
Revectorization-based Shadow Mapping

- Step 5 – Normalize distance to aliasing;



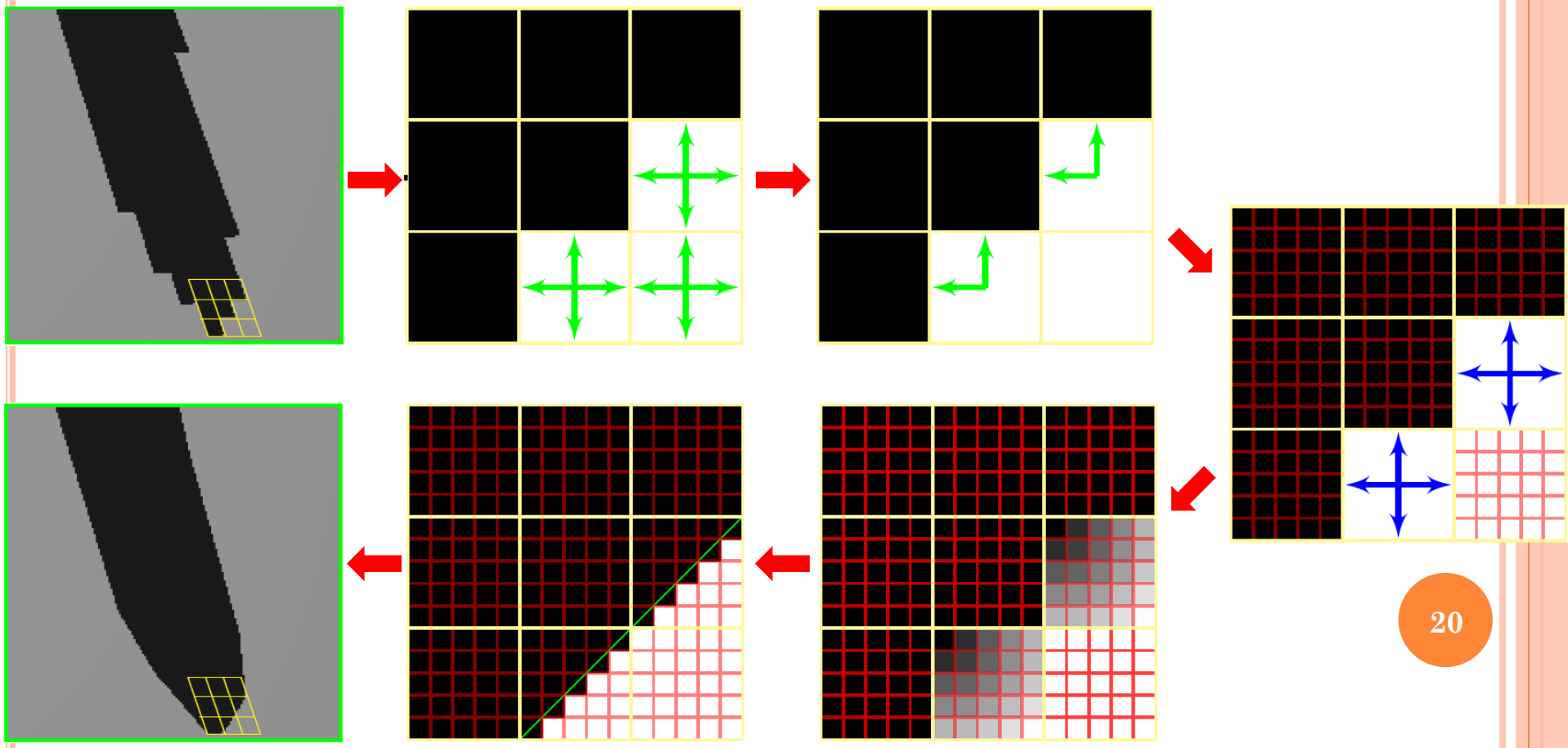
Revectorization-based Shadow Mapping

- Step 6 – Revectorize the shadow boundary;



