2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Spatio-Temporal Depth Interpolation (STDI)

Matthias Ochs[1], Henry Bradler[1], and Rudolf Mester[1,2]

***Abstract*— In the area of autonomous driving, sensing the environment is most important for self-localization and egomotion estimation. Visual odometry/SLAM methods have proven capable to achieve good results, even in real-time applications by operating in a sparse mode. Running on a sequence, these methods need to continuously incorporate new features well distributed over the image. Therefore, the performance of these methods can be further improved, if they are supplied with coarse but dense initial depth information, that can be utilized at arbitrary sparse image positions. Previously triangulated depths and even high quality depth measurements of a LIDAR sensor are not suitable for this task, since they only provide a sparse depth map. To solve this issue, we introduce a novel interpolation method called *Spatio-Temporal Depth Interpolation* (STDI), which exploits spatial *and* temporal correlations of the data (e.g. sequences of sparse depth maps) to give a consistent dense output including associated uncertainties. STDI is a fused approach, which makes use of the most important components of a principal component analysis (PCA) (spatial information) *and* additionally is capable to re-use information of previously interpolated depth maps in a regression based approach (temporal information). We evaluate the quality of STDI on the KITTI visual odometry benchmark, where a sequence of extremely sparsely sampled depth maps ($\approx$ 40 depth values) is densified and on the KITTI depth completion benchmark. The latter deals with the densification of sparse LIDAR input. Of course, our method is not limited to these applications and can be used for any densification of sparse sequential data which is expected to contain spatial and/or temporal correlations (e.g. initialization for dense optical flow methods based on a sparse measurement).**

I. Introduction

In this paper, we present a novel statistical interpolation approach, which applied to sequences of sparse input data is designed to exploit not only spatial correlations but additionally temporal information encoded in the sequence and by this give a dense and consistent spatio-temporal enforced interpolation. We call this method *Spatio-Temporal Depth Interpolation* (STDI). It is particularly interesting for real-time sequential applications that typically process data only in an efficient sparse mode but need to initialize input at arbitrary new positions.

We focus on an application for visual odometry/SLAM methods in autonomous driving where we need a coarse but dense initialization of the depth structure to increase performance and robustness. At each frame, we have already very limited depth information of previously estimated 3D points or additionally coming from a LIDAR sensor. To be able to continuously estimate egomotion for self-localization, it is crucial for SLAM to regenerate new 3D points at arbitrary positions. This regeneration can be extremely simplified if we compute a dense interpolation that even if only based on very sparse input preserves the main print of the correct depth map. This can not be achieved by interpolation methods that do not possess any prior knowledge about the data in use.

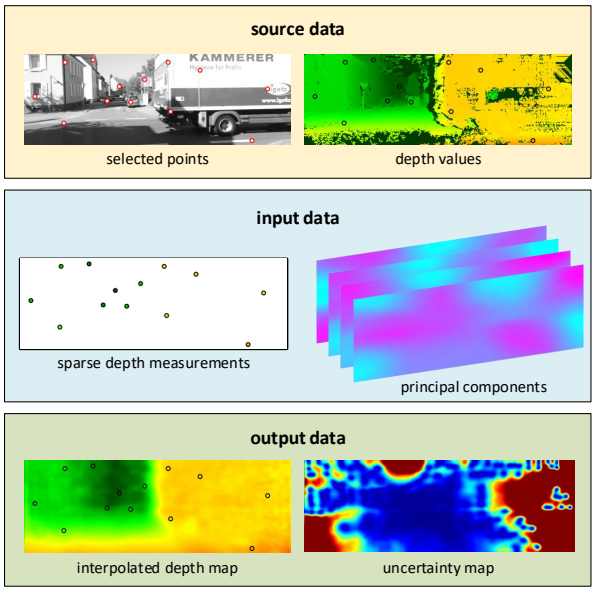

Fig. 1: In the first row, you can see an image with possible feature positions indicated by red circles and the corresponding ground truth depth map. The second row illustrated the very sparse depth measurement and a set of the most important principal basis depth images computed by PCA. Finally, the last row shows the resulting interpolation of the sparse depth measurement and a corresponding uncertainty map with red values in areas of high uncertainty and blue color in reliable areas.

For this purpose, we utilize a representative training sequence of depth maps (e.g. KITTI dataset) and learn its spatial and temporal correlations. Based on this sequence, the principal component analysis (PCA) is used to compute a new basis that is more powerful at expressing depth maps. We will use the most important components (basis depth maps) to perform the interpolation not in the original domain but in this statistical optimal truncated basis representation. Unlike other interpolation approaches, we will not only exploit spatial information by this statistical change of basis, but we will also learn temporal dependencies of those PCA basis coefficients. To do so, we expand the training sequence of depth maps into the truncated PCA basis and yield a

[1]Visual Sensorics & Information Processing Lab, Goethe University, Frankfurt am Main, Germany
[2]Computer Vision Laboratory, ISY, Linköping University, Sweden

sequence of the coefficients of the dominant basis depth maps. We then deploy statistical linear regression to obtain a predictor that is able to give an optimal estimate of the current depth map based on the preceding basis coefficients. By this it is even possible to predict a reasonable depth map of the current frame even if no current input of the depth is available.

