CATEGORY: **COMPUTER VISION**

POSTER
C009

CONTACT NAME
Torkel Haufmann: torkel.haufmann@sintef.no

GPU TECHNOLOGY CONFERENCE

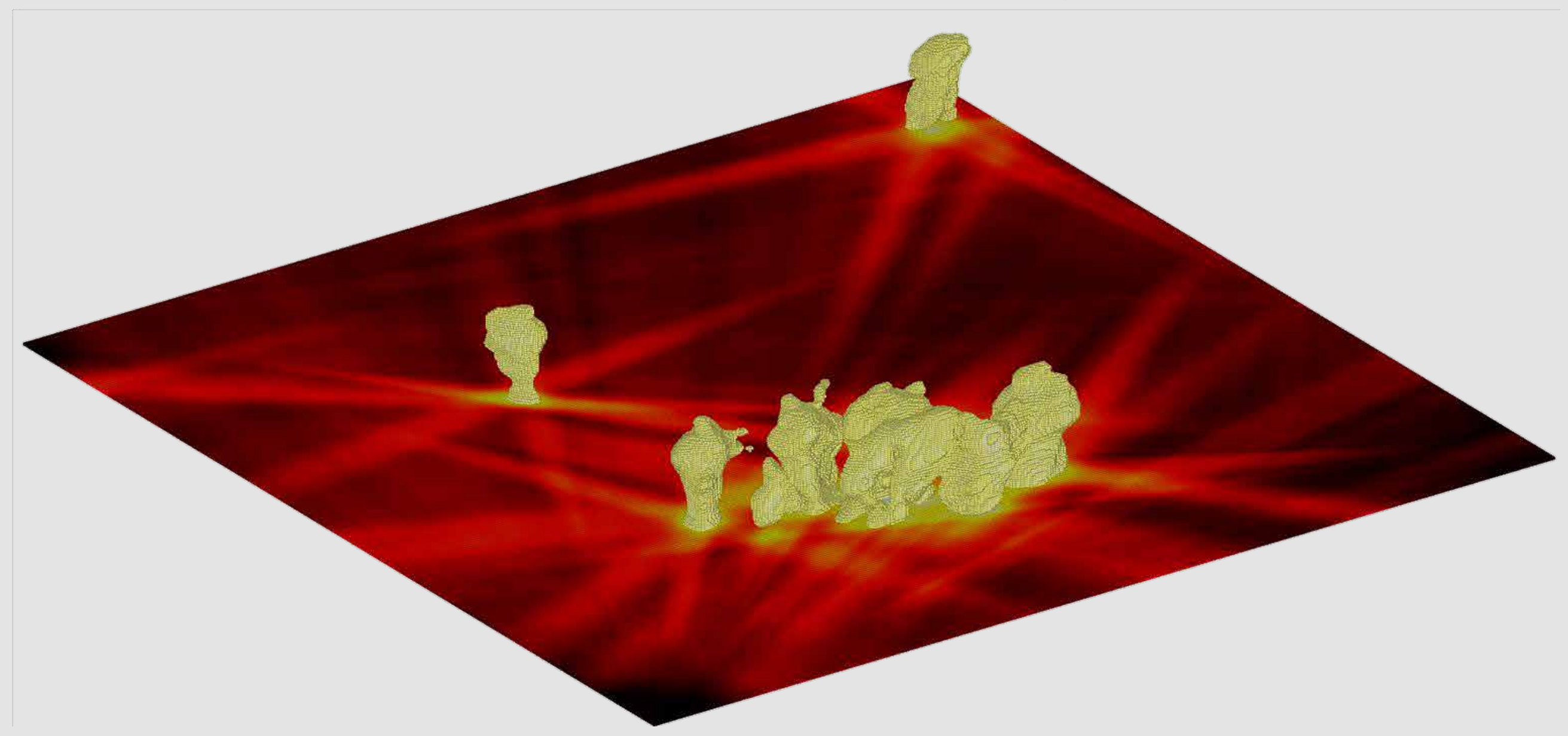
Real-time voxel carving with automatic synchronization