Revectorization-based Shadow Mapping

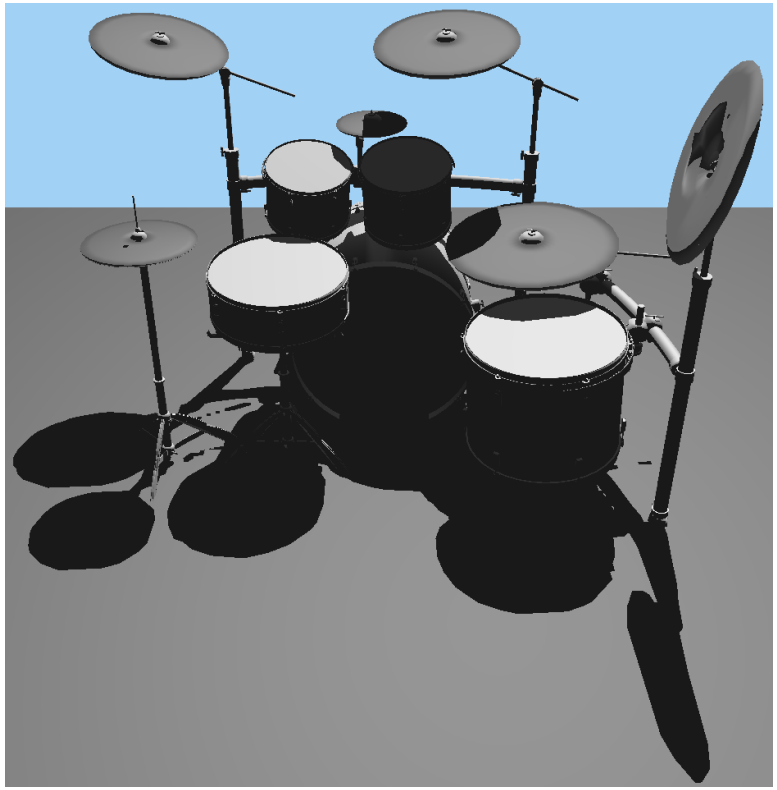
- Step 7 – Render the anti-aliased shadow;