Finally, we fuse spatial and temporal knowledge in an MAP-like approach, where sparsely observed depth input is interpolated within the efficient truncated PCA basis in a least squares manner including the temporal predictor as prior knowledge. The dense interpolation can then simply be computed by transforming back to the original domain of the depth map. STDI further supplies an uncertainty map that gives the reliability of the interpolation. Exemplary source data, input and output of STDI is shown in figure 1.

The performance of STDI is evaluated in our experiments. In the KITTI odometry benchmark, it is used to sequentially predict depth maps of size 1098×370 based on a varying number of input depth values (uniform [20, 60]). Up to three of the preceding time steps are utilized to temporally enforce the interpolation. In a second experiment, STDI is used for non-guided (no input camera image) depth completion of single sparse depth maps coming from a LIDAR sensor. Due to the non-existing temporal dependency in this benchmark, STDI can not be performed in a temporal enforced mode and only exploit spatial correlations for the interpolation task.

II. Related Work

Interpolation of sparse measurements has been studied in the literature in a broad field of interest in context of computer vision methods (e.g. optical-flow, lidar data, stereo vision or sparse depth maps from visual SLAM/odometry methods). In this work, we focus on the interpolation of sparse depth maps, which can be retrieved from any depth sensor. Wulff and Black [1] also use the principal component analysis (PCA) to determine a significant basis to estimate missing values in an optical-flow field. A quite similar approach was proposed by Ochs et al. [2] to interpolate very sparse depth maps with a maximum a posteriori method, where a static learned prior from the training data was used. The main contribution of this paper is the transformation of this static prior into a dynamic one, which is inspired by the work of Bradler et al. [3].

Recently, KITTI released a new depth completion benchmark [4]. In this challenge, missing depth values need to be interpolated based on measured sparse LIDAR data. The task of depth completion is strongly related to image or depth super-resolution [5]. The main difference between both methods is the location of the sampling points. In super-resolution, the sampling data is normally arranged on a regular grid, while this is not the case for depth completion. Classical super-resolution methods that are formulated as an optimization problem or exploit specific filters were applied in the work of [6], [7], [8], [9], [10]. Modern approaches that incorporate convolutional neural networks (CNN) lead to even better results [11], [12], [13].

Besides the similar task of super-resolution, there are also works on depth completion which can be subdivided into image guided and non-guided approaches. Representatives of the first class, which need an additional RGB image, are [14] and [15]. Both approaches use CNNs for depth completion. The CNN of Jampani et al. learns edge-aware bilateral filters to preserve structural properties. These high dimensional filters can also be applied to sparse input data. In the method from Schneider et al. [14], they use semantic labels as additional guidance, which also allows better preserving of edges and local context. Uhrig et al. [4] proposed a CNN for depth completion without any further guidance. To be able to process the sparse depth data as input, they introduced a sparse convolutional layer which is explicitly aware of missing depth values so that the kernel does not incorporate such invalid values.

III. Approach

In the following subsections, we describe a procedure that aims at interpolating a depth map $\vec{d}$ within a sequence by incorporating not only a very sparse measurement $\tilde{\vec{d}}$ of the dense map $\vec{d}$ but also prior knowledge about temporal correlations. For this purpose, we extend [2] which uses principal components of the PCA as a basis and the corresponding eigenvalues as static prior knowledge. We introduce a temporal dynamic prior which is analogous to [3]. The prediction of the depth map in its PCA basis representation is performed by including temporal information of up to three of the preceding time steps. We use a similar notation as [2].

A. Basis Transformation

A simple basis transformation in a center-of-mass ($\vec{m}$) system for a complete orthonormal basis $\mathbf{U}$ is given by

$$\vec{y} = \mathbf{U}^T \left(\vec{d} - \vec{m} \right) \quad , \quad \vec{d} = \mathbf{U}\, \vec{y} + \vec{m}. \tag{1}$$

In the case of a PCA, the basis $\mathbf{U}$ consists of the principal components and $\vec{m}$ is the mean of the training samples. As in [2], we do not use a complete orthonormal basis ($\mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I}$) but an incomplete orthonormal basis ($\mathbf{B}^T\mathbf{B} = \mathbf{I}$ but $\mathbf{B}\mathbf{B}^T \neq \mathbf{I}$). This basis consists only of the l most important principal component vectors computed for a sequence of n training samples each of dimension s ($l \ll n \ll s$).