T. A. Haufmann*, A. R. Brodtkorb and A. Berge

SINTEF, Dept. Appl. Math., P.O. Box 124, Blindern, NO-0314 Oslo, Norway

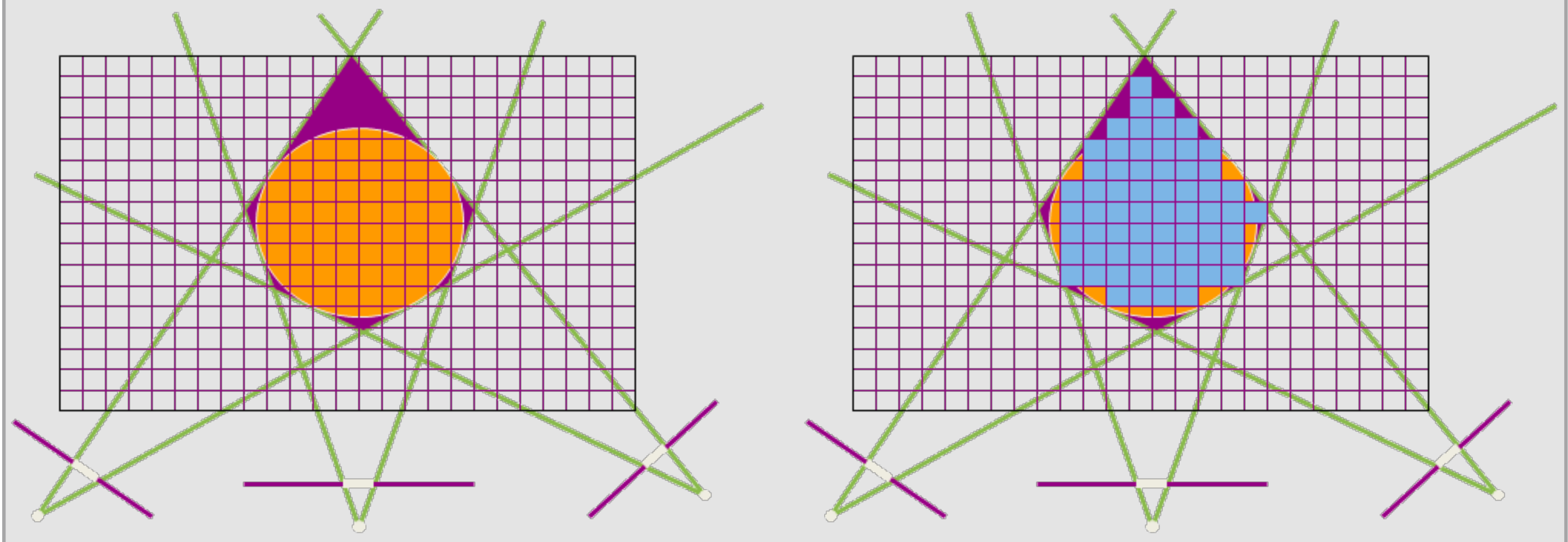
Abstract: We present (see [2]) a system for performing 3D reconstruction using a voxel carving algorithm. The system corrects automatically for frame drift using simultaneous event detections along epipolar lines as proposed by Pundik and Moses [1]. Using an Nvidia GTX 580 the system performs faster than real-time, and using several GPUs the approach can handle observation of several scenes at once.

Volume Carving



Assuming a scene is being observed by several cameras whose positions and characteristics are known, it is possible to reconstruct the 3D information about objects as determined by a segmentation algorithm.

In *voxel carving* a 3D voxel grid is imagined to be located in the centre of the observed scene. Each voxel then corresponds to an actual position in world space. For every frame we iterate through the voxel grid in the *z* direction, letting each GPU thread be responsible for one complete column of voxels. The segmentation in each of the cameras is then used to fill in the voxels, and after normalization the voxels whose values exceed a certain threshold are assumed to be occupied.

