Learning Generative Models and the Inference Process in Low Level Vision

Thesis submitted for the degree of

"Doctor of Philosophy"

by

Dan Rosenbaum

Submitted to the Senate of the Hebrew University of Jerusalem

August 2016

This work was carried out under the supervision of
Prof. Yair Weiss

Acknowledgments

First I would like to thank my advisor Yair Weiss. I have learned a lot from Yair about research in general and particularly about the balance between having a long-term vision and making rigorous step-by-step progress. Our discussions were always fascinating to me and I feel very lucky to have had the chance to work with him. I am also grateful to the members of my thesis committee, Michal Irani and Amir Globerson for their time and help.

I would like to thank all my friends in the learning and vision lab for the good company in the last years. Special thanks to Daniel who offered me a lot of help with my research, Elad and Elad for all the corridor discussions, and to all my roommates: Cobi, Alon, Nir, Yossi, Yoav, Maya, and for the last two years, Ofer. Thank you all for making it fun to come to the lab.

I would also like to thank my parents, Stella and Nicolas, for all their support along the way.
Most importantly, I would like to send my thanks and love to Noa, Yuval and Uri. Thank you for always reminding me what are the important things in life.

Abstract

In low level vision problems such as image restoration and optical flow estimation the desired output is an image consisting of high dimensional data arranged in a two dimensional structure. Most of those problems are ill-posed, meaning that additional assumptions on the structure of data need to be incorporated in methods that solve them. The high dimension and rich structure of images make it hard to develop handcrafted methods, and call for a data-driven approach where all assumptions are directly learned from data. It is therefore natural to use machine learning for low level vision, allowing the structure of images to be automatically discovered and exploited. However the use of machine learning in low level vision is still not delivering the anticipated improvements. Although for easier problems such as image restoration it has outperformed the handcrafted methods, for harder problems like optical flow estimation the performance of both handcrafted methods and data-driven methods is still not satisfactory.

Machine learning can be used in different ways. In the generative approach, the assumptions about the structure of data are formulated as probabilistic models and learned from data. The estimation is then cast as an inference problem using Bayes' rule. In the discriminative approach, a predictor is learned directly for the estimation problem, avoiding the need to perform an inference process at test time. The advantage of generative learning is that sometimes it makes it more natural to incorporate prior knowledge about the structure of the problem, allowing faster training and a modular usage of the predictor where different components can be changed at test time. The advantage of the discriminative approach is that it usually results in a faster predictor. This is because while the inference of a generative model typically involves a hard optimization problem at test time, in the discriminative approach all the optimization is performed in advance, by finding the best predictor for a given architecture and running time constraints.

In the work presented here, we demonstrate the different ways in which machine learning can be used for low level vision problems. By taking a more general view on the generative approach, it can be divided to three components: (1) the prior which models the structure of the hidden data we want to estimate, (2) the likelihood which models the generation of the observed input given the hidden data, and (3) the inference process which uses the prior and likelihood to estimate the hidden data.

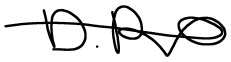
Our results are presented in four papers. In the first and second papers we show how the assumptions made by different handcrafted optical flow methods can be extracted and formulated as probabilistic models of the prior and likelihood. We then evaluate the different models and show how they can be improved by learning them directly from ground-truth data. In the third paper we show how the inference process can be learned from data for image restoration. We show that this results in a predictor that has the advantages of both the discriminative and generative approaches by being fast at test time while retaining the modularity property that allows the same predictor to be used for different tasks. The forth paper deals with the problem of enhancing the depth map output of RGBD cameras. We show that by evaluating and learning prior models on ground-truth data we can improve the state-of-the-art in depth enhancement.

August 2016

Declaration of Author Contribution

I declare that all the research presented in my thesis was performed by me under the guidance of my supervisor, Yair Weiss.
The thesis consists of four papers in which I am the main author.
All papers appear under the name of my supervisor as well.
One of the papers, “Learning the local statistics of optical flow” (chapter 2.1), appears under an additional author, Daniel Zoran, who gave me additional guidance on training and evaluation methods for the models used in the paper.

Dan Rosenbaum



Yair Weiss

Contents

1 Introduction **1**

1.1 Low level vision . . . 1

1.1.1 Image restoration . . . 2

1.1.2 Optical flow estimation . . . 3

1.1.3 Depth enhancement . . . 4

1.2 The generative learning approach . . . 5

1.2.1 Image priors . . . 7

1.2.2 Patch priors . . . 7

1.3 The discriminative learning approach . . . 8

1.3.1 Partial discriminative methods . . . 8

1.3.2 Generative vs. discriminative . . . 9

1.4 A generalized approach to generative learning . . . 10

1.4.1 Learning the inference . . . 11

1.5 Interim summary . . . 12

2 Results **13**

2.1 Learning the local statistics of optical flow . . . 14

2.2 Beyond brightness constancy: learning noise models for optical flow . . . 24

2.3 The return of the gating network: combining generative models and discriminative training in natural image priors . . . 34

2.4 Statistics of RGBD images . . . 44

3 Discussion **57**

3.1 Future work . . . 58

Chapter 1

Introduction

1.1 Low level vision

Low level vision is a set of vision problems including image restoration, depth estimation, optical flow estimation, and more. In such problems the goal is more related to the structure of the image rather than to its semantic content, and the resulting output is in itself an image, consisting of high dimensional data arranged in a two dimensional structure.

We start by describing three different low level vision problems, and methods to solve them. The methods we present in this section are handcrafted, i.e. they were engineered by computer vision researchers rather than learned from data. Low level vision is usually an ill-posed problem, which means that some prior assumptions on the output should be incorporated in the solution method. Typical assumptions are the self-similarity of local patches within the output image, or the smoothness of the output. Such assumptions are sometimes formulated as penalties in an energy function that is being minimized.

Since low level vision has been a topic of much research in the last four decades, many good methods have been developed. While handcrafted methods have achieved a good level for some of the easier problems such as image denoising, for more challenging problems like optical flow estimation the results are still far from being satisfactory. After reviewing some examples of handcrafted methods, in the following sections we present different methods based on machine learning approaches. In recent years, machine learning and data-driven methods have been able to overcome many challenges in computer vision. A prime example is the success in classifying objects in an image using deep neural networks that were trained discriminatively over a huge amount of labeled data [27]. This success of machine learning is still not replicated for low level vision problems. As we will see, although machine learning approaches have achieved state-of-the-art performance for image restoration, for harder problems it is still lacking in performance.

In our work we aim to get a better understanding of the role of machine learning in problems with high dimensional output and rich structure like in low level vision. The hope is that understanding the different ways in which machine learning can be used, will ultimately lead to improved methods to solve low level vision problems.

Figure 1.1: Image restoration. The original image (left), and examples of different types of corruption in the input image: noise, blur, and missing parts.

1.1.1 Image restoration

Perhaps the most basic low level vision problem is image restoration, where one needs to recover the original image given a corrupted version of it. The corruption can be of different kinds including additive noise, blur and missing parts of the image (figure 1.1). Although usually the corruption model is unknown (sometimes called "blind" image restoration), here we focus on the easier problem and assume that it is given ("non-blind" image restoration). In what follows we describe some methods for image denoising, and image deblurring.

Image denoising: One way to remove the noise from noisy images is using the coring approach [39, 21]. This approach takes a wavelet transform of the given image, and zeros out the wavelet coefficients that are smaller than some threshold. A more recent method that is very successful and popular is BM3D [9]. The method is based on the assumption that small patches in the clean image appear several times within the image. It works by forming three dimensional blocks consisting of similar patches in the image, and then running a filter on the blocks in a manner that cancels noise and preserves the signal, which is the original image. It is an extension of the non-local means method [6] which simply averages similar patches.

Image deblurring: Methods for image deblurring typically work by formulating and minimizing an energy function [38, 18, 14, 28, 26]. The energy function contains a data term and a smoothness term such as the following:

$$J(x) = \rho_d \left(b(x) - y\right) + \lambda \sum_{ij \in adj} \rho_s(x_i - x_j) \tag{1.1}$$

The data term encourages the output image x to be such that under the given blur process $b(x)$ would be similar to the input blurry image y. The blur process is sometimes formulated as a convolution with a blur kernel, i.e. $b(x) = Kx$ where the matrix K is a 2D convolution matrix based on the blur kernel. The smoothness term serves as a regularization of this ill-posed problem, and encourages adjacent pixels in x to be similar under some measure.

Figure 1.2: Optical Flow estimation. Using the MPI-Sintel dataset of synthetic images [8] allows for high quality ground-truth (middle). Each of The top figures shows two consecutive frames overlayed on top of each other. The estimated results (bottom) were computed using EpicFlow [36], which first finds sparse matches of image patches and uses them to initialize a global energy function minimization.

1.1.2 Optical flow estimation

In optical flow estimation we seek to find a flow field, consisting of the two dimensional motion of every pixel between one frame to another in a video sequence. Figure 1.2 shows some examples of images and flow fields.

A classic solution to the problem was introduced by Horn and Schunck in 1981 [22], and is based on minimizing the following energy function:

$$J(v) = \|I_1 - I_2^{\rightarrow v 1}\|_2^2 + \lambda \sum_{ij \in adj} \|v_i - v_j\|_2^2 \tag{1.2}$$

where I_1 is the first image, v is the flow field containing the motion of every pixel in the image, and $I_2^{\rightarrow v 1}$ is the second image warped back to the time of I_1 according to the proposed motion v. The first term in the energy function is a data term that formulates the "brightness constancy" assumption, i.e. that when following each pixel according to its motion from the first image to the second, we should expect to see the same color or brightness. The second term in the energy function is a smoothness term which serves as a regularization of this ill-posed problem, stating that all adjacent pixels should have similar motion.

Since both terms in the Horn and Schunck energy function are measured by a quadratic penalty, it is not robust to outliers, and thus encourages overly smooth solutions. In order to cope with this problem and allow sudden sharp changes that typically occur in the boundary between objects with different motion (see figure 1.2), Black and Anandan [3] proposed to replace the quadratic penalty with a more robust one such as the absolute value.

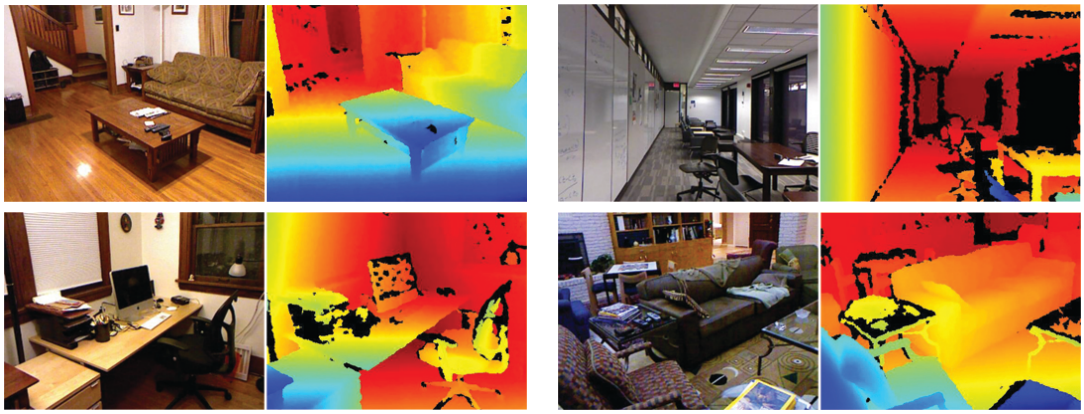


Figure 1.3: Depth Enhancement. Data from the NYU Depth dataset [33] collected using the Kinect RGBD camera. The depth channel (right) is usually of lower quality than the RGB channels (left), because of noise and missing data (black regions).

One drawback of the energy minimization approach is that the global minimization can be a very hard problem. This is true specially because the warping function is highly non-convex. Typically the optimization is performed using Gauss-Newton and coarse-to-fine iterations, but the resulting motion is still only an approximate solution to the energy function.

In order to avoid the difficulty of finding global solutions, other optical flow methods are based on local matching of image patches. One of the earliest methods that was proposed by Lucas and Kanade [31] simply searches for local matches that minimize the sum of square distance of all the pixels in a patch, and then averages the results for all overlapping patches (or takes only the middle pixel from every patch). More recent methods use the results of local matches to initialize a global energy minimization, and often use more sophisticated features for the matching such as SIFT [30] and others [5, 45, 36].

Although much progress has been made in the last decades, the results are still far from being satisfactory. Many methods still rely on minimizing an energy function similar to Horn and Schunck, and typical problems are over-smoothing of motion boundaries, and failure on small objects with large motions (see for example the arms and spear in the right image of figure 1.2). One of the differences compared to image restoration, is the difficulty of obtaining ground-truth data. Whereas million of images can be easily collected from the internet, labeling the motion of every pixel in a video sequence is a much harder problem. Recently, the usage of synthetic data has gained popularity through the availability of high quality graphics engines. One example is the MPI-Sintel dataset [8] which is based on an open source animation movie, and allows access to video sequences along with corresponding flow fields, depth maps and more.

1.1.3 Depth enhancement

The depth of pixels in an image can be estimated using different queues. In stereo imaging for example, the depth of every pixel is a function of the disparity in its appearance between the left and the right image. In recent years, RGBD cameras that output both the RGB channels of the image, and a depth map for all pixels, have become extremely popular. Most cameras extract depth using either the “structured light” method or the “time-of-flight” method [1]. In structured light cameras, some texture is projected onto the scene and then captured by the sensor. The depth is then estimated using the deformation of the

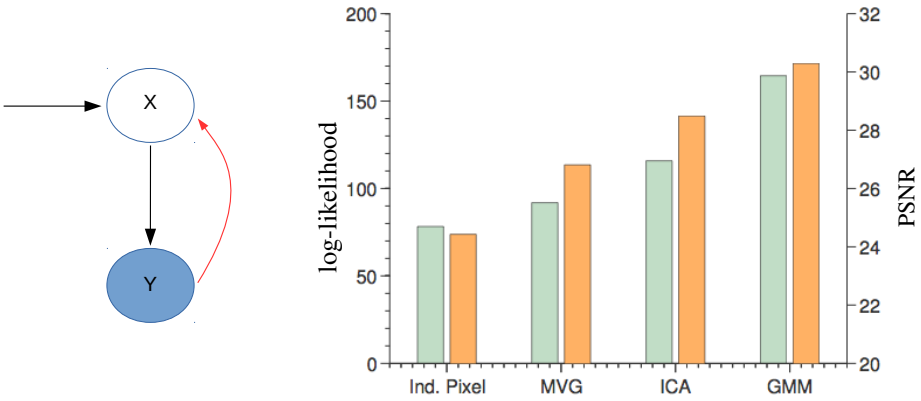


Figure 1.4: The generative learning approach. Left: black arrows illustrate the generative model and red arrow illustrates the inference process. Right: Models with better log-likelihood on patches (green) lead to better image restoration (orange). (replotted from [48]).

known texture and the disparity in each pixel similarly to stereo imaging. Time-of-flight cameras also use the projection of light onto the scene, but the depth is estimated by measuring the time of flight of the projected signal between the camera and the location of each pixel in the scene.

It is usually the case that the resulting output in both of those methods is in lower quality than the RGB image. The D channel typically contains noise and has many missing parts due to different kinds of occlusions and specularities (figure 1.3).

Therefore, there is a need for methods to improve the quality of the depth map. Many such methods rely on the fact that depth discontinuities occur in the same location as color discontinuities. Following a similar method for a colorization task [29], one of the most successful methods to enhance the depth map is based on minimizing an energy function containing a smoothness term which is conditioned on the color image [33]:

$$J(d) = \|m \odot (d - \tilde{d})\|_2^2 + \lambda \sum_{ij \in adj} w_{ij}(c)(d_i - d_j)^2 \tag{1.3}$$

Where d is the depth map, $\tilde{d}$ is the low quality observation containing noise and holes, and m is a mask consisting of zeros where $\tilde{d}$ contains holes and ones otherwise. The regularization depends on the weights $w_{ij}(c)$ which are conditioned on the color image c:

$$w_{ij}(c) = e^{-\frac{1}{\sigma^2}\|c_i - c_j\|^2} \tag{1.4}$$

This way pairs of adjacent pixels with a large color difference are given a smaller weight in the regularization and therefore depth boundaries are encouraged to co-occur with color boundaries.

