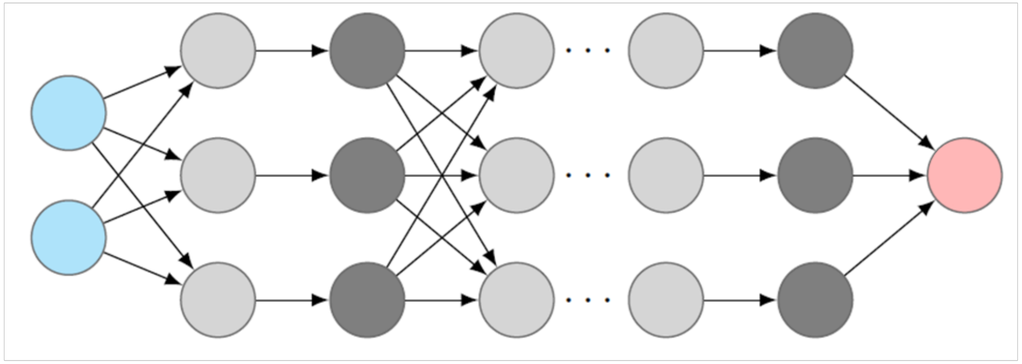
Advanced network architectures with prior knowledge

- From variational methods to machine learning -

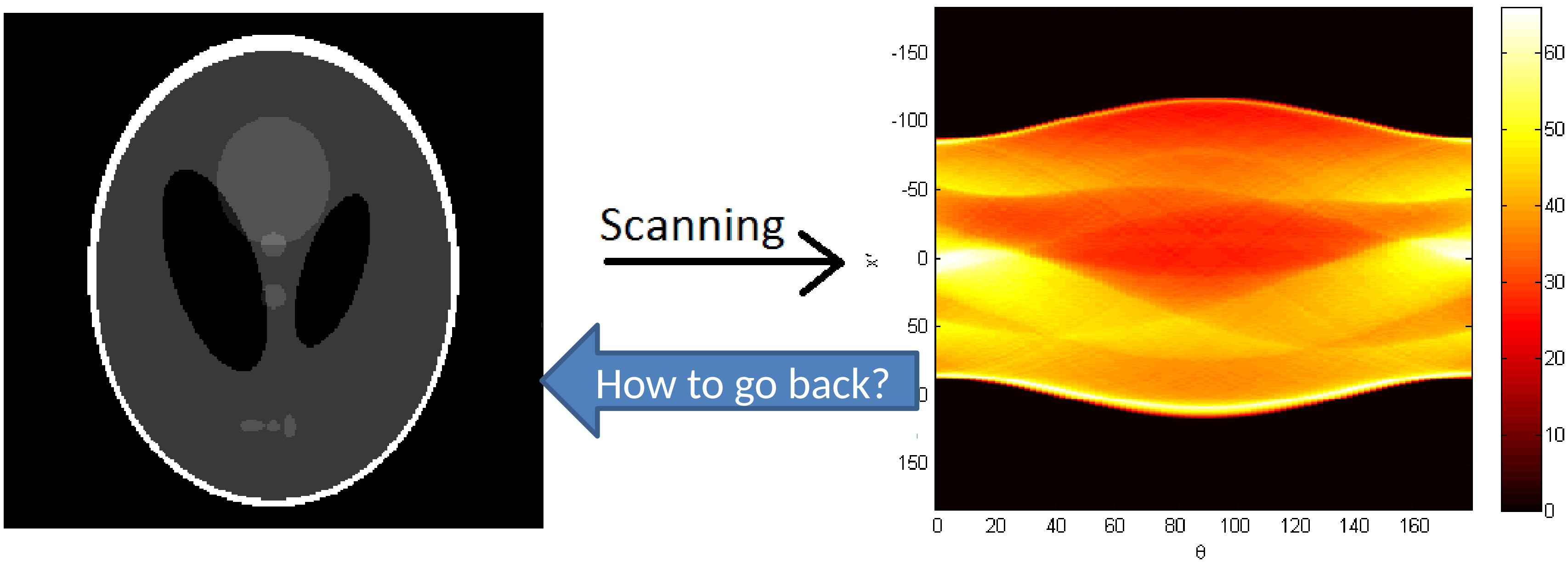
Lecturer: Michael Möller – michael.moeller@uni-siegen.de
Exercises: Hartmut Bauermeister – hartmut.bauermeister@uni-siegen.de



Inverse problems

In many applications the desired quantity cannot be observed directly, but we have a thorough understanding of what exactly the relation is.

