# Real-time Spatiotemporal Stereo Matching Using the Dual-Cross-Bilateral Grid


Christian Richardt — University of Cambridge
Douglas Orr — University of Cambridge
Ian Davies — University of Cambridge
Antonio Criminisi — Microsoft Research Cambridge
Neil A. Dodgson — University of Cambridge


## Overview

We introduce a real-time stereo matching technique based on a reformulation of Yoon and Kweon's adaptive support weights algorithm. We use the bilateral grid to achieve a speedup of 200× compared to a straightforward full-kernel GPU implementation, making our technique **the fastest on the Middlebury website**.

Based on our technique, we present a spatiotemporal stereo matching approach that **incorporates temporal evidence in real time** (>14 fps). Our technique visibly reduces flickering and outperforms per-frame approaches in the presence of image noise. Source code for all our techniques and datasets are available on our project website:

http://www.cl.cam.ac.uk/research/rainbow/projects/dcbgrid/

## Motivation

Yoon and Kweon's adaptive support weights are a popular non-global stereo matching technique. Results are good, but the algorithm is slow, taking about a minute for Tsukuba. Our aim is to **speed up their technique by several orders of magnitude**, hence making it practical for real-time use.

## Adaptive Support Weights

Yoon & Kweon's technique relies on aggregation of support over large window sizes and weights that adapt according to similarity and proximity to the central pixel in the window. The weight between two pixels is given by

$$w(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{\Delta E(\mathbf{p},\mathbf{q})}{\gamma_c} - \frac{\|\mathbf{p}-\mathbf{q}\|}{\gamma_p}\right).$$

Starting from cost space $C(\mathbf{p}, d)$, with pixel $\mathbf{p} = (x, y)$ in the left image and disparity hypothesis $d$, the aggregated costs are

$$C'(\mathbf{p}, d) = \frac{1}{k} \cdot \sum_{\mathbf{q}\in N_\mathbf{p}} w(\mathbf{p},\mathbf{q}) \cdot w(\underline{\mathbf{p}},\underline{\mathbf{q}}) \cdot C(\mathbf{q}, d),$$

where $\underline{\mathbf{p}} = (x - d, y)$ is the corresponding pixel in the right image and $N_\mathbf{p}$ ranges over the 35×35 pixel support window.

## Dual-Cross-Bilateral Aggregation

Yoon & Kweon's technique is similar to a bilateral filter in that it smoothes the cost space while preserving edges in both input images. In the bilateral filtering framework, we call this kind of filter a **dual-cross-bilateral filter** (DCB).

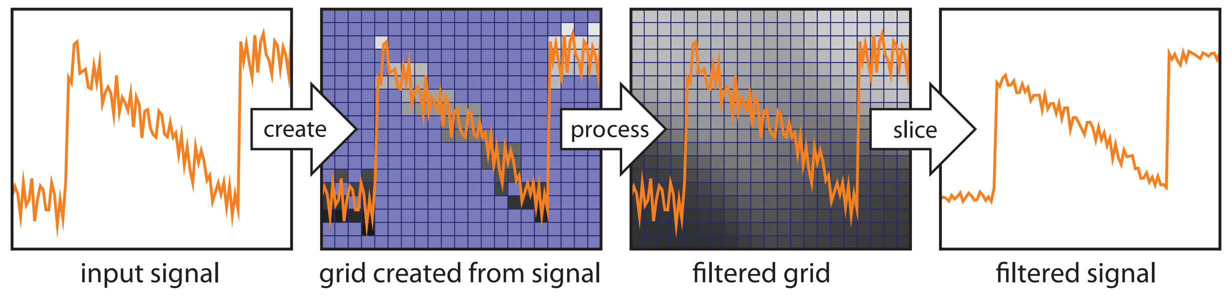
We reformulate their approach using Gaussian weights, the *de facto* standard in bilateral filtering. This yields

$$w(\mathbf{p},\mathbf{q}) = G_{\sigma_r}(\Delta E(\mathbf{p},\mathbf{q})) \cdot \sqrt{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)},$$

where $\sigma_r$ and $\sigma_s$ are similarity and proximity parameters. Our DCB aggregation improves on our implementation of Yoon & Kweon in the *nonocc* and *all* categories in almost all cases.

