Supplementary material of ”Semi-supervised Regression using Hessian Energy with an Application to Semi-supervised Dimensionality Reduction”

Kwang In Kim[1]**, Florian Steinke**[2,3]**, and Matthias Hein**[1]
[1]Department of Computer Science, Saarland University Saarbrücken, Germany
[2]Siemens AG Corporate Technology Munich, Germany, [3]MPI for Biological Cybernetics, Germany
{kimki,hein}@cs.uni-sb.de, Florian.Steinke@siemens.com

1 Additional experimental results

Figures 1–4 show additional results of the proposed Hessian energy-based regression method.

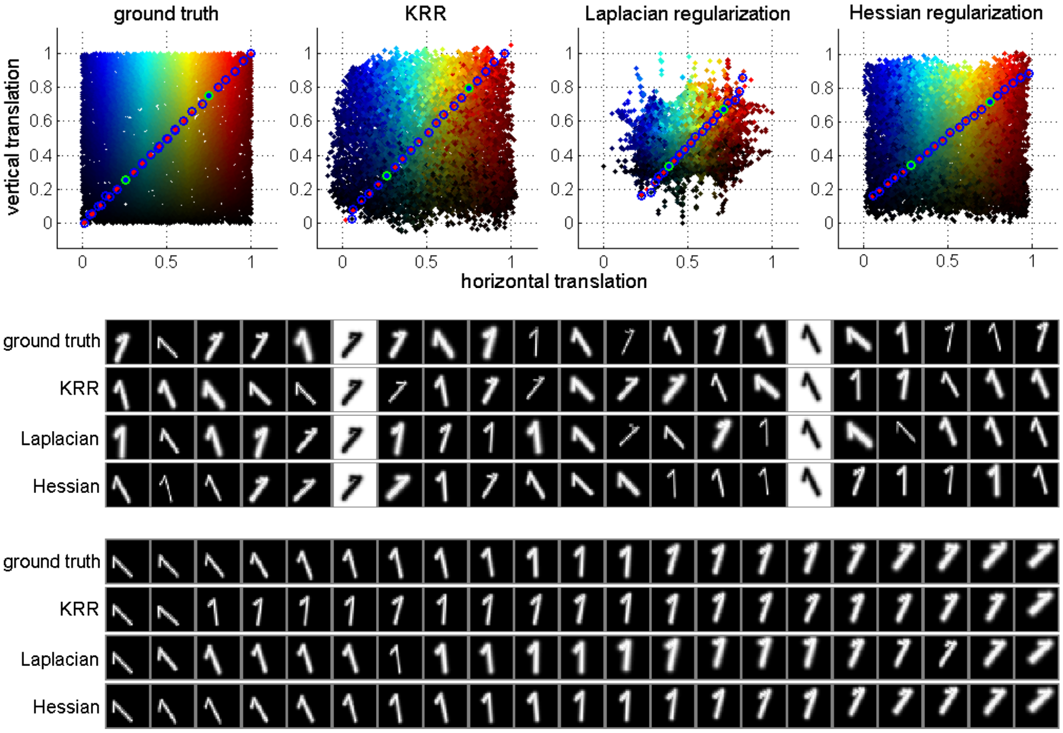

Figure 1: Results of regression on the artificial digit 1 dataset. First row: the two-dimensional embedding of the digits obtained by regression for the horizontal and vertical translation with 100 labels. Second row: 21 digit images sampled at regular intervals in the estimated translation parameter spaces. Please refer to the corresponding figure in the paper. Third row: 21 digit images sampled at regular intervals on the line between (0,0,0,0) and (1,1,1,1) in the four dimensional parameter space (horizontal and vertical translations, thickness variation, and rotation).