Example 1: CT reconstruction



Inverse problems

In many applications the desired quantity cannot be observed directly, but we have a thorough understanding of what exactly the relation is.

Simpler example: Removing a known blur from a photograph

True image

Recorded image

Certain problems cannot be solved directly

We might know the relation between the data and the unknown,

e.g. $$\begin{array}{ccccccccc} \text{blurry image} & = & \text{convolution kernel} & * & \text{sharp image} & + & \text{noise} \\ f & = & k & * & u & + & n \end{array}$$

Why do we not just solve a least squares problem for the unknown?

$$\hat{u} = \arg\min_u \; \|k * u - f\|_2^2$$

Possible issue 1: The solution might not be unique,

e.g. recover a and b from c = a + b.

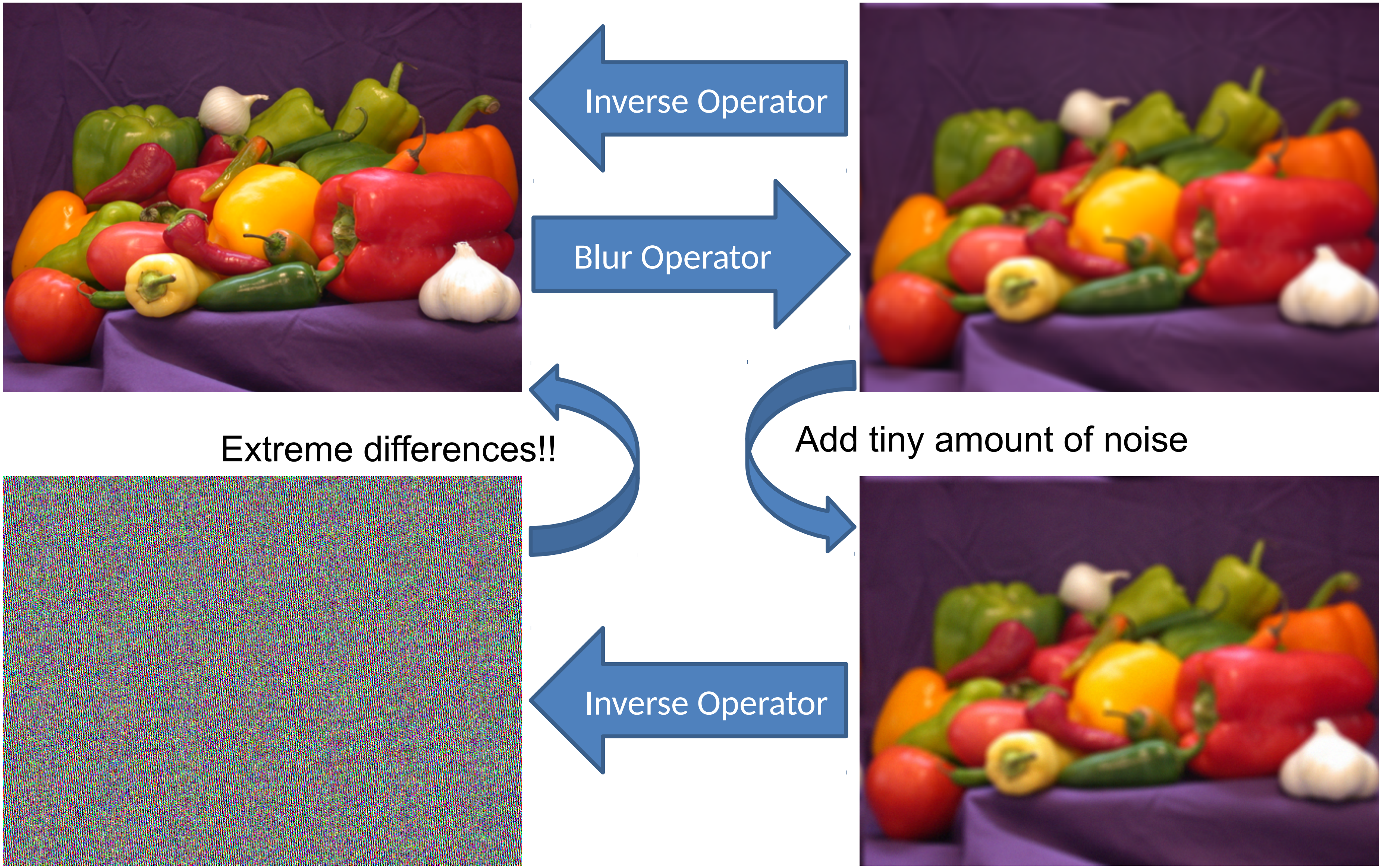
Possible issue 2: The solution might not be depend on the data