1.2 The generative learning approach

While in the above examples, the method to solve low level vision tasks is based on some assumptions made about the problem, a machine learning approach would mean that those assumptions are automatically learned from data. In the generative approach one assumes the data was generated by the model

in figure 1.4, i.e. that first the target variable x is generated, and then the observable input y is generated based on x. For low level vision, this direction is usually the easier one to model. For example in image restoration, it is easier to model how a noisy or blurry image y was generated given the clean sharp image x, than vice-versa. In optical flow, it is easier to model how the second image is generated given the first image and the motion at every pixel, rather than modeling how the motion is generated given a pair of images.

Generative learning is usually approached using probabilistic models. Looking at figure 1.4 again we see that this requires two models (denoted by the black arrows):

1. The prior, $Pr(x)$, which captures properties of the target variable x. For image restoration this would be a model of natural images; for optical flow this would be a model of flow fields; and for depth estimation this would be a model of depth maps.

2. The noise/likelihood $Pr(y|x)$, which models how the observed input y is generated given the target x. For image restoration this would be a model of the noise, or blur. For optical flow this would model the warping and the generation of the second image (e.g. the deviation from constant brightness).

Those models are usually trained by maximizing the likelihood over a training set of clean images, which is equivalent to finding the probability model which best matches the empirical distribution using the KL-divergence.

Once those two models are given, we can define the inference task using Bayes' rule:

$$Pr(x|y) = \frac{1}{Z} Pr(y|x) Pr(x) \tag{1.5}$$

which gives us the full posterior probability of the target variable that interests us x, given the observed input to the problem y. Calculating the full posterior is usually infeasible for low level vision since x is a high dimensional image, so one can use either a Bayesian least square (BLS) approach which seeks the mean of the posterior probability (and also minimizes the expected square loss from the true x), or alternatively find the value of x that maximizes the posterior probability known as MAP inference.

Using the generative approach with MAP inference is very similar to the handcrafted energy minimization examples we presented in the previous section. Since the normalization constant can be disregarded when maximizing equation 1.5 with respect to x, this becomes equivalent to minimizing the following energy function:

$$J(x) = -\log \Pr(y|x) - \log Pr(x) \tag{1.6}$$

where the likelihood model serves as the data term and the prior model serves as the regularization/smoothness term. In this view it is therefore natural to extend the handcrafted methods to a generative approach where the energy function terms are learned from data rather than assumed by researchers. The separate modeling of the prior and the likelihood allows for different settings in which they are either both learned from data or only one of them is learned and the other assumed to be given. For example in non-blind image restoration, a noise model is assumed to be given at inference time, therefore only the prior model needs to be learned in advance from prior data. In the following we present some examples of prior work using a generative approach for non-blind image restoration.

1.2.1 Image priors

As the likelihood term is assumed to be given in non-blind image restoration, what's left to be found is a prior model of natural images. The fact that quadratic smoothness terms and Gaussian models in general were found inappropriate for natural images [34, 42, 35], and the extremely large dimension of the data, pose a very hard challenge to the problem of statistically modeling images. The work of Zhu and Mumford [47] that was later extended to the Fields-of-Experts (FoE) model by Roth and Black [40] are examples of image models that are based on a Markov field on top of local linear filters:

$$Pr(X) = \frac{1}{Z} \prod_i \prod_k e^{\phi_k(A_k x^i)} \tag{1.7}$$

where i is an index of all the local patches x^i in the image, and k indexes the different experts which are based on different linear filters A_k.

The parameters of the model are learned by training on a dataset of clean images. Since this model is intractable, and the normalization factor Z cannot be computed in closed form, training the model becomes a hard problem and the likelihood is maximized only approximately (e.g using contrastive divergence).

A different kind of generative models for images are models based on deep neural networks that can be used to generate random images. Recently, impressive results have been demonstrated for training such networks [12, 19]. Although these models can be used to generate random images with typical characteristics of natural images, it is not clear how to use them to compute the probability of a given image, and therefore, they are not used as prior models for inference tasks.

1.2.2 Patch priors

In contrast to the FoE which is a model for whole images, the Expected Patch Log Likelihood (EPLL) method by Zoran and Weiss [48] shows how image restoration can be performed using models of small patches only. They show that given a good model of local image patches one can use a naive approximation that assumes independence between all patches, and perform global MAP inference that achieves very good results:

$$\arg\max_x \sum_i \log Pr(x_i) + \lambda \log Pr(y|x) \tag{1.8}$$

where the sum in the log prior is over all patches and comes from the independence assumption.

The idea behind EPLL is that learning a good model for small patches and doing the approximation at inference time is better than compromising for an approximate model for whole images (due to the difficulty of learning good models for such high dimensional data).

Popular models for image patches include models based on Gaussian scale mixtures [35, 43], ICA and sparse coding [34, 2, 13, 32] and others [24, 44]. The patch model that Zoran and Weiss use is an unconstrained Gaussian mixture model (GMM) with 200 components on 8×8 patches. This model is very expressive and has almost half a million parameters, much more than other patch models and also more than the FoE model for whole images which has less than a thousand parameters.

Figure 1.4 shows that better patch models in terms of log-likelihood result in better image restoration when used in the EPLL method. The GMM model which has the best log-likelihood also results in the best image restoration. Image denoising with the EPLL method and the trained GMM also outperforms image

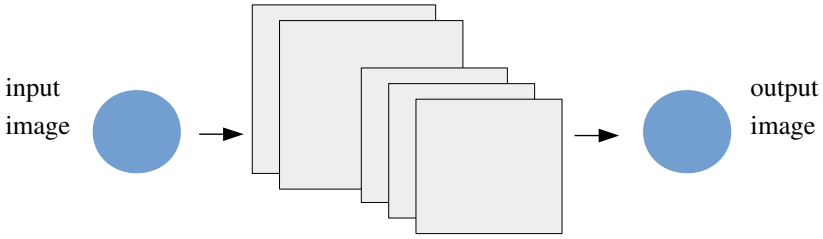


Figure 1.5: The discriminative learning approach. Given a set of input images and corresponding output images, search for the best predictor within a predefined architecture.

models such as the FoE and the very popular handcrafted method BM3D. In addition, it significantly outperforms the handcrafted methods for image deblurring.

1.3 The discriminative learning approach

In contrast to generative learning, the discriminative approach aims to directly model a predictor for the inference problem, avoiding the need to apply Bayes' rule at test time. Looking at the model in figure 1.4, this means directly learning the red arrow.

For low level vision this would consist of gathering a dataset of input/output image pairs, coming up with some hypothesis class of possible predictors and searching for the best predictor from the input to the output within the class (figure 1.5). Specifically for image denoising, a dataset of clean images can serve as the desired output, and one can create the corresponding input examples by applying noise to each image. Then, after coming up with a suitable class of predictors one needs to define a search method to find the best predictor. A possible choice for the predictor is a feed-forward neural network with several layers, trained using stochastic gradient descent.

Although this approach was tested in the past [23], perhaps the first example of a discriminative approach to image denoising that surpassed the results of the handcrafted BM3D was the MLP model [7] which uses a multi-layer perceptron (i.e. a fully connected feed-forward neural network) for small image patches. The model is used to denoise full images by running on all patches and averaging the results.

One of the reasons that allowed the MLP model to outperform BM3D is that in contrast to previous attempts, it was trained on a very large dataset of natural images (millions of images from the ImageNet dataset [11]). However, in order to outperform BM3D for various noise levels, the MLP model was trained separately for different noise levels, resulting in a different model for each noise level. Figure 1.6 shows the performance of models trained on different noise levels, compared to BM3D. The performance of models that were trained on a certain noise level drops quickly when the noise level is changed at test time.

1.3.1 Partial discriminative methods

Although using end-to-end discriminative learning in other low-level vision tasks is still rare, many successful methods combine discriminative training in different aspects of the predictor. One example is the

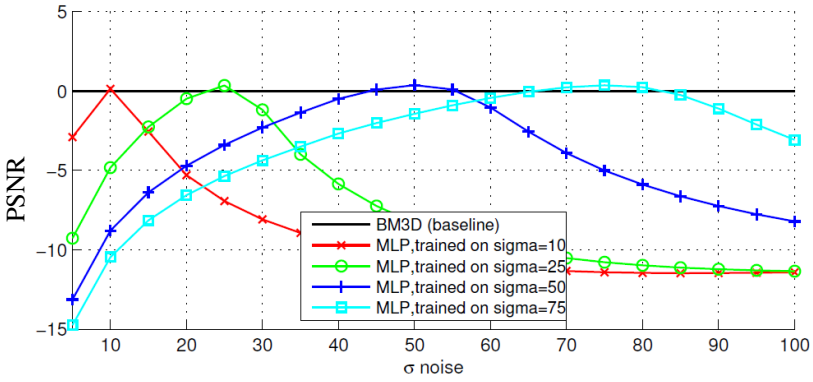


Figure 1.6: Quality of Image denoising compared to BM3D. Discriminative models that were trained on certain noise levels used for other noise levels. (replotted from [7]).

FlowNet model [15], which is a neural network that predicts the optical flow field given a pair of images. Although the FlowNet results are impressive, it is shown in the paper that the output of the discriminative model is still worse than the output of handcrafted methods and is only good as a starting point to other optical flow methods.

Another way to use discriminative training in low level vision is to train a predictor of the data term in the energy function. Examples such as [46, 17] train neural networks to predict the quality of match between two image patches in an optical flow or stereo setting. For inference, they use different search methods to find a solution that gives the best results under the trained predictor.

1.3.2 Generative vs. discriminative

A natural question to ask at this point is which approach is better for low level vision, generative or discriminative? If we compare the EPLL and MLP methods presented above, we see that both approaches can achieve good results but they both have different advantages and disadvantages.

The discriminative approach can lead to better performance with a controlled running time in inference, since it optimizes in advance the best choice within a given predictor architecture for a given task. This is not the case for the generative approach which at inference time needs to apply Bayes' rule, usually involving a hard optimization problem.

However, the advantage of the generative approach is in the separate modeling of the prior and the likelihood term which leads to a modular predictor. For example in non-blind image restoration, one can learn a single model for images and use it for different tasks such as denoising with different noise levels, deblurring with different blur kernel and inpainting. Training the prior model is done only once on clean images, and then for each task it is paired to the corresponding likelihood and used in Bayes' rule. The discriminative approach in this setting would require to re-train a model for each task, such as different noise levels (figure 1.6), and probably would require to train different architectures for other tasks like deblurring. The strengths of the modularity property in the generative approach, is that it can even be used for new tasks that were not conceived yet in the time of training.

The modularity property of generative learning can also be significant at training time, reducing the sample complexity for a specific task and loss. When the target variable x is of high dimension and has

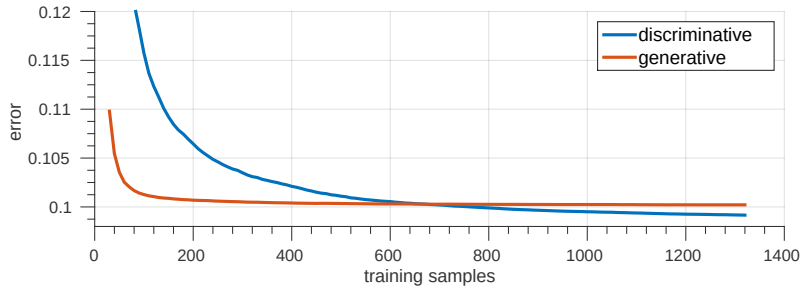


Figure 1.7: Comparing discriminative and generative training of linear predictors for non-Gaussian data. For small training sets, the generative modeling which decouples the prior from the noise, serves as a regularization and results in better generalization.

some interesting structure, and the observed variable y is really generated by some process based on x (i.e. when the data really comes from the graphical model in figure 1.4), this prior knowledge can be helpful at training time.

Figure 1.7 illustrates this effect, using synthetic data of ten dimensional vectors x and ten dimensional observations y. For different sizes of the training set, the data is used to (1) train a discriminative linear regressor, and (2) fit a Gaussian model to x (prior), a Gaussian model to $y - x$ (noise), and use them to compute the BLS linear predictor $C_x(C_x + C_{y-x})^{-1}$. Both the target vectors x and the noise added to generate y are not Gaussian and thus neither of the predictors is optimal. When evaluating both linear predictors on a hold-out test set, we see that indeed for small training sets up to a certain size, the generative approach achieves better results. When the dataset is too small to train a predictor end-to-end, de-coupling the prior of x from the noise that generated the observation y can serve as a good regularization.

1.4 A generalized approach to generative learning

Taking the generative approach and using MAP inference boils down to the following optimization problem:

$$\arg\max_x \quad \log Pr(y|x) + \log Pr(x) \tag{1.9}$$

Examining this equation we see that the generative approach has three components:

1. The prior: $Pr(x)$.
2. The likelihood/noise: $Pr(y|x)$.
3. The inference: $\arg\max_x$.

   Here we focus on simple maximization (MAP inference), but in general the inference could be any other process applied to the prior and likelihood, such as finding the mean of the posterior (BLS inference).

Taking a more general approach for generative learning, we can use ground-truth data to not only learn a prior or noise model, but also to learn how to use these models at inference time. Viewed this way, when tackling different low level vision tasks we have the choice to either fix in advance or learn from data

each one of the above components. In our work we aim to extend the success of the generative approach as achieved by the EPLL method for image restoration [48] in 2 different aspects:

- We apply the approach to other low level vision problems such as optical flow and depth estimation.
- We demonstrate how to improve all the three components of the generative approach by directly learning them from ground-truth data.

1.4.1 Learning the inference

How is learning the inference of a generative model different from the end-to-end discriminative approach which directly learns an inference process?

The first advantage of the generative model is that it can be used as a starting point for the discriminative method or at least it can define the class of possible predictors to search within. If there is some natural way to model the problem in the generative direction (e.g. when the noise model is known in advance), one can formulate the predictor as some inference procedure performed on the unknown generative model. Since this defines a class of predictors with unknown parameters, it can be trained discriminatively to minimize a certain loss. If the generative model was trained beforehand, then it can be used as a starting point for gradient descent for example.

This approach, also known as "unrolling the inference", has become popular lately for low level vision. For example, Gregor and LeCun [20] discriminatively train a predictor for super-resolution which is based on a sparse prior and different inference methods (such as coordinate descent). Another example is the discriminative training of the Fields-of-Experts model for image deblurring [41], where the predictor is based on half-quadratic splitting inference [18] with five iterations.

The unrolling approach results in an interpretable predictor since its architecture was designed in advance based on a prior component and a likelihood component. Therefore it can be used in a modular way, e.g. by changing the noise model parameters at test time. However, since it was trained on specific inference tasks and loss it is expected to be biased towards the distribution of tasks and the loss it has encountered at training time.

One of the basic principles in the generative approach is that training is performed independently of a certain inference task or loss. Instead, at training time one seeks to maximize the likelihood over the training data which can be interpreted as directly trying to match the data distribution as good as possible using the KL-divergence. Our aim is to use this approach also when learning the inference. Although it might be a harder problem than optimizing the inference for a specific task, the hope is that this will lead to more modular predictors that can even work for new tasks and losses that were not known at training time.

When using complex generative models, one of the difficulties in inference is the need to predict the state of hidden variables. Learning a predictor for the hidden states can facilitate and accelerate inference at test time. Although it was not applied to low level vision problems before, the idea of learning a predictor for hidden states was already introduced in the Helmholtz machine [10]. The Helmholtz machine is a hierarchical feed-forwrd generative model coupled with a "backwards" inference model. Since the generative model consists of several layers of hidden variables, direct learning is intractable, and instead the model is trained by simultaneously training the inference model to predict the hidden states.

We show in our work how the idea of learning to predict hidden states can be used for image restoration. Specifically, we show that predicting the posterior over the components of a mixture model prior, termed "gating", leads to accelerated image restoration while retaining the modularity property (section 2.3).

1.5 Interim summary

In this chapter we discussed the problem of low level vision and three approaches to tackle it: the handcrafted approach, generative learning and discriminative learning. We described different advantages of each approach, and showed that although discriminative learning can lead to better performance for a a given running time and pre-defined task, the advantage of generative learning is in its modularity. Separately modeling the prior and likelihood can lead to lower sample complexity at training, and it also allows a model to be used for different tasks at test time (e.g. when given different noise models). We suggested to take a more general view of the generative approach where each of its three components, the prior, the likelihood, and the inference process, can be either handcrafted or learned from data.

In the next chapter we present four papers which form the main results of our research. The papers demonstrate on different domains of low level vision, how the three components of the generative approach can be improved by statistical learning from data. In the first and second papers we discuss prior models and likelihood models for optical flow estimation. We show how the assumptions made by different handcrafted methods can be extracted and formulated as probabilistic models of the prior and likelihood, allowing them to be evaluated on ground-truth data. We compare the different handcrafted models and show how better models can be learned directly from the ground-truth. In the third paper we show how the inference process can be learned from data, for image restoration problems. We show that this results in a predictor that combines the advantages of the generative and discriminative approaches by being both modular and fast at test time. The forth paper deals with the problem of enhancing the depth map output of RGBD cameras. We compare different prior models extracted from handcrafted methods, and show that by evaluating and learning the models on ground-truth data we can improve the state-of-the-art in depth enhancement.