## Bilateral Grid

Full-kernel implementations of the bilateral filter are slow, so we use the bilateral grid. It has the interesting property that it **runs faster and uses less memory** as $\sigma$ increases.

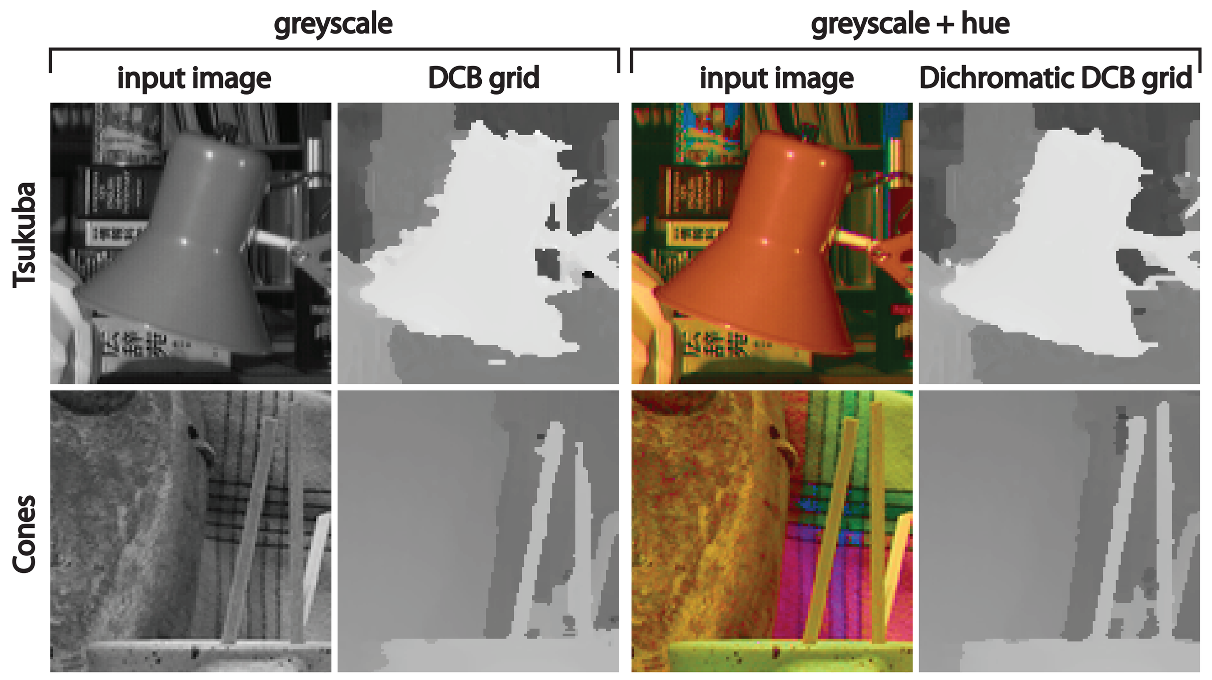



## The DCB Grid

The bilateral grid can also be used for cross-bilateral filtering. We extend the bilateral grid to take into account the input images as edge images, and to accumulate cost space values instead of pixel values. We call our extension the **DCB grid**.

Our DCB grid runs at 13 fps or higher on all datasets, which is **more than 200× faster** than the full-kernel implementation.

