Interactive Adaptive Cubistic Images

Intelligent Systems Laboratory — Winter Term 2013/2014

Tristan Walter & Nicolai Ernst & Akshat Tandon

Supervisor: Eduard Zell

Bielefeld University, Faculty of Technology

Universität Bielefeld

Abstract

Our project is about rendering 3D scenes into images in the style of cubistic paintings. To accomplish this, we have implemented several features of cubistic art that are applied to the 3D scene. For the future it is planned to make the images interactive and adaptive to their surroundings.

Introduction

In the early 20th century the art movement called Cubism evolved in Europe. Cubistic paintings were abstract, the artists driven by order and rationality. The absence of depth illusion and the decomposition of objects into simple shapes were typical characteristics of their works.

This project is aimed at combining that style of art with computer graphics by rendering 3D scenes into images that resemble cubistic paintings. In addition, those images are intended to be interactive and able to adapt to their surroundings.

Main Objectives

1. Render 3D scenes into images in the style of cubistic art
2. Make the images adapt to their surroundings
3. Allow interaction with the images

Materials, Methods, Architecture

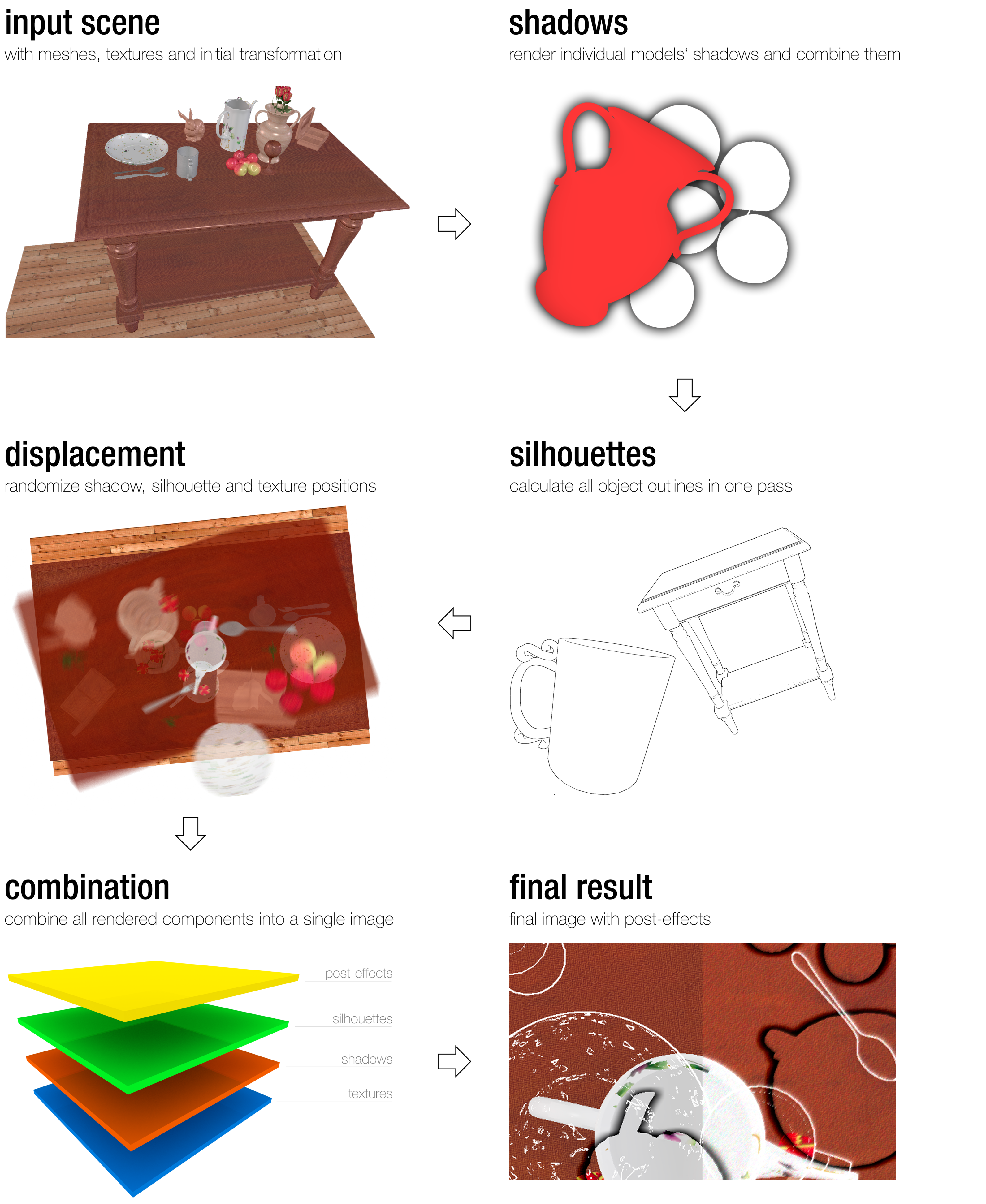
We chose a painting by Juan Gris as a reference and selected features from it for our implementation.