continuously!

Remark: The two issues above are reasons to call the underlying problem ***ill-posed****!*

Michael Möller – michael.moeller@uni-siegen.de

Lacking continuous dependence on the data



Michael Möller – michael.moeller@uni-siegen.de

Regularization

How can we modify a least squares problem

$$\hat{u} = \arg\min_u \; \|k * u - f\|_2^2$$

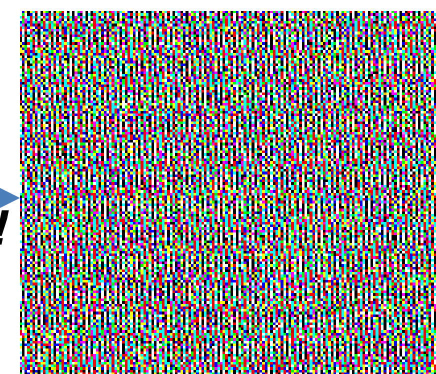
to fight the ill-posedness?

Use *regularization*! Not only minimize a least squares problem, but encourage an image that is not too oscillatory, e.g.

$$\hat{u} = \arg\min_u \; \|k * u - f\|_2^2 + \alpha \|\nabla u\|_1$$

for a parameter α that controls the trade-off between *fidelity* and *regularity*!

very expensive!



Variational Methods

Energy minimization methods

candidate image

$$\hat{u} = \arg\min_u \; E(u)$$

Small if u is has desirable properties

Large, otherwise

Data fidelity term *Regularization*

$$= \arg\min_u \; H_f(u) + R(u)$$

Often known from physics, e.g. the radon transform for CT reconstruction

Often difficult! Common energy minimization approaches for instance use $TV(u) = \|\nabla u\|_1$

More details to be discussed in the lecture!

Why not plain machine learning?

If we know how the data is formed, what type of noise to expect and have a data base of representative clean images, why not use machine learning?

E.g. simulate lots of noisy, blurry images from the clean ones and train a network on inverting this process.

Not a bad idea in general, but

- one uses the data formation process to simulate data from which one partly learns things that are known a-priori already,
- one might have to retrain the network as soon as the data formation process, the type or amount of noise, or the class of representative images change.

Learning vs. Modelling

Energy minimization methods

Data fidelity term — *Regularization*

$$\hat{u} = \arg\min_u \boxed{H_f(u)} + \boxed{R(u)}$$

Describe how the data is formed — *Describe 'natural' or 'typical' images*

Pro: Active control over how the data is formed and what type of noise to expect

Con: Good regularizers are difficult to design.

Machine learning methods

- Simulate 100,000 pairs of u and f with the data formation process and expected noise level
- Learn how to map f to u

Pro: Powerful!

Con: Costly training. Does not explicitly use prior knowledge about data formation.

Idea: Create hybrid methods between energy minimization and machine learning!

Common approaches

1. Learning a regularization

Step 1, parameterize a regularization, e.g. set $R(u;\theta) = \|\text{vec}(\ \text{conv}(u;\theta)\ )\|_1$

Determine the parameters θ in a reasonable way, e.g. such that

- $\text{vec}(\ \text{conv}(u;\theta)\ )$ has as many zero entries as possible for natural image u. This leads to the theory of *compressed sensing*, see e.g. the corresponding lecture by Prof. Loffeld.