Chapter 2

Results

In this chapter we present the main results of our research. The upcoming sections consist of two papers which were published and two which were submitted for review.

2.1 Learning the local statistics of optical flow

This section includes the following publication:

Dan Rosenbaum, Daniel Zoran and Yair Weiss. *Learning the local statistics of optical flow.* Advances in Neural Information Processing Systems, 2013.

Learning the Local Statistics of Optical Flow

Dan Rosenbaum[1], **Daniel Zoran**[2], **Yair Weiss**[1,2]
[1] CSE , [2] ELSC , Hebrew University of Jerusalem
{danrsm,daniez,yweiss}@cs.huji.ac.il

Abstract

Motivated by recent progress in natural image statistics, we use newly available datasets with ground truth optical flow to learn the local statistics of optical flow and compare the learned models to prior models assumed by computer vision researchers. We find that a Gaussian mixture model (GMM) with 64 components provides a significantly better model for local flow statistics when compared to commonly used models. We investigate the source of the GMM's success and show it is related to an explicit representation of flow boundaries. We also learn a model that jointly models the local intensity pattern and the local optical flow. In accordance with the assumptions often made in computer vision, the model learns that flow boundaries are more likely at intensity boundaries. However, when evaluated on a large dataset, this dependency is very weak and the benefit of conditioning flow estimation on the local intensity pattern is marginal.

1 Introduction

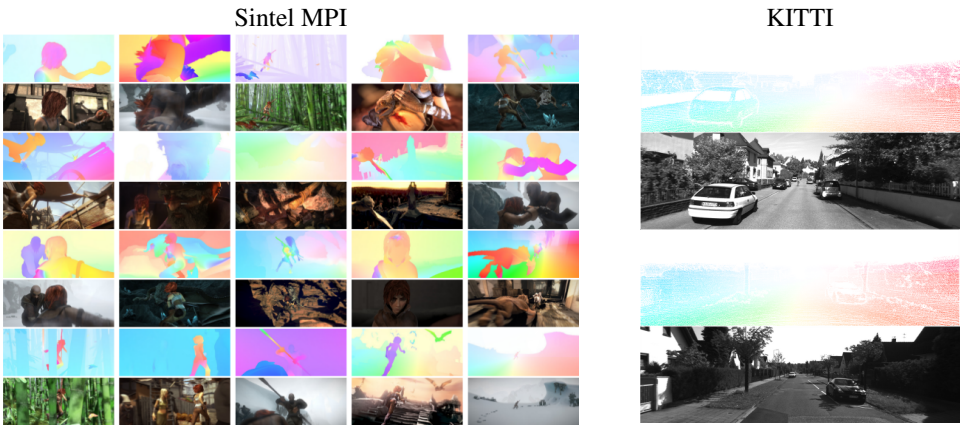


Figure 1: Samples of frames and flows from new flow databases. We leverage these newly available resources to learn the statistics of optical flow and compare this to assumptions used by computer vision researchers.

The study of natural image statistics is a longstanding research topic with both scientific and engineering interest. Recent progress in this field has been achieved by approaches that systematically compare different models of natural images with respect to numerical criteria such as log likelihood on held-out data or coding efficiency [1, 10, 14]. Interestingly, the best models in terms of log likelihood, when used as priors in image restoration tasks, also yield state-of-the-art performance [14].

Many problems in computer vision require good priors. A notable example is the computation of *optical flow*: a vector at every pixel that corresponds to the two dimensional projection of the motion

at that pixel. Since local motion information is often ambiguous, nearly all optical flow estimation algorithms work by minimizing a cost function that has two terms: a local data term and a "prior" term (see. e.g. [13, 11] for some recent reviews).

Given the success in image restoration tasks, where learned priors give state-of-the-art performance, one might expect a similar story in optical flow estimation. However, with the notable exception of [9] (which served as a motivating example for this work and is discussed below) there have been very few attempts to learn priors for optical flow by modeling local statistics. Instead, the state-of-the-art methods still use priors that were formulated by computer vision researchers. In fact, two of the top performing methods in modern optical flow benchmarks use a hand-defined smoothness constraint that was suggested over 20 years ago [6, 2].

One big difference between image statistics and flow statistics is the availability of ground truth data. Whereas for modeling image statistics one merely needs a collection of photographs (so that the amount of data is essentially unlimited these days), for modeling flow statistics one needs to obtain the ground truth motion of the points in the scene. In the past, the lack of availability of ground truth data did not allow for learning an optical flow prior from examples. In the last two years, however, two ground truth datasets have become available. The Sintel dataset (figure 1) consists of a thousand pairs of frames from a highly realistic computer graphics film with a wide variety of locations and motion types. Although it is synthetic, the work in [3] convincingly show that both in terms of image statistics and in terms of flow statistics, the synthetic frames are highly similar to real scenes. The KITTI dataset (figure 1) consists of frames taken from a vehicle driving in a European city [5]. The vehicle was equipped with accurate range finders as well as accurate localization of its own motion, and the combination of these two sources allow computing optical flow for points that are stationary in the world. Although this is real data, it is sparse (only about 50% of the pixels have ground truth flow).

In this paper we leverage the availability of ground truth datasets to learn explicit statistical models of optical flow. We compare our learned model to the assumptions made by computer vision algorithms for estimating flow. We find that a Gaussian mixture model with 64 components provides a significantly better model for local flow statistics when compared to commonly used models. We investigate the source of the GMM's success and show that it is related to an explicit representation of flow boundaries. We also learn a model that jointly models the local intensity pattern and the local optical flow. In accordance with the assumptions often made in computer vision, the model learns that flow boundaries are more likely at intensity boundaries. However, when evaluated on a large dataset, this dependency is very weak and the benefit of conditioning flow estimation on the local intensity pattern is marginal.

1.1 Priors for optical flow

One of the earliest methods for optical flow that is still used in applications is the celebrated Lucas-Kanade algorithm [7]. It overcomes the local ambiguity of motion analysis by assuming that the optical flow is constant within a small image patch and finds this constant motion by least-squares estimation. Another early algorithm that is still widely used is the method of Horn and Schunck [6]. It finds the optical flow by minimizing a cost function that has a data term and a "smoothness" term. Denoting by u the horizontal flow and v the vertical flow, the smoothness term is of the form:

$$J_{HS} = \sum_{x,y} u_x^2 + u_y^2 + v_x^2 + v_y^2$$

where u_x, u_y are the spatial derivatives of the horizontal flow u and v_x, v_y are the spatial derivatives of the vertical flow v. When combined with modern optimization methods, this algorithm is often among the top performing methods on modern benchmarks [11, 5].

Rather than using a quadratic smoothness term, many authors have advocated using more robust terms that would be less sensitive to outliers in smoothness. Thus the Black and Anandan [2] algorithm uses:

$$J_{BA} = \sum_{x,y} \rho(u_x) + \rho(u_y) + \rho(v_x) + \rho(v_y)$$

where $\rho(t)$ is a function that grows slower than a quadratic. Popular choices for ρ include the Lorentzian, the truncated quadratic and the absolute value $\rho(x) = |x|$ (or a differentiable approximation to it $\rho(x) = \sqrt{\epsilon + x^2}$)[11]. Both the Lorentzian and the absolute value robust smoothness

terms were shown to outperform quadratic smoothness in [11] and the absolute value was better among the two robust terms.

Several authors have also suggested that the smoothness term be based on the local intensity pattern, since motion discontinuities are more likely to occur at intensity boundaries. Ren [8] modified the weights in the Lucas and Kanade least-squares estimation so that pixels that are on different sides of an intensity boundary will get lower weights. In the context of Horn and Shunck, several authors suggest using weights to the horizontal and vertical flow derivatives, where the weights had an inverse relationship with the image derivatives: large image derivatives lead to low weight in the flow smoothness (see [13] and references within for different variations on this idea). Perhaps the simplest such regularizer is of the form:

$$J_{HSI} = \sum_{x,y} w(I_x)(u_x^2 + v_x^2) + w(I_y)(u_y^2 + v_y^2) \tag{1}$$

As we discuss below, this prior can be seen as a Gaussian prior on the flow that is *conditioned on the intensity*.

In contrast to all the previously discussed priors, Roth and Black [9] suggested learning a prior from a dataset. They used a training set of optical flow obtained by simulating the motion of a camera in natural range images. The prior learned by their system was similar to a robust smoothness prior, but the filters are not local derivatives but rather more random-looking high pass filters. They did not observe a significant improvement in performance when using these filters, and standard derivative filters are still used in most smoothness based methods.

Given the large number of suggested priors, a natural question to ask is: what is the best prior to use? One way to answer this question is to use these priors as a basis for an optical flow estimation algorithm and see which algorithm gives the best performance. Although such an approach is certainly informative it is difficult to get a definitive answer using it. For example, Sun et al. [11] reported that adding a non-local smoothness term to a robust smoothness prior significantly improved results on the Middlebury benchmark, while Geiger et al. [5] reported that this term decreased performance on KITTI benchmark. Perhaps the main difficulty with this approach is that the prior is only one part of an optical flow estimation algorithm. It is always combined with a non-convex likelihood term and optimized using a nonlinear optimization algorithm. Often the parameters of the optimization have a very large influence on the performance of the algorithm.

In this paper we take an alternative approach. Motivated by recent advances in natural image statistics and the availability of new datasets, we compare different priors in terms of (1) log likelihood on held-out data and (2) inference performance with tractable posteriors. Our results allow us to rigorously compare different prior assumptions.

2 Comparing priors as density models

In order to compare different prior models as density models, we generate a training set and test set of optical flow patches from the ground truth databases. Denoting by f a single vector that concatenates all the optical flow in a patch (e.g. if we consider 8×8 patches, f is a vector of length 128 where the first 64 components denote u and the last 64 components denote v). Given a prior probability model $\Pr(f;\theta)$ we use the training set to estimate the free parameters of the model θ and then we measure the log likelihood of *held out* patches from the test set.

From Sintel, we divided the pairs of frames for which ground truth is available into 708 pairs which we used for training and 333 pairs which we used for testing. The data is divided into scenes and we made sure that different scenes are used in training and testing. We created a *second test set* from the KITTI dataset by choosing a subset of patches for which full ground truth flow was available. Since we only consider full patches, this set is smaller and hence we use it only for testing, not for training.

The priors we compared are:

- Lucas and Kanade. This algorithm is equivalent to the assumption that the observed flow is generated by a constant (u_0, v_0) that is corrupted by IID Gaussian noise. If we also assume

that u_0, v_0 have a zero mean Gaussian distribution, $\Pr(f)$ is a zero mean multidimensional Gaussian with covariance given by $\sigma_p^2 OO^t + \sigma_n^2 I$ where O is a binary 128×2 matrix and σ_p the standard deviation of u_0, v_0 and σ_n the standard deviation of the noise.

- Horn and Schunck. By exponentiating J_{HS} we see that $\Pr(f;\theta)$ is a multidimensional Gaussian with covariance matrix λDD^T where D is a 256×128 derivative matrix that computes the derivatives of the flow field at each pixel and λ is the weight given to the prior relative to the data term. This covariance matrix is not positive definite, so we use $\lambda DD^T + \epsilon I$ and determine λ, ϵ using maximum likelihood.
- L1. We exponentiate J_{BA} and obtain a multidimensional Laplace distribution. As in Horn and Schunck, this distribution is not normalizeable so we multiply it by an IID Laplacian prior on each component with variance $1/\epsilon$. This again gives two free parameters (λ, ϵ) which we find using maximum likelihood. Unlike the Gaussian case, the solution of the ML parameters and the normalization constant cannot be done in closed form, and we use Hamiltonian Annealed Importance Sampling [10].
- Gaussian Mixture Models (GMM). Motivated by the success of GMMs in modeling natural image statistics [14] we use the training set to estimate GMM priors for optical flow. Each mixture component is a multidimensional Gaussian with full covariance matrix and zero mean and we vary the number of components between 1 and 64. We train the GMM using the standard Expectation-Maximization (EM) algorithm using mini-batches. Even with a few mixture components, the GMM has far more free parameters than the previous models but note that we are measuring success on *held out* patches so that models that overfit should be penalized.

The summary of our results are shown in figure 2 where we show the mean log likelihood on the Sintel test set. One interesting thing that can be seen is that the local statistics validate some assumptions commonly used by computer vision researchers. For example, the Horn and Shunck smoothness prior is as good as the *optimal Gaussian prior* (GMM1) even though it uses local first derivatives. Also, the robust prior (L1) is much better than Horn and Schunck. However, as the number of Gaussians increase the GMM is significantly better than a robust prior on local derivatives.

A closer inspection of our results is shown in figure 3. Each figure shows the histogram of log likelihood of held out patches: the more shifted the histogram is to the right, the better the performance. It can be seen that the GMM is indeed much better than the other priors including cases where the test set is taken from KITTI (rather than Sintel) and when the patch size is 12×12 rather than 8×8.

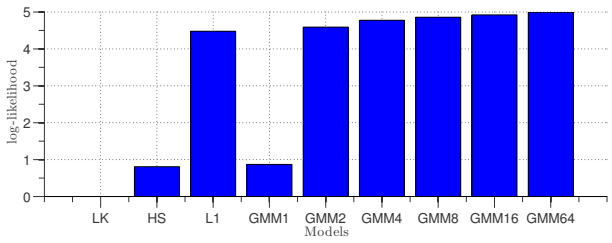


Figure 2: mean log likelihood of the different models for 8×8 patches extracted from held out data from Sintel. The GMM outperforms the models that are assumed by computer vision researchers.

2.1 Comparing models using tractable inference

A second way of comparing the models is by their ability to restore corrupted patches of optical flow. We are not claiming that optical flow restoration is a real-world application (although using priors to "fill in" holes in optical flow is quite common, e.g. [12, 8]). Rather, we use it because for the models we are discussing the inference can either be done in closed form or using convex optimization, so we would expect that better priors will lead to better performance.

We perform two flow restoration tasks. In "flow denoising" we take the ground truth flow and add IID Gaussian noise to all flow vectors. In "flow inpainting" we add a small amount of noise to all

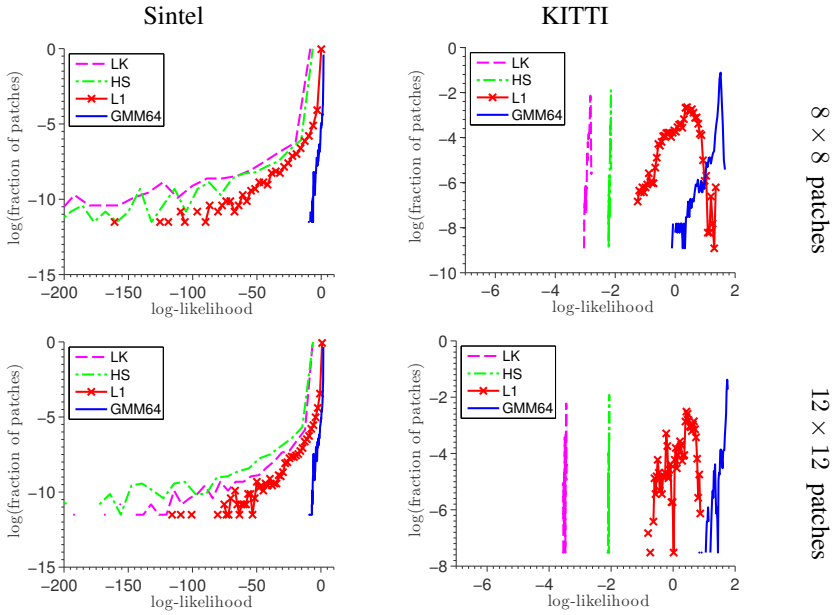


Figure 3: Histograms of log-likelihood of different models on the KITTI and Sintel test sets with two different patch sizes. As can be seen, the GMM outperforms other models in all four cases.

flow vectors and a very big amount of noise to some of the flow vectors (essentially meaning that these flow vectors are not observed). For the Gaussian models and the GMM models the Bayesian Least Squares (BLS) estimator of f given y can be computed in closed form. For the Laplacian model, we use MAP estimation which leads to a convex optimization problem. Since MAP may be suboptimal for this case, we optimize the parameters λ, ϵ for MAP inference performance.

