FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Improved Persistent Grid Mapping

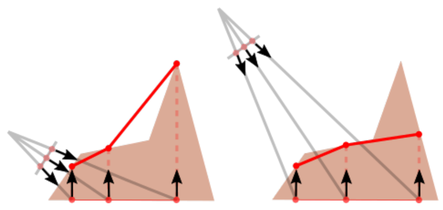
Peter Houska

Masterstudium:
Visual Computing

Technische Universität Wien
Institute of Visual Computing and Human-Centered Technology
Arbeitsbereich: Computergraphik
Betreuer: Assoc. Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer
Mitwirkung: Prof. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Daniel Scherzer

Motivation & Problem

Terrains are an essential part of applications involving interactive visualizations of outdoor scenes. Terrain surfaces typically cover the entire virtual environment, making it impossible to render the landscape as a single high-detail object at interactive frame rates. A widespread approach is to only keep the terrain data which surrounds the camera in video memory, and adapt triangle density based on camera distance to maintain the illusion of rendering the entire surface at full detail. Consequently, the terrain mesh needs to be re-adapted constantly as the camera roams freely. *Persistent Grid Mapping* (PGM) addresses these challenges by projecting the terrain mesh from the camera's point of view, through a second projection camera, onto the ground plane, thereby achieving continuous camera-distance-based adaption of the mesh's detail level, as well as automatic view-frustum culling. After projection, mesh vertices are displaced vertically by the sampled heightmap value.
While PGM is an elegant algorithm, it unfortunately suffers from **vertex swimming**, meaning that especially peaks and ridges flicker under camera movement, as shown in the figure below.



We describe four GPU-based techniques to increase the fidelity of the reconstructed terrain.