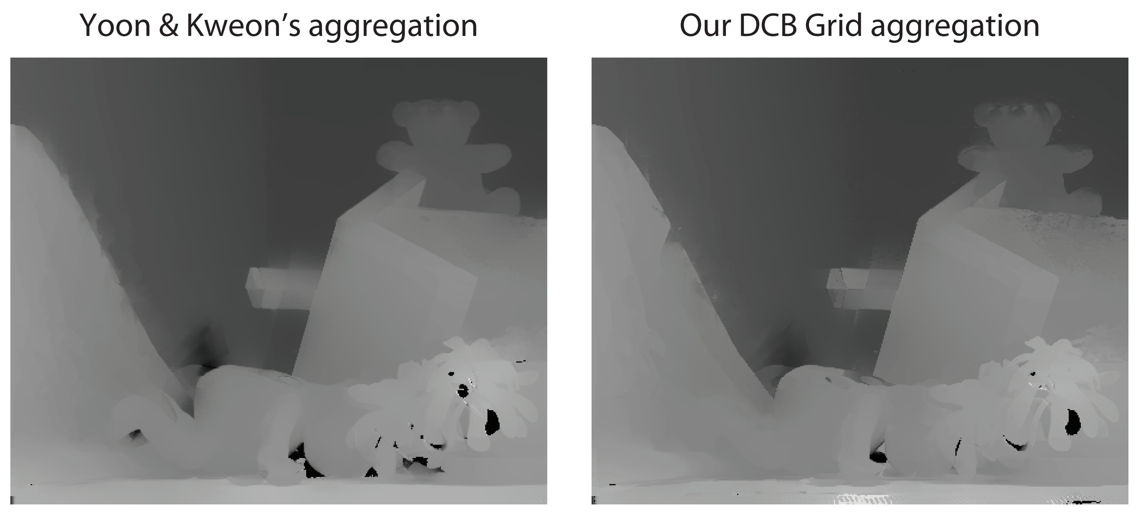
## Dichromatic DCB Grid




## Temporal DCB Grid

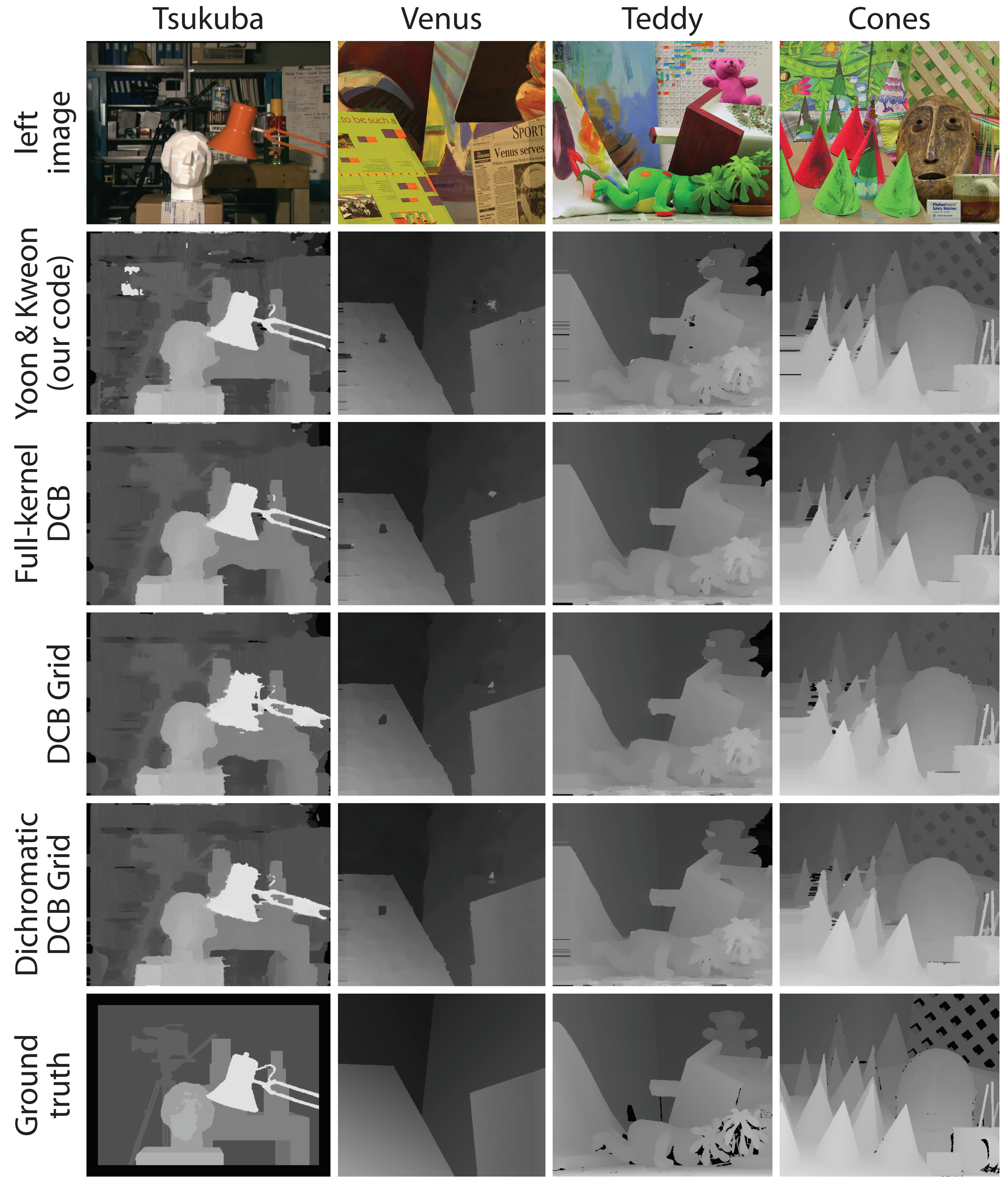
Per-frame techniques are insufficient to achieve temporally coherent disparity maps from stereo videos. We aggregate costs over a 3D spatiotemporal support window of 5 frames. The **run time that is sublinear** in the number of frames: processing 5 frames only takes 76% longer than one frame.