The Effects

- shadows - objects cast shadows on objects benath them
- silhouettes - objects are drawn as outlines
- displacement - object textures are not placed on their actual object
- lighting - diffuse and specular lighting are reduced to make objects appear more flat
- post-effects - some smearing and grain are added to make the image look more like a painting

The Rendering Pipeline

- Every object in the scene will either be rendered as a texture with a shadow or as a silhouette. It is randomly selected which option will be used.
- The positions and rotations of the shadows, silhouettes and textures are randomized. Bigger objects are moved less than smaller ones.



input scene

with meshes, textures and initial transformation

shadows

render individual models' shadows and combine them

displacement

randomize shadow, silhouette and texture positions

silhouettes

calculate all object outlines in one pass

combination

combine all rendered components into a single image

final result

final image with post-effects

Figure 1: The rendering pipeline of the program. The final image is composed of several parts in the order shown above.

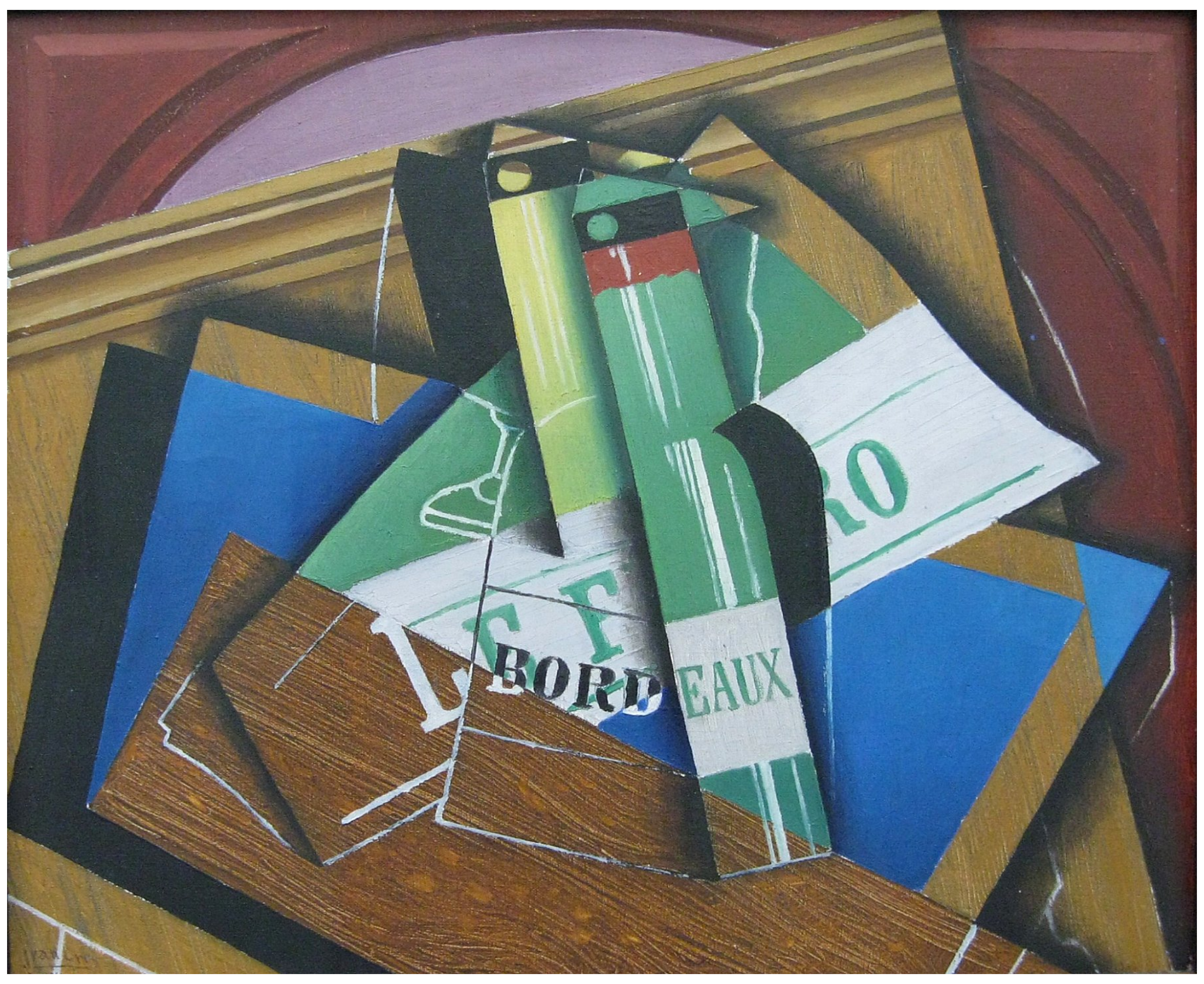


Figure 2: Juan Gris, The Bordeaux bottle (1915)

Results

Figure 3 shows what is the final result of the rendering pipeline shown in figure 1. As mentioned in the methods-section, several key-features of cubistic paintings have been identified and planned to be implemented. Each aspect can be seen in the final result below.

Being one of the main objectives for this project, the interaction- and adaptation-parts are still missing. These tasks have been scheduled for the second term.