Results are shown in figures 4,5. The standard deviation of the ground truth flow is approximately 11.6 pixels and we add noise with standard deviations $10, 20$ and 30 pixel. Consistent with the log likelihood results, L1 outperforms the Gaussian methods but is outperformed by the GMM. For small noise values the difference between L1 and the GMM is small, but as the amount of noise increases L1 becomes similar in performance to the Gaussian methods and is much worse than the GMM.

3 The secret of the GMM

We now take a deeper look at how the GMM models optical flow patches. The first (and not surprising) thing we found is that the covariance matrices learned by the model are block diagonal (so that the u and v components are independent given the assignment to a particular component).

More insight can be gained by considering the GMM as a local subspace model: a patch which is generated by component k is generated as a linear combination of the eigenvectors of the kth covariance. The coefficients of the linear combination have energy that decays with the eigenvalue: so each patch can be well approximated by the leading eigenvectors of the corresponding covariance. Unlike global subspace models, different subspace models can be used for different patches, and during inference with the model one can infer which local subspace is most likely to have generated the patch.

Figure 6 shows the dominant leading eigenvectors of all 32 covariance matrices in the GMM32 model: the eigenvectors of u are followed by the eigenvectors of v. The number of eigenvectors displayed in each row is set so that they capture 99% of the variance in that component. The rows are organized by decreasing mixing weight. The right hand half of each row shows (u,v) patches that are sampled from that Gaussian.

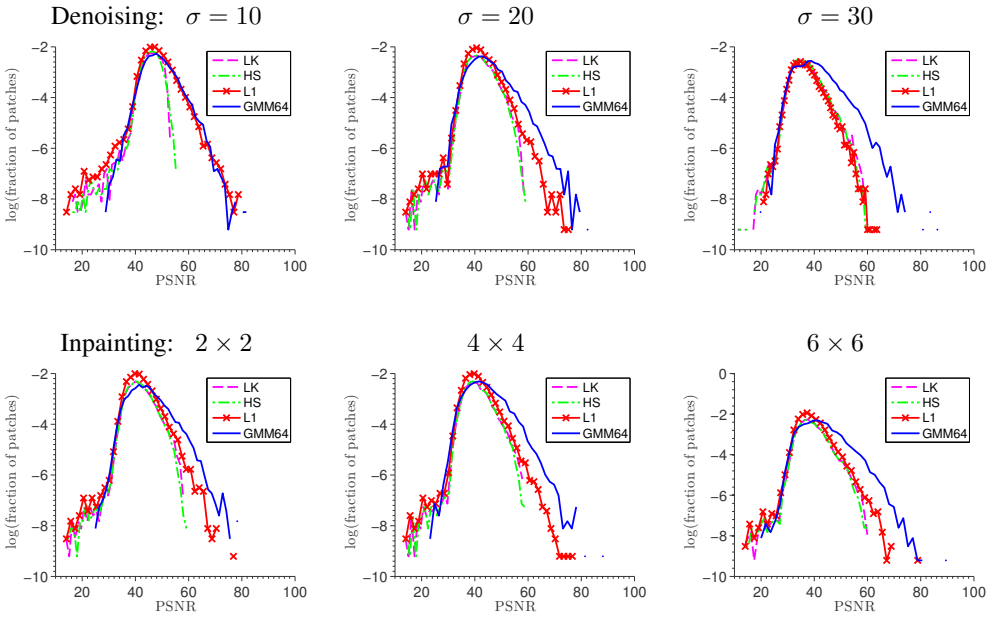


Figure 4: Denoising with different noise values and inpainting with different hole sizes.

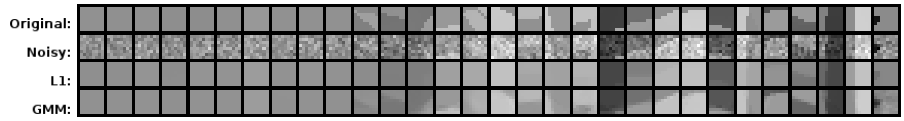


Figure 5: Visualizing denoising performance ($\sigma = 30$).

It can be seen that the first 10 components or so model very smooth components (in fact the samples appear to be completely flat). A closer examination of the eigenvalues shows that these ten components correspond to smooth motions of different speeds. This can also be seen by comparing the v samples on the top row which are close to gray with those in the next two rows which are much closer to black or white (since the models are zero mean, black and white are equally likely for any component).

As can be seen in the figure, almost all the energy in the first components is captured by uniform motions. Thus these components are very similar to a *non-local* smoothness assumption similar to the one suggested in [11]): they not only assume that derivatives are small but they assume that all the 8×8 patch is constant. However, unlike the suggestion in [11] to enforce non-local smoothness by applying a median filter at *all pixels*, the GMM only applies non-local smoothness at a subset of patches that are inferred to be generated by such components.

As we go down in the figure towards more rare components. we see that the components no longer model flat components but rather motion boundaries. This can be seen both in the samples (rightmost rows) and also in the leading eigenvectors (shown on the left) which each control one side of a boundary. For example, the bottom row of the figure illustrates a component that seems to generate primarily diagonal motion boundaries.

Interestingly, such local subspace models of optical flow have also been suggested by Fleet et al. [4]. They used synthetic models of moving occlusion boundaries and bars to learn linear subspace models of the flow. The GMM seems to support their intuition that learning separate linear subspace models for flat vs motion boundary is a good idea. However, unlike the work of Fleet et al. the separation into "flat" vs. "motion boundary" was learned in an unsupervised fashion directly from the data.

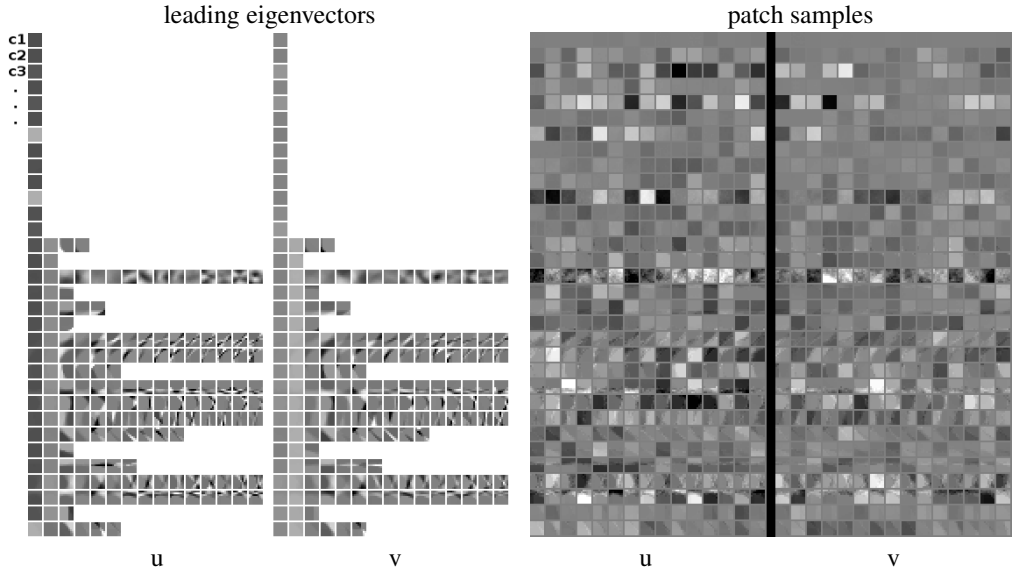


Figure 6: The eigenvectors and samples of the GMM components. GMM is better because it explicitly models edges and flat patches separately.

4 A joint model for optical flow and intensity

As mentioned in the introduction, many authors have suggested modifying the smoothness assumption by conditioning it on the local intensity pattern and giving a higher penalty for motion discontinuities in the absence of intensity discontinuities. We therefore ask, does conditioning on the local intensity give better log likelihood on held out flow patches? Does it give better performance in tractable inference tasks?

We evaluated two flow models that are conditioned on the local intensity pattern. The first one is a conditional Gaussian (eq. 1) with exponential weights, i.e. $w(I_x) = \exp(-I_x^2/\sigma^2)$ and the variance parameter σ^2 is optimized to maximize performance. The second one is a Gaussian mixture model that simultaneously models both intensity and flow.

The simultaneous GMM we use includes a 200 component GMM to model the intensity together with a 64 dimensional GMM to model the flow. We allow a dependence between the hidden variable of the intensity GMM and that of the flow GMM. This is equivalent to a hidden Markov model (HMM) with 2 hidden variables: one represents the intensity component and one represents the flow component (figure 8). We learn the HMM using the EM algorithm. Initialization is given by independent GMMs learned for the intensity (we actually use the one learned by [14] which is available on their website) and for the flow. The intensity GMM is not changed during the learning. Conditioned on the intensity pattern, the flow distribution is still a GMM with 64 components (as in the previous section) but the mixing weights depend on the intensity.

Given these two conditional models, we now ask: will the conditional models give better performance than the unconditional ones? The answer, shown in figure 7 was surprising (to us). Conditioning on the intensity gives basically zero improvement in log likelihood and a slight improvement in flow denoising only for very large amounts of noise. Note that for all models shown in this figure, the denoised estimate is the Bayesian Least Squares (BLS) estimate, and is optimal given the learned models.

To investigate this effect, we examine the transition matrix between the intensity components and the flow components (figure 8). If intensity and flow were independent, we would expect all rows of the transition matrix to be the same. If an intensity boundary always lead to a flow boundary, we would expect the bottom rows of the matrix to have only one nonzero element. By examining the learned transition matrix we find that while there is a dependency structure, it is not very strong.

Regardless of whether the intensity component corresponds to a boundary or not, the most likely flow components are flat. When there is an intensity boundary, the flow boundary in the same orientation becomes more likely. However, even though it is more likely than in the unconditioned case, it is still less likely than the flat components.

To rule out that this effect is due to a local optimum found by EM, we conducted additional experiments whereby the emission probabilities were held fixed to the GMMs learned independently for flow and motion and each patch in the training set was assigned one intensity and one flow component. We then estimated the joint distribution over flow and motion components by simply counting the relative frequency in the training set. The results were nearly identical to those found by EM.

In summary, while our learned model supports the standard intuition that motion boundaries are more likely at intensity boundaries, it suggests that when dealing with a large dataset with high variability, there is very little benefit (if any) in conditioning flow models on the local intensity.

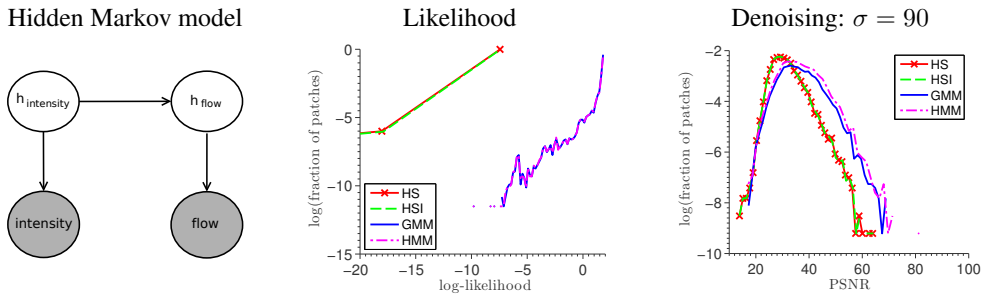


Figure 7: The hidden Markov model we use to jointly model intensity and flow. Both log likelihood and inference evaluations show almost no improvement of conditioning flow on intensity.

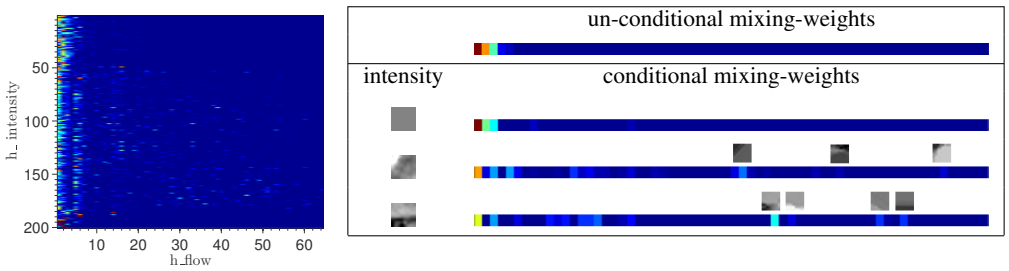


Figure 8: Left: the transition matrix learned by the HMM. Right: comparing rows of the matrix to the unconditional mixing weights. Conditioned on an intensity boundary, motion boundaries become more likely but are still less likely than a flat motion.

5 Discussion

Optical flow has been an active area of research for over 30 years in computer vision, with many methods based on assumed priors over flow fields. In this paper, we have leveraged the availability of large ground truth databases to learn priors from data and compare our learned models to the assumptions typically made by computer vision researchers. We find that many of the assumptions are actually supported by the statistics (e.g. the Horn and Schunck model is close to the optimal Gaussian model, robust models are better, intensity discontinuities make motion discontinuities more likely). However, a learned GMM model with 64 components significantly outperforms the standard models used in computer vision, primarily because it explicitly distinguishes between flat patches and boundary patches and then uses a different form of nonlocal smoothness for the different cases.

Acknowledgments

Supported by the Israeli Science Foundation, Intel ICRI-CI and the Gatsby Foundation.

References

[1] M. Bethge. Factorial coding of natural images: how effective are linear models in removing higher-order dependencies? 23(6):1253–1268, June 2006.

[2] Michael J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.

[3] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV (6)*, pages 611–625, 2012.

[4] David J. Fleet, Michael J. Black, Yaser Yacoob, and Allan D. Jepson. Design and use of linear models for image motion analysis. *International Journal of Computer Vision*, 36(3):171–193, 2000.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[6] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[7] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[8] Xiaofeng Ren. Local grouping for optical flow. In *CVPR*, 2008.

[9] Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50, 2007.

[10] J Sohl-Dickstein and BJ Culpepper. Hamiltonian annealed importance sampling for partition function estimation. 2011.

[11] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE, 2010.

[12] Li Xu, Zhenlong Dai, and Jiaya Jia. Scale invariant optical flow. In *Computer Vision–ECCV 2012*, pages 385–399. Springer, 2012.

[13] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. Optic flow in harmony. *International Journal of Computer Vision*, 93(3):368–388, 2011.

[14] Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. In *NIPS*, pages 1745–1753, 2012.

2.2 Beyond brightness constancy: learning noise models for optical flow

This section includes the following paper that was submitted for review:

Dan Rosenbaum and Yair Weiss. *Beyond brightness constancy: learning noise models for optical flow.*

Beyond Brightness Constancy: Learning Noise Models for Optical Flow

Dan Rosenbaum
School of Computer Science and Engineering
Hebrew University of Jerusalem
www.cs.huji.ac.il/~danrsm

Yair Weiss
School of Computer Science and Engineering
Hebrew University of Jerusalem
www.cs.huji.ac.il/~yweiss

Abstract

Optical flow is typically estimated by minimizing a "data cost" and an optional regularizer. While there has been much work on different regularizers many modern algorithms still use a data cost that is not very different from the ones used over 30 years ago: a robust version of brightness constancy or gradient constancy. In this paper we leverage the recent availability of ground-truth optical flow databases in order to learn a data cost. Specifically we take a generative approach in which the data cost models the distribution of noise after warping an image according to the flow and we measure the "goodness" of a data cost by how well it matches the true distribution of flow warp error. Consistent with current practice, we find that robust versions of gradient constancy are better models than simple brightness constancy but a learned GMM that models the density of patches of warp error gives a much better fit than any existing assumption of constancy. This significant advantage of the GMM is due to an explicit modeling of the spatial structure of warp errors, a feature which is missing from almost all existing data costs in optical flow. Finally, we show how a good density model of warp error patches can be used for optical flow estimation on whole images. We replace the data cost by the expected patch log-likelihood (EPLL), and show how this cost can be optimized iteratively using an additional step of denoising the warp error image. The results of our experiments are promising and show that patch models with higher likelihood lead to better optical flow estimation.

1 Introduction

Despite being a longstanding topic of study in computer vision, the current state-of-the-art optical flow estimation results are far from being satisfactory. This is especially evident when performance is evaluated on outdoor scenes with large occlusions and fast motions [5, 4]. In the last two years ground truth flow for such scenes has been made available either using synthetic scenes [4] or by accurate laser range finders that provide flow for stationary points in the scene [5].

Like many problems in computer vision, optical flow estimation is commonly solved by optimizing a function derived from a certain assumed model. The assumed model can be typically divided to a data cost model which reflects the assumptions on the way the flow should correspond to the images, and a regularizer that reflects the prior assumptions on typical flow fields. Since the functions optimized are usually not convex, most algorithms only achieve approximate solutions and so another critical component in the algorithm is the optimization procedure.