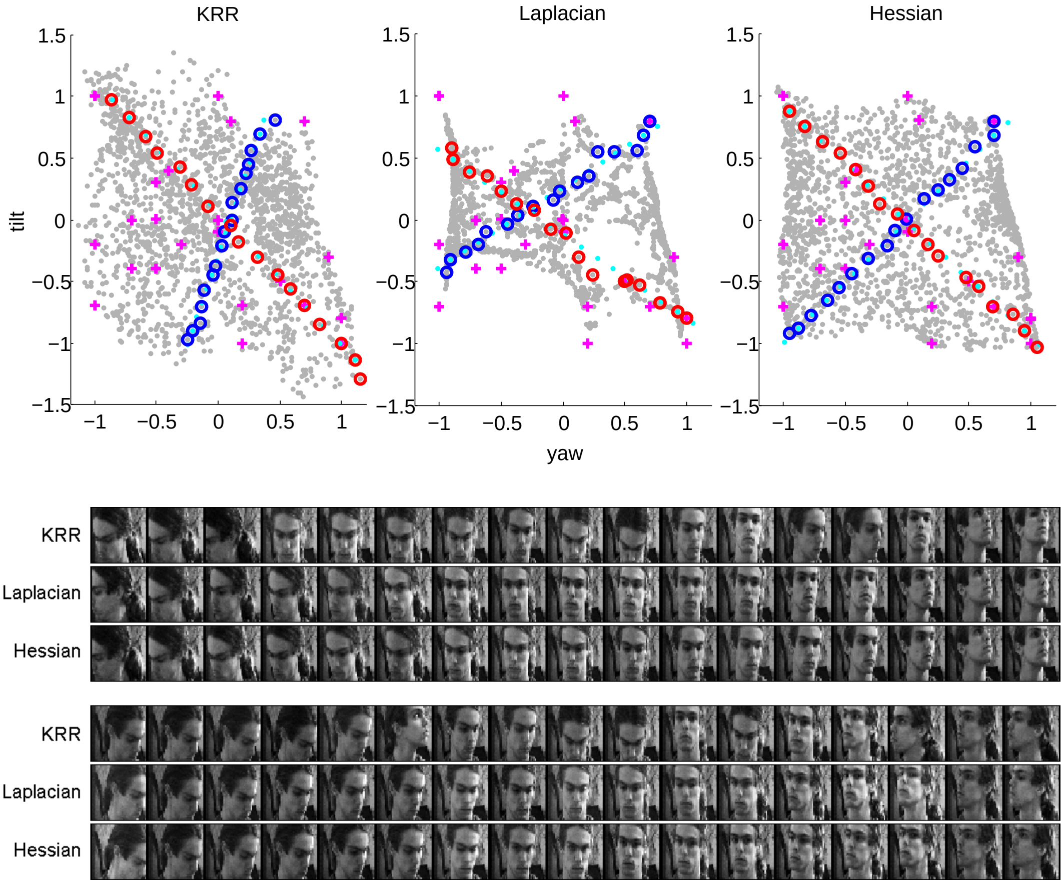

Figure 2: Regression of face images into the parameter space corresponding to tilt and yaw of the faces. The database contains 2000 images of size 40×40 of faces looking in different directions (the data was kindly provided by Jakob J. Verbeek and has been used in [13]). Before regression the dimensionality of the data is reduced using PCA. Twenty-five images were randomly sampled and manually labeled (tilt and yaw). Note, that we do not have ground truth for this dataset and the labels might be noisy as it is not an easy task to manually estimate the true parameters based on a single view. First row: the two-dimensional embedding of the faces for KRR and SSR with Laplacian and Hessian regularization. All parameters including number of PCA components are chosen by 5-fold cross-validation. Labeled points are marked with plus sign. Second row: two sets of 17 images sampled at regular intervals in the estimated parameter spaces, marked as blue circles and red circles, respectively in the first row. Images for the sequences were found in the same way as for the digit dataset. The reference points generating the line in the parameter space correspond to the second and second last images in the row.

