Supplementary Material - Functionally Modular and Interpretable Temporal Filtering for Robust Segmentation

Jörg Wagner[1,2]
Joerg.Wagner3@de.bosch.com
Volker Fischer[1]
Volker.Fischer@de.bosch.com
Michael Herman[1]
Michael.Herman@de.bosch.com
Sven Behnke[2]
behnke@cs.uni-bonn.de

[1] Bosch Center for Artificial Intelligence, 71272 Renningen, Germany

[2] University of Bonn, Computer Science Institute VI, Autonomous Intelligent Systems, Endenicher Allee 19 A, 53115 Bonn, Germany

1 Simulation of Aleatoric Perturbations

Aleatoric failures originate from perturbations inherent in the data. To simulate such perturbations, we add noise, clutter, and changes in lighting conditions to all sequences. In the following, we give a detailed desciption of the process used to generate these perturbations. After applying the perturbations to the clean sequences generated from the *SceneNet RGB-D* dataset, we clip pixels to the interval $[0,1]$ to get valid images. Example sequences are shown in Fig. 1.

Noise is simulated by adding independent Gaussian noise with zero mean to each pixel. The variance of the noise is independently sampled for each sequence from the interval $[0, 0.001]$.

Clutter is introduced by setting subregions of each image to the pixel mean μ, computed on a per-sequence basis. The clutter is generated once per-sequence and applied to each frame. Thus, the resulting clutter pattern is the same in each frame, comparable to dirt on the camera lens. The perturbed images $\mathbf{x}'_j$ are calculated by:

$$\mathbf{x}'_j = \mathbf{x}_j \cdot (1 - \mathbf{m}) + \mu \cdot \mathbf{m}, \tag{1}$$

where $\mathbf{m}$ is a per-sequence clutter mask, μ the per-sequence pixel mean, and $\mathbf{x}_j$ the clean image. The clutter mask is generated by summing N_g Gaussian kernels whose centers are randomly placed (uniformly sampled) within the image dimensions. Each kernel is normalized to the maximum value one. The number of kernels N_g is uniformly sampled for each sequence from the interval $[0, 8]$. In addition, we uniformly sample the standard deviation of each dimension independently from the interval $[10, 36]$. The kernels are truncated at three times the standard deviation.

Changes in lighting conditions are simulated by increasing or decreasing the intensity of frames. For each sequence, we uniformly sample one frame $\mathbf{x}_i$ and a scaling factor s from the interval $[0.5, 1.0]$. In addition, we draw a multiplier p which with a probability of 0.5 is either 1 or -1. The perturbed images $\mathbf{x'}_j$ are calculated by:

$$\forall j < i : \mathbf{x}'_j = \mathbf{x}_j, \tag{2}$$

$$\forall j \geq i : \mathbf{x}'_j = \mathbf{x}_j + p \cdot 0.3^{j-i} \cdot s. \tag{3}$$