In order to improve performance of flow estimation one can choose to improve any of these three components: the regularizer, the data cost and the optimizer. While much recent work has explored using different regularizers (e.g. [19, 13]) or different optimizers (e.g. [16, 11, 17]) there has been relatively little work on the data term. A notable exception is the recent work of Vogel and Roth [14]

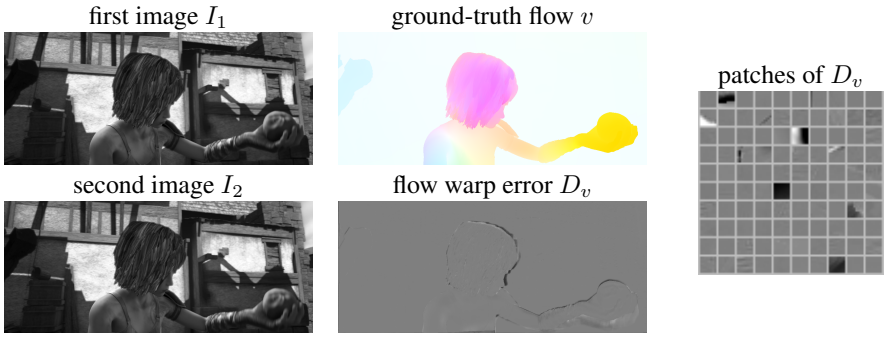


Figure 1: Flow warp error. The second image is warped backwards according to the ground-truth flow and subtracted from the first image. The resulting warp error image is then divided to small patches. In this paper we use a database of such patches in order to learn a data term for optical flow. In contrast to models used in common algorithms, the warp error has an evident spatial structure and is far from being isotropic noise.

which compares the effect of different versions of brightness or gradient constancy on the performance of optical flow algorithms.

Brightness constancy and gradient constancy are in a sense "hand-crafted" data costs. Is it possible to leverage the availability of ground truth flow datasets in order to *learn* a data cost for optical flow? A step in this direction was taken by Sun et al [12] who used a Fields of Experts distribution over the error term and learned 3×3 filters that defined the data cost. The learned cost was similar to gradient constancy but with irregular filters.

In this paper we take a generative approach in which the data cost models the distribution of noise after warping an image according to the flow. Under the ideal brightness constancy assumption, when we backwards-warp the second image according to the optical flow we should obtain an image that is identical to the first one (figure 1). In real images, of course, we never get exact matches and we call the difference between the warped second image and the first image the "flow warp error." Different data costs for optical flow give different penalties for this flow warp error.

Here we measure the "goodness" of a data cost by how well it matches the true distribution of flow warp error. By focusing on patches of flow warp errors we can measure this "goodness" robustly and efficiently. Consistent with current practice, we find that robust versions of gradient constancy are better models than simple brightness constancy but a learned Gaussian Mixture Model (GMM) density model of the error gives a much better fit than any existing assumption of constancy. This significant advantage of the GMM is due to an explicit modeling of the spatial structure of warp errors, a feature which to the best of our knowledge is missing from the vast majority of existing data costs in optical flow.

A second question we address here, is how a patch model of flow warp error can be used for flow estimation of whole images. To do so, we replace the image data cost by the *expected patch log-likelihood* (EPLL) term introduced by [20]. We also propose a method for optimizing this cost, which is based on *half-quadratic splitting* [15, 7]. Our method boils down to an iterative algorithm consisting of two steps. In the first step we solve a flow estimation problem with a simple brightness constancy cost, and in the second step we "denoise" the resulting image of flow warp error using a patch density model. The results of our experiments are promising and show that patch models with higher likelihood lead to better optical flow estimation.

1.1 Optical flow data costs

Brightness Constancy

Perhaps the most common data cost simply penalizes the gray-scale distance between every pixel in the first image I_1 and its corresponding location in the second image I_2 according to the flow v. This is equivalent to creating a warped image I_2^v by warping back I_2 according to v and then

subtracting the warped image from I_1. In classic optical flow algorithms like Horn and Schunck [6] and Lucas-Kanade [8] the squared distance is summed over all pixels (SSD) resulting in:

$$J_{BCL2} = \sum_p (I_1(p) - I_2^v(p))^2 \ . \tag{1}$$

In later extensions like Black and Anandan [2], more robust functions are used, e.g. the sum of absolute distances (SAD),

$$J_{BCL1} = \sum_p |I_1(p) - I_2^v(p)| \ . \tag{2}$$

Gradient Constancy

A second approach is to measure the distance of the image spatial derivatives rather than gray-scale values [3], thus allowing a constant change in gray-scale. Denoting by I_{1x},I_{1y} and I_{2x},I_{2y} the horizontal and vertical derivatives of the first and second image, this is equivalent to warping I_{2x} and I_{2y} according to v and subtracting them from I_{1x} and I_{1y},

$$J_{GCL2} = \sum_p (I_{1x}(p) - I_{2x}^v(p))^2 + \sum_p (I_{1y}(p) - I_{2y}^v(p))^2 \ . \tag{3}$$

Once again, the quadratic function can be replaced by a more robust function like the absolute value,

$$J_{GCL1} = \sum_p |I_{1x}(p) - I_{2x}^v(p)| + \sum_p |I_{1y}(p) - I_{2y}^v(p)| \ . \tag{4}$$

Census

An increasingly popular approach to deal with smooth changes of gray-scale between images is to replace the gray-scale by some monotone ranking in a certain neighborhood. In the Census transform [18], the data cost at a pixel p counts the number of neighboring pixels q that change their sign relative to p,

$$J_{CEN} = \sum_p \sum_q 1_{\left[sign(I_1(q) - I_1(p)) \neq sign(I_2^v(q) - I_2^v(p))\right]} \ , \tag{5}$$

A convex approximation of the Census transform can be formulated by replacing the indicator and sign functions by the absolute value, resulting in the centralized sum of absolute distance (CSAD) data cost [14]

$$J_{CSAD} = \sum_p \sum_q |(I_1(q) - I_1(p)) - (I_2^v(q) - I_2^v(p))| \ . \tag{6}$$

One drawback of all the above costs is that they are all sums of local costs and lack the modeling of spatial structure. Figure 1 shows the warp error $D_v = I_1 - I_2^v$ of images from the Sintel dataset [4], using the provided ground-truth flow. The warp error images show an evident spatial structure. Even when looking at small random patches from the dataset, the structure is clearly observed. In particular patches tend to be flat and close to zero but occasionally contain an edge in some orientation.

2 The data cost as a noise model

Using a generative approach to flow estimation from a pair of images I_1 and I_2, it can be assumed that the first image I_1 is generated as

$$I_1 = I_2^v + w \tag{7}$$

where w is a random noise image generated from some density model. In this view, different data costs that are functions of the warp error $D_v = I_1 - I_2^v$, are equivalent to different density models of w:

$$Pr(I_1|I_2; v) = Pr(D_v) = \frac{1}{Z} e^{-\lambda J(D_v)} \tag{8}$$

Notice that according to equation 7, the warp error D_v is equal to the noise w and thus equation 8 is also a density model over the additive noise w.

The data costs we consider above: brightness constancy (BC), gradient constancy (GC) and centralized sum of absolute differences (CSAD) are all functions of the warp error. In particular, they can all be expressed as the l_2-norm or l_1-norm of a linear transformation of D_v. Therefore we can formulate them as density models as follows:

Brightness Constancy L2 Exponentiating J_{BCL2} (equation 1), we obtain a multidimensional Gaussian which is a product of independent Gaussians with variance $1/2\lambda$.

$$Pr(D_v) = \frac{1}{Z} e^{-\lambda \sum_p D_v(p)^2} = \frac{1}{Z} e^{-\lambda ||d_v||_2^2} \tag{9}$$

where d_v is a vector created by concatenating all pixels in D_v.

Brightness Constancy L1 Exponentiating J_{BCL1} (equation 2), we obtain a multidimensional Laplace distribution which is a product of independent Laplacians with variance $1/2\lambda$.

$$Pr(D_v) = \frac{1}{Z} e^{-\lambda \sum_p |D_v(p)|} = \frac{1}{Z} e^{-\lambda ||d_v||_1} \tag{10}$$

Gradient Constancy L2 Exponentiating J_{GCL2} (equation 3), we obtain a multidimensional Gaussian with inverse covariance matrix $\lambda A^\top A$ where A is a derivative matrix that computes the horizontal and vertical derivatives at each pixel. Since this matrix is not invertible we add ϵI.

$$Pr(D_v) = \frac{1}{Z} e^{-\lambda \sum_p D_{vx}(p)^2 + D_{vy}(p)^2} \approx \frac{1}{Z} e^{-d_v^\top (\lambda A^\top A + \epsilon I) d_v} \tag{11}$$

Gradient Constancy L1 Exponentiating J_{GCL1} (equation 4), we obtain a multidimensional Laplace distribution. As in GCL2, we add ϵI to make this distibution normalizable, and since the normalization constant Z cannot be found in closed form we use Hamiltonian Annealed Importance Sampling to approximate it [10].

$$Pr(D_v) = \frac{1}{Z} e^{-\lambda \sum_p |D_{vx}(p)| + |D_{vy}(p)|} \approx \frac{1}{Z} e^{-||(\lambda A + \epsilon I) d_v||_1} \tag{12}$$

Centralized Sum of Absolute Differences Exponentiating J_{CSAD} using a 5×5 neighborhood around each pixel p (equation 4), we obtain a multidimensional Laplace distribution. Now the derivative matrix A contains more rows corresponding to all the differences between p and each pixel q in the 5×5 neighborhood. Like in GCL1 we need to add ϵI and approximate the normalization constant using Hamiltonian Annealed Importance Sampling.

$$Pr(D_v) = \frac{1}{Z} e^{-\lambda \sum_p \sum_q |D_v(q) - D_v(p)|} \approx \frac{1}{Z} e^{-||(\lambda A_{5\times 5} + \epsilon I) d_v||_1} \tag{13}$$

2.1 Comparing different data costs

Perhaps the most direct way of comparing different data costs is by evaluating the relative performance of optical flow algorithms that use these costs. This is the approach taken in [14, 12]. The main drawback of this approach is that the flow predicted by an algorithm is usually the result of a complicated, nonconvex optimization and many parameters can influence the final result. For example, Sun et al [11] reported that changing the number of levels in the pyramid used for coarse to fine optimization can dramatically change the performance of some algorithms on the Sintel benchmark.

Here we take an alternative approach. We consider the data costs as density models on D_v, and ask: which of these density models best fits the distribution of actual patches of flow warp errors? The primary method we use to estimate the goodness of fit is the *average log likelihood on held out data*. It is well known that this log likelihood can be equivalently written as a constant minus the KL divergence between the empirical distribution and the density model. Thus the model that gives highest log likelihood to held out data is also the model whose distribution is most similar to the empirical distribution.

We create a dataset of flow warp error D_v using the Sintel dataset. First we use the ground-truth flow to warp the images backwards, then we subtract the warped images from their corresponding

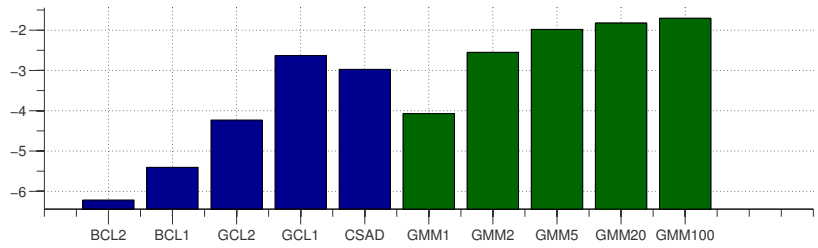


Figure 2: Log-likelihood on held-out data from Sintel's final pass. The learned GMM noise models (green) are compared to the common data cost noise models (blue). Consistent with common practice, gradient constancy better fits the data than brightness constancy and robust (L1) costs are better than Gaussian (L2). However, a GMM with 100 components outperforms all other models.

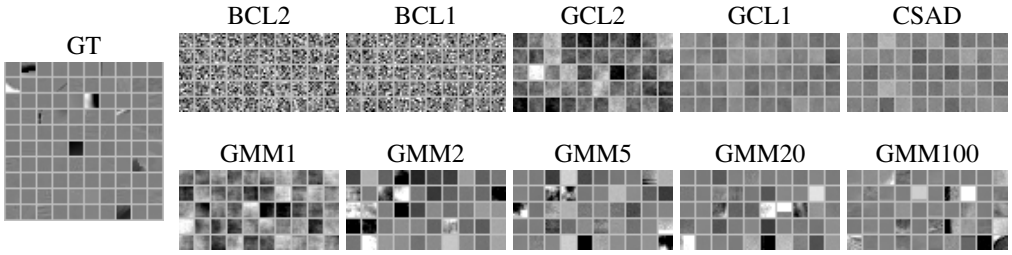


Figure 3: Patch samples of warp error from Sintel's final pass (GT) and randomly generated using the different noise models. Top: samples generated using the common data cost models. Bottom: samples generated using the learned GMM models. The patches generated from GMM100 demonstrate a very similar structure to the ground-truth patches.

preceding images, and finally we divide the resulting images to 8×8 patches. Following [9] we divide the training set of Sintel into two parts: 708 image pairs in training and 333 pairs were used for testing. All model parameters (e.g. λ, ϵ discussed above) were learned on the training set using maximum likelihood. We then compare the likelihood of different density models on a random sample of patches from the test set. We repeat this process for each of the three passes of Sintel: *albedo*, *clean* and *final*, resulting in three separate training sets and test sets. Since all our results are very similar on all the three passes we focus here only on the *final* pass.

The resulting likelihood for the above models are shown in figure 2 (in blue). The main things to note are that the l_1-norm is better than the l_2-norm, that the constant gradient assumption is better than the constant brightness assumption, and that the convex approximation of the census transform is very similar to the gradient constancy assumption. These findings agree with the comparison of optical flow estimation using different data costs reported by [14].

Another way to measure how well models capture the true statistics is by *generating samples*. Patches created from a certain model, typically satisfy the underlying assumptions of the model. Therefore, a visual resemblance to the ground-truth suggests that the patches were generated from a better model. We can see in figure 3 (top row), that the patches generated from GCL1 and CSAD are the most similar to the ground-truth patches. Although those patches seem to model the flatness correctly, evidently, they fail to model the occasional structure that is present in the ground-truth.

3 Learning the noise model

Following the recent success of learning Gaussian mixture models (GMM) in natural image statistics [21] and as prior models for optical flow [9], we use the training set to estimate GMMs with a different number of components. Every component of the GMM is a multivariate Gaussian with zero mean and a full covariance matrix. We train the GMM using the Expectation Maximization (EM) algorithm on mini-batches from the training set. It is important to note that the GMM has

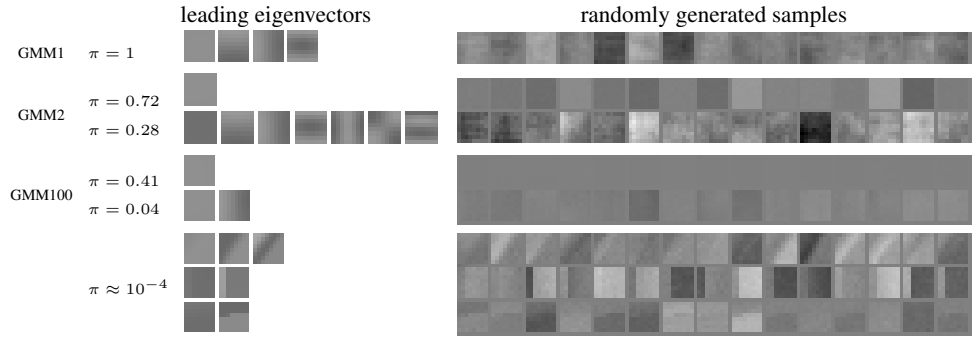


Figure 4: The leading eigenvectors and randomly generated samples for the components of GMM1 and GMM2, and for some selected components of GMM100. GMM2 captures outliers by modeling flat patches and noisy patches separately. GMM100 explicitly distinguishes between flat patches with different variance, and patches with different types of edges.

far more parameters than the common data cost models and thus we emphasize that the models are tested on the held-out test set to assure no overfitting occurs.

The resulting likelihood of all models is shown in figure 2. The results show that: 1. a single Gaussian model (GMM1) has a very similar likelihood to the L2 Gradient Constancy model (GCL2); 2. a GMM with 2 components (GMM2) is similar to the robust L1 Gradient Constancy model (GCL1); and 3. a GMM model with 100 components (GMM100) outperforms all other models.

We also use the learned GMMs to generate random patches and compare them to the ground-truth patches and to random patches generated by the common data cost models. Figure 3 shows that the patches generated by GMM100 resemble the ground-truth patches more than other models do.