- Or, on a set of training pairs $(f_i, \hat{u}_i)$, optimize

$$\min_\theta \sum_{i \in \text{training examples}} \|u_i(\theta) - \hat{u}_i\|^2$$

subject to

$$u_i(\theta) = \arg\min_u H_{f_i}(u) + R(u;\theta)$$

Such approaches are called *bilevel optimization problems* or *structured inference*.

Common approaches

2. Network architectures motivated by energy minimization

For example, if one modifies $\min_u H_f(u) + R(u)$ to

$$\min_{u,z} H_f(u) + R(z) + \frac{\lambda}{2}\|u - z\|^2$$

and considers an alternating minimization

$$z^{k+1} = \sout{\arg\min_u R(z) + \frac{\lambda}{2}\|z - u^k\|^2} = \mathcal{G}(u^k;\theta)$$

← Small parameterized network

$$u^{k+1} = \arg\min H_f(u) + \frac{\lambda}{2}\|u - z^{k+1}\|^2$$

Now roll out the algorithmic scheme for a fixed number of iterations.
Possibly use approximations for the u update, e.g.

$$u^{k+1} = z^{k+1} - \frac{1}{\lambda}\nabla H_f(z^{k+1})$$

Then consider the overall algorithm as a network.

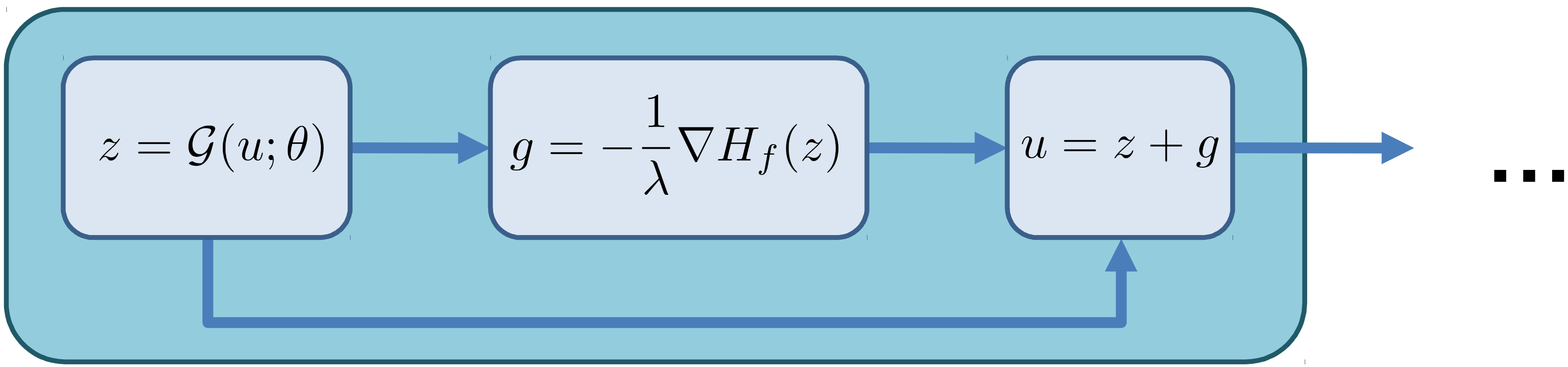
Common approaches

Let us write the following updates as a network:

$$z^{k+1} = \mathcal{G}(u^k; \theta)$$

$$u^{k+1} = z^{k+1} - \frac{1}{\lambda} H_f(z^{k+1})$$

Let us pick a starting point, e.g. u = 0



Keep repeating this block X-times!

Such an architecture could be interpreted as / called *recurrent neural network*.

Instead, it is also quite common to allow a different $\mathcal{G}(u; \theta)$ (or at least different θ) in each block.