## Spatial-Depth Super-Resolution




| fps | nonocc | all | disc | fps | nonocc | all | disc |
|---|---|---|---|---|---|---|---|
| 0.15 | 17.1 | 17.4 | 41.7 | 17.0 | 19.9 | 19.2 | 42.5 |

## Results




## Run times (in ms)

We benchmarked our techniques on an Nvidia Quadro FX 5800. Asterisks (*) indicate estimated run times.

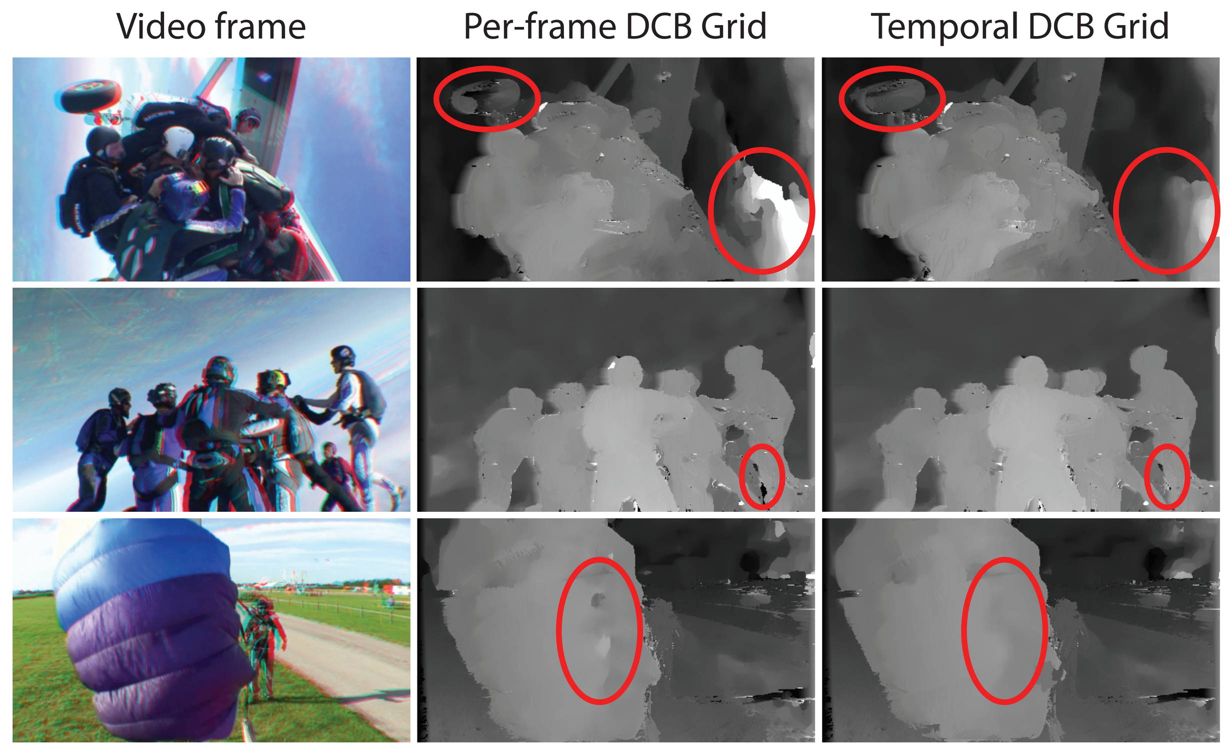
| Technique | Tsukuba 384 × 288 ×16 | Venus 434 × 383 × 20 | Teddy 450 × 375 × 60 | Cones 450 × 375 × 60 |
|---|---|---|---|---|
| **DCB Grid** | 14.2 | 25.7 | 75.8 | 75.0 |
| Real-time GPU | 30* | 60* | 200* | 200* |
| Reliability DP | 42 | 109 | 300* | 300* |
| **Dichromatic DCB Grid** | 188 | 354 | 1,070 | 1,070 |
| Plane-fit BP | 200* | 400* | 1,000* | 1,000* |
| Y&K (our GPU impl.) | 2,350 | 4,480 | 13,700 | 13,700 |
| **Full-kernel DCB** | 2,990 | 5,630 | 17,700 | 17,600 |
| Yoon & Kweon | 60,000 | 100,000* | 300,000* | 300,000* |