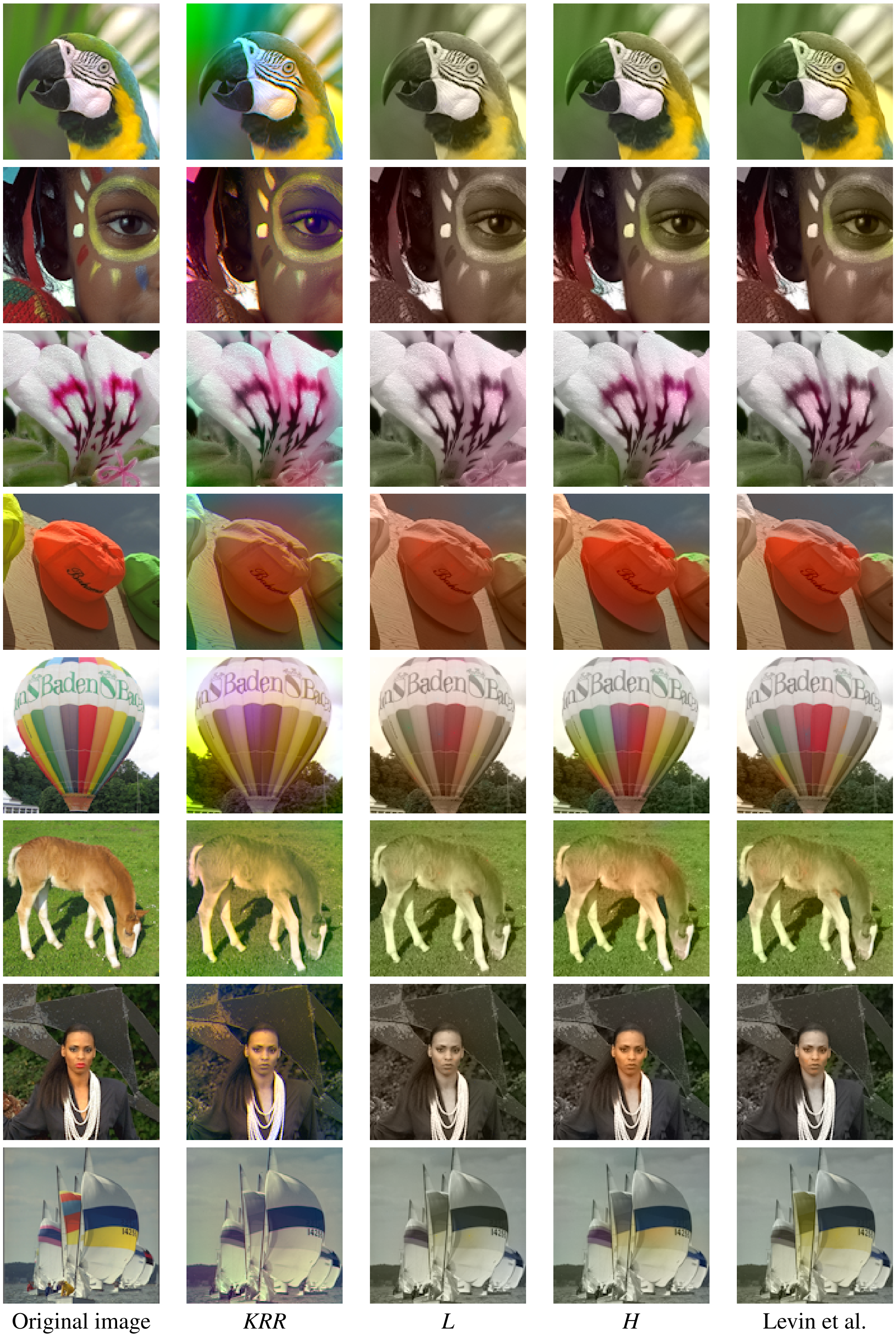

Figure 3: More example of image colorization with 30 labels. See main paper for details.