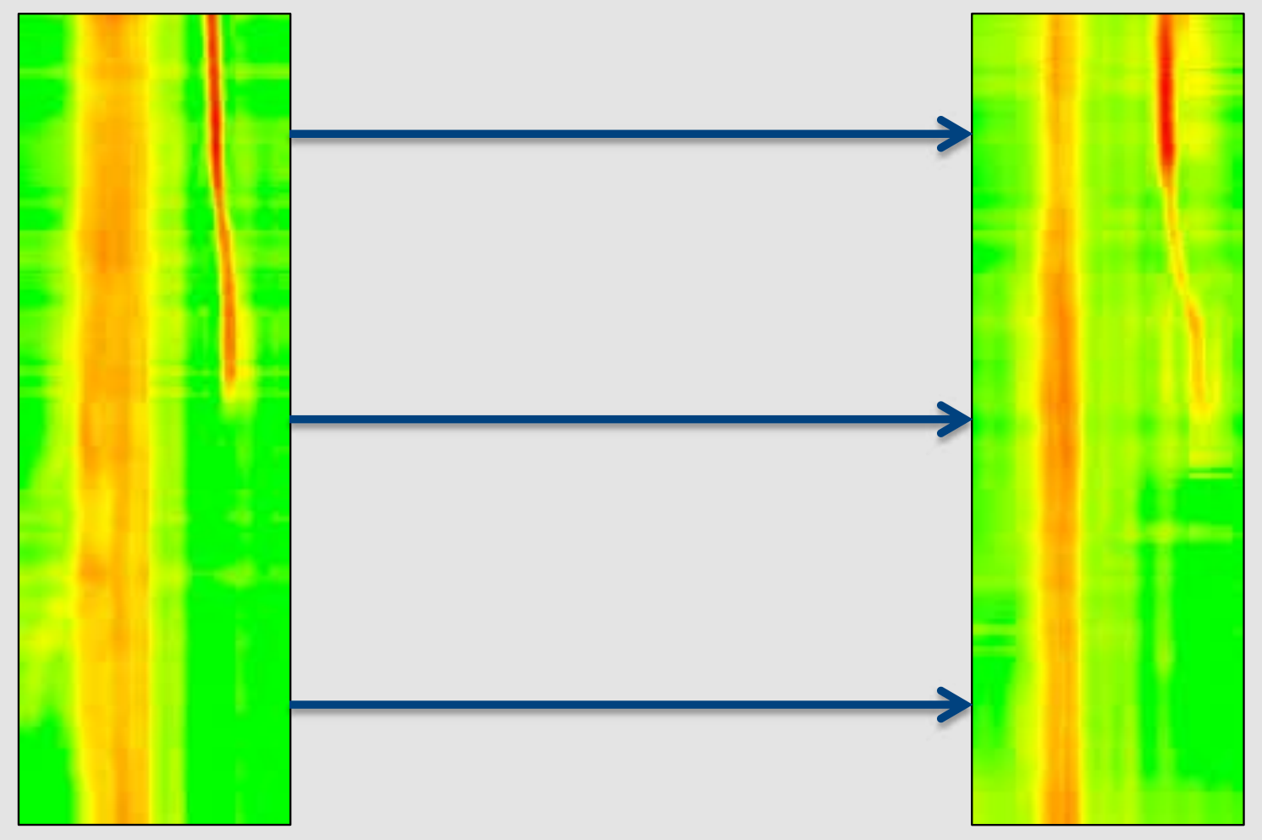


Left: Intersection of observations. Right: Discretized to voxel grid.

Our implementation of voxel carving is essentially projective texture mapping (used extensively in shadow mapping), making our algorithm output sensitive.