## Performance

Comparison of our techniques to Yoon & Kweon and selected real-time techniques using the Middlebury benchmark.

| Technique | Rank | Tsukuba | | | Venus | | | Teddy | | | Cones | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | nonocc | all | disc | nonocc | all | disc | nonocc | all | disc | nonocc | all | disc |
| Plane-fit BP | 19.4 | 0.97 | 1.83 | 5.26 | 0.17 | 0.51 | 1.71 | 6.65 | 12.1 | 14.7 | 4.17 | 10.7 | 10.6 |
| Yoon & Kweon | 32.8 | 1.38 | 1.85 | 6.90 | 0.71 | 1.19 | 6.13 | 7.88 | 13.3 | 18.6 | 3.97 | 9.79 | 8.26 |
| **Full-kernel DCB** | 47.7 | 3.96 | 4.75 | 12.9 | 1.36 | 2.02 | 10.4 | 9.10 | 15.9 | 18.4 | 3.34 | 9.60 | 8.26 |
| Y&K (our GPU impl.) | 48.2 | 4.39 | 5.29 | 8.10 | 1.30 | 2.07 | 8.31 | 9.39 | 16.3 | 18.4 | 3.68 | 9.96 | 8.42 |
| **Dichromatic DCB Grid** | 52.9 | 4.28 | 5.44 | 14.1 | 1.20 | 1.80 | 9.69 | 9.52 | 16.4 | 19.5 | 4.05 | 10.4 | 10.3 |
| Real-time GPU | 56.2 | 2.05 | 4.22 | 10.6 | 1.92 | 2.98 | 20.3 | 7.23 | 14.4 | 17.6 | 6.42 | 13.7 | 16.5 |
| Reliability DP | 59.7 | 1.36 | 3.39 | 7.25 | 2.35 | 3.48 | 12.2 | 9.82 | 16.9 | 19.5 | 12.9 | 19.9 | 19.7 |
| **DCB Grid** | 64.9 | 5.90 | 7.26 | 21.0 | 1.35 | 1.91 | 11.2 | 10.5 | 17.2 | 22.2 | 5.34 | 11.9 | 14.9 |

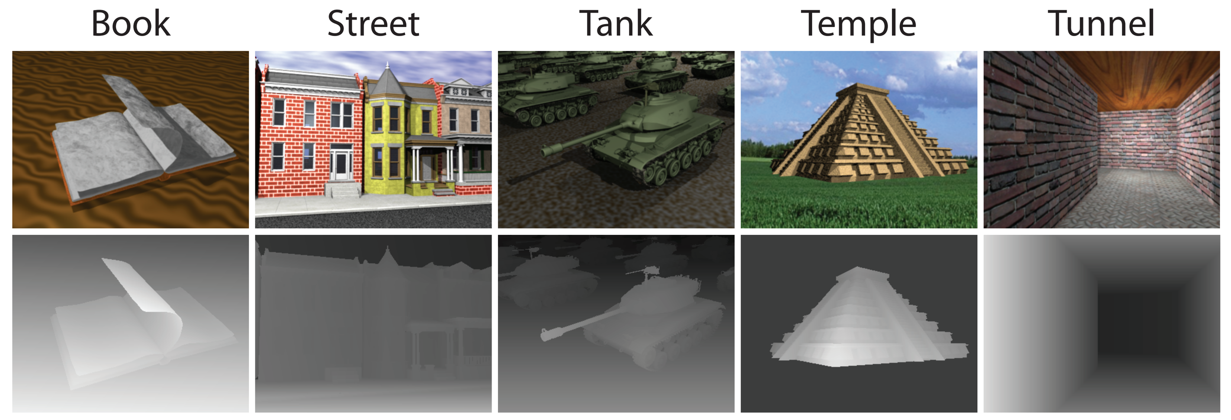
## Qualitative Evaluation on Stereo Videos