3.1 What does the GMM learn?

We investigate the learned GMM models to understand what makes them better than the common data cost models, In figure 4 we show the components of GMM1 (which contains only one component), GMM2 and some selected components of GMM100. Each component is shown in a different row. Each of those components is a Gaussian, and to illustrate its preference, we show the leading eigenvectors of the covariance matrix corresponding to 95% of the cumulative eigenvalues (i.e. corresponding to 95% of the variance) re-organized as patches. In addition, we show in each row a set of patches that were randomly generated using the corresponding Gaussian.

For the single Gaussian model (GMM1) which essentially estimates the covariance of the patches, we can see that the leading eigenvectors of the covariance correspond to smooth changes in patches. This is also seen in the randomly generated patches. This behavior is similar to what the gradient constancy with l_2 norm (GCL2) models.

For GMM2, the figure shows that the first component favors flat patches in a much stronger manner than GMM1. This can be seen both in the generated samples and by the fact that 95% of the variance is expressed by the single flat eigenvector. In contrast, the second component of GMM2 allows the patches to be much more noisy than GMM1, and needs more eigenvectors to reach 95% of the variance. Similarly to the robustness characteristic of GCL1, the behavior of GMM2, can be viewed as a form of outlier detection where 72% of the errors are essentially just an additive constant and 28% of them are allowed to be very noisy.

For GMM100, we show only a few selected components ordered by decreasing mixing weights. The first components, capture the flatness assumption, and each component allows a random constant change with a different variance. Looking at components with lower mixing weights we see components that capture more interesting structure. *Most components are dedicated to edges in certain orientations and shifts.* Intuitively this model learns that most of the time the warped patch and the true patch will differ by an additive constant, but when this is not the case, the difference is not simply white noise. *Rather this "noise" is extremely structured and is well approximated locally*

by an oriented edge. In retrospect, this assumption is very intuitive and is related to the process of occlusion. Differences between the original patches and warped patches that are not simple additive constants are most commonly the result of occlusion and disocclusion. Since the occluded objects have spatial structure, so does the warp error. While this assumption is very intuitive, we are not aware of any optical flow data cost that utilizes it.

4 Optical Flow Estimation

We now show how a density model of warp error patches can be used for optical flow estimation. A common way to estimate optical flow from a pair of images is by minimizing an energy function containing a data cost depending on the input images and a regularizer on the flow field $R(f)$. Using our generative assumption (equations 7,8), the data cost we wish to minimize is equal to the log density model of the warp error of the whole image $Pr(D_v)$.

Given a patch density model, one way to define the image density model, introduced in [20], is to measure the *expected patch log-likelihood* (EPLL) in the image. Recall that d_v is a vector representation of the warp error image, and denote by P_i a matrix that extracts the i'th patch from it. The EPLL cost can be written as:

$$J(v) = -\sum_i \log Pr(P_i\ d_v) + \lambda R(v) \tag{14}$$

The exact minimization of the cost defined in equation 14 is not tractable. The first reason is that the warp error d_v is a non-convex function of the flow v. The common way to overcome this is by iteratively approximating d_v as a linear function of the flow (by taking the Taylor expansion of the image intensities around the current warp). A second problem is that even after the linearization of d_v the density model might cause the minimization to be intractable. To solve this for any density model we use the method of *half-quadratic splitting* as presented in [15, 7], combined with the EPLL image denoising method of [20]. In *half-quadratic splitting*, we introduce a new variable r, resulting in the following new cost:

$$J(v, r) = -\sum_i \log Pr(P_i\ r) + \beta||d_v - r||_2^2 + \lambda R(v) \tag{15}$$

This cost is approximately minimized by alternatingly solving for v and for r and by gradually increasing β. Note that once β is big enough, r is forced to be close to d_v and we return to the original EPLL cost (equation 14). We next describe the 2 steps performed in each iteration:

r-step: When v is fixed, the third term in equation 15 is constant and solving for r is equivalent to the problem of image denoising using a prior on clean patches. The "noisy" image in this case is the warp error image d_v, and the cost function on the difference $d_v - r$ is equivalent to the assumption that the noise model is Gaussian, isotropic and with variance $1/\beta$. Solving for r can be done using the EPLL denoising algorithm introduced in [20], where any patch model can be used (assuming that a patch denoising method is provided).

v-step: When r is fixed, solving for v is equivalent to estimating the optical flow using a simple brightness constancy data cost on the image, where the first image is "fixed" according to r. To see this recall that $d_v = I_1 - I_2^v$ and so defining a new image $I_1^r = I_1 - r$ results in the cost: $\text{argmin}_v\ ||I_1^r - I_2^v||_2^2 + \frac{\lambda}{\beta} R(v)$.

4.1 Experiments

To test the method proposed above, we use it to estimate the optical flow in the Sintel dataset using different warp error patch models. The estimation is performed in a coarse-to-fine manner such that in every level we run 20 iterations, each consisting of one r-step and one v-step. During the 20 iterations we gradually increase β to assure the original cost (equation 14) is decreasing. We use a common regularizer that penalizes the spatial derivatives of the flow using the l_1 norm [14, 11], and optimize it in each v-step using the *iteratively reweighted least squares* (IRLS) method. In each v-step we perform one image warp and linearization. The r-step is performed using the EPLL denoising software published in [20]. We start the process using an initial flow estimate in the coarsest level that was computed using a standard gradient constancy algorithm.

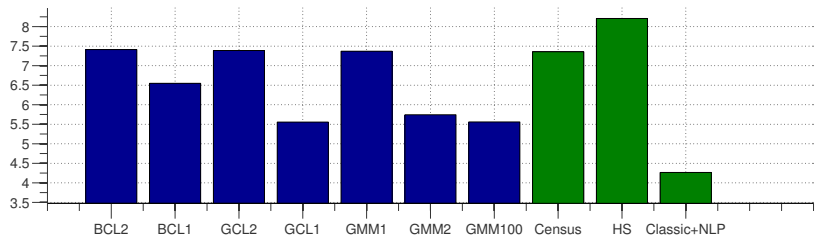


Figure 5: Average end-point-error of the optical flow estimated on 20 Sintel images. Blue: Our proposed EPLL method using different warp error patch models. Models with higher likelihood generally lead to lower error. Green: Reference algorithms. Even with a fairly simple flow regularizer, our EPLL method, combined with a good warp error patch model, is able to reach an error that is comparable to some of the top performing algorithms.

For reference, we also compare the EPLL method to other algorithms, which we run on the same 20 Sintel images. We use the software provided by [11] to run their implementation of Horn and Schunck (HS) and Classic+NLP which is one of the top performing algorithms in the Sintel and KITTI datasets. We also use the software by [14] which implements the Census transform data cost and is also one of the top performing algorithms in Sintel and KITTI. For all those algorithms we use the default parameters as suggested in their software kits.

The results are shown in figure 5. It can be seen that the performance of the EPLL with different warp error models is correlated to the likelihood of the models as shown in figure 2. In general, models with higher likelihood lead to flow estimation with smaller average end-point-error. The results also show that our EPLL method, combined with a good warp error patch model, is able to estimate the optical flow with error that is comparable to the reference algorithms: with a good warp error model EPLL outperforms the classic Horn and Schunck algorithm and the Vogel et al. implementation that also uses an L1 regularizer. The Classic+NLP algorithm uses a stronger, nonlocal regularizer and outperforms all the methods that use an L1 regualrizer in these experiments.

While we have found that all other things being equal, better warp noise models lead to better optical flow performance, our experiments indicate that the optimization method and the regularizers can be just as important. In particular, we find that the result of our "v-step" which uses a standard coarse-to-fine optimization procedure is often suboptimal and gives higher cost than the ground truth flow. This suggests that more powerful optimization methods are needed.

5 Discussion

In this paper we use a generative approach to evaluate and learn optical flow data costs. By focusing on patches of flow warp errors we measure the likelihood of different models robustly and efficiently. We show that evaluating the likelihood of existing data costs, largely agrees with common practice. We find that a learned GMM gives a better fit to the true distribution and show that it is related to the seperate representation of flat patches and different edge orientations. This intuitive structure that mirrors the spatial structure of occluding objects in natural scenes, has not been used in existing data costs for optical flow. Finally, we show how good patch models of warp error can lead to better performance in flow estimation. We define a new data cost which models the expected patch log likelihood and propose a method to optimize it. The results of our experiments show that using models with higher likelihood leads to better estimation. Even though we use a fairly simple flow regularizer, our EPLL method, combined with a good warp error patch model, is able to estimate the optical flow with error that is comparable to some of the top performing algorithms. We are confident that further research on improving the optimization, and combining our novel data cost with a strong regularizer, can lead to improved optical flow estimation.

References

[1] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[2] Michael J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.

[3] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer, 2004.

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV (6)*, pages 611–625, 2012.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[6] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[7] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, volume 22, pages 1–9, 2009.

[8] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[9] Dan Rosenbaum, Daniel Zoran, and Yair Weiss. Learning the local statistics of optical flow. In *Advances in Neural Information Processing Systems*, pages 2373–2381, 2013.

[10] J Sohl-Dickstein and BJ Culpepper. Hamiltonian annealed importance sampling for partition function estimation. 2011.

[11] Deqing Sun, Stefan Roth, and Michael J Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

[12] Deqing Sun, Stefan Roth, JP Lewis, and Michael J Black. Learning optical flow. In *Computer Vision–ECCV 2008*, pages 83–97. Springer, 2008.

[13] Deqing Sun, Erik B Sudderth, and Michael J Black. Layered segmentation and optical flow estimation over time. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1768–1775. IEEE, 2012.

[14] Christoph Vogel, Stefan Roth, and Konrad Schindler. An evaluation of data costs for optical flow. In *Pattern Recognition*, pages 343–353. Springer, 2013.

[15] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[16] Li Xu, Zhenlong Dai, and Jiaya Jia. Scale invariant optical flow. In *Computer Vision–ECCV 2012*, pages 385–399. Springer, 2012.

[17] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Robust monocular epipolar flow estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1862–1869. IEEE, 2013.

[18] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer VisionECCV'94*, pages 151–158. Springer, 1994.

[19] Henning Zimmer, Andrés Bruhn, and Joachim Weickert. Optic flow in harmony. *International Journal of Computer Vision*, 93(3):368–388, 2011.

[20] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.

[21] Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. In *NIPS*, pages 1745–1753, 2012.

2.3 The return of the gating network: combining generative models and discriminative training in natural image priors

This section includes the following publication:

Dan Rosenbaum and Yair Weiss. *The return of the gating network: combining generative models and discriminative training in natural image priors.* Advances in Neural Information Processing Systems, 2015.

The Return of the Gating Network: Combining Generative Models and Discriminative Training in Natural Image Priors

Dan Rosenbaum
School of Computer Science and Engineering
Hebrew University of Jerusalem

Yair Weiss
School of Computer Science and Engineering
Hebrew University of Jerusalem

Abstract

In recent years, approaches based on machine learning have achieved state-of-the-art performance on image restoration problems. Successful approaches include both generative models of natural images as well as discriminative training of deep neural networks. Discriminative training of feed forward architectures allows explicit control over the computational cost of performing restoration and therefore often leads to better performance at the same cost at run time. In contrast, generative models have the advantage that they can be trained once and then adapted to any image restoration task by a simple use of Bayes' rule.

In this paper we show how to combine the strengths of both approaches by training a discriminative, feed-forward architecture to predict the state of latent variables in a generative model of natural images. We apply this idea to the very successful Gaussian Mixture Model (GMM) of natural images. We show that it is possible to achieve comparable performance as the original GMM but with two orders of magnitude improvement in run time while maintaining the advantage of generative models.

1 Introduction

Figure 1 shows an example of an image restoration problem. We are given a degraded image (in this case degraded with Gaussian noise) and seek to estimate the clean image. Image restoration is an extremely well studied problem and successful systems for specific scenarios have been built without any explicit use of machine learning. For example, approaches based on "coring" can be used to successfully remove noise from an image by transforming to a wavelet basis and zeroing out coefficients that are close to zero [7]. More recently the very successful BM3D method removes noise from patches by finding similar patches in the noisy image and combining all similar patches in a nonlinear way [4].

In recent years, machine learning based approaches are starting to outperform the hand engineered systems for image restoration. As in other areas of machine learning, these approaches can be divided into *generative* approaches which seek to learn probabilistic models of clean images versus *discriminative* approaches which seek to learn models that map noisy images to clean images while minimizing the training loss between the predicted clean image and the true one.

Two influential generative approaches are the fields of experts (FOE) approach [16] and KSVD [5] which assume that filter responses to natural images should be sparse and learn a set of filters under this assumption. While very good performance can be obtained using these methods, when they are trained generatively they do not give performance that is as good as BM3D. Perhaps the most successful generative approach to image restoration is based on Gaussian Mixture Models (GMMs) [22]. In this approach 8x8 image patches are modeled as 64 dimensional vectors and a

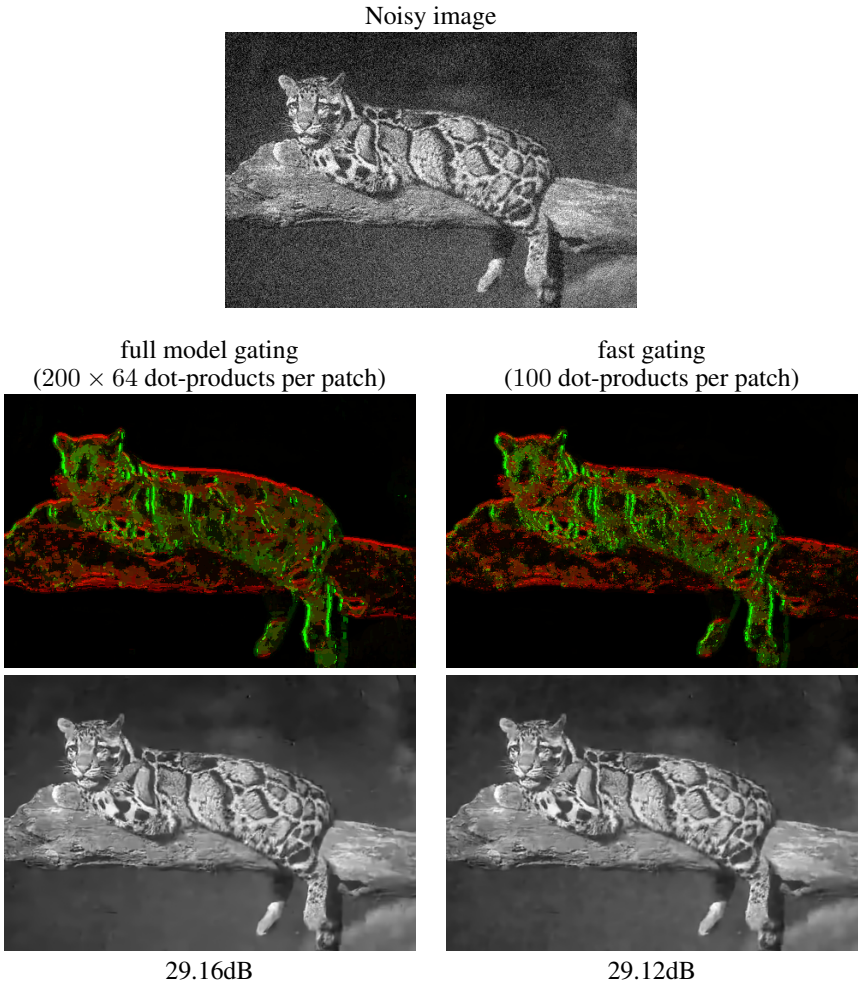


Figure 1: Image restoration with a Gaussian mixture model. Middle: the most probable component of every patch calculated using a full posterior calculation vs. a fast gating network (color coded by embedding in a 2-dimensional space). Bottom: the restored image: the gating network achieves almost identical results but in 2 orders of magnitude faster.

simple GMM with 200 components is used to model the density in this space. Despite its simplicity, this model remains among the top performing models in terms of likelihood given to left out patches and also gives excellent performance in image restoration [23, 20]. In particular, it outperforms BM3D on image denoising and has been successfully used for other image restoration problems such as deblurring [19]. The performance of generative models in denoising can be much improved by using an "empirical Bayes" approach where the parameters are estimated from the noisy image [13, 21, 14, 5].