If data is only sampled sparsely $\vec{d} \rightarrow \tilde{\vec{d}}$, the basis transformation is given by the solution of a least squares problem equivalent to a maximum likelihood (ML) method:

$$\left\| \tilde{\mathbf{B}}\, \vec{y} + \tilde{\vec{m}} - \tilde{\vec{d}} \right\|_{\mathbf{C}_{\tilde{\vec{d}}}} \rightarrow \min . \tag{2}$$

Depending on the level of sparsity, this transformation might be ill-posed when the number of components that are measured is less than the number of basis vectors (size($\tilde{\vec{d}}$) $< l$).

B. MAP Extension

Extending the data term by an additional prior in a maximum a posteriori probability (MAP) fashion can help to make the transformation in (2) well-posed again. If prior knowledge is given, it can be included by

$$\left\|\tilde{\mathbf{B}}\,\vec{y}+\tilde{\vec{m}}-\tilde{\vec{d}}\right\|_{\mathbf{C}_{\tilde{\vec{d}}}}+\left\|\vec{y}-\hat{\vec{y}}\right\|_{\mathbf{C}_{\hat{\vec{y}}}}\rightarrow\min. \tag{3}$$

In this case $\hat{\vec{y}}$ is the prediction of $\vec{y}$ and $\mathbf{C}_{\hat{\vec{y}}}$ is the covariance matrix of the prediction residuals $\vec{y}-\hat{\vec{y}}$. Following equation (3), the optimal estimate $\vec{y}^*$ and the Cramér-Rao lower bound of its covariance $\mathrm{Cov}\left[\vec{y}^*\right]$ are given by

$$\left(\tilde{\mathbf{B}}^T\mathbf{C}_{\tilde{\vec{d}}}^{-1}\tilde{\mathbf{B}}+\mathbf{C}_{\hat{\vec{y}}}^{-1}\right)\vec{y}^*=\tilde{\mathbf{B}}^T\mathbf{C}_{\tilde{\vec{d}}}^{-1}(\tilde{\vec{d}}-\tilde{\vec{m}})+\mathbf{C}_{\hat{\vec{y}}}^{-1}\hat{\vec{y}},$$
$$\mathrm{Cov}\left[\vec{y}^*\right]=\left(\tilde{\mathbf{B}}^T\mathbf{C}_{\tilde{\vec{d}}}^{-1}\tilde{\mathbf{B}}+\mathbf{C}_{\hat{\vec{y}}}^{-1}\right)^{-1}. \tag{4}$$

Transforming back to the original domain of the data $\vec{d}$ yields a dense estimate $\vec{d}^*$ and its uncertainty $\vec{\xi}(\vec{d}^*)$

$$\vec{d}^*=\mathbf{B}\,\vec{y}^*+\vec{m},$$
$$\vec{\xi}(\vec{d}^*)=\mathrm{diag}\left(\mathrm{Cov}\left[\vec{d}^*\right]\right)=\mathrm{diag}\left(\mathbf{B}\,\mathrm{Cov}\left[\vec{y}^*\right]\mathbf{B}^T\right). \tag{5}$$

C. Static Prediction

In case of static PCA prior knowledge, the prediction of the coefficient vector is always zero ($\hat{\vec{y}}=\vec{0}$), which is equivalent to the mean $\vec{m}$ being the data to be observed most likely as the next measurement. The covariance is given by the diagonal PCA eigenvalue matrix $\mathbf{C}_{\hat{\vec{y}}}=\mathbf{\Lambda}=\mathrm{diag}\left(\lambda_0,\ldots,\lambda_{l-1}\right)$. The eigenvalues λ_i give the variance of the training samples in the subspace of the basis vectors $\vec{u}_i$

$$\lambda_i=\mathrm{var}\left[\vec{u}_i^T\vec{d}\right]=\mathrm{var}\left[y_i\right]. \tag{6}$$

D. Dynamic Prediction

When working on a sequence there might be a better estimate of the next measurement, given the preceding k coefficient vectors $\vec{x}_k^T=(\vec{y}_{-1}^T,\ldots,\vec{y}_{-k}^T)$. For this purpose, we built an affine linear predictor $\hat{\vec{y}}=\mathbf{A}\cdot\vec{x}_k+\vec{b}$ of k-th order similar to [3]. If we enforce unbiasedness and minimum variance, we get the MVUE of $\vec{y}$ as

$$\hat{\vec{y}}=\mathrm{Cov}\left[\vec{y},\vec{x}_k\right]\mathrm{Cov}\left[\vec{x}_k\right]^{-1}\left(\vec{x}_k-\mathbb{E}\left[\vec{x}_k\right]\right)+\mathbb{E}\left[\vec{y}\right],$$
$$\mathrm{Cov}\left[\hat{\vec{y}}\right]=\mathrm{Cov}\left[\vec{y},\vec{x}_k\right]\mathrm{Cov}\left[\vec{x}_k\right]^{-1}\mathrm{Cov}\left[\vec{x}_k,\vec{y}\right],$$
$$\mathrm{Cov}\left[\vec{y}-\hat{\vec{y}}\right]=\mathrm{Cov}\left[\vec{y}\right]-\mathrm{Cov}\left[\hat{\vec{y}}\right]. \tag{7}$$