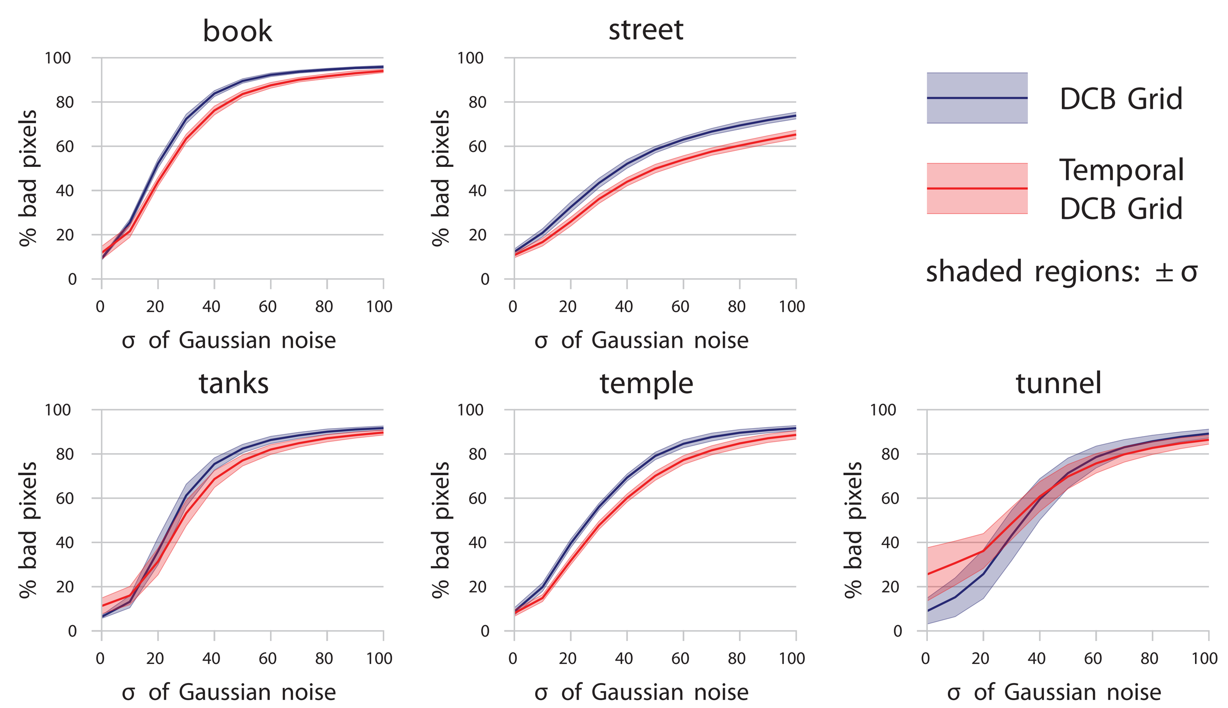



## Grund Truth Stereo Videos

As there are no stereo videos with ground truth disparities, we created a set of 5 synthetic stereo videos with ground truth disparity maps, which we make available.




## Quantitative Evaluation on Stereo Videos





## Acknowledgements

We are grateful to Andrew Fitzgibbon for helpful discussions. We further thank the anonymous reviewers for their valuable feedback, and Nvidia for donating the Quadro graphics card through their CUDA Centre of Excellence at Cambridge.

Christian Richardt and Ian Davies were supported by the Engineering and Physical Sciences Research Council (EPSRC). Douglas Orr was supported by Presenccia (FP6-FET-27731).


UNIVERSITY OF CAMBRIDGE

Microsoft Research