



Silesian  
University  
of Technology

## MASTER THESIS

Selected techniques of rendering shadow models in interactive computer  
graphics

**Radosław Jacek RZECZKOWSKI**  
**Student identification number: 290776**

**Programme:** Informatyka  
**Specialisation:** Interaktywna Grafika Trójwymiarowa

### SUPERVISOR

**Dr hab. inż. Agnieszka Szczęsna**

**DEPARTMENT Wydział Automatyki, Elektroniki i Informatyki**  
**Faculty of Automatic Control, Electronics and Computer Science**

Gliwice 2024



**Thesis title**

Selected techniques of rendering shadow models in interactive computer graphics

**Abstract**

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

**Key words**

(2-5 keywords, separated by commas)

**Tytuł pracy**

Wybrane techniki renderowania modelu cieni w interaktywnej grafice komputerowej

**Streszczenie**

(Streszczenie pracy – odpowiednie pole w systemie APD powinno zawierać kopię tego streszczenia.)

**Słowa kluczowe**

(2-5 słów (fraz) kluczowych, oddzielonych przecinkami)



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Problem analysis</b>	<b>3</b>
2.1	The task of rendering . . . . .	3
2.2	The role of shadows in computer generated images . . . . .	5
2.3	Shadow rendering techniques . . . . .	6
2.3.1	Naming conventions . . . . .	6
2.3.2	Coarse distinction of shadow rendering techniques . . . . .	8
2.3.3	Planar shadows . . . . .	9
2.3.4	Technique 1 . . . . .	10
2.3.5	Technique 2 . . . . .	10
2.3.6	Technique 3 . . . . .	10
2.3.7	Technique 4 . . . . .	10
<b>3</b>	<b>Subject of the thesis</b>	<b>13</b>
3.1	Test application . . . . .	13
3.2	Implemented shadow rendering techniques . . . . .	13
3.2.1	Planar shadows implementation . . . . .	13
3.3	[Section title] . . . . .	15
3.4	[Subsection title] . . . . .	15
<b>4</b>	<b>Experiments</b>	<b>19</b>
4.1	Methodology . . . . .	19
4.2	Data sets . . . . .	19
4.3	Results . . . . .	19
4.4	testing of multiple figures . . . . .	19
<b>5</b>	<b>Summary</b>	<b>21</b>
<b>Bibliography</b>		<b>23</b>
<b>Technical documentation</b>		<b>27</b>

<b>List of abbreviations and symbols</b>	<b>29</b>
<b>List of additional files in electronic submission (if applicable)</b>	<b>31</b>
<b>List of figures</b>	<b>33</b>
<b>List of tables</b>	<b>35</b>

# **Chapter 1**

## **Introduction**

- introduction into the problem domain
- settling of the problem in the domain
- objective of the thesis
- scope of the thesis
- short description of chapters
- clear description of contribution of the thesis's author



# Chapter 2

## Problem analysis

### 2.1 The task of rendering

In this thesis, rendering is understood as the process of obtaining an image from a description of a three-dimensional scene. A scene can be rendered in a multitude of ways, with differences both in the specifics of the initial scene description and the rendered result. The rendered visuals can range from stylized to photorealistic. Rendering happens everywhere where a computer-generated imagery (CGI) is created for a viewer to see.

The wide spectrum of possible rendering results hints at the multitude of ways in which rendering itself is performed. There are various techniques employed during the rendering process, which can be utilized together to create a desired look and fit within performance constraints. In more complex processes, there are many techniques used to render a scene, each responsible for modeling the visual aspects of different real-life phenomena or artificial effects. A renderer could be capable of adding stylized edge detection and cell-shading to an image, rendering glossy and rough surfaces, simulating shadows, reflections, caustics and refraction, dealing with hair and fur or volumetric participating media such as fog, smoke and clouds. Each of the mentioned effects can be rendered using one of many techniques, and many of them are being actively improved upon and researched.

The fact that there are many techniques currently in use to render a single type of effect creates an advantageous situation, where the techniques used can be chosen for each application depending on its characteristic. Two main factors can be discerned as defining the needs of an application: the performance requirements and the desired visual style.