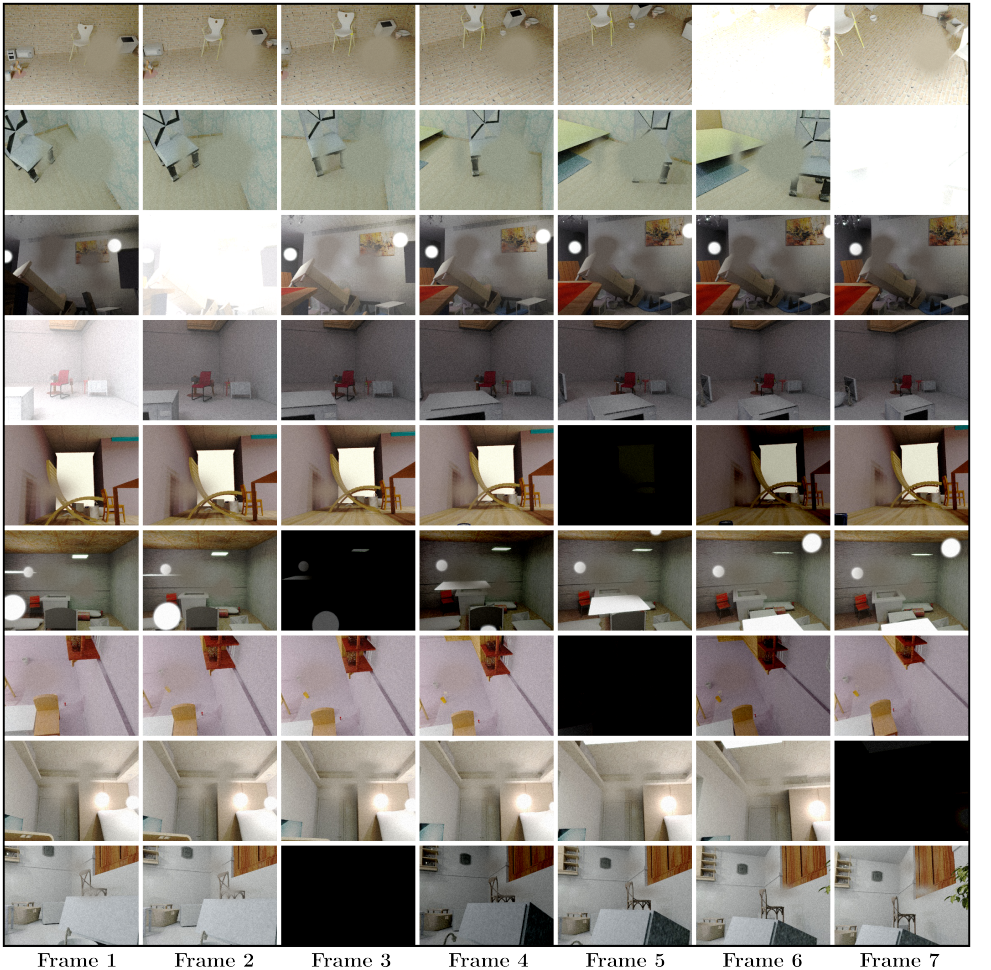

Figure 1: Example sequences of the data used for training and in the evaluation. One sequence of length 7 is shown per row. Each sequence is perturbed with noise, clutter, and changes in lighting conditions.

2 Example Prediction of FMTNet

In Fig. 2, we show an example prediction of our FMTNet. In addition to visualizing the predicted semantic segmentation (Fig. 2c), we also show the predicted depth map (Fig. 2e) and the update gate (Fig. 2f), which are two of the human interpretable representations computed within our functionally modularized temporal filter. The model is able to predict a meaningful depth map as well as camera motion, which are required to propagate information over time. This is especially visible in the last frame of the sequence – although the last frame is missing, the model is still able to produce a meaningful semantic segmentation. In Fig. 2f, we show the gate $\mathbf{i}_t$ of our update module. A white pixel corresponds to a gate value of one, which means that the model uses information provided by the current input frame. A black pixel, on the other hand, corresponds to a gate value of zero – the model relies on prior knowledge of previous frames. As expected, the gate of the first frame is fully white, since the filter has to rely on new information. In the last frame, the gate is mainly black, since no meaningful information is provided in that frame. The gate values at the right border of all frames are more white, as the model has never seen these areas before due to camera motion.

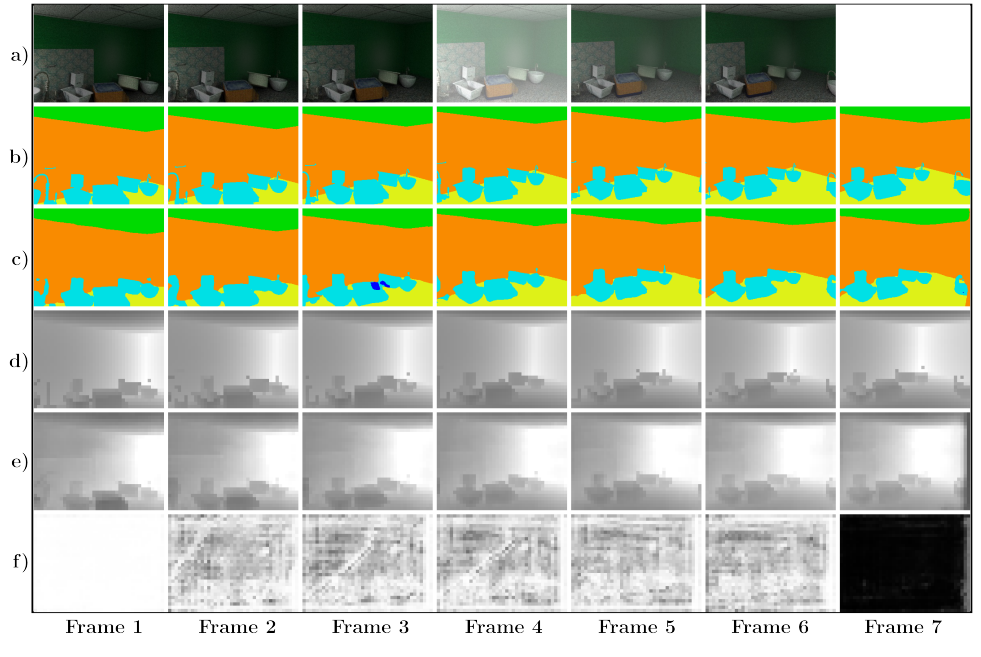

Figure 2: Example prediction of our FMTNet. a): Input sequence; b): Ground-truth semantic segmentation. c): Predicted semantic segmentation; d): Ground-truth deph; e): Predicted depth; f): Update gate $\mathbf{i}_t$, white corresponds to a value of one (use new information).