In figure 2, the correlations of the components of the auto-covariance $\mathrm{Cov}\left[\vec{y},\vec{y}\right]$ (spatial) and cross-covariance $\mathrm{Cov}\left[\vec{y},\vec{x}_k\right]$ (temporal) are visualized. In agreement with the orthogonality of the PCA basis the components of the coefficient vector $\vec{y}$ are uncorrelated. In temporal direction, a strong correlation can only be observed for the most important principal components and their temporal preceding counterparts. This correlation strongly decreases with increasing temporal distance.

Comparing section III-C to the equations (7), the static PCA predictor can be identified to be in fact the simplest implementation of a *dynamic* predictor, i.e. a predictor of 0-th order, which actually is *static* of course. An overview of how predictors of different orders compare against each other is listed in table I.

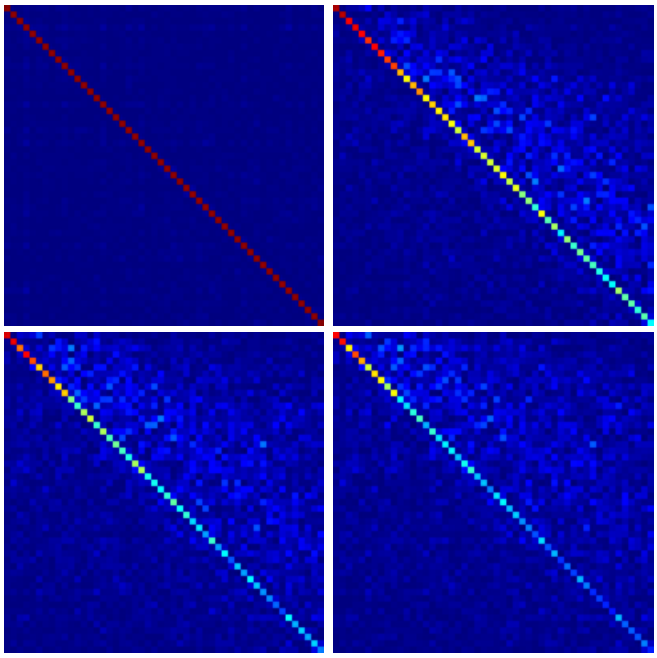

Fig. 2: A visualization of the correlation coefficients of $\mathrm{Cov}\left[\vec{y},\vec{y}\right]$ (tl), $\mathrm{Cov}\left[\vec{y},\vec{y}_{-1}\right]$ (tr), $\mathrm{Cov}\left[\vec{y},\vec{y}_{-2}\right]$ (bl) and $\mathrm{Cov}\left[\vec{y},\vec{y}_{-3}\right]$ (br) is depicted using a color map where red indicates perfect correlation and blue indicates no correlation. Correlations are only visualized for the components of $\vec{y}$ that correspond to the 100 most important principal components.

Predictor	**RMSE** $l=500$	**MAE** $l=500$	**RMSE** $l=50$	**MAE** $l=50$
$\hat{\vec{y}}_{0th}$	1.150	0.415	3.425	1.805
$\hat{\vec{y}}_{simple}$	0.720	0.393	1.712	1.058
$\hat{\vec{y}}_{1st}$	0.510	0.280	1.319	0.821
$\hat{\vec{y}}_{2nd}$	0.481	0.273	1.253	0.793
$\hat{\vec{y}}_{3rd}$	0.468	0.270	1.242	0.788

TABLE I: Prediction residuals / errors for the simple predictor (current state is always assumed to be the same as the preceding one) and predictors of zeroth to third order for different sizes l of the limited basis $\mathbf{B}$. Even the simple predictor performs much better than the static zeroth order predictor. Prediction performance increases up to third order but saturates already at first order which is already indicated by the visualization of temporal correlations in figure 2.

IV. Experiments

In the experiments and evaluations, we show that the proposed enhancement of the PCA guided depth interpolation of Ochs et al. [2] with a dynamic temporal predictor is beneficial, if temporal correlations between consecutive frames can be exploited. Particularly, this is the case in the context of driving scenes.

