Spatio-thermal depth correction of RGB-D sensors based on Gaussian Processes in real-time

Christoph Heindl

PROFACTOR®

PROFACTOR GmbH, Austria

Introduction

Figure 1: RGBD-Sensor components

Main vision sensor in many robotic / computer vision applications these days.

RGBD - Commodity 3D Sensing

Advantages

- Dense depth maps in real-time
- Simultaneous color and depth streams
- Affordable devices

Drawbacks

- Inaccurate compared to industrial cameras
- Operating temperature sensitivity
- No time synchronization

Improve the accuracy of dense depth maps by correcting in a joint spatio-thermal domain.

Gaussian Processes

Gaussian Process: a probabilistic model used in supervised machine learning for classification and regression[1, 2].

Regression searches for an unknown function $\mathcal{F}$ that satisfies

$$\mathcal{F}(\mathsf{X}) = \mathsf{y} + \epsilon$$

where $\mathsf{X} \in \mathbb{R}^{N \times d}$ are training features, $\mathsf{y} \in \mathbb{R}^{N \times 1}$ are the target scalar observations and $\epsilon \sim \mathcal{N}$ is i.i.d. Gaussian observation noise.

In a probabilistic framework we would like to compute a distribution over functions based on training data and use the posterior predictive distribution for new predictions.

$$p(\mathbf{y}_\star|\mathbf{X}_\star, \mathbf{X}, \mathbf{y}) = \int p(\mathbf{y}_\star|\mathcal{F}, \mathbf{X}_\star)p(\mathcal{F}|\mathbf{X}, \mathbf{y})d\mathcal{F}$$

where $\mathbf{X}_\star$ are test inputs, $\mathbf{y}_\star$ are the predictions, $\mathbf{X}$ are training inputs, $\mathbf{y}$ are training target values.

Generally intractable to integrate over all possible functions and their parametrizations.

Assumption: all variables are jointly Gaussian

$$\begin{bmatrix} y \\ y_\star \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathcal{M}(X) \\ \mathcal{M}(X_\star) \end{bmatrix}, \begin{bmatrix} \Sigma(X, X) & \Sigma(X, X_\star) \\ \Sigma(X, X_\star)^T & \Sigma(X_\star, X_\star) \end{bmatrix} \right)$$

where $\mathcal{M} \colon \mathbb{R}^{K \times d}$ is a mean function, and Σ is a positive-definite function encoding the covariance between samples.

Gaussian Process Regression

Posterior predictive now simplifies due to Gaussian identities

$$p(y_\star | X_\star, X, y) \sim \mathcal{N}(u_\star, K_\star),$$
$$u_\star = \mathcal{M}(X_\star) + \Sigma(X, X_\star)^T \Sigma(X, X)^{-1}(y - \mathcal{M}(X)),$$
$$K_\star = \Sigma(X_\star, X_\star) - \Sigma(X, X_\star)^T \Sigma(X, X)^{-1} \Sigma(X, X_\star)$$

Computationally interesting: just linear algebra.

Covariance functions encode assumptions about the form of functions that are being modelled. Squared exponential kernel function

$$\Sigma_{ij} = \sigma_s^2 \exp(-0.5(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)) + \delta_{ij}\sigma_y^2.$$
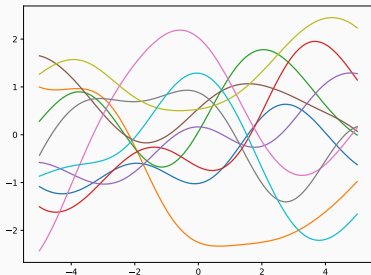

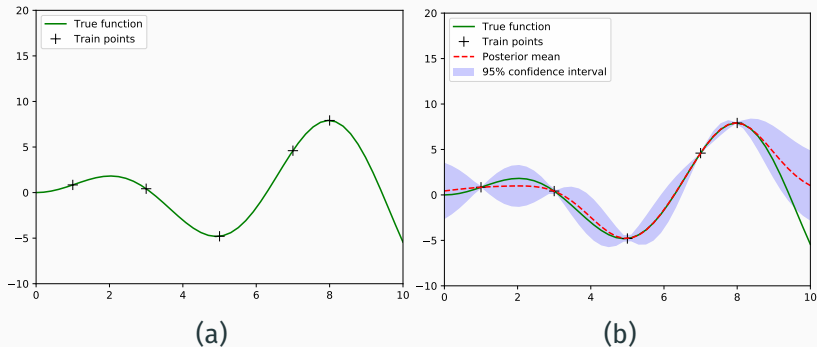**Figure 2:** Ten samples from the squared exponential prior.

Figure 3: Illustration of Gaussian Process regression. Left: Input samples and true function, right: Predictions including confidence intervals. Output generated by [3].

