In this thesis, the generation and use of soft shadows in real-time rendering is discussed:
- Hard shadow algorithms are already widely used in games & applications.
- The Fast & correct calculation of soft shadows is a complex task & area of active research.
- Soft shadows significantly increase the realism of the generated images.

We present a new algorithm, which is capable of rendering physically accurate soft shadows in real-time by exploiting the temporal coherence between neighboring frames.

The calculation of hundreds of shadow maps per frame is very costly and makes real-time frame rates nearly impossible.
- Only a single shadow map is evaluated per frame!
  - exploit temporal coherence between frames
  - store the shadowing information in a screen-space shadow buffer - a second render target texture with 4 channels, in which the shadow data can be saved.

Fragments are reprojected (to account for camera movement) from the new frame into the old.
- Their depths values are compared to the stored depths
- If the depth difference is smaller than a predefined threshold, the new and the old fragment are considered equal
- Fragment data from the previous frame is reused!

Upper left image: visualization of the shadow buffer:
- Red channel: depth
- Green Channel: number successful shadow tests
- Blue channel: number of all shadow tests
- alpha channel: penumbra size estimation

Our algorithm is based on the widely-used shadow-mapping algorithm, which generates hard shadows from a point light source:
- First, the scene is viewed from the position of the light source
- The depth values of the fragments are stored in a shadow map (SM)
- The shadow map has to be updated whenever a movement occurs

In the second pass, the scene is rendered from the camera position:
- Every fragment is transformed into light space
- Its distance to the light is compared to the corresponding value in the shadow map
- If the distance to the current fragment is larger than the shadow map value, it lies in shadow; otherwise it has to be illuminated.

Camera movement  disocclusions
- Detected in depth comparison step (red)
- No shadowing information from previous frames in the shadow buffer for them!
- We sample the neighboring fragments in the shadow buffer, use these values to generate an estimate, and blend it together with the correct solution:
  - Sampling kernel size depends on penumbra size
  - estimated in a blocker search in the SM
  - stored in e-channel  reuse in next frame(s)

For area light sources, things are more complicated:
- If the light source is partly visible from the position of the affected point, it is neither completely lit, nor completely shadowed, but in the penumbra:
  - Soft shadow = union of umbra & penumbra
  - The exact calculation of umbra and penumbra is complex. It implicates solving a three-dimensional visibility problem.
  - Umbra & penumbra extents depend on the relationship between light source size, distance from occluder to receiver and distance from light source to occluder.

The exact calculation of umbra and penumbra extents is very costly and makes real-time frame rates nearly impossible.

Calculation of hundreds of shadow maps per frame!