The desired visual style is partially just a matter of preference defined by the style of the project. More importantly however it is a matter of clearly and efficiently conveying visual information, in a way that is consistent with the rest of the application and with what the end user expects. This means that realism, often touted the pinnacle and goal of

computer graphics, is not necessarily always the best approach. A CAD (computer-aided design) application or a 3D modelling tool would be made less useful by including realistic reflections, highly contrasting full shadows and motion blur in the rendered viewport. These programs need to clearly convey information about the shape and design of 3D objects, without distractions and obstructions. On the other hand, when a player starts a modern action-adventure game, they expect a level of realism in the game's graphics that allows for immersion in the presented world. The intricate visuals also make for a more engaging experience and can mean a better reception of a game. At the same time, design choices should be made to ensure that the realism or intricacy of the presented graphics does not get in the way of enjoying the gameplay, which in the end should be the main attribute of a video game.

The performance requirements, or performance constraints, are usually better defined and divide rendering into two general categories: real-time rendering and offline rendering. When designing an offline renderer the performance constraints would most likely concern memory usage and possibly general time, or energy, efficiency, as with offline rendering the time it takes to render an image is of small importance. Most important are image quality and fidelity, often also realism. Techniques used in this context can spend as much time performing calculations as is necessary for the desired output. Offline rendering work can also easily be distributed between many machines, so-called render farms. Real-time rendering on the other hand works with very strict and small time budgets. Because a real-time application needs to be responsive to user inputs and give the illusion of continuous motion it is expected to render at least 30 frames per second (FPS), giving the time budget for a single frame of at most 0.03 seconds. This time cannot all be spent on rendering, as user inputs need to be handled and application logic performed in the same time frame. Because of that, real-time applications need to balance visual complexity and performance. They are also often created for more casual consumers than offline rendering programs, so the hardware on which the application will run is expected to be moderately powerful.

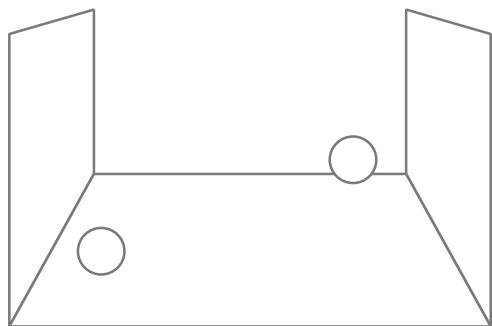
It is apparent that the choice of rendering techniques is complex and requires in-depth knowledge about each of them as early as the design stages of an application. This thesis aims to provide a guide through the rendering techniques used to render cast shadows in scenes, focusing on those applicable in real-time contexts. Some of the chosen techniques are the most widely adopted techniques, but some, while possibly less popular, are still noteworthy due to their promised outstanding performance or visual quality. This thesis introduces the techniques, gives detailed explanations of their algorithms, shows exemplary implementations and compares them in a series of tests. The comparison is made based on measures of performance (execution time), memory consumption, visual fidelity, realism and the amount of aliasing.

## 2.2 The role of shadows in computer generated images

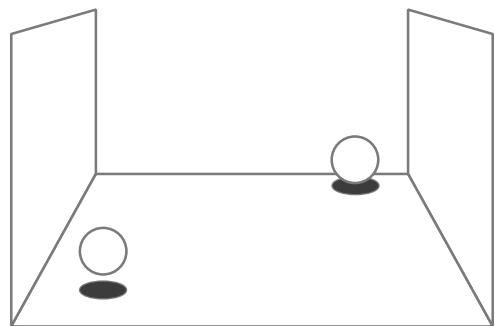
Computer graphics usually aim to reflect our surrounding reality, with techniques for rendering phenomena modeled from physical laws and after observations made in the real world. When not aiming for realism, the displayed graphics need to be at least rooted in the common human observation experience for the information to be understandable and interpretable. Shadows are a phenomenon that is very commonly present in our daily lives, deeply rooted in the common experience. An 1997 article [3] shows that even simple shadows provide the viewer essential information on the position and movement of objects, especially movement in depth. In some cases this information even overrides other cues such as change of size due to perspective. This effect is illustrated by Fig. 2.1.

The importance of shadows for understanding the 3D composition of a scene is especially great when an observer is devoid of stereo visual information, such as in the case of observing an image displayed on a regular monitor. Shadows, lighting and perspective then become the sole sources of information about depth and the relative positions of objects in the scene.