Synchronization

IP cameras cannot be relied upon for constant framerates – the h.264 encoding takes a variable amount of time depending on many hard-to-predict factors. As a consequence, a network of cameras can slowly go out of sync over time, which in the context of volume carving will cause the 3D information to lose a great deal of quality.



In [1] an approach to synchronizing pairs of cameras using epipolar geometry is given. In this paper, a *line signal* is computed for each line according to the formula

$$S_r(t) = \sum_{p \in l_r} \|I(t,p) - B(t,p)\|,$$

where l_r is the set of pixels on the line *r*, and *I* and *B* denote foreground intensity and background estimate respectively; in our system we are using a segmentation algorithm that also provides a background. The matrices *S* and *S'* are then defined to be the sets of line signals for all *r* over some time interval. The best possible matching is computed according to the formula

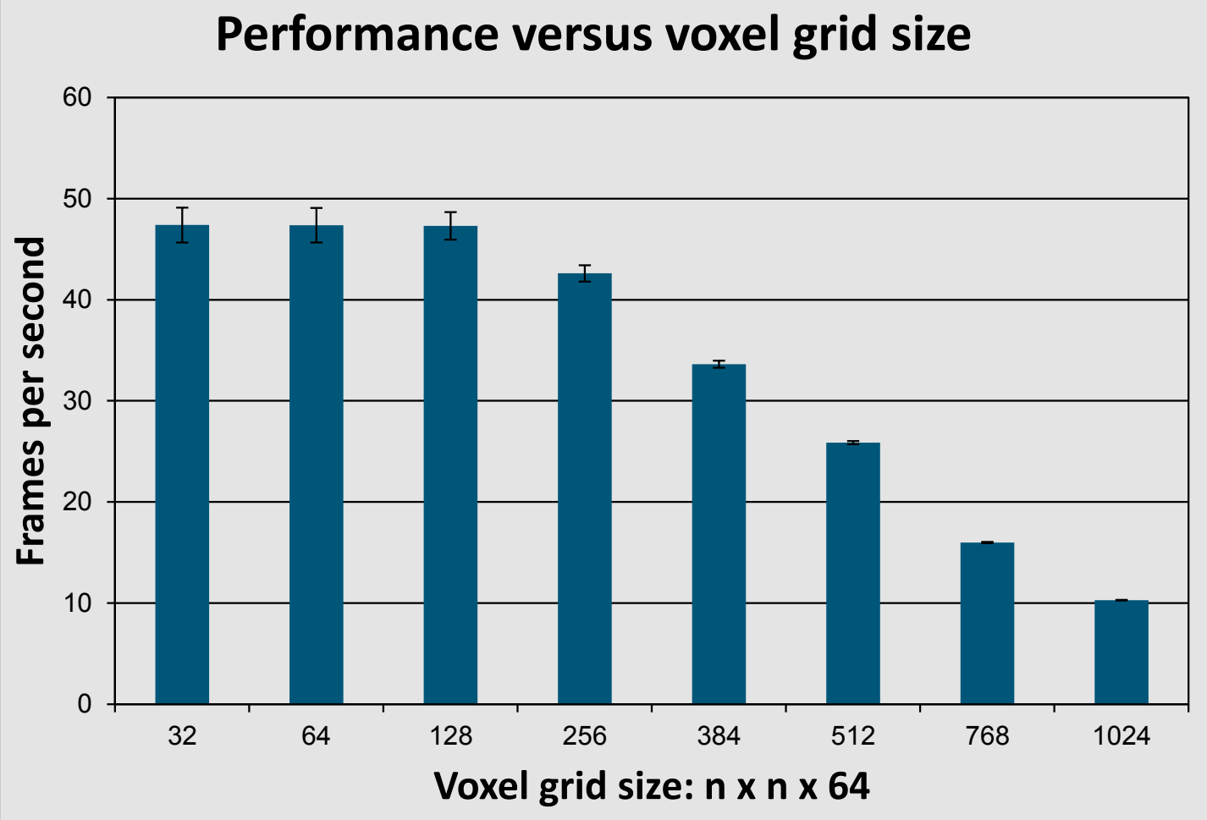
$$\arg \max_{-c \leq \Delta t \leq c} P(\Delta t) e^{-\sum_r \sum_{t=t_0-k}^{t_0} \frac{(S_{r,t} - S'_{r,t+\Delta t})^2}{2\sigma^2}},$$

where σ is a parameter adjusting the willingness of the algorithm to consider large shifts in the cameras and *P* is a prior.