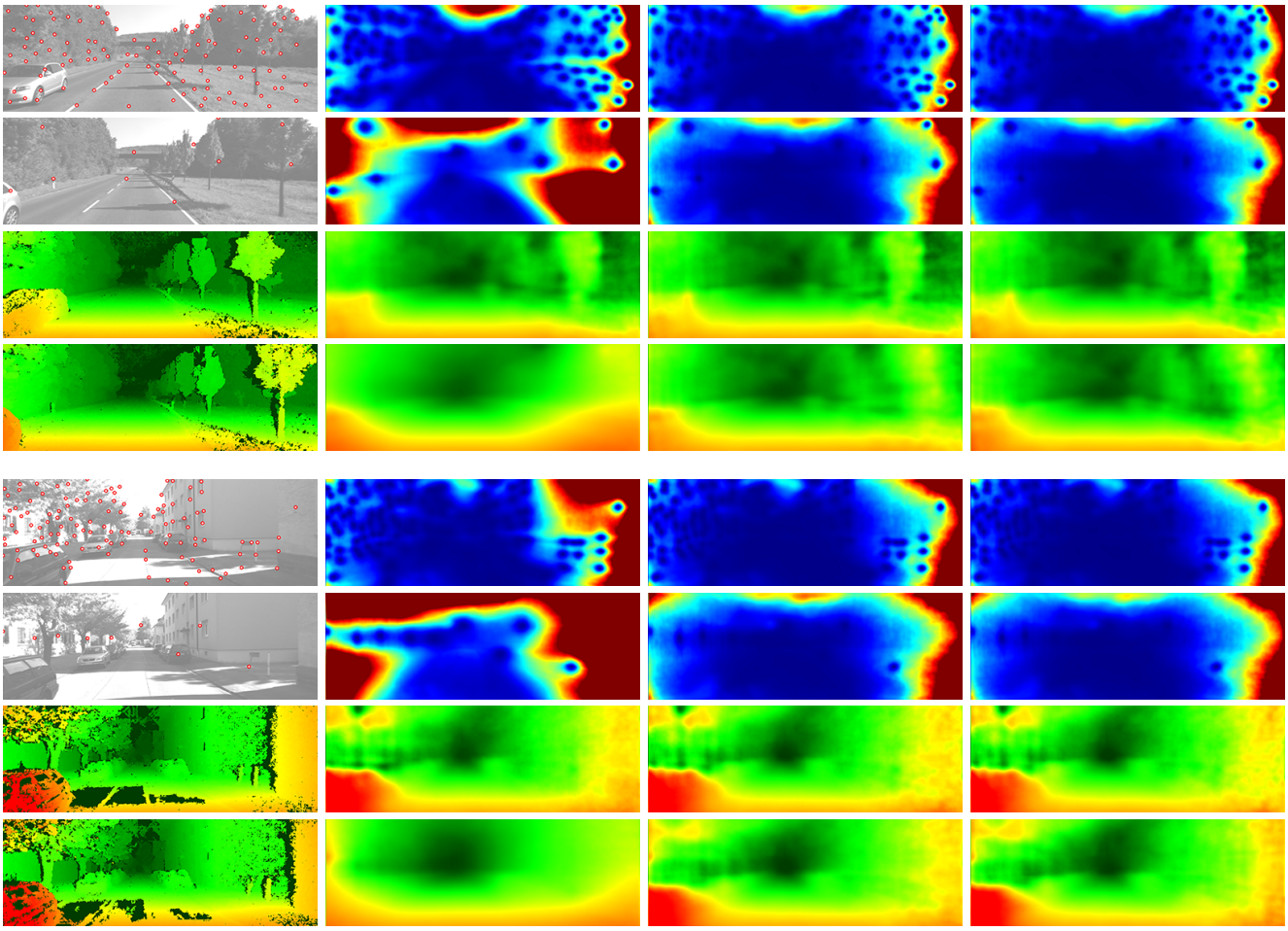

Fig. 3: Exemplary results of the KITTI odometry test dataset [16] (sequence 17 and 19) showing an urban and rural scenario. In the first column, the selected measurement points (row: 1-2 and 5-6) and the corresponding SGBM depth maps (row: 3-4 and 7-8; near points are encoded in red)) of two consecutive frames are shown. The depth values of these points are used for our STDI-ZO, STDI-FO and STDI-TO methods to interpolate the depth and uncertainty maps (dark red indicates high uncertainty), which are shown in the second, third and fourth column. We used $l = 500$ basis vectors for all three methods.

We demonstrate the positive effect of this enhancement on the KITTI odometry dataset. This dataset consists of a training and test set. Hence, we used the training sequences to train our predictor (see section III-D) and to learn the principal components. For learning the PCA basis, we follow the steps described in [2] with some slight modifications. To obtain the depth maps, Ochs at al. use semi-global block matching (SGBM) of Hirschmueller [17], where invalid positions are interpolated with the nearest-neighbor algorithm. As opposed to this, we use a linear interpolation method to get rid of the invalid positions and use only nearest-neighbor interpolation, where the linear interpolation is not capable to retrieve a correct value, which can occur for instance near border locations of the image.