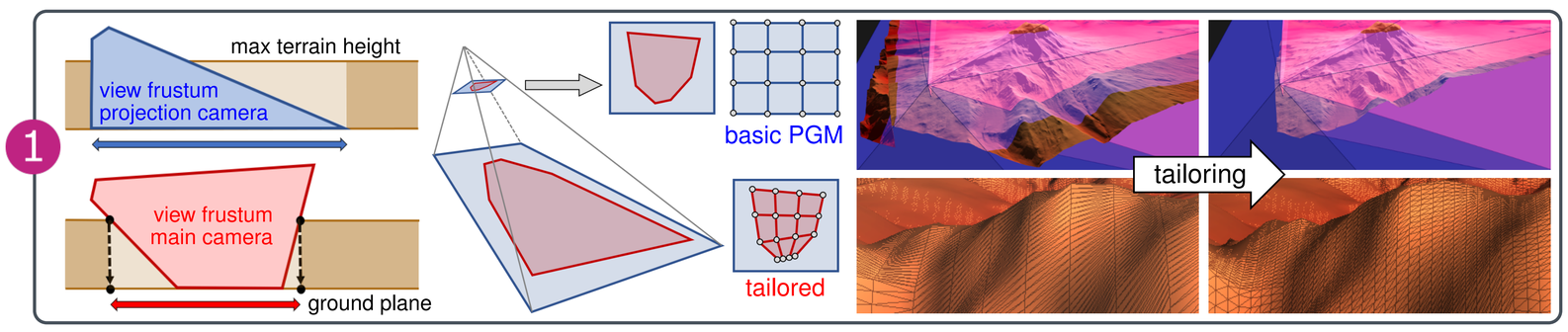
Our Method

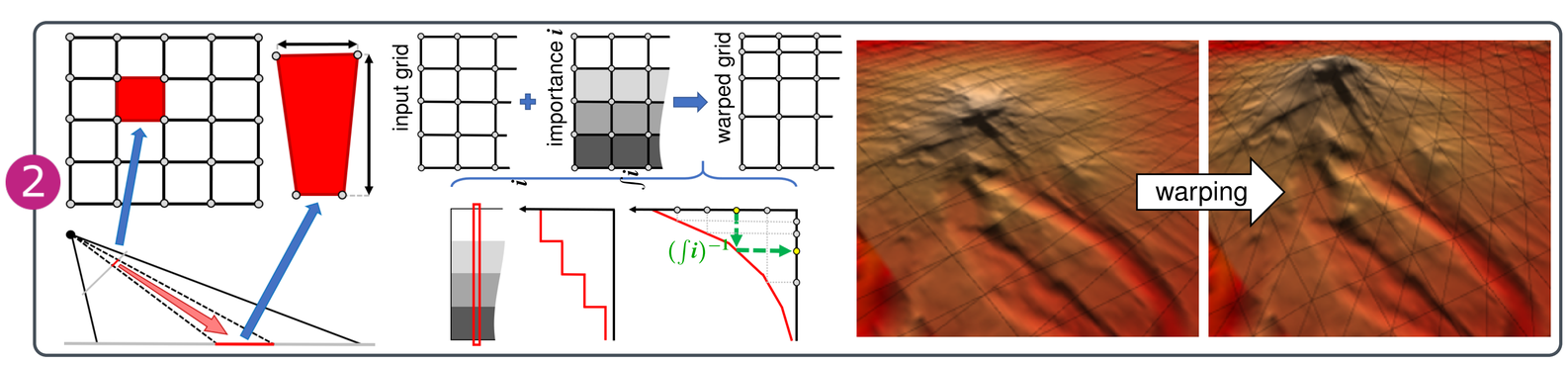
1 While PGM performs view-frustum culling by definition, many vertices still end up off screen, as the grid is projected to an unnecessarily large area on the ground plane. **Grid tailoring** shrinks the grid such that every grid vertex is potentially visible after displacement, which is achieved by intersecting the view frustum with the volume that the terrain potentially occupies. Setting the height of all intersection-volume vertices to the ground-plane height gives the minimal area on the ground plane that the grid needs to cover.

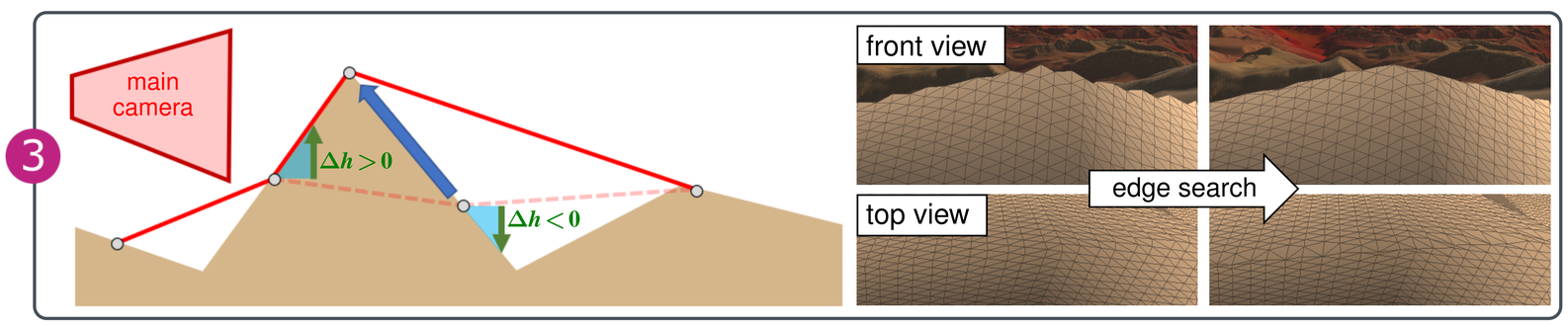
2 **Grid warping** redistributes grid vertices from the camera's vicinity toward the horizon, thereby reducing excessive grid stretching in the distance. The redistribution process is based on the aspect ratio of the projected grid quads.