Some exemplary literature

Learning a regularization

- Fields of experts model by Roth and Black (2005)
- K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, Aharon, Elad and Bruckstein (2006)
- Analysis operator learning by Chen, Ranftl, and Pock (2014)

Learning an algorithm

- Shrinkage Fields by Schmidt and Roth (2014)
- From a PDE perspective: On learning optimized reaction diffusion processes for effective image restoration by Chen, Yu, and Pock (2015)

Extensions that decouple the learned regularity from the data formation process

- Learning proximal operators: Using denoising networks for regularizing inverse imaging problems by Meinhardt, Moeller, Hazirbas, Cremers (2017).
- One network to solve them all – solving linear inverse problems using deep projection models by Chang et al. (2017).
- Learning deep CNN denoiser prior for image restoration by Zhang, Zuo, Gu, Zang (2017).

Generalizing today's ideas

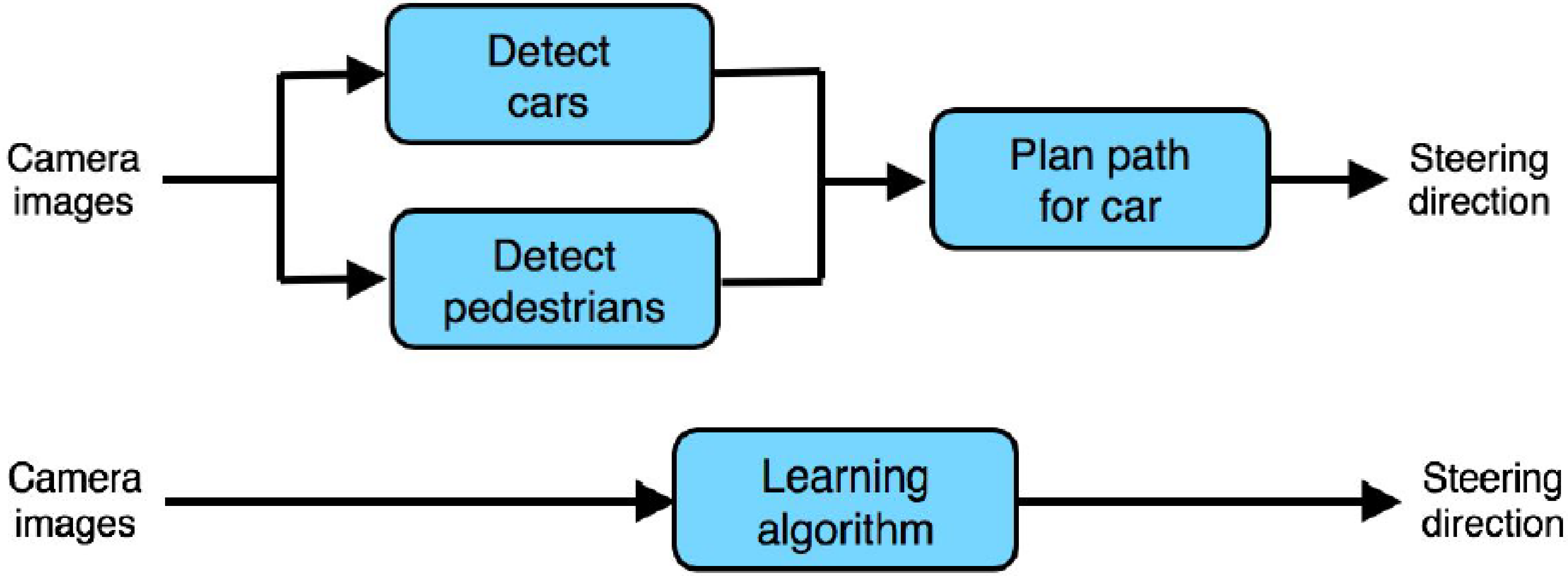
If you have an algorithm that already works quite well, incorporates prior knowledge about your problem at hand, but should be improved with machine learning, try to develop network architectures along the design of your existing algorithm. Parameterize only those parts that are difficult to model!

Any type of prior knowledge you can incorporate, will potentially reduce the amount of training data you need!

Interesting discussion in the draft of ***machine learning yearning*** by ***Andrew Ng***:

Option 1:

Option 2:



For getting machine learning solutions to work, I highly recommend to read Andrew's book!

Michael Möller – michael.moeller@uni-siegen.de

Take home message

Use machine learning only for those parts that are unknown and difficult to model faithfully!

Do not use a sledgehammer
for cracking a nut!

Michael Möller – michael.moeller@uni-siegen.de