Depth completion of sparse data of a LIDAR scanner is another application field next to the densification of sparse depth maps, which can be obtained by a visual SLAM/odometry framework as in [19], [20], [18]. In this case, we use the LIDAR data points from the KITTI depth completion benchmark [4] to fill missing depth values with our proposed method.

During our experiments, we evaluate our method with prior terms of different order (see section III-D) and for a basis size of $l = 50$ and $l = 500$:

- Zero-order predictor (**STDI-ZO**)
- First-order predictor (**STDI-FO**)
- Third-order predictor (**STDI-TO**)

These variants of our method are evaluated in terms of the following four evaluation measures, where $w_{\text{est}}/w_{\text{ref}}$ is the estimated depth and the reference depth, respectively, and N is the total number of evaluation samples:

- Root mean square error of inverse depth w^{-1}:
  iRMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(w_{i,\text{ref}}^{-1} - w_{i,\text{est}}^{-1})^2}$
- Mean absolute error of the inverse depth w^{-1}:
  iMAE $= \frac{1}{N}\sum_{i=1}^{N}|w_{i,\text{ref}}^{-1} - w_{i,\text{est}}^{-1}|$
- Root mean square error of depth w:
  RMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(w_{i,\text{ref}} - w_{i,\text{est}})^2}$
- Mean absolute error of depth w:
  MAE $= \frac{1}{N}\sum_{i=1}^{N}|w_{i,\text{ref}} - w_{i,\text{est}}|$

A. KITTI Odometry Dataset

For training the basis and evaluating the performance of our proposed methods, we use depth maps, which are computed by SGBM, because KITTI does not provide any depth maps for these sequences. Unfortunately, SGBM yields depth maps, where some depth values do not match with the true depth value. Thus, the evaluation with SGBM depth maps is not as convincing as the evaluation with LIDAR data, like in section IV-B.

Similar to Ochs et al. [2], we use a *good-feature-to-track*-like algorithm [21] to get a sparse sampling of the depth map. For each frame of the test sequences, we randomly change the configuration of good-feature-to-track, so that the number of sampling points uniformly varies between $[20, 60]$. This random sampling ensures that the improvements by our temporal predictor are exhibited on frames with only a very few points. On these frames, STDI-FO, and STDI-TO can predict the coefficient vector $\hat{\vec{y}}$ based on the history of the preceding frames.

The results of the different configurations of our approach are shown in table II. Note, STDI-ZO with basis size $l = 500$ is exactly the same method as proposed in [2].

Method	**iRMSE** [1/km]	**iMAE** [1/km]	**RMSE** [m]	**MAE** [m]
STDI-ZO ($l = 50$)	24.20	13.71	16732	47.13
STDI-FO ($l = 50$)	23.06	13.08	12536	40.40
STDI-TO ($l = 50$)	22.89	12.98	11327	39.08
STDI-ZO ($l = 500$) [2]	22.28	12.20	1091	13.11
STDI-FO ($l = 500$)	21.05	11.45	2596	16.46
STDI-TO ($l = 500$)	21.03	11.46	2916	18.83

TABLE II: The performance of our methods is evaluated on the KITTI odometry test sequences 11 - 21. To generate ground truth depth values, we used the SGBM algorithm. [17].

In general, the results of table II show that using more basis functions yield better performance on all configurations of the proposed interpolation approaches. All methods are capable to reconstruct finer details, if more basis functions are used, which explains the lower errors. But taking more basis functions into account comes also with a higher computational cost and runtime. Including temporal knowledge into the prior term, like it is done in STDI-FO and STDI-TO also increases the performance on most of the error measures. Surprisingly, the RMSE and MAE of STDI-FO and STDI-TO with $l = 500$ are worse than the errors of STDI-ZO. We do not have a conclusive explanation for this behavior right now, but we think that this could be also explained with the bad or misleading SGBM depth values in the evaluation.

Some exemplary results of STDI-ZO, STDI-FO and STDI-TO are shown in figure 3. In the top block of this figure, a basis size of $l = 500$ is used to reconstruct the depth maps with the measured points, which are shown in the first column of that figure. At the lower block, only 50 basis vectors are deployed. For a better visualization of the improvements provided by the dynamic temporal predictor, we sample an average number of measurement points in first frame and only very few points in the next frame. While the results of STDI-ZO, STDI-FO and STDI-TO are quite similar, if the coverage of measurements is good throughout the image, the results of STDI-ZO are much worser compared to methods which use a temporal predictor, when only very few measurements exist. The resulting depth map which is computed by STDI-ZO tends to the mean of the learned principal components. Instead, STDI-FO and STDI-TO uses temporal information of the past which can be exploited to reproduce the depth map of the scene quite well despite the fact that only a very few measurements are available as input.