Discriminative approaches for image restoration typically assume a particular feed forward structure and use training to optimize the parameters of the structure. Hel-Or and Shaked used discriminative training to optimize the parameters of coring [7]. Chen et al. [3] discriminatively learn the parameters of a generative model to minimize its denoising error. They show that even though the model was trained for a specific noise level, it acheives similar results as the GMM for different noise levels. Jain and Seung trained a convolutional deep neural network to perform image denoising. Using the same training set as was used by the FOE and GMM papers, they obtained better results than FOE but not as good as BM3D or GMM [9]. Burger et al. [2] trained a deep (non-convolutional) multi layer perceptron to perform denoising. By increasing the size of the training set by two orders of magnitude relative to previous approaches, they obtained what is perhaps the

best stand-alone method for image denoising. Fanello et al. [6] trained a random forest architecture to optimize denoising performance. They obtained results similar to the GMM but at a much smaller computational cost.

Which approach is better, discriminative or generative? First it should be said that the best performing methods in both categories give excellent performance. Indeed, even the BM3D approach (which can be outperformed by both types of methods) has been said to be close to optimal for image denoising [12]. The primary advantage of the discriminative approach is its *efficiency* at run-time. By defining a particular feed-forward architecture we are effectively constraining the computational cost at run-time and during learning we seek the best performing parameters for a fixed computational cost. The primary advantage of the generative approach, on the other hand, is its *modularity*. Learning only requires access to clean images, and after learning a density model for clean images, Bayes' rule can be used to peform restoration on any image degradation and can support different loss functions at test time. In contrast, discriminative training requires separate training (and usually separate architectures) for every possible image degradation. Given that there are literally an infinite number of ways to degrade images (not just Gaussian noise with different noise levels but also compression artifacts, blur etc.), one would like to have a method that maintains the modularity of generative models but with the computational cost of discriminative models.

In this paper we propose such an approach. Our method is based on the observation that the most costly part of inference with many generative models for natural images is in estimating latent variables. These latent variables can be abstract representations of local image covariance (e.g. [10]) or simply a discrete variable that indicates which Gaussian most likely generated the data in a GMM. We therefore discriminatively train a feed-forward architecture, or a "gating network" to predict these latent variables using far less computation. The gating network need only be trained on "clean" images and we show how to combine it during inference with Bayes' rule to perform image restoration for any type of image degradation. Our results show that we can maintain the accuracy and the modularity of generative models but with a speedup of two orders of magnitude in run time.

In the rest of the paper we focus on the Gaussian mixture model although this approach can be used for other generative models with latent variables like the one proposed by Karklin and Lewicki [10]. Code implementing our proposed algorithms for the GMM prior and Karklin and Lewicki's prior is available online at `www.cs.huji.ac.il/~danrsm`.

2 Image restoration with Gaussian mixture priors

Modeling image patches with Gaussian mixtures has proven to be very effective for image restoration [22]. In this model, the prior probability of an image patch x is modeled by: $\Pr(x) = \sum_h \pi_h \mathcal{N}(x; \mu_h, \Sigma_h)$. During image restoration, this prior is combined with a likelihood function $\Pr(y|x)$ and restoration is based on the posterior probability $\Pr(x|y)$ which is computed using Bayes' rule. Typically, MAP estimators are used [22] although for some problems the more expensive BLS estimator has been shown to give an advantage [17].

In order to maximize the posterior probability different numerical optimizations can be used. Typically they require computing the *assignment probabilities*:

$$\Pr(h|x) = \frac{\pi_h \mathcal{N}(x; \mu_h, \Sigma_h)}{\sum_k \pi_k \mathcal{N}(x; \mu_k, \Sigma_k)} \tag{1}$$

These assignment probabilities play a central role in optimizing the posterior. For example, it is easy to see that the gradient of the log of the posterior involves a weighted sum of gradients where the assignment probabilities give the weights:

$$\begin{aligned} \frac{\partial \log \Pr(x|y)}{\partial x} &= \frac{\partial \left[\log \Pr(x) + \log \Pr(y|x) - \log \Pr(y)\right]}{\partial x} \\ &= -\sum_h \Pr(h|x)(x - \mu_h)^\top \Sigma_h^{-1} + \frac{\partial \log \Pr(y|x)}{\partial x} \end{aligned} \tag{2}$$

Similarly, one can use a version of the EM algorithm to iteratively maximize the posterior probability by solving a sequence of reweighted least squares problems. Here the assignment probabilities define the weights for the least squares problems [11]. Finally, in auxiliary samplers for performing

BLS estimation, each iteration requires sampling the hidden variables according to the current guess of the image [17].

For reasons of computational efficiency, the assignment probabilities are often used to calculate a hard assignment of a patch to a component:

$$\hat{h}(x) = \arg\max_{h} \Pr(h|x) \tag{3}$$

Following the literature on "mixtures of experts" [8] we call this process *gating*. As we now show, this process is often the most expensive part of performing image restoration with a GMM prior.

2.1 Running time of inference

The successful EPLL algorithm [22] for image restoration with patch priors defines a cost function based on the simplifying assumption that the patches of an image are independent:

$$J(x) = -\sum_{i} \log \Pr(x_i) - \lambda \log \Pr(y|x) \tag{4}$$

where $\{x_i\}$ are the image patches, x is the full image and λ is a parameter that compensates for the simplifying assumption. Minimizing this cost when the prior is a GMM, is done by alternating between three steps. We give here only a short representation of each step but the full algorithm is given in the supplementary material. The three steps are:

- Gating. For each patch, the current guess x_i is assigned to one of the components $\hat{h}(x_i)$
- Filtering. For each patch, depending on the assignments $\hat{h}(x_i)$, a least squares problem is solved.
- Mixing. Overlapping patches are averaged together with the noisy image y.

It can be shown that after each iteration of the three steps, the EPLL splitting cost function (a relaxation of equation 4) is decreased.

In terms of computation time, the gating step is by far the most expensive one. The filtering step multiplies each d dimensional patch by a single $d \times d$ matrix which is equivalent to d dot-products or d^2 flops per patch. Assuming a local noise model, the mixing step involves summing up all patches back to the image and solving a local cost on the image (equivalent to 1 dot-product or d flops per patch).[1] In the gating step however, we compute the probability of all the Gaussian components for every patch. Each computation performs d dot-products, and so for K components we get a total of $d \times K$ dot-products or $d^2 \times K$ flops per patch. For a GMM with 200 components like the one used in [22], this results in a gating step which is 200 times slower than the filtering and mixing steps.

3 The gating network

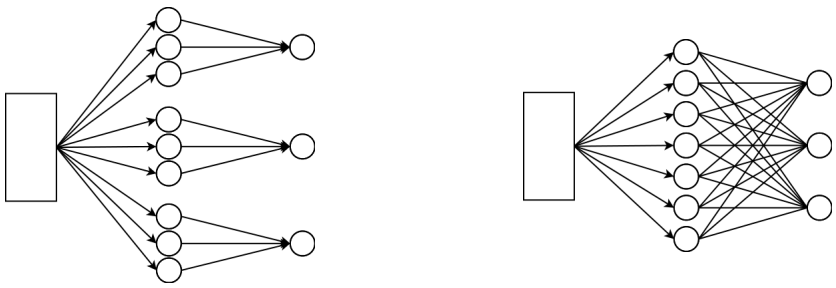


Figure 2: Architecture of the gating step in GMM inference (left) vs. a more efficient gating network.

[1] For non-local noise models like in image deblurring there is an additional factor of the square of the kernel dimension. If the kernel dimension is in the order of d, the mixing step performs d dot-products or d^2 flops.

The left side of figure 2 shows the computation involved in a naive computing of the gating. In the GMM used in [22], the Gaussians are zero mean so computing the most likely component involves multiplying each patch with all the eigenvectors of the covariance matrix and squaring the results:

$$\log Pr(x|h) = -x^\top \Sigma_h^{-1} x + const_h = -\sum_i \frac{1}{\sigma_i^h}(v_i^h x)^2 + const_h \tag{5}$$

where σ_i^h and v_i^h are the eigenvalues and eigenvectors of the covariance matrix. The eigenvectors can be viewed as templates, and therefore, the gating is performed according to weighted sums of dot-products with different templates. Every component has a different set of templates and a different weighting of their importance (the eigenvalues). Framing this process as a feed-forward network starting with a patch of dimension d and using K Gaussian components, the first layer computes $d \times K$ dot-products (followed by squaring), and the second layer performs K dot-products.

Viewed this way, it is clear that the naive computation of the gating is inefficient. There is no "sharing" of dot-products between different components and the number of dot-products that are required for deciding about the appropriate component, may be much smaller than is done with this naive computation.

3.1 Discriminative training of the gating network

In order to obtain a more efficient gating network we use discriminative training. We rewrite equation 5 as:

$$\log \Pr(x|h) \approx -\sum_i w_i^h (v_i^T x)^2 + const_h \tag{6}$$

Note that the vectors v_i are required to be shared and do not depend on h. Only the weights w_i^h depend on h.

Given a set of vectors v_i and the weights w the posterior probability of a patch assignment is approximated by:

$$\Pr(h|x) \approx \frac{\exp(-\sum_i w_i^h (v_i^T x)^2 + const_h)}{\sum_k \exp(-\sum_i w_i^k (v_i^T x)^2 + const_k)} \tag{7}$$

We minimize the cross entropy between the approximate posterior probability and the exact posterior probability given by equation 1. The training is done on 500 mini-batches of 10K clean image patches each, taken randomly from the 200 images in the BSDS training set. We minimize the training loss for each mini-batch using 100 iterations of `minimize.m` [15] before moving to the next mini-batch.

Results of the training are shown in figure 3. Unlike the eigenvectors of the GMM covariance matrices which are often global Fourier patterns or edge filters, the learned vectors are more localized in space and resemble Gabor filters.

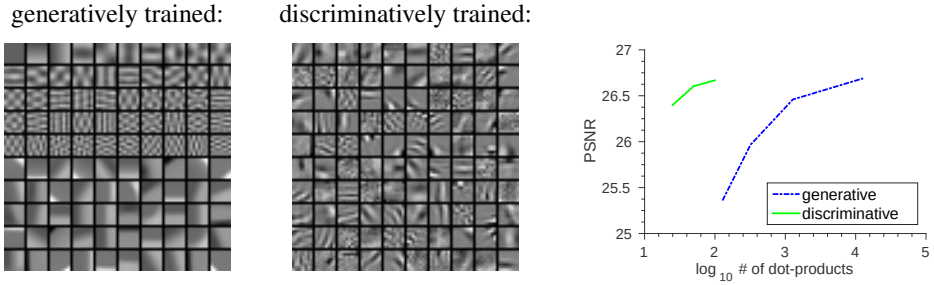


Figure 3: Left: A subset of the 200×64 eigenvectors used for the full posterior calculation. Center: The first layer of the discriminatively trained gating network which serves as a shared pool of 100 eigenvectors. Right: The number of dot-products versus the resulting PSNR for patch denoising using different models. Discrimintively training smaller gating networks is better than generatively training smaller GMMs (with less components).

Figure 1 compares the gating performed by the full network and the discriminatively trained one. Each pixel shows the predicted component for a patch centered around that pixel. Components are color coded so that dark pixels correspond to components with low variance and bright pixels to high variance. The colors denote the preferred orientation of the covariance. Although the gating network requires far less dot-products it gives similar (although not identical) gating.

Figure 4 shows sample patches arranged according to the gating with either the full model (top) or the gating network (bottom). We classify a set of patches by their assignment probabilities. For 60 of the 200 components we display 10 patches that are classified to that component. It can be seen that when the classification is done using the gating network or the full posterior, the results are visually similar.

The right side of figure 3 compares between two different ways to reduce computation time. The green curve shows gating networks with different sizes (containing 25 to 100 vectors) trained on top of the 200 component GMM. The blue curve shows GMMs with a different number of components (from 2 to 200). Each of the models is used to perform patch denoising (using MAP inference) with noise level of 25. It is clearly shown that in terms of the number of dot-products versus the resulting PSNR, discriminatively training a small gating network on top of a GMM with 200 components is much better than a pure generative training of smaller GMMs.

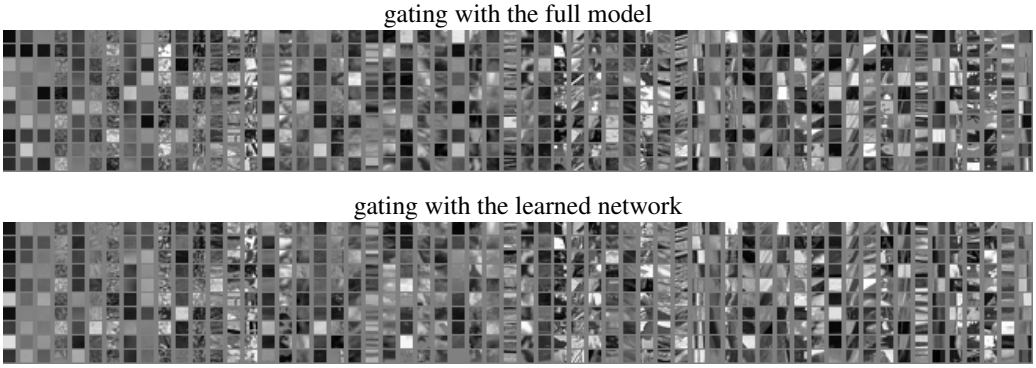


Figure 4: Gating with the full posterior computation vs. the learned gating network. Top: Patches from clean images arranged according to the component with maximum probability. Every column represents a different component (showing 60 out of 200). Bottom: Patches arranged according to the component with maximum gating score. Both gating methods have a very similar behavior.

4 Results

We compare the image restoration performance of our proposed method to several other methods proposed in the literature. The first class of methods used for denoising are "internal" methods that do not require any learning but are specific to image denoising. A prime example is BM3D. The second class of methods are generative models which are only trained on clean images. The original EPLL algorithm is in this class. Finally, the third class of models are discriminative which are trained "end-to-end". These typically have the best performance but need to be trained in advance for any image restoration problem.

In the right hand side of table 1 we show the denoising results of our implementation of EPLL with a GMM of 200 components. It can be seen that the difference between doing the full inference and using a learned gating network (with 100 vectors) is about 0.1dB to 0.3dB which is comparable to the difference between different published values of performance for a single algorithm. Even with the learned gating network the EPLL's performance is among the top performing methods for all noise levels. The fully discriminative MLP method is the best performing method for each noise level but it is trained explicitly and separately for each noise level.

The right hand side of table 1 also shows the run times of our Matlab implementation of EPLL on a standard CPU. Although the number of dot-products in the gating has been decreased by a factor of

σ	20	25	30	50	75
internal					
BM3D[22]		28.57		25.63	
BM3D[1]		28.35		25.45	23.96
BM3D[6]	29.25		27.32	25.09	
LSSC[22]		28.70		25.73	
LSSC[6]	29.40		27.39	25.09	
KSVD[22]		28.20		25.15	
generative					
FoE[22]		27.77		23.29	
KSVDG[22]		28.28		25.18	
EPLL[22]		28.71		25.72	
EPLL[1]		28.47		25.50	24.16
EPLL[6]	29.38		27.44	25.22	
discriminative					
$CSF^5_{7\times7}$[18]		28.72			
MLP[1]		28.75		25.83	24.42
FF[6]	29.65		27.48	25.25	

EPLL with different gating methods

σ	25	50	75	sec.
full	28.52	25.53	24.02	91
gating	28.40	25.37	23.79	5.6
gating_3	28.36	25.30	23.71	0.7

full: naive posterior computation.
gating: the learned gating network.
gating_3: the learned network calculated with a stride of 3.

Table 1: Average PSNR (dB) for image denoising. Left: Values for different denoising methods as reported by different papers. Right: Comparing different gating methods for our EPLL implementation, computed over 100 test images of BSDS. Using a fast gating method results in a PSNR difference comparable to the difference between different published values of the same algorithm.

noisy: 20.19 MLP: 27.31 full: 27.01 gating: 26.99

noisy: 20.19 MLP: 30.37 full: 30.14 gating: 30.06

Figure 5: Image denoising examples. Using the fast gating network or the full inference computation, is visually indistinguishable.

128, the effect on the actual run times is more complex. Still, by only switching to the new gating network, we obtain a speedup factor of more than 15 on small images. We also show that further speedup can be achieved by simply working with less overlapping patches ("stride"). The results show that using a stride of 3 (i.e. working on every 9'th patch) leads to almost no loss in PSNR. Although the "stride" speedup can be achieved by any patch based method, it emphasizes another important trade-off between accuracy and running-time. In total, we see that a speedup factor of more than 100, lead to very similar results than the full inference. We expect even more dramatic speedups are possible with more optimized and parallel code.

Figures 5 gives a visual comparison of denoised images. As can be expected from the PSNR values, the results with full EPLL and the gating network EPLL are visually indistinguishable.