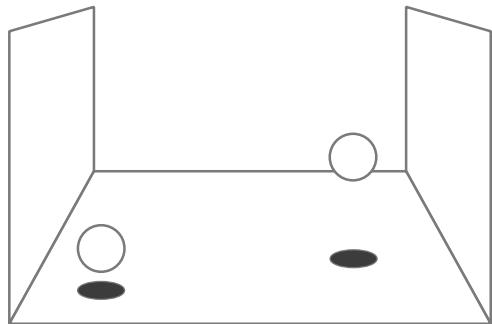
These observations are also supported by a more recent 2018 review on the perception



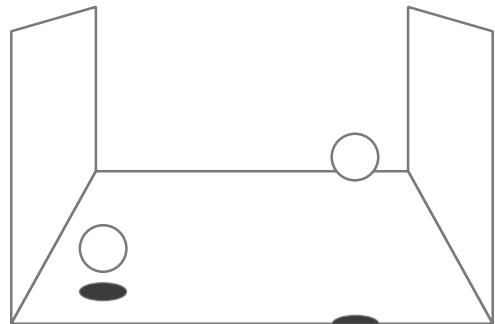
(a) Shadowless scene, the positions of the spheres are ambiguous.



(b) The upper ball seems placed on the surface.



(c) The upper ball seems to hover halfway.



(d) The upper ball seems high in the air and closer to the observer.

Figure 2.1: The position and scale of the spheres do not change, yet the perceived locations in 3D space are dictated by the shadow.

of shadows [4]. It suggests more possible roles that shadows play in human perception of the world. Shadows help observers position objects in 3D space. Sometimes shadows are perceived as being a part of the object, so the lack of a shadow would make the object incomplete. Information about the shape of an object can also be derived from the shadow that it casts, as presented in Fig. 2.2.



Figure 2.2: The shadow can convey information on the shape of the object. It additionally describes whether an object is placed on a surface or above it.

The review also mentions that shadows in art, even if depicted in a way that is impossible in the real world, are an important element for an image to be perceived as realistic and of high quality. This also applies to computer graphics, making rendering shadows a necessary element for any application that requires elevated realism or sophistication of the visual presentation. The difference between a scene with and without cast shadows is presented in Fig. 2.3. It is worth noting that since observers are not great at spotting physically incorrect shadows, as is stated in the studies, so graphics programmers can utilize this fact and take liberties with the realism of their shadows. This can lead to simplifying the algorithms producing them and improving their performance, with negligible impact on the apparent visual quality.

## 2.3 Shadow rendering techniques

### 2.3.1 Naming conventions

In this thesis the term ‘shadows’ is used interchangeably with term ‘cast shadows’, not to be confused with shading on the surface of objects, which stems from the orientation of the surface with regard to the light source.

As presented in the “Real-Time Rendering” book [1], a shadow is cast by a shadow caster, also called an occluder, onto a shadow receiver, when the caster occludes a line of sight from a point on the surface of the receiver to a light source.

Shadows can be differentiated into hard and soft shadows. Perfectly hard shadows are not observed in reality, as such a shadow would require an infinitely small light source. Hard shadows however are often found in computer-generated images, as they are relatively simple and cost-effective to render. Soft shadows require more complex solutions,



(a) The scene rendered with no cast shadows, despite the high quality of models and materials, looks unrealistic and flat.



(b) The scene rendered with cast shadows is realistic and reads well.

Figure 2.3: The same scene rendered twice using Blender's Cycles renderer, with and without cast shadows. *Scene source: "Classroom" by Christophe Seux, Blender demo files.*

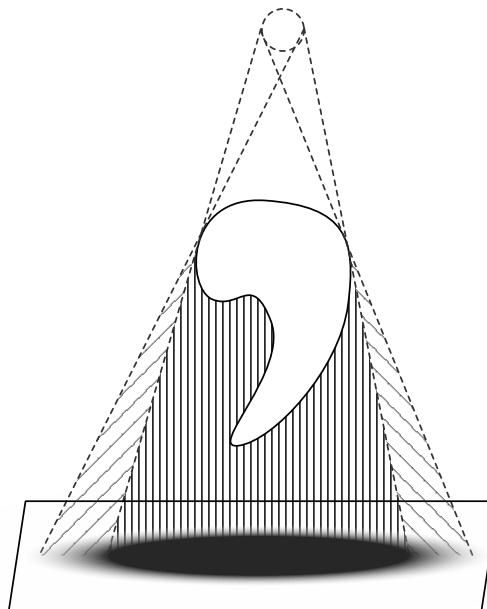


Figure 2.4: The umbra (under vertical lines) and penumbra (under slant lines) create a soft shadow.

but enhance the realism of the rendered scene. A soft shadow, as shown in Fig. 2.4 consists of an *umbra*, being the fully shadowed region, and *penumbra*, meaning the region that is partially in shadow. The appearance of a soft shadow, the sizes of the umbra and penumbra defining its softness, depend on the relative distances between the shadow caster, the shadow receiver, the light source and also the size of the light source.