B. KITTI Depth Completion Dataset

Another important application of the proposed method, next to the densification of very sparse depth points of a visual SLAM component, is the completion of data points of a LIDAR scan. Those LIDAR sensors, which are currently used in many vehicles for autonomous driving, cannot reconstruct a dense depth map, which is needed for a full perception of the environment.

In the KITTI depth completion benchmark, depth values of missing locations have to be estimated, where the LIDAR scanner is not capable to retrieve information about the depth. A drawback of this dataset is that it does not consists of consecutive frames. Hence, we can only apply STDI-ZO because we cannot utilize temporal information with our proposed predictor, if the frames are not temporally correlated. Some examples of the interpolated depth map with STDI-ZO are shown in figure 4. The results of the first two rows of this figure are computed with $l = 500$, however, for the last two examples only $l = 50$ is used. Even, if there are only very few points on an object, for instance on the bottom left vehicle in the last example, our method is capable to reconstruct the depth of this car correctly.

Besides these quantitative results, we have also evaluated the performance of STDI-ZO at the depth completion challenge. Unfortunately, there are some restriction in our method, which does not allow us to evaluate on the test dataset. Our method is trained with the SGBM depth maps on the KITTI odometry training sequences, which cannot deliver depth values for the whole image due to stereo matching. Thus, we have to crop our basis to this resolution, where SGBM computes valid values. So, we are only able to estimate the depth values for this region. This prevents us to evaluate the performance of our method at the test dataset because KITTI assumes that you can estimate a depth value of all missing locations. Hence, we used the validation dataset to manually compute the error measures, which are shown in table III.

In comparison to NN+CNN [4] and SGDU [14], we do not use any additional image guidance. STDI-ZO interpolates the missing depth values only based on the raw LIDAR measurements. Furthermore, all these comparable methods except the regression approach, which is based on the work of Nadaraya [22] and Watson [23], use CNN techniques, which are computationally expensive. In terms of the inverse

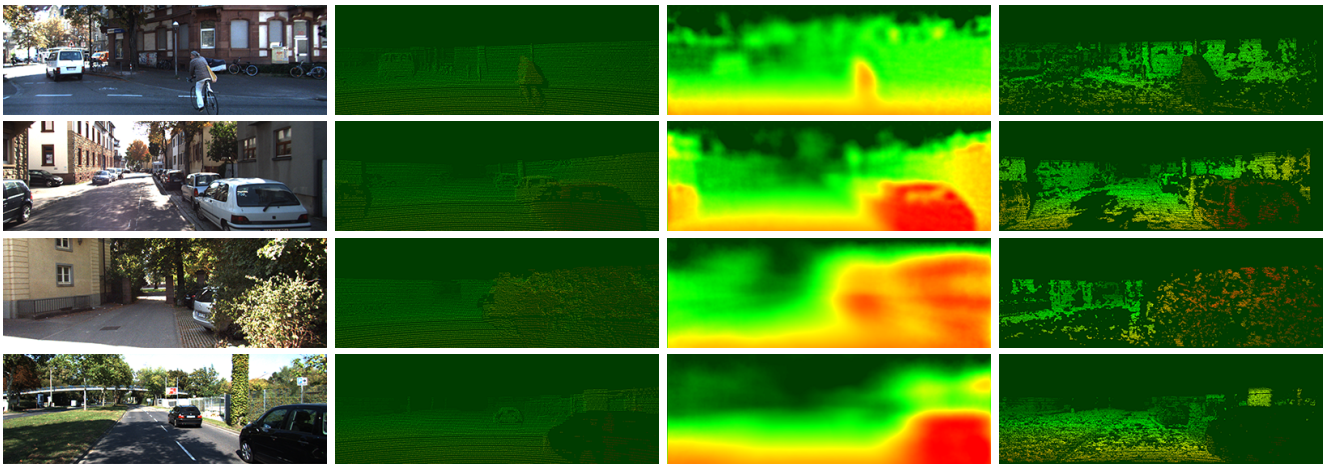

Fig. 4: This figure represents four examples of the KITTI depth completion validation dataset [4]. In the first column, the image of the scene is shown which are not used by our method. The task of this benchmark is to interpolate the missing locations of the sparse LIDAR measurements in the second column. The result of STDI-ZO using these sparse measurements is shown in the third column. Accumulated LIDAR points of consecutive frames, which serves as ground truth, are visualized in the last column. We use a basis size of $l = 500$ for the first two examples in row 1 and 2 and $l = 50$ for the last two examples in the third and fourth row.

TABLE III: Comparison to other depth completion methods on the KITTI depth completion benchmark [4] using LIDAR data (as of Jan 29th, 2018). Note: We could only evaluate our performance on the validation dataset because our trained PCA basis cannot reconstruct the whole depth map, which is needed for participating at test benchmark.