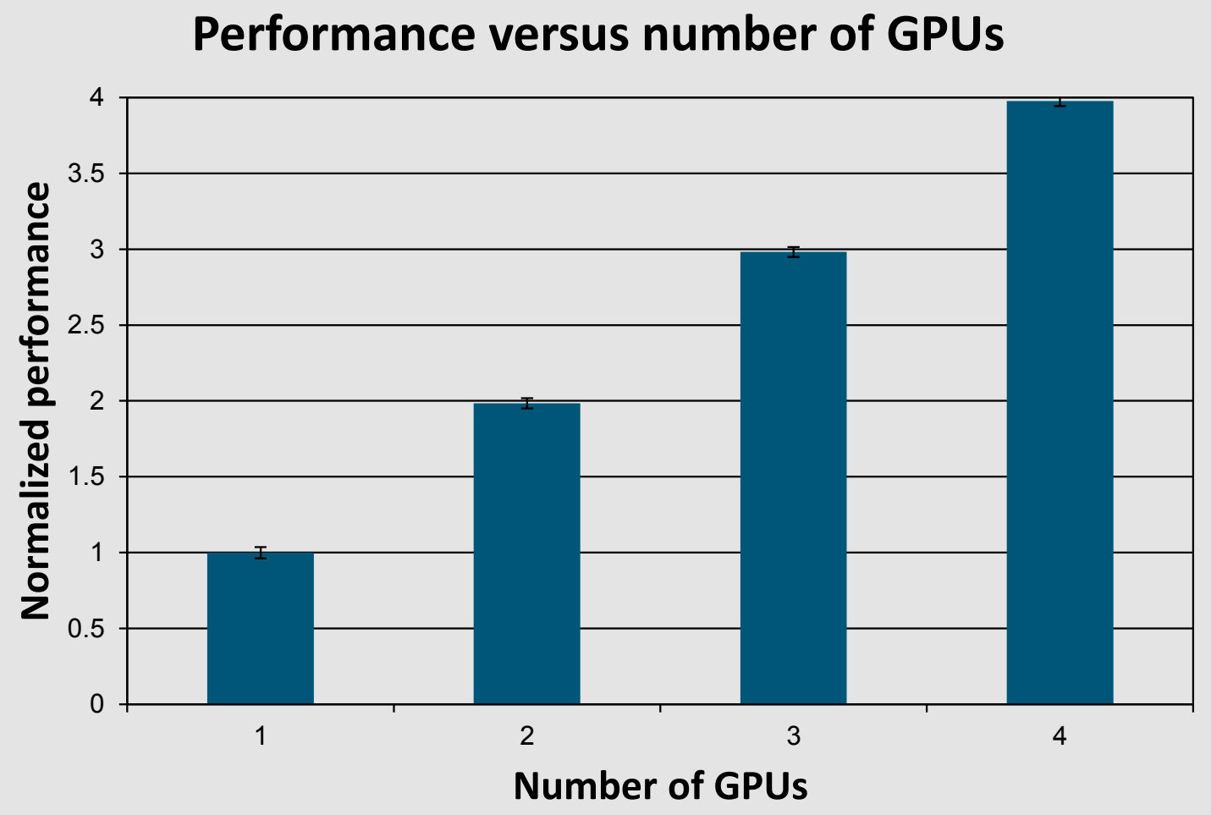
We perform the computations on the GPU, using a single-block kernel to iterate through all the lines in each camera and computing their signal every frame. Every few hundred frames the signals are downloaded to the host and shift computation is performed; then the cameras that are ahead of the system correct. In our system the synchronization computations consume a negligible amount of time.