Approach and Results

Per pixel dense depth correction using Gaussian Process regression based on spatio-thermal input. For a given depth map $\mathcal{D}$ and temperature reading t we compute a corrected depth map $\mathcal{D}_\star$ by

$$\mathcal{D}_\star(i,j) = \mathcal{D}(i,j) + \Delta_{ij}$$

where Δ_{ij} is given by a Gaussian Process Regression $\Delta_{ij} = \mathcal{G}(\mathsf{x}_{ij})$.

The input to the Gaussian Process $\mathcal{G}$ is given by

$$\mathbf{x}_{ij} = \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} \mathcal{D}(i,j)\mathbf{K}^{-1}[i,j,1]^T \\ t \end{bmatrix}$$

$$y_{ij} = \mathcal{D}_{\mathrm{RGB}}(i,j) - \mathcal{D}_{\mathrm{IR}}(i,j)$$

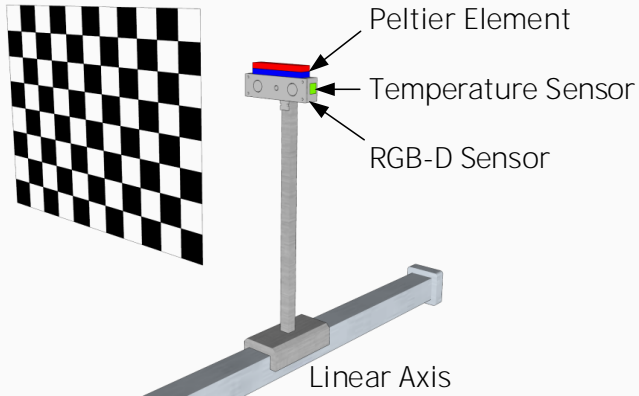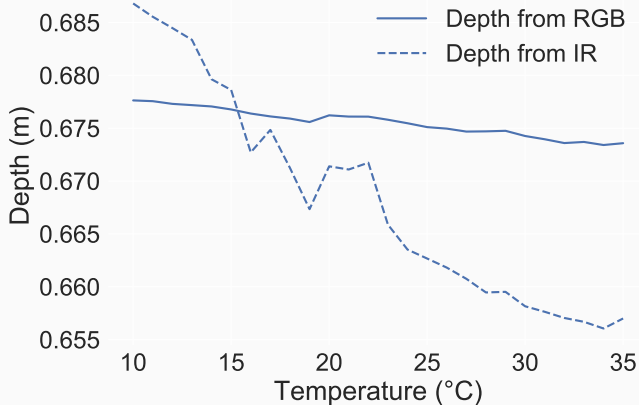**Figure 4:** Hardware capture setup

Figure 5: Effects of varying temperature on depth estimation at a fixed distance over a small window region.

Figure 6: Depth correction by Gaussian Process regression. Before/after correction effects. The white-speckles in after-images are due to missing depth sensor readings for which no correction can be computed. The error is significantly reduced independent from chosen position and temperature.

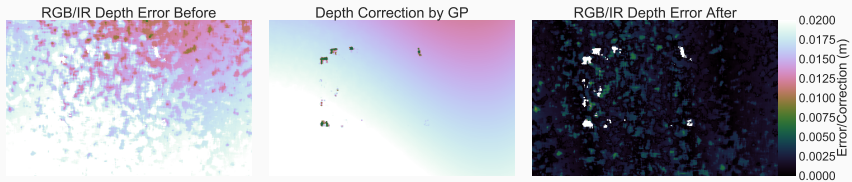**Figure 7:** Depth correction by Gaussian Process regression.

Figure 8: Depth correction by Gaussian Process regression.

Table 1: RMSE before and after correction in Cartesian space across all temperature and distance captures. The correction yields an improvement by one order of magnitude.

	x (mm)	y (mm)	z (mm)
RMSE before correction	5.7	3.7	16.0
RMSE after correction	0.7	0.5	2.2

Gaussian Process predications can leverage GPUs. Mostly boils down to linear algebra matrix multiplications.

Table 2: Execution times for correction per depth map frame of size 640 times 480. The GPU optimized version outperforms the other variants.

	Time (s)	FPS (1/s)
CPU	20	0.05
GPU optimized	0.14	7.1

For reproducibility we made our dataset and source code publicly available at

`https://github.com/cheind/rgbd-correction`

References

📄 Carl Edward Rasmussen and Christopher KI Williams.
Gaussian processes for machine learning, volume 1.
MIT press Cambridge, 2006.

📄 Kevin P Murphy.
Machine learning: a probabilistic perspective.
MIT press, 2012.

📄 Christoph Heindl.
Gaussian processes in python.
`https://github.com/cheind/py-gaussian-process`, 2017.

This work was supported by

*Federal Ministry
for Transport,
Innovation and Technology*