To highlight the *modularity* advantage of generative models, figure 6 shows results of image deblurring using the same prior. Even though all the training of the EPLL and the gating was done on clean sharp images, the prior can be combined with a likelihood for deblurring to obtain state-of-the-art deblurring results. Again, the full and the gating results are visually indistinguishable.

9×9 blur: 19.12 Hyper-Laplacian: 24.69 full: 26.25 gating: 26.15

9×9 blur: 22.50 Hyper-Laplacian: 25.03 full: 25.77 gating: 25.75

Figure 6: Image deblurring examples. Using the learned gating network maintains the modularity property, allowing it to be used for different restoration tasks. Once again, results are very similar to the full inference computation.

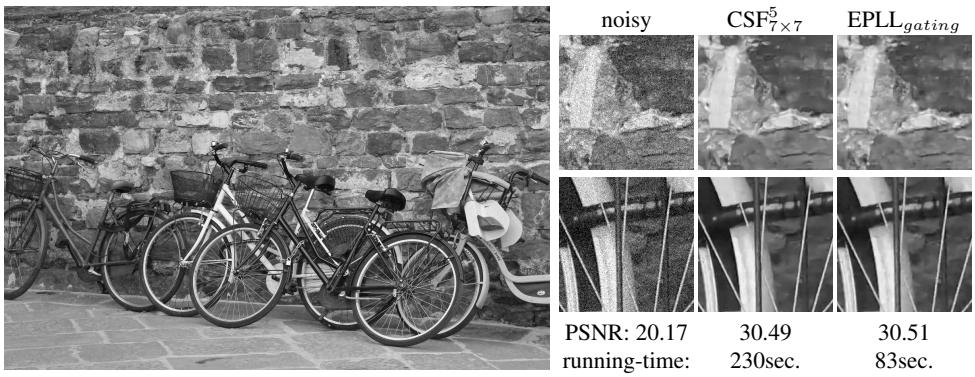


Figure 7: Denoising of a 18mega-pixel image. Using the learned gating network and a stride of 3, we get very fast inference with comparable results to discriminatively "end-to-end" trained models.

Finally, figure 7 shows the result of performing resotration on an 18 mega-pixel image. EPLL with a gating network achieves comparable results to a discriminatively trained method (CSF) [18] but is even more efficient while maintaining the modularity of the generative approach.

5 Discussion

Image restoration is a widely studied problem with immediate practical applications. In recent years, approaches based on machine learning have started to outperform handcrafted methods. This is true both for generative approaches and discriminative approaches. While discriminative approaches often give the best performance for a fixed computational budget, the generative approaches have the advantage of modularity. They are only trained on clean images and can be used to perform one of an infinite number of possible resotration tasks by using Bayes' rule. In this paper we have shown how to combine the best aspects of both approaches. We discriminatively train a feed-forward architecture to perform the most expensive part of inference using generative models. Our results indicate that we can still obtain state-of-the-art performance with two orders of magnitude improvement in run times while maintaining the modularity advantage of generative models.

Acknowledgements

Support by the ISF, Intel ICRI-CI and the Gatsby Foundation is greatfully acknowledged.

References

[1] Harold Christopher Burger, Christian Schuler, and Stefan Harmeling. Learning how to combine internal and external denoising methods. In *Pattern Recognition*, pages 121–130. Springer, 2013.

[2] Harold Christopher Burger, Christian J Schuler, and Stefan Harmeling. Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. *arXiv preprint arXiv:1211.1544*, 2012.

[3] Yunjin Chen, Thomas Pock, René Ranftl, and Horst Bischof. Revisiting loss-specific training of filter-based mrfs for image restoration. In *Pattern Recognition*, pages 271–281. Springer, 2013.

[4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

[5] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[6] Sean Ryan Fanello, Cem Keskin, Pushmeet Kohli, Shahram Izadi, Jamie Shotton, Antonio Criminisi, Ugo Pattacini, and Tim Paek. Filter forests for learning data-dependent convolutional kernels. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1709–1716. IEEE, 2014.

[7] Yacov Hel-Or and Doron Shaked. A discriminative approach for wavelet denoising. *Image Processing, IEEE Transactions on*, 17(4):443–457, 2008.

[8] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[9] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009.

[10] Yan Karklin and Michael S Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457(7225):83–86, 2009.

[11] Effi Levi. *Using natural image priors-maximizing or sampling?* PhD thesis, The Hebrew University of Jerusalem, 2009.

[12] Anat Levin and Boaz Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840. IEEE, 2011.

[13] Siwei Lyu and Eero P Simoncelli. Statistical modeling of images with fields of gaussian scale mixtures. In *Advances in Neural Information Processing Systems*, pages 945–952, 2006.

[14] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[15] Carl E Rassmusen. minimize.m, 2006. http://learning.eng.cam.ac.uk/carl/code/minimize/.

[16] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 860–867. IEEE, 2005.

[17] Uwe Schmidt, Qi Gao, and Stefan Roth. A generative perspective on mrfs in low-level vision. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1751–1758. IEEE, 2010.

[18] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2774–2781. IEEE, 2014.

[19] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[20] Benigno Uria, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pages 2175–2183, 2013.

[21] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *Image Processing, IEEE Transactions on*, 21(5):2481–2499, 2012.

[22] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.

[23] Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. In *NIPS*, pages 1745–1753, 2012.

2.4 Statistics of RGBD images

This section includes the following paper that was submitted for review:

Dan Rosenbaum and Yair Weiss. *Statistics of RGBD images.*

Statistics of RGBD Images

Dan Rosenbaum and Yair Weiss
School of Computer Science and Engineering
Hebrew University of Jerusalem

Abstract

Cameras that can measure the depth of each pixel in addition to its color have become easily available and are used in many consumer products worldwide. Often the depth channel is captured at lower quality compared to the RGB channels and different algorithms have been proposed to improve the quality of the D channel given the RGB channels. Typically these approaches work by assuming that edges in RGB are correlated with edges in D.

In this paper we approach this problem from the standpoint of natural image statistics. We obtain examples of high quality RGBD images from a computer graphics generated movie (MPI-Sintel) and we use these examples to compare different probabilistic generative models of RGBD image patches. We then use the generative models together with a degradation model and obtain a Bayes Least Squares (BLS) estimator of the D channel given the RGB channels. Our results show that learned generative models outperform the state-of-the-art in improving the quality of depth channels given the color channels in natural images even when training is performed on artificially generated images.

1 Introduction

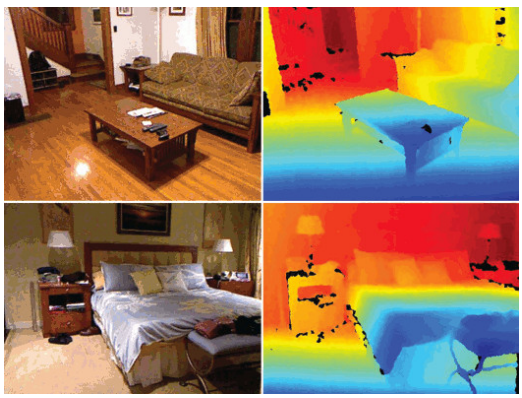


Figure 1: Examples of RGBD images from the NYU Depth V2 datatset. The depth channel often contains missing values and the depth is typically of lower resolution and more noisy than the RGB. In this paper we approach the problem of improving the D channel given RGB using natural image statistics.

Figure 1 shows examples from the NYU Depth V2 dataset [1]. Each scene is captured with a Kinect sensor and a color image is available along with a depth image. Ten years ago it may have been hard to believe that a depth image of such quality will be attainable with a sensor that costs less than 200 dollars, but today RGBD cameras are ubiquitous and have enabled a large suite of consumer

applications. Despite the impressive improvement in RGBD technology, the quality of the depth channel is still lacking. As can be seen in the figure, the depth channel often has missing pixels. Many of these missing pixels occur at object discontinuities where the different sensors used to measure depth have a viewpoint disparity. Others occur at specular objects. In addition, the depth image is often noisy and at a poorer resolution compared to the RGB channels.

In recent years, several authors have proposed improving the quality of the D channel based on the RGB channel [2, 3]. The vast majority of these approaches are based on assuming that depth edges are more likely to occur at intensity edges and this leads to a natural use of the joint bilateral filter [4, 5]. Silverman and Fergus [1] used the colorization by optimization framework of Levin et al. [6] to obtain a weighted least squares problem for filling in missing pixels where the weights are based on the assumption that neighboring pixels with similar colors should have similar depths.

As pointed out by Lu et al. [7], the assumption of correlation between color edges and depth edges may be insufficient to improve the quality of the depth image. In particular, they pointed out that both the color and the depth image are often subject to noise and that previous approaches did not handle this noise well. They suggested a statistical model of RGBD patches which is based on the assumption that similar patches in the image define a low rank matrix. Their approach outperformed approaches such as joint bilateral filtering, even when the color image was first denoised using a denoising algorithm.

In this paper we approach the problem of RGBD restoration from the standpoint of natural image statistics. We are motivated by the success of learning based methods that achieve excelllent performance in image restoration [8, 9, 10] by learning from a large database of clean images. In the case of RGBD the challenge is to obtain clean examples and we take advantage of a computer graphics generated movie (MPI-Sintel [11]) for this task. We use the clean examples to compare existing approaches and to learn new generative models for the patches. We then use the generative models together with a degradation model and obtain a Bayes Least Squares (BLS) estimator of the D channel given the RGB channels. Our results show that learned generative models outperform the state-of-the-art in improving the quality of depth channels given the color channels in natural images even when training is performed on artificially generated images.

2 Density models for depth

All methods for depth enhancement incorporate some assumption about the depth itself and sometimes about its dependence on the color channels. Typical assumptions are that the depth is usually smooth and that depth boundaries are correlated to color boundaries.

One way to compare different assumptions is to formulate them as density models for depth. Instead of using depth values in meters, we use the common representation of $1/depth$ or *disparity*. This has the advantage that background pixels with depth infinity which are very common translate to a mode in zero, and the precision is higher for closer objects.

We will evaluate the following density models, where d is a vector of disparity pixels:

DL2
The smoothness is modeled by giving a quadratic penalty to the spatial derivatives of disparity:

$$J(d) = \sum_p d_x(p)^2 + d_y(p)^2$$

where $d_x(p)$ and $d_y(p)$ are the x and y derivatives of disparity at pixel p. This can be formulated as a multivariate Gaussian over the disparity using a matrix A that takes all the derivatives of d. To make the covariance positive definite we add the indentity matrix times a small constant.

$$Pr(d) = \frac{1}{Z} e^{-\lambda \sum_p d_x(p)^2 + d_y(p)^2} \approx \frac{1}{Z} e^{-d^\top (\lambda A^\top A + \epsilon I) d} \tag{1}$$

DL1
The smootheness is modeled by giving an absolute value penalty to the spatial derivatives of disparity:

$$J(d) = \sum_p |d_x(p)| + |d_y(p)|$$

Figure 2: The Sintel dataset. Top: color images. Bottom: disparity=1/depth images. Using high quality depth images allows us to evaluate and learn density models.

This can be formulated as a multivariate Laplacian over d using the same derivative matrix A as above:

$$Pr(d) = \frac{1}{Z} e^{-\lambda \sum_p |dx(p)|+|dy(p)|} \approx \frac{1}{Z} e^{-\|(\lambda A+\epsilon I)d\|_1} \quad (2)$$

Here the normalization cannot be computed in closed form, making this model hard to use for measuring likelihood.

DL2|int

Here we use a weighted quadratic penalty on the derivatives of disparity, where the weights $w(p)$ depend on the color image:

$$J(d) = \sum_p w_x(p)\, d_x(p)^2 + w_y(p)\, d_y(p)^2$$

In order to encourage disparity edges to correlate with color edges, the weights are computed as a function of the color derivatives in the same location $c_x(p)$ and $c_y(p)$ as following:

$$w_x(p) = e^{-\frac{1}{\sigma^2} c_x(p)^2} \quad w_y(p) = e^{-\frac{1}{\sigma^2} c_y(p)^2}$$

giving derivatives that cross color edges a lower weight. This is the model of the colorization by optimization code [6] used in [1].

The model can be formulated as a conditional multivariate Gaussian over d using the same derivative matrix A and an additional diagonal weight matrix that depends on the color $W(c)$:

$$Pr(d|c) = \frac{1}{Z} e^{-\lambda \sum_p w_x(p) d_x(p)^2 + w_y(p) d_y(p)^2} \approx \frac{1}{Z} e^{-d^\top (\lambda A^\top W(c) A+\epsilon I)d} \quad (3)$$

For simplicity, and since we haven't noticed any significant difference, we reduce the RGB channels to a single intensity channel.

2.1 Evaluation of density models

The challenge in applying learning techniques to RGBD data is to obtain a large dataset of clean images. Previous works (e.g. [12]) used the output of a depth sensor in order to estimate the statistics but these statistics themselves may already be corupted. Here we use a highly realistic computer graphics generated dataset, the MPI-Sintel dataset [11] (figure 2). We divided the 23 scenes of Sintel to 16 training set scenes and 7 test set scenes. We follow roughly the approach of Rosenbaum and Weiss [13] and use the training set to tune the parameters λ and ϵ for each model and we use the test set to evaluate the different models.

2.1.1 Likelihood

The first way to evaluate the density models is by the likelihood on the test set. Since all density models need to integrate to 1 over all possible values, models that give high likelihood to a set of ground truth disparity images are models that capture frequent properties of the data. Figure 3 shows the resulting log-likelihood per pixel for the different models. We can see that the log-likelihood for DL2 and DL2|int are very similar. Since we can't compute exactly the noramlization contstant of DL1 we don't use it here.

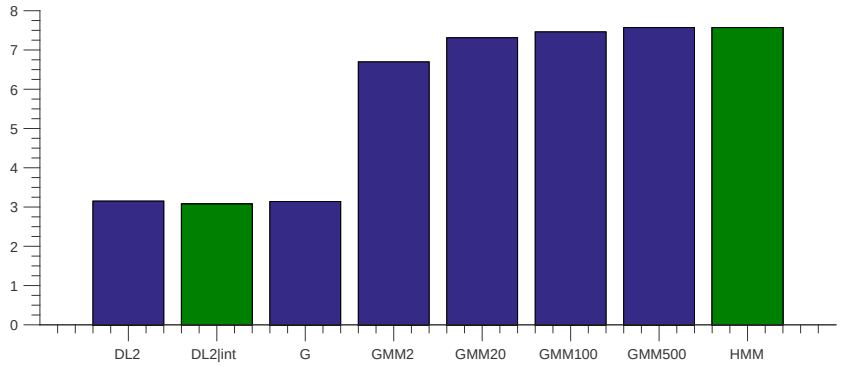


Figure 3: The log-Likelihood of hand-crafted density models and learned density models of disparity. A GMM model with enough components outperforms other models. Models that are conditioned on the intensity (shown in green) have a very similar log-likelihood to the unconditional models.

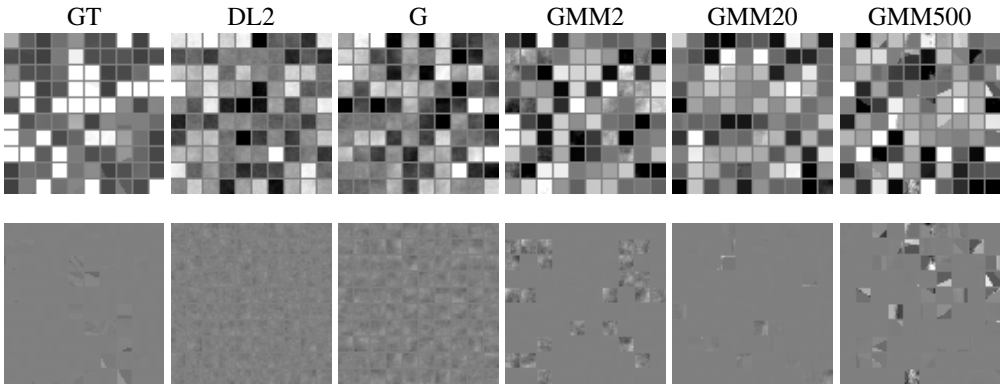


Figure 4: Patches from the ground truth (GT) vs. patches that were randomaly generated from different models. For better visibility, the bottom line shows the same patches with the DC substracted from each patch. Patches generated from a GMM with enough components exhibit similar properties as the ground truth: patches are usually very flat, and occasionally contain an edge.

Figure 5: Ground truth patches of disparity together with the corresponding intensity patch (all patches are shown without the DC). The correlation between intensity and disparity is not very strong: Intensity edges can occur with no corresponding disparity edge (due to texture), and disparity edges can occur with no