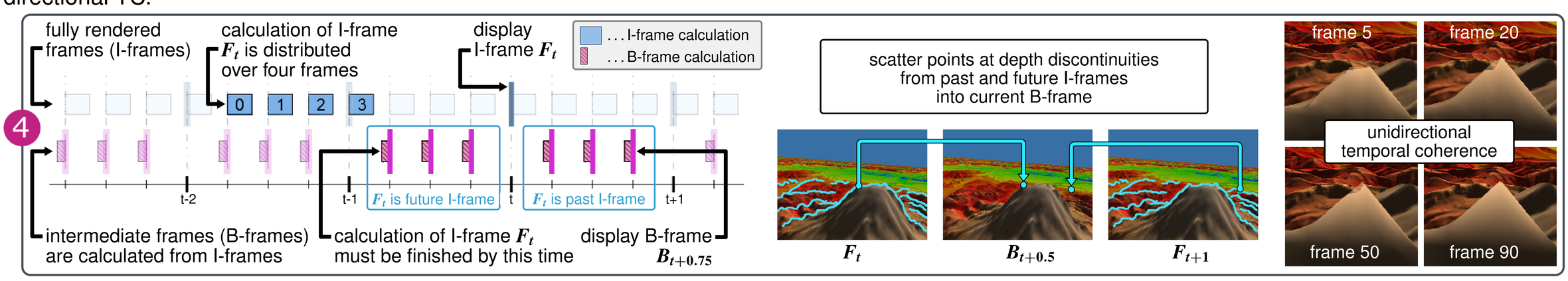
3 **Local edge search** moves grid vertices that missed a terrain peak toward the viewer onto the peak's ridge.

4 Image-based bidirectional *temporal coherence* (TC) is extended by **scattering points on terrain silhouettes** (as seen from the camera) from fully rendered I-frames to interpolated B-frames to reduce reconstruction artifacts. Further, whenever the camera stops translating, **jittered grid projections are accumulated** through unidirectional TC.









Results

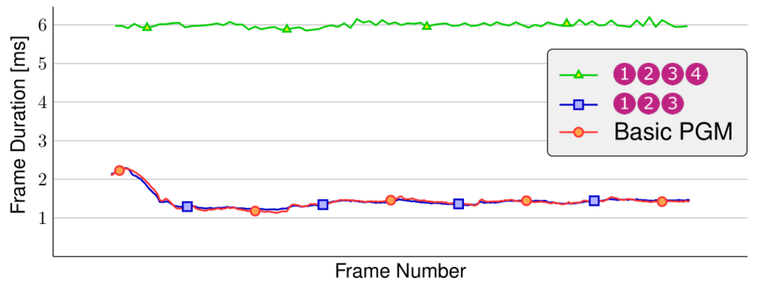
We compared PGM and our method to a reference solution in a virtual flight over Puget Sound, rendering 500K triangles per frame to a 1600 x 900 screen. The reference solution was generated by rendering a densely tessellated terrain mesh without any kind of resolution adaption. The quality of each method was quantified with the *peak-signal-to-noise-ratio* (PSNR) metric (larger values mean better quality).

Method	Min PSNR	Avg PSNR	Max PSNR
PGM	25.45 dB	27.69 dB	31.65 dB
1 2 3	29.22 dB	31.25 dB	34.75 dB
1 2 3 4	26.33 dB	36.41 dB	41.01 dB

The following images show zoomed-in sample screenshots from the virtual fly-through, along with a difference picture of the reference solution to our method (all four improvements are enabled).

Reference | 1 2 3 4 | 38.47 dB PSNR

Techniques 1 2 3 incur virtually no performance penalty, while technique 4 drops the framerate considerably, and also increases memory- and bandwidth requirements. We measured the following frame durations on a computer with a 3 GHz dual-core CPU, 3 GB RAM, and a GeForce GTS 450 with 1 GB GDDR5 VRAM (128-bit memory interface).



Memory requirements for 1080p and 2160p screen resolutions are given below.

Screen Resolution	1 2 3	1 2 3 4
1920 x 1080	126.5 KB	77.2 MB
3840 x 2160	126.5 KB	308.6 MB

Kontakt: peter.houska@gmail.com