### 2.3.2 Coarse distinction of shadow rendering techniques

In this thesis a coarse distinction of the shadow rendering techniques is proposed, dividing them into three groups. The division is decided based on the main mechanism utilized in a technique. Additionally, techniques in each group share similar issues that researchers have attempted and are still attempting to resolve with newer iterations of the original algorithms.

The first group consists of shadow map-based, or image-based, techniques. They render the scene from the viewpoint of the light and store how far away scene geometry is in a depth buffer. This is then used when rendering the actual view of the scene to determine which parts are in shadow and which are lit. A common issue for these techniques is aliasing, which occurs when signals are sampled at different rates, causing artifacts. In the case of computer graphics, spatial aliasing is most often observed, where the source signal, being geometry, a texture, or a depth buffer as in the case of shadow mapping, is sampled in discrete locations in space at varying sample densities. It is not trivial to counteract aliasing in shadow map techniques and there are many approaches to solving this problem. On the other hand, it is relatively simple to achieve approximate soft shadows, with more

advanced techniques allowing for more realistic results.

The second group contains geometry-based techniques. These approaches modify the scene geometry to create shadow volumes, which contain areas where light from a source does not fall. The shadow volumes are then rendered with a clever use of a stencil buffer, producing the information whether a surface point is in light or not. Such techniques suffer less aliasing problems than image-based ones, producing perfectly sharp shadow information on a per-pixel basis. They however introduce a lot of new geometry to the scene, the rendering time of which can be large and difficult to predict. It is also more difficult to achieve the effect of soft shadows with this approach.

The third group are ray-based techniques. They have only recently started being utilized in real-time scenarios, as they can be computationally costly and require powerful, modern hardware to run at interactive rates. This group of techniques is possibly the most intuitive, as it does not really rely on clever utilization of the rasterization hardware of a graphics card. To determine if a surface point is in shadow, a ray is cast in a straight line from the point to a light source. If any geometry is encountered along the ray, then the light source is occluded and the point is in shadow, otherwise it is lit. These techniques, apart from relying on powerful hardware, utilize specialized scene description data structures that accelerate these ray queries. They do not suffer from aliasing and can be extended to give photorealistic and physically-correct results.

The following sections introduce the techniques based on the publications which first described them. The sections attempt to mention them in logical order, highlighting the improvements and additions made to each group over the years. More detailed descriptions of the algorithms used to implement these methods will be given in chapter 3. Additionally, the next section presents a now largely unused approach, that does not really fit into any of these categories, but is useful to build a geometric intuition of how shadows form.

### 2.3.3 Planar shadows

Planar shadows are one of the first techniques used for implementing shadow rendering, so they are mentioned in this thesis for historic reasons. They can be a help with understanding the geometry involved with shadows and their shape. That is because planar shadows are rendered using basic geometric projection.

Introduced by James Blinn in 1988 [2], planar shadows are created by projecting the shadow caster onto a flat surface of the shadow receiver. The caster vertices are projected along a vector between the vertex and a light source onto a planar surface of the receiver. This is achieved with a projection matrix, which multiplied with the vertex positions of the caster gives a new mesh, flattened onto the planar surface. This mesh can then be rendered on top of the shadow receiver with a black material, giving the illusion of a shadow. The technique is described in more detail in section 3.2.1.

While very simple and intuitive, this approach has many shortcomings that nowadays render it obsolete. The application needs to keep track of shadow casters and receivers separately, to know which objects should be projected onto which surfaces. Issues occur when a light source is between the caster and receiver. In such a situation an incorrect anti-shadow is created, as the operation of projection through a point is still valid. The receivers have to be planes, and special care needs to be taken to only draw the shadow on top of a receiver. Since the shadow is a separate mesh, it could be rendered beyond the mesh of a receiver. Additionally, objects cannot shadow themselves. On the other hand planar shadows give a lot of artistic control over the appearance of shadows and even their shape, as they can be manipulated just as any other mesh.