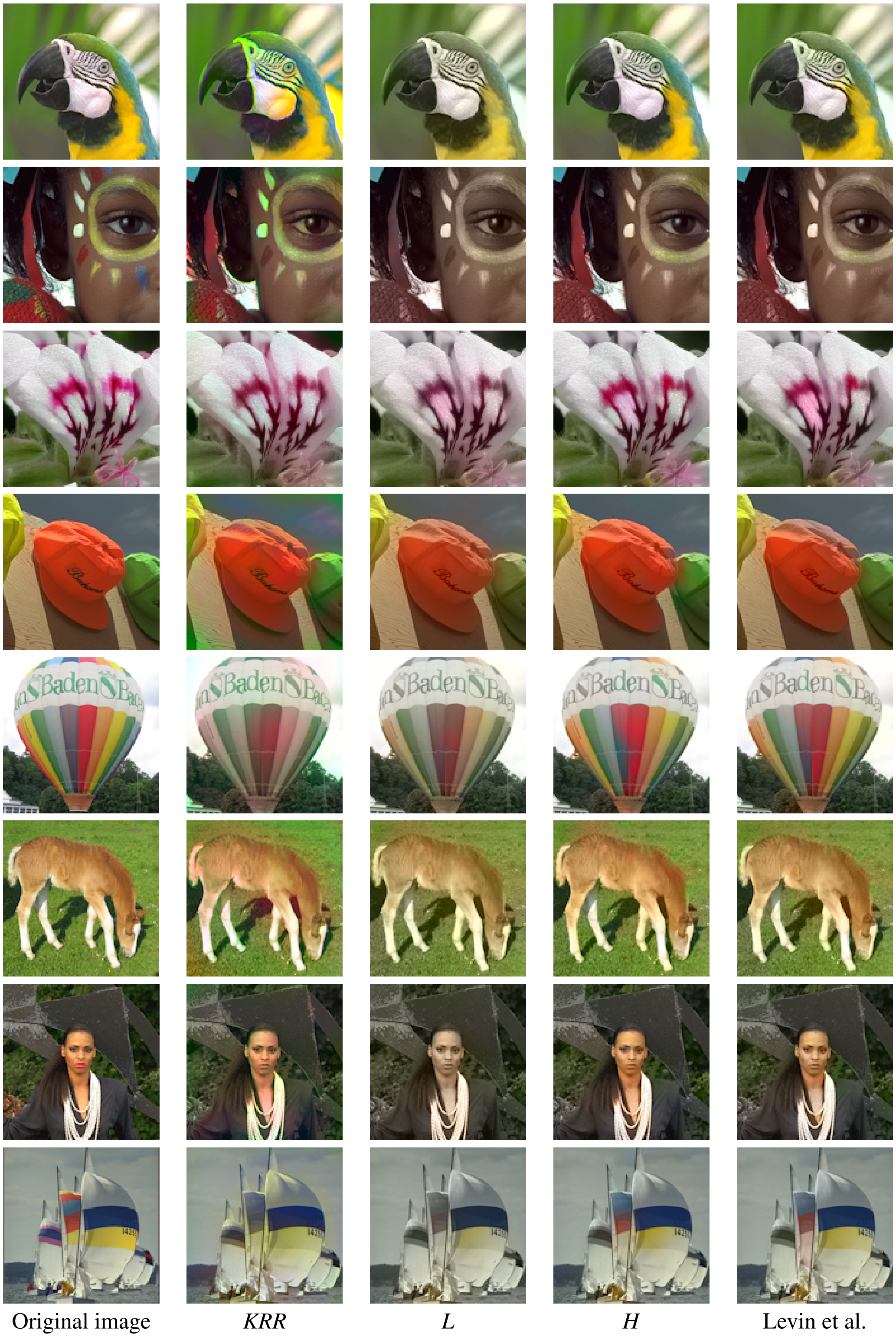

Figure 4: Example of image colorization with 100 labels. See main paper for details.

2 Ruling out unstable behaviour of Hessian regularizer using an additional regularizer

While fixing the zeroth-order terms of the local fitting polynomials worked in all the experiments, it does not completely rule out the possibility of producing heavy oscillations. Actually, we were able to construct an example (unrealistically pessimistic though) which show such a behavior. One approach for this problem is to introduce a second regularizer (or a stabilizer) $S_2(f)$ which penalizes the deviation of f from the value of the local polynomial $p^{(i)}$ not only at the point of approximation X_i (which was done by fixing the zeroth-order term of $p^{(i)}$ at $f(X_i)$), but also in the neighborhood of it.

$$S_2(f) \quad = \quad \sum_{i=1}^{n} \left(\frac{1}{k} \sum_{j=1}^{k} \left(p^{(i)}(X_j) - f(X_j) \right)^2 W_j^i \right),$$

where $W_j^i = k(\|X_i - X_j\|^2, \sigma_i^2)$, $\sigma_i = \max_{j=1}^{k}(\|X_i - X_j\|^2)$, and

$$k(d^2, \sigma^2) = \begin{cases} \exp\left(\frac{\sigma^2}{d^2 - \sigma^2}\right) & \text{if } d^2 < \sigma^2 \\ 0 & \text{otherwise.} \end{cases}$$

Here, we used a slight abuse of notation such that X_j actually denotes the j-th element of $N_k(X_i)$.

This regularizer enforces a certain smoothing of the function values and particularly kicks in when due to underfitting, $p^{(i)}$ is not equal to f in the local neighbor hood of X_i. Note, that S_2 does not penalize constant or "geodesic" functions. Moreover, it is mainly influential in the estimation of Hessian when the function is very irregular. Since we are anyway not interested to fit such functions, the additional regularizer does not change significantly the original objective.