Performance

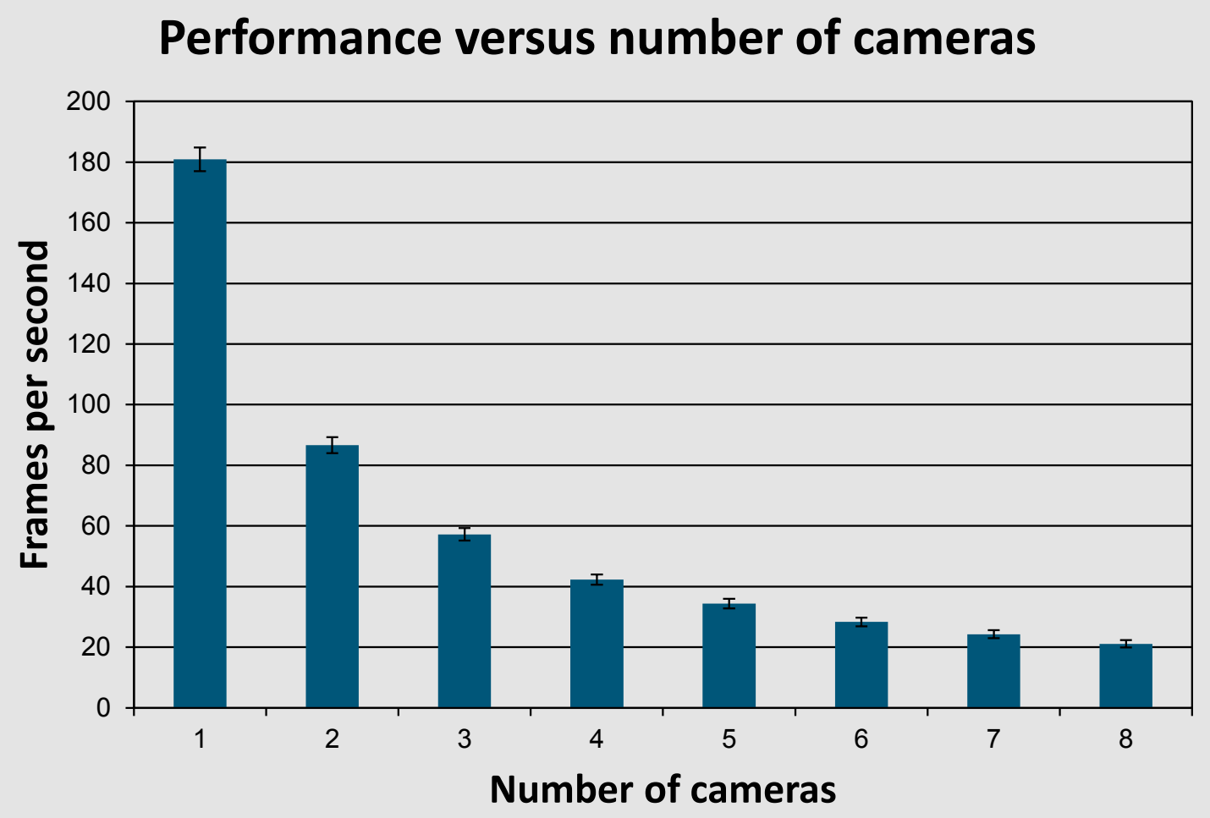
The system runs on a single GPU, performing the video decoding using the GTX 580's inbuilt video processor and then computing the per-frame segmentation. Next the volume carving is performed and the line signal matrices are updated, after which the process repeats for the next frame. The system can process several scenes independently using multiple GPUs.