### 2.3.4 Technique 1

This is the first technique.

### 2.3.5 Technique 2

This is the second technique.

### 2.3.6 Technique 3

This is the third technique.

### 2.3.7 Technique 4

- What problem do I want (have to :-) to solve?
- Why the problem is important?
- How do others solve the problem?
- What are pros and cons of my solution?

References to book [5], scientific papers in journals [6], conference papers [7], and web pages [8].

Equations should be numbered:

$$y = \frac{\partial x}{\partial t} \tag{2.1}$$

- problem analysis, problem statement
- state of the art, literature research (all sources in the thesis have to be referenced, eg journal article [6] book [5], conference paper [7], internet source [8])

- description of known solutions, algorithms
- location of the thesis in scientific domain
- The title of this chapter is similar to the title of the thesis.

**Definition 1.** *body of the definitions*

**Theorem 1** (optional name). *body of the theorem*

**Example 1** (optional name). *body of the example*



# Chapter 3

## Subject of the thesis

### 3.1 Test application

### 3.2 Implemented shadow rendering techniques

Below, all the shadow rendering techniques that have been implemented during work on this thesis are described, along with explanations of their algorithms.

#### 3.2.1 Planar shadows implementation

Planar shadows, as described in section 2.3.3, are a very simple technique to achieve hard shadows, introduced by Blinn [2]. The idea for the algorithm stems from the geometric understanding of how a shadow is formed. The shape of a shadow is basically the shadow caster projected onto the surface of a shadow receiver along rays from a light source, as shown in Fig. 3.1.

Using this geometric understanding and similar triangles, marked in Fig. 3.1 in dashed lines, the following equations and a projection matrix can be derived. Assuming  $l$  to be the position of a light source,  $v$  to be the position of a vertex of the shadow caster and  $p$  the position of the projected shadow vertex, a projection onto the  $y = 0$  plane is described by:

$$\frac{p_x - l_x}{v_x - l_x} = \frac{l_y}{l_y - v_y} \iff p_x = \frac{l_y v_x - l_x v_y}{l_y - v_y} \quad (3.1)$$

The  $z$  component is derived in analogous way:

$$\frac{p_z - l_z}{v_z - l_z} = \frac{l_y}{l_y - v_y} \iff p_z = \frac{l_y v_z - l_z v_y}{l_y - v_y} \quad (3.2)$$

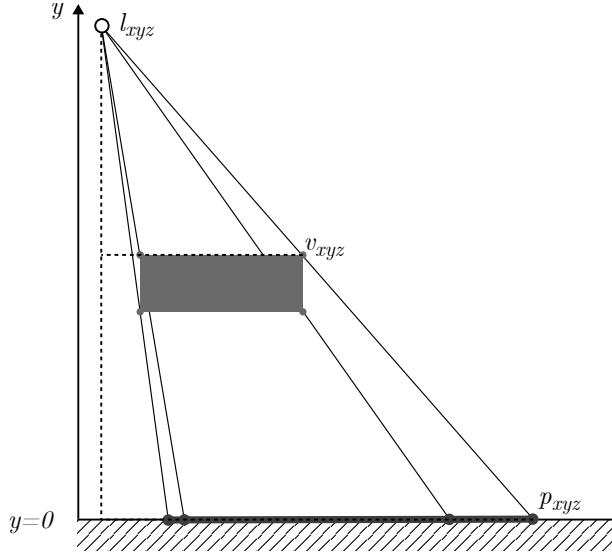


Figure 3.1: Vertices of a shadow caster projected onto  $y = 0$  plane create a planar shadow. Dashed lines visualize the similar triangles.

The  $y$  component will be  $y = 0$  since the projection happens onto the  $xz$  plane. From that the projection matrix can be constructed:

$$\mathbf{M} = \begin{pmatrix} l_y & -l_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -l_z & l_y & 0 \\ 0 & -1 & 0 & l_y \end{pmatrix} \quad (3.3)$$

The  $l_y - v_y$  factor in the denominator is obtained by utilizing the property of homogenous coordinates, where all components of a vector  $\mathbf{v} = [x \ y \ z \ w]^T$  get divided by the  $w$  component, which gives  $\mathbf{v} = [x/w \ y/w \ z/w \ 1]^T$ .