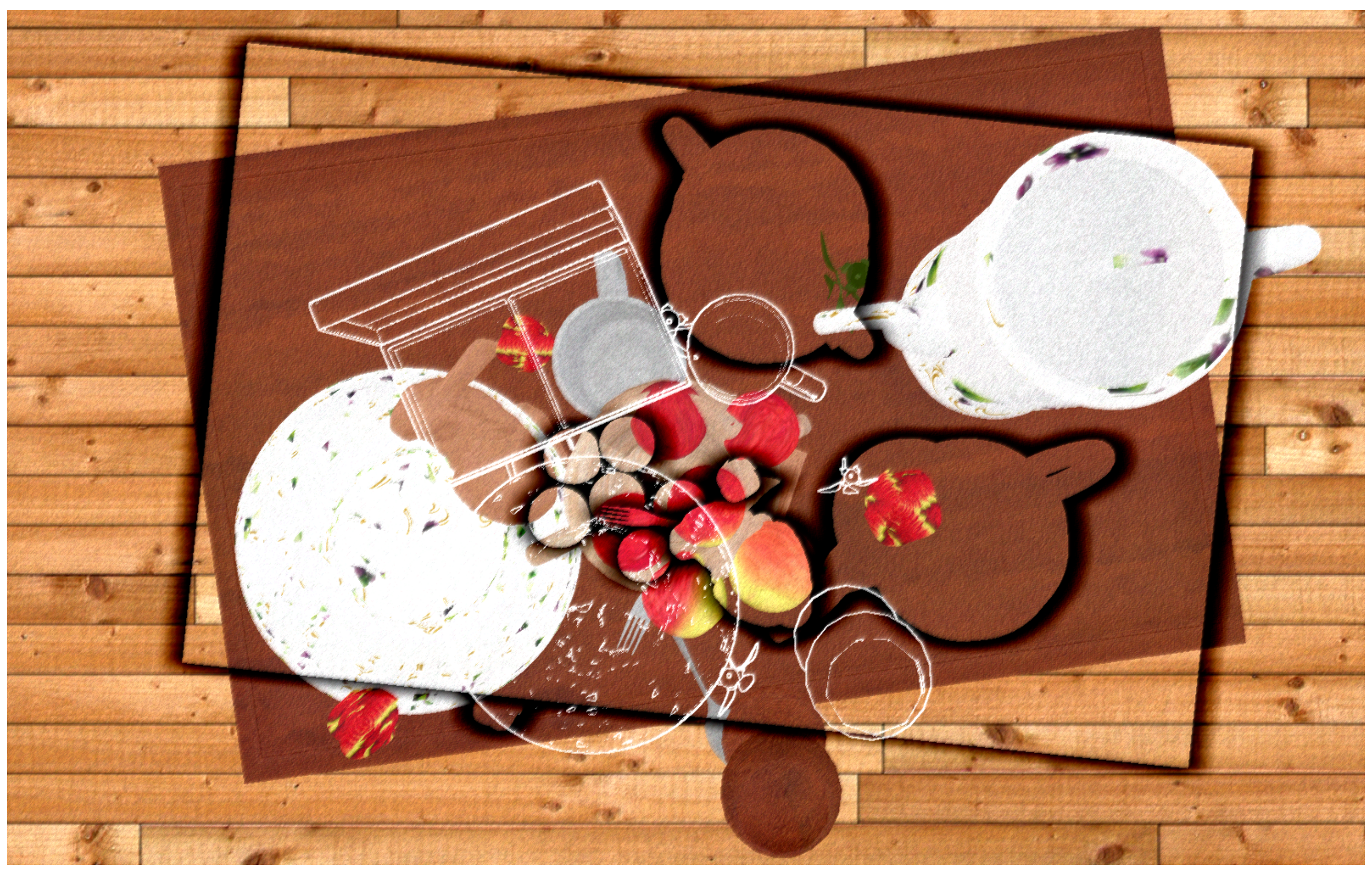


Figure 3: The resulting image after combining all features and applying post-effects.

Conclusions

Being generated only by modifying and thus also reinterpreting the underlying scene, the result shows close resemblance with the style of the painting by Juan Gris. However, certain processes have been identified as being unexpectedly complicated. One of those was the arrangement of objects in the final scene. As can be seen from the pipeline illustration, the meshes inside the scene must be rearranged in order to render shadows and silhouettes. This means for the objects not only to be translated, but also rotated and possibly being scaled. According to the initial plan, those objects should have been distributed randomly which has been found to be a very poor solution to the problem. Parameters had to be introduced for controlling the movement of unequally sized meshes to prevent bigger objects from completely occluding smaller objects when being transformed.

Even with overcoming this problem, there were still other issues (for example performance issues) to be dealt with. Some of them are still present in the current version of the program and need to be fixed in future versions. A good example are object shadows. Computing these shadows is currently very expensive and must be optimized during the second term. Only then will the program be able to compute in real-time (even on average-performance machines) and allow the user to interact with the scene.

References

[1] José Pierre. *Der Kubismus*. Ed. Rencontre, 1967.

[2] K. Ruhrberg, M. Schneckenburger, C. Fricke, and K. Honnef. *Kunst des 20. Jahrhunderts*, volume I. Taschen GmbH Köln, 2005.