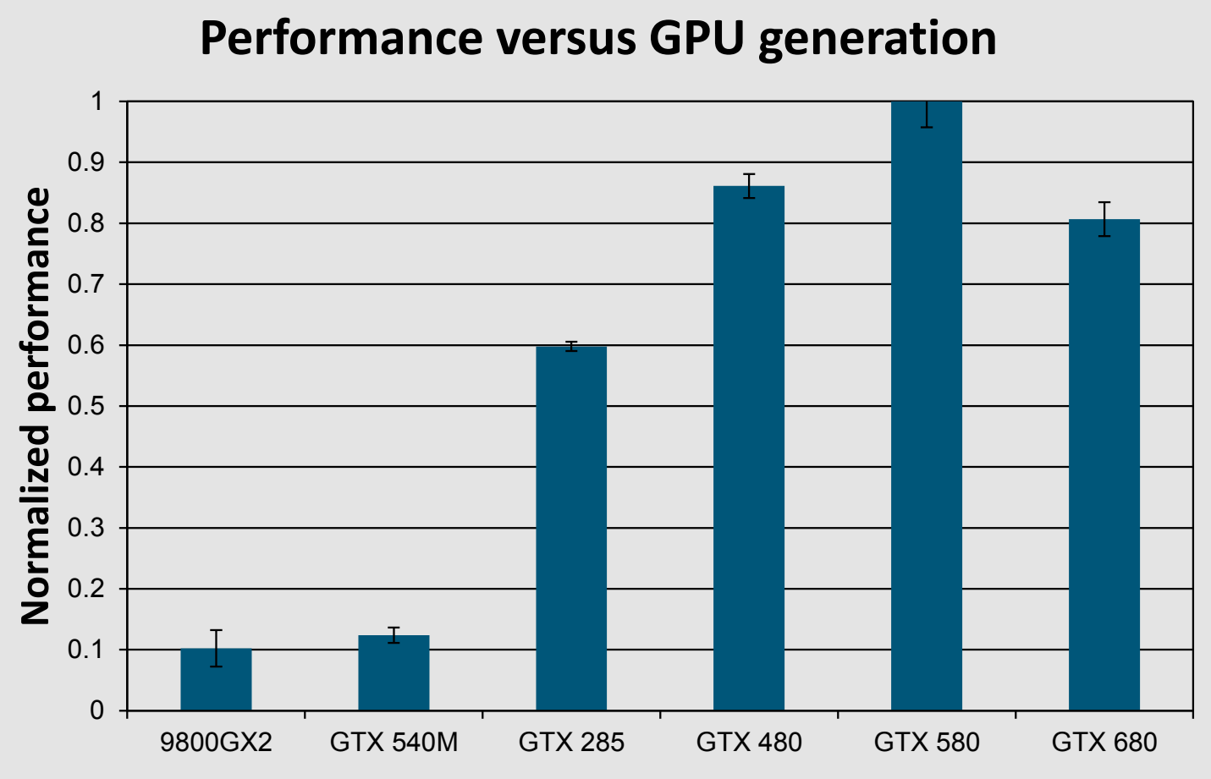
Performance versus voxel grid size



Performance versus number of GPUs



Performance versus number of cameras



Performance versus GPU generation

We observe that the generation for which the system is written has the highest performance of the currently available cards using our setup, and that the system using the GTX 580 can handle up to 512x512x64 (roughly 16M) voxels in real time, assuming 24 frames per second. There is no substantial performance hit involved with using multiple GPUs to observe several scenes.

References

1. Pundik, D., and Moses, Y. Video synchronization using temporal signals from epipolar lines. *Computer Vision – ECCV 2010* (2010), 15-28.
2. Berge, A. et al. Recommendations and guidelines for image processing on heterogeneous hardware. *Technical report* (2013).

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 218 197.

About Us

SINTEF ICT
Department of Applied
Mathematics
www.hetcomp.com

SINTEF ICT
Department of Optical
Measurement Systems
And Data Analysis
www.sintef.com/omd

Technology for a better society

SINTEF