Since locally

$$\frac{1}{k} \sum_{j=1}^{k} \left(p^{(i)}(X_j) - f(X_j) \right)^2 W_j^i \quad = \quad \sum_{\alpha \in N_k(X_i)} \sum_{\beta \in N_k(X_i)} \mathbf{f}_\alpha \mathbf{f}_\beta ({B_S}^{(i)})_{\alpha\beta},$$

where

$$B_S^{(i)} = \frac{1}{k} (\Phi_i \Phi_i^+ - I)^\top \mathrm{diag}(W^i) (\Phi_i \Phi_i^+ - I)$$

by summing up the contributions from all points we can construct a matrix B_S which has the same sparsity structure as B and $S_2(f) = \langle f, B_S f \rangle$:

$$S_2(f) = \sum_{i=1}^{n} \sum_{\alpha \in N_k(X_i)} \sum_{\beta \in N_k(X_i)} \mathbf{f}_\alpha \mathbf{f}_\beta ({B_S}^{(i)})_{\alpha\beta} = \langle f, B_S f \rangle .$$

The regression is then performed by solving

$$\operatorname*{arg\,min}_{f \in \mathbb{R}^n} \; \frac{1}{l} \sum_{i=1}^{l} (Y_i - f(X_i))^2 + \lambda_1 \langle f, Bf \rangle + \lambda_2 \langle f, B_S f \rangle ,$$

or equivalently, by solving the following sparse linear system,

$$(\mathbb{I}' + l\,\lambda_1 B + l\,\lambda_2 B_S) f = Y.$$

Figure 5 and table 1 show the performance of this new system (denoted as $H2$) where λ_2, as an additional parameter, was optimized based on cross-validation. Overall, the performance of this system was slightly better than the one proposed in the main paper. However, it introduced an additional parameter to be tunned. For practical applications, one could choose either one of them by trading the computational complexity off with the performance.

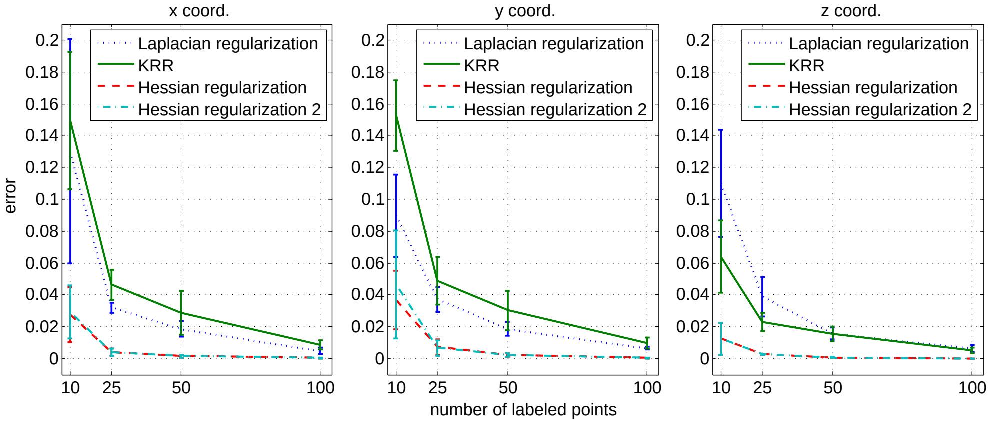

Figure 5: Error plots for regression on the figure dataset.

Table 1: Results on digits: mean squared error (standard deviation) (both in units 10^{-3}).

	50 labeled points				100 labeled points			
	h-trans.	v-trans.	rotation	thickness	h-trans.	v-trans.	rotation	thickness
K	0.78(0.13)	0.85(0.14)	45.49(7.20)	0.02(0.01)	0.39(0.10)	0.48(0.08)	26.02(2.98)	0.01(0.00)
L	2.41(0.26)	3.91(0.59)	64.56(3.90)	0.39(0.02)	1.17(0.13)	2.20(0.22)	30.73(6.05)	0.34(0.01)
H	0.34(0.03)	0.88(0.07)	4.03(1.15)	0.15(0.02)	0.16(0.03)	0.39(0.07)	1.48(0.26)	0.06(0.01)
$H2$	0.33(0.04)	0.84(0.11)	3.98(1.14)	0.11(0.02)	0.16(0.03)	0.39(0.07)	1.43(0.28)	0.04(0.01)