Revectorization-based Shadow Mapping

Shadow Mapping

RBSM



Results and Discussion

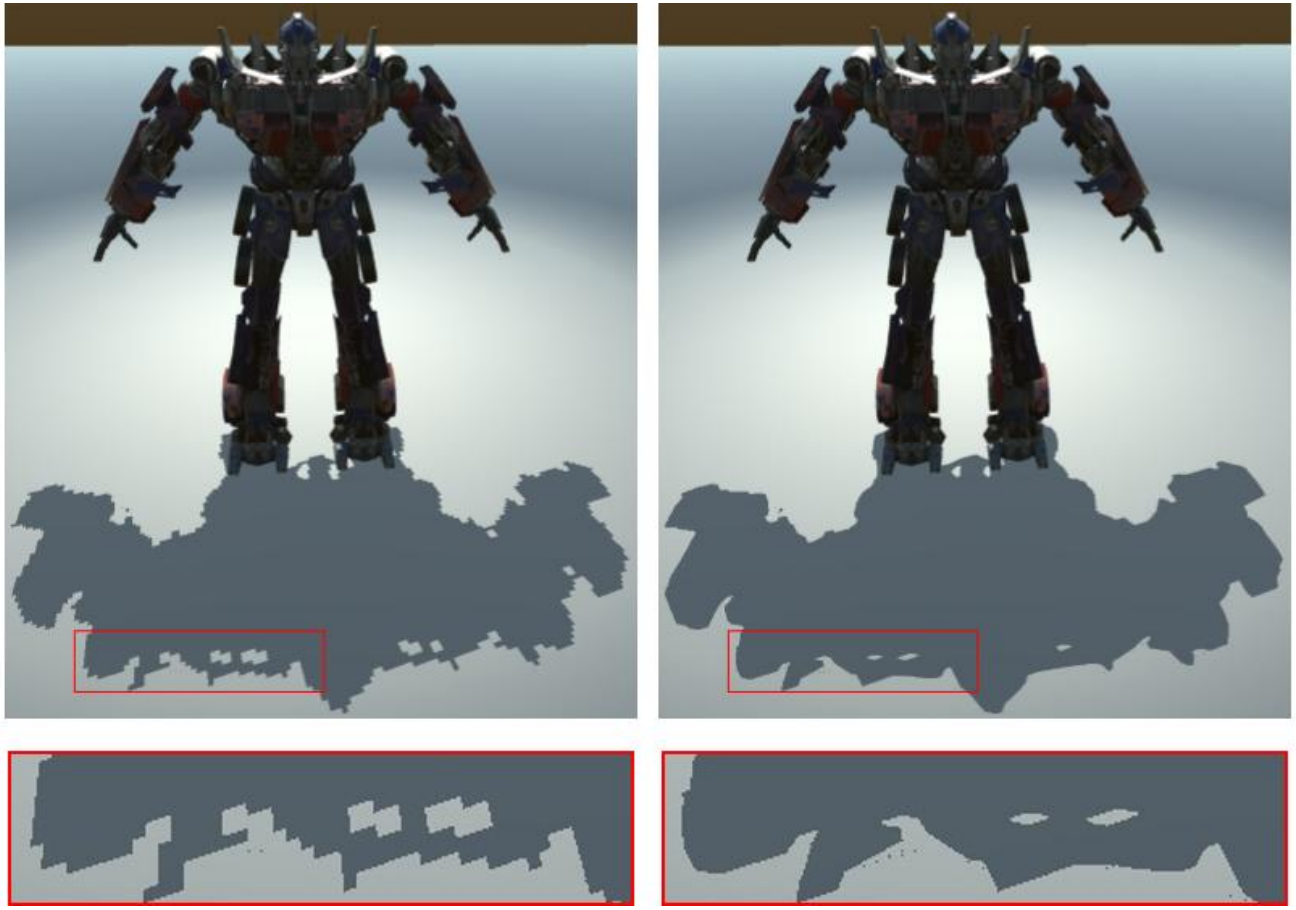
Experimental Setup

- Hardware:
  - NVIDIA GeForce GTX Titan X graphics card;
  - Intel CoreTM i7-3770K CPU (3.50 GHz);
  - 8GB RAM;

- Software:
  - Unity3D Pro version 5.6.0.f3;

Results

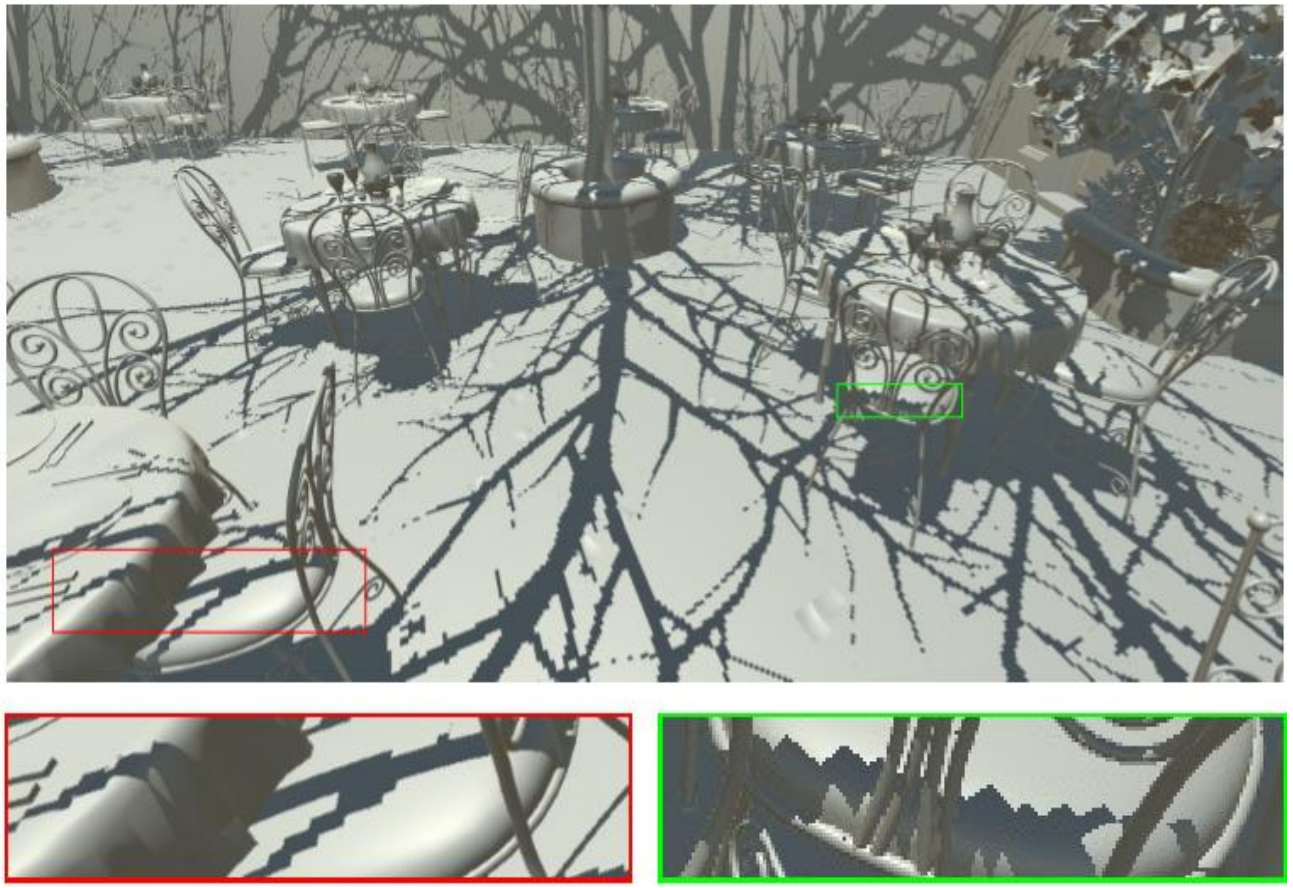
- Visual Quality:



Shadow Mapping RBSM

Results

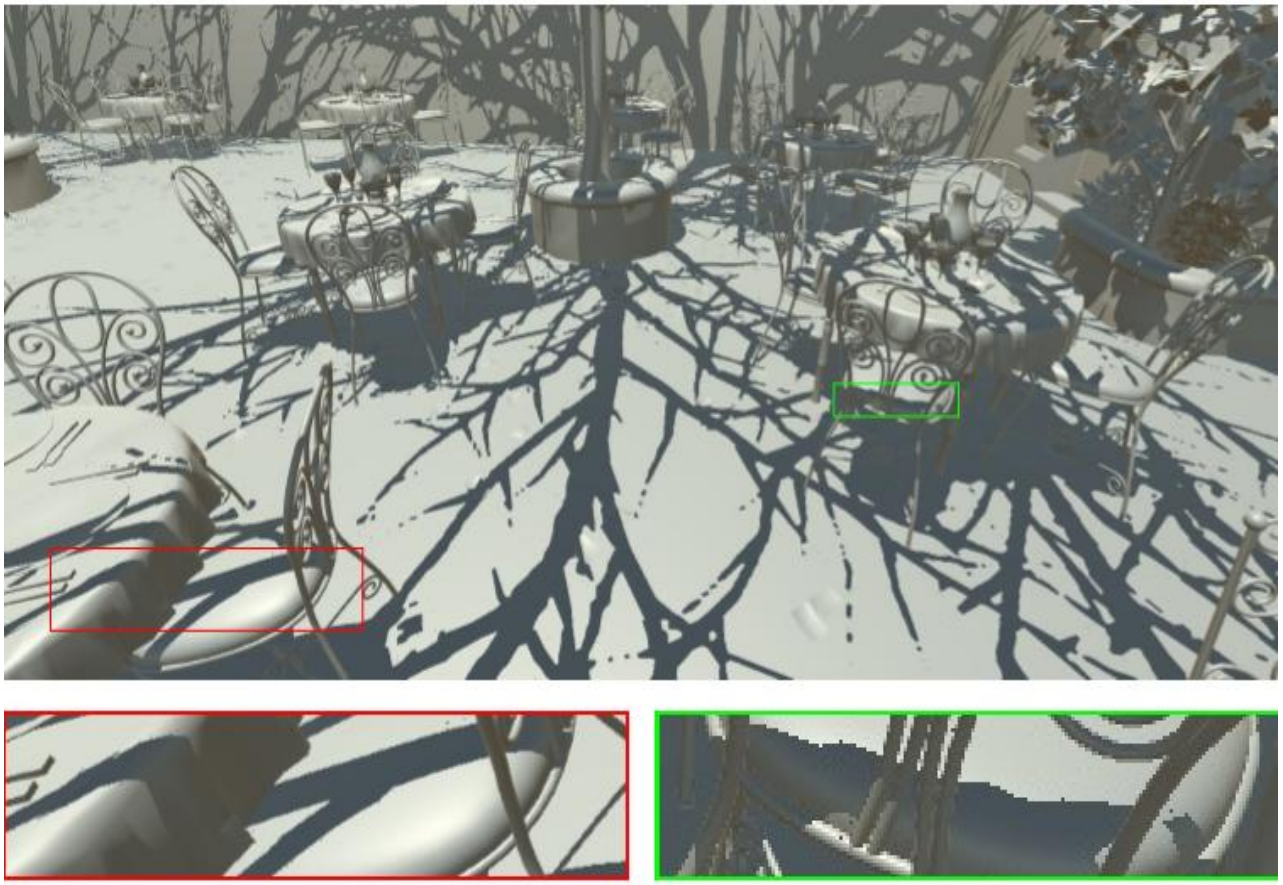
- Visual Quality:



Shadow Mapping

Results

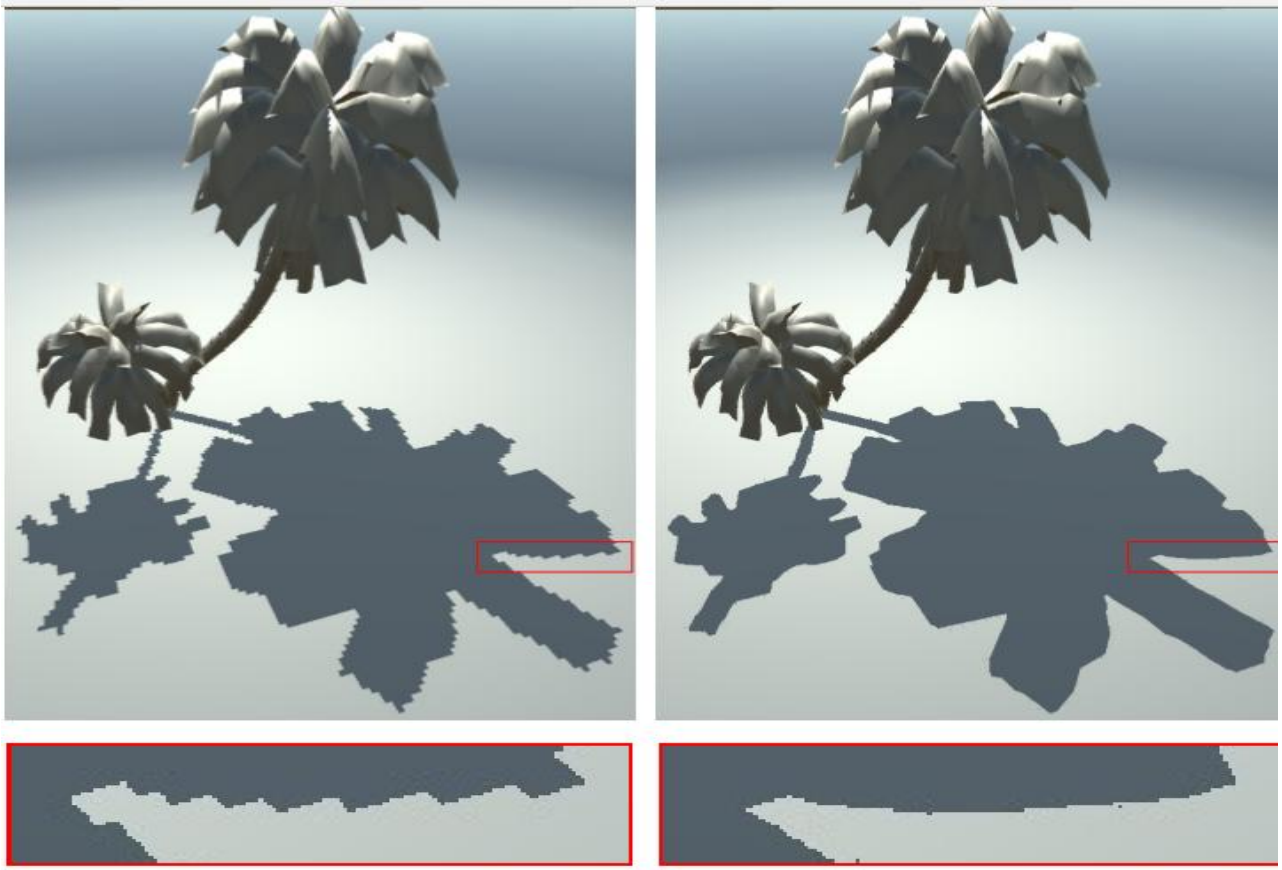
- Visual Quality:



RBSM

Results

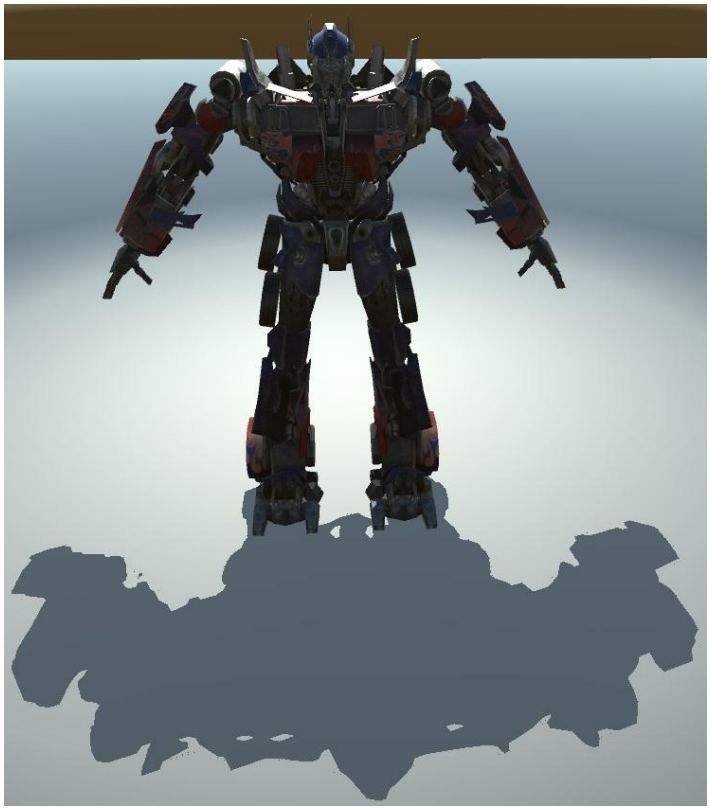
- Visual Quality:

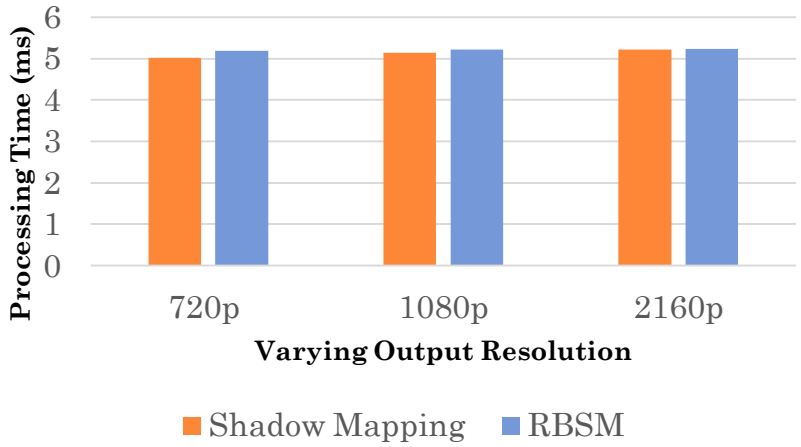


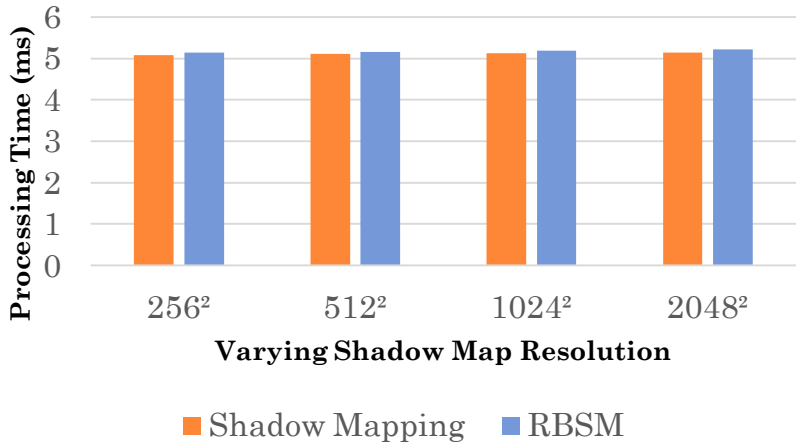
Shadow Mapping RBSM

Results

- Rendering Time:

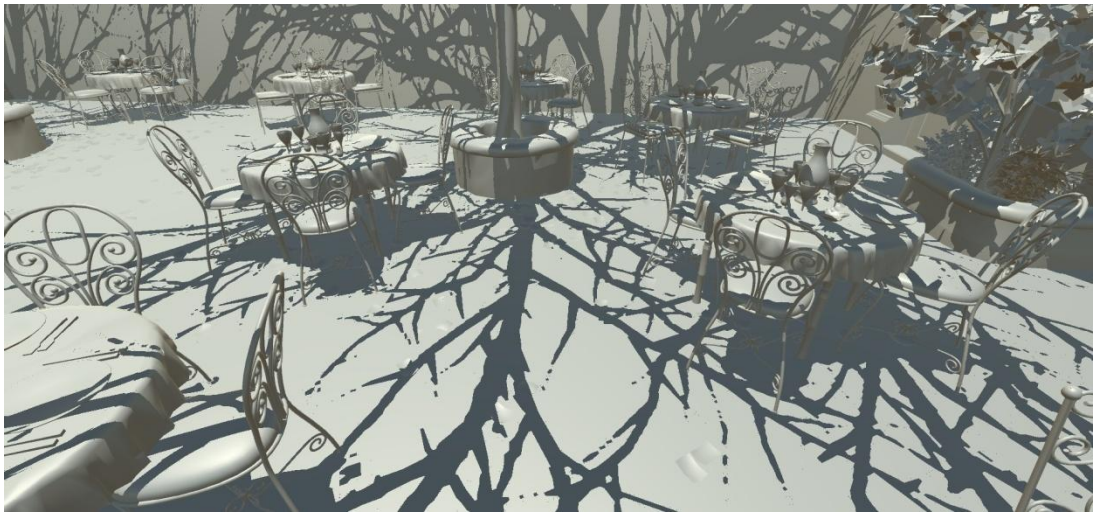


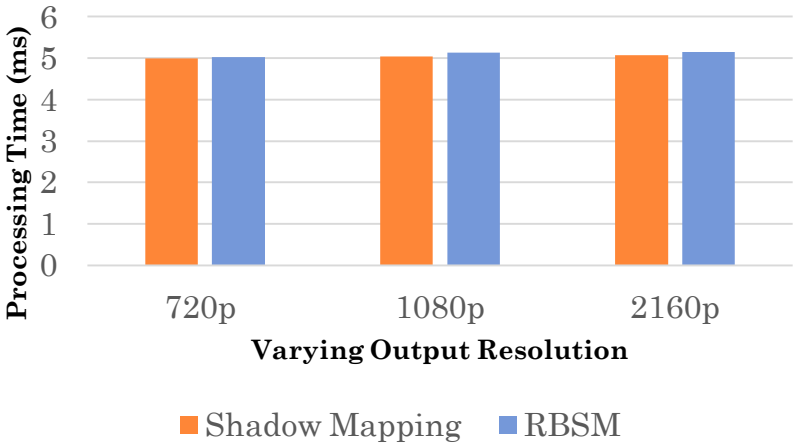


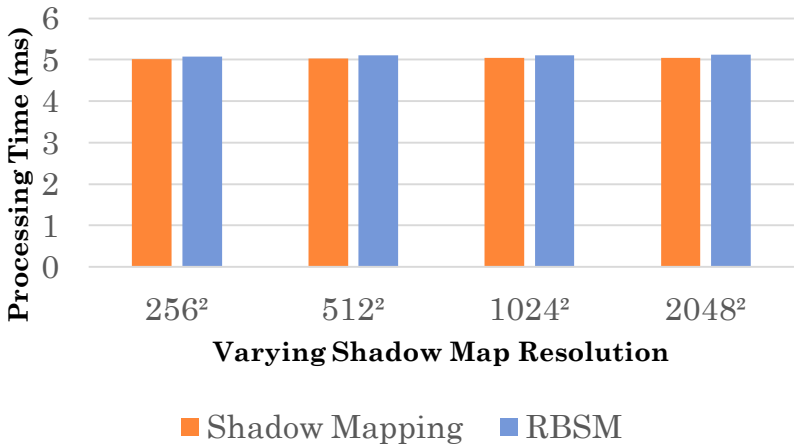


Results

- Rendering Time:

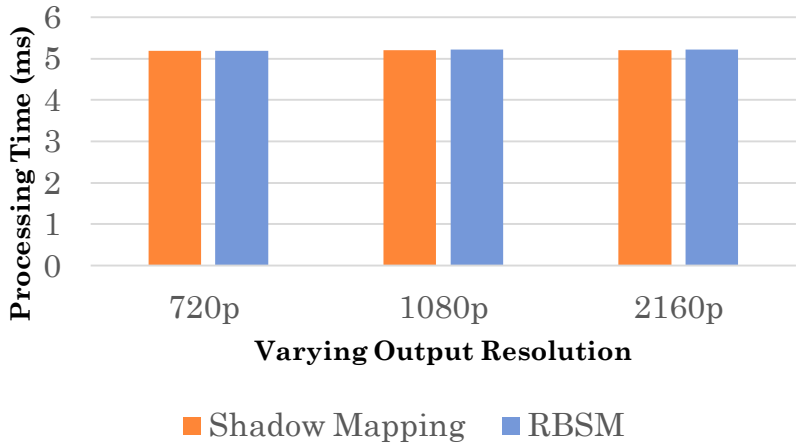


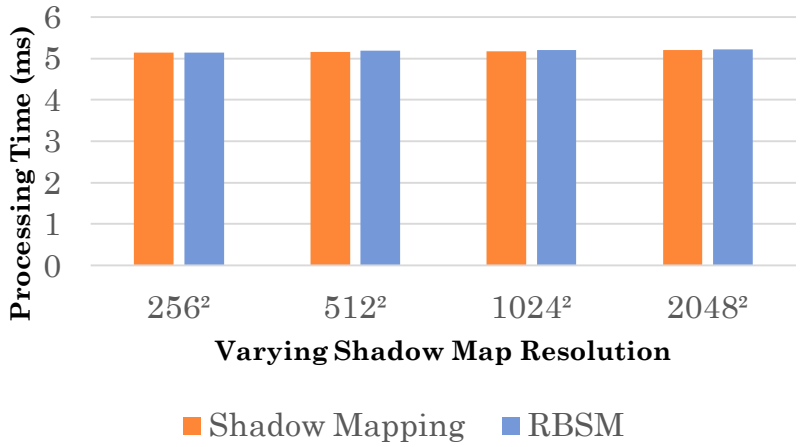




Results

- Rendering Time:





Conclusion and Future Work

Final Considerations

- Conclusion:
  - We have shown that, by the use of the proposed implementation of RBSM, we are able to minimize the aliasing artifacts, generating high-quality hard shadows at little additional cost;
  - Due to the limited source code access provided by Unity, we were not able to implement RBSM for other types of light sources;
  - We could not implement RBSM for soft shadows that simulate the penumbra effect;
- Future Work:
  - The implementation of RBSM using the full Unity source code;
  - The implementation of RBSM in other game engines;

References

- [Williams1978] – L. Williams. “Casting Curved Shadows on Curved Surfaces”. In *Proceedings of the ACM SIGGRAPH*, pages 270-274, 1978;
- [Macedo2016] – M. Macedo and A. Apolinário. “Revectorization-based Shadow Mapping”. In *Proceedings of the Graphics Interface*, pages 75-83, 2016.