This matrix can be generalized to project points onto any plane, but this was not implemented.

In the implementation the matrix is created in the vertex shader based on the position of the light source. The vertices of the caster, after being transformed into world space, are multiplied by it, projecting them onto the  $y = 0$  plane. Then they are transformed with the view and perspective projection matrices as usual. The resulting mesh is rendered in black, with a pixel shader that just returns the black color. Special care should be taken when rendering the shadow mesh, since it perfectly overlaps the ground plane at  $y = 0$ . To avoid z-fighting this mesh needs to be rendered after the receiver and before the caster, with z depth testing turned off in the rendering pipeline, or, as is the case in the implementation, a negative depth offset needs to be added to the mesh to bring it above the ground plane.

Multiple issues arise with this technique. The receiver must be a flat plane, objects

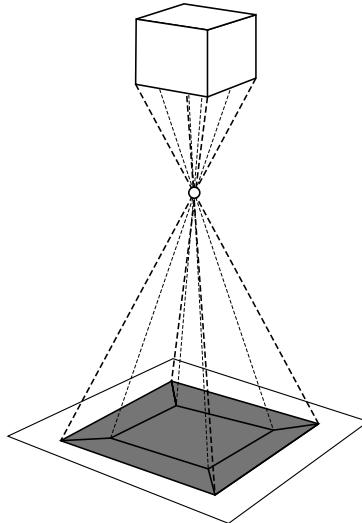


Figure 3.2: The shadow caster projected through a light source, creating an anti-shadow.

cannot shadow themselves, casters and receivers need separate treatment. Additionally, care needs to be taken to only render the shadow meshes on top of the geometry of a receiver. Since the shadow is a separate mesh, it could be cast beyond the actual surface of the receiver, creating an impossible shadow in the air. There is another possibility here for an impossible shadow. If the light source is between the caster and receiver, an anti-shadow will be cast as shown in Fig 3.2. This creates another special case that needs to be checked for by the application. Soft shadows are also difficult and costly to achieve, created by projecting the caster geometry multiple times using different discrete locations sampled over the area of the light source. The shadow meshes can then be blended, producing a darker color where more shadows are present. This gives very accurate results with a high number of samples, but is not a viable option for real-time rendering. All these problems make this technique virtually unused in modern rendering engines.

- solution to the problem proposed by the author of the thesis
- theoretical analysis of proposed solutions
- rationale of applied methods, algorithms, and tools

Here I should describe my reasoning for the testing, reasoning for selecting the techniques that I do select, my program for testing.

### 3.3 [Section title]

#### 3.4 [Subsection title]

Each figure in the document should be referred to at least once (fig. 3.3).

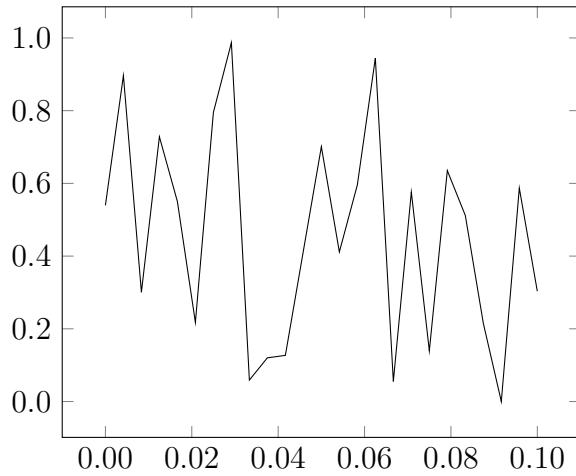


Figure 3.3: Figure caption.

Table 3.1: A caption of a table is ABOVE it.

$\zeta$	method							
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$		
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$	
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365	
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630	
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045	
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614	
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217	
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640	
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209	
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059	
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768	
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362	
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724	



Figure 3.4: Caption of a figure is always below the figure akakak.

Each table in the document should be referred to at least once (Tab. 3.1).  
some citation of the above 3.4



# **Chapter 4**

## **Experiments**

This chapter presents the experiments. It is a crucial part of the thesis and has to dominate in the thesis. The experiments and their analysis should be done in the way commonly accepted in the scientific community (eg. benchmark datasets, cross validation of elaborated results, reproducibility and replicability of tests etc).