Method	**iRMSE** [1/km]	**iMAE** [1/km]	**RMSE** [mm]	**MAE** [mm]
NN+CNN [4]	3.25	1.29	1419	416
SparseConvs [4]	4.94	1.78	1601	481
Nadaraya [22], [23]	6.34	1.84	1852	416
SGDU [14]	7.38	2.05	2312	605
STDI-ZO* ($l = 50$)	9.21	5.36	36483	2142
STDI-ZO* ($l = 500$)	6.28	2.96	24775	1150

depth errors, we can even keep up with them. On the other hand, we score quite worse in terms of the depth error metrics. The reason for this could be that we have trained the principal components on the SGBM depth which do not necessarily coincide with the measured depth of LIDAR. For instance, the maximum resolvable depth of SGBM and LIDAR is certainly not the same.

V. Summary & Conclusion

In this work, we propose a novel non-guided interpolation method to estimate dense depth map from very sparsely measured depth values based on a learned PCA basis. The approach of [2] used only this spatial information of the PCA to compute the dense depth map. We extend this pure spatial PCA interpolation method by a temporal predictor, which utilizes temporal knowledge of previous frames. This temporal information is incorporated by a dynamic prior term into our maximum a posteriori approach, which yields significant better result on the KITTI odometry dataset.

The depth completion of sparse LIDAR data is another application field, where our proposed method can be applied. In the KITTI depth completion benchmark where we cannot exploit temporal information, STDI achieves in a non-temporal mode and as a non-guided approach competitive results to approaches, which uses computational expensive deep learning techniques. In general LIDAR data exhibits temporal dependencies in reality, thus the enhancement of a dynamic prior term can be also utilized in depth completion task of LIDAR data as well.

References

[1] J. Wulff and M. J. Black, "Efficient sparse-to-dense optical flow estimation using a learned basis and layers," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 120–130.

[2] M. Ochs, H. Bradler, and R. Mester, "Learning Rank Reduced Interpolation with Principal Component Analysis," in *Intelligent Vehicles Symposium (IV)*, 2017, pp. 1126–1133.

[3] H. Bradler, B. Wiegand, and R. Mester, "The Statistics of Driving Sequences - and what we can learn from them," in *International Conference on Computer Vision - Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving (ICCV-W CVRSUAD)*, 2015.

[4] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity Invariant CNNs," in *International Conference on 3D Vision (3DV)*, 2017.

[5] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image Super-Resolution Via Sparse Representation," *Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[6] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-Depth Super Resolution for Range Images," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[7] J. T. Barron and B. Poole, "The Fast Bilateral Solver," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 617–632.

[8] M. Horncek, C. Rhemann, M. Gelautz, and C. Rother, "Depth Super Resolution by Rigid Body Self-Similarity in 3D," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1123–1130.

[9] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 1623–1630.

[10] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H. Bischof, "Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 993–1000.

[11] X. Song, Y. Dai, and X. Qin, "Deep Depth Super-Resolution: Learning Depth Super-Resolution Using Deep Convolutional Neural Network," in *Asian Conference on Computer Vision (ACCV)*, 2017, pp. 360–376.

[12] T.-W. Hui, C. C. Loy, and X. Tang, "Depth Map Super-Resolution by Deep Multi-Scale Guidance," in *European Conference on Computer Vision (ECCV)*, 2016.

[13] G. Riegler, M. Rüther, and H. Bischof, "ATGV-Net: Accurate Depth Super-Resolution," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 268–284.

[14] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, "Semantically Guided Depth Upsampling," in *German Conference on Pattern Recognition (GCPR)*, 2016, pp. 37–48.

[15] V. Jampani, M. Kiefel, and P. V. Gehler, "Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4452–4461.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354 – 3361.

[17] H. Hirschmueller, "Stereo processing by semiglobal matching and mutual information," *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 2, pp. 328–341, 2008.

[18] N. Fanani, A. Stuerck, M. Ochs, H. Bradler, and R. Mester, "Predictive monocular odometry (PMO): What is possible without RANSAC and multiframe bundle adjustment?" *Image and Vision Computing*, vol. 68, pp. 3–13, 2017.

[19] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2017.

[20] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *Transactions on Robotics*, vol. 33, no. 2, pp. 249 – 265, 2017.

[21] J. Shi and C. Tomasi, "Good Features to Track," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593 – 600.

[22] E. A. Nadaraya, "On Estimating Regression," *Theory of Probability and Its Application*, vol. 9, pp. 141–142, 1964.

[23] G. S. Watson, "Smooth Regression Analysis," *Sankhy: The Indian Journal of Statistics*, vol. 26, pp. 359–372, 1964.