### **4.1 Methodology**

- description of methodology of experiments
- description of experimental framework (description of user interface of research applications – move to an appendix)

### **4.2 Data sets**

- description of data sets

### **4.3 Results**

- presentation of results, analysis and wide discussion of elaborated results, conclusions

### **4.4 testing of multiple figures**

some citation of the above 4.1

some citation of the above 4.2



Figure 4.1: Caption of a figure is always below the figure akakak.



Figure 4.2: Caption of a figure is always below the figure akakak.

# **Chapter 5**

## **Summary**

- What problem have I solved?
- How have I solved the problem?
- What are pros and cons of my solutions?
- Can I state some recommendations?
- synthetic description of performed work
- conclusions
- future development, potential future research
- Has the objective been reached?



# Bibliography

- [1] Thomas Akenine-Moller, Eric Haines and N. Hoffman. *Real-Time Rendering, Fourth Edition*. Boca Raton: Taylor & Francis Ltd, 2018. ISBN: 978-1-1386-2700-0.
- [2] J. Blinn. ‘Me and My (Fake) Shadow’. In: *IEEE Comput. Graph. Appl.* 8.1 (1988), 82–86.
- [3] Daniel Kersten, Pascal Mamassian and D. Knill. ‘Moving Cast Shadows Induce Apparent Motion in Depth’. In: *Perception* 26.2 (1997). PMID: 9274752, pp. 171–192. DOI: 10.1068/p260171. eprint: <https://doi.org/10.1068/p260171>. URL: <https://doi.org/10.1068/p260171>.
- [4] Paulo E. Santos, Roberto Casati and P. Cavanagh. ‘Perception, cognition and reasoning about shadows’. In: *Spatial Cognition & Computation* 18.2 (2018), pp. 78–85. DOI: 10.1080/13875868.2017.1377204. eprint: <https://doi.org/10.1080/13875868.2017.1377204>. URL: <https://doi.org/10.1080/13875868.2017.1377204>.
- [5] Name Surname and Name Surname. *Title of a book*. Hong Kong: Publisher, 2017. ISBN: 83-204-3229-9-434.
- [6] Name Surname and Name Surname. ‘Title of an article in a journal’. In: *Journal Title* 157.8 (2016), pp. 1092–1113.
- [7] Name Surname, Name Surname and N. Surname. ‘Title of a conference article’. In: *Conference title*. 2006, pp. 5346–5349.
- [8] Name Surname, Name Surname and N. Surname. *Title of a web page*. 2021. URL: <http://somewhere/on/the/internet.html> (visited on 30/09/2021).



# Appendices



# Technical documentation



# List of abbreviations and symbols

CGI computer-generated imagery

CAD computer-aided design

FPS frames per second

DNA deoxyribonucleic acid

MVC model–view–controller

$N$  cardinality of data set

$\mu$  membership function of a fuzzy set

$\mathbb{E}$  set of edges of a graph

$\mathcal{L}$  Laplace transformation



# **List of additional files in electronic submission (if applicable)**

Additional files uploaded to the system include:

- source code of the application,
- test data,
- a video file showing how software or hardware developed for thesis is used,
- etc.

---

Appendix . List of additional files in electronic submission (if applicable)

# List of Figures

2.1	The position and scale of the spheres do not change, yet the perceived locations in 3D space are dictated by the shadow. . . . .	5
2.2	The shadow can convey information on the shape of the object. It additionally describes whether an object is placed on a surface or above it. . . . .	6
2.3	The same scene rendered twice using Blender’s Cycles renderer, with and without cast shadows. <i>Scene source: “Classroom” by Christophe Seux, Blender demo files.</i> . . . . .	7
2.4	The umbra (under vertical lines) and penumbra (under slant lines) create a soft shadow. . . . .	8
3.1	Vertices of a shadow caster projected onto $y = 0$ plane create a planar shadow. Dashed lines visualize the similar triangles. . . . .	14
3.2	The shadow caster projected through a light source, creating an anti-shadow. . . . .	15
3.3	Figure caption. . . . .	16
3.4	Caption of a figure is always below the figure akakak. . . . .	17
4.1	Caption of a figure is always below the figure akakak. . . . .	20
4.2	Caption of a figure is always below the figure akakak. . . . .	20



# List of Tables

3.1 A caption of a table is ABOVE it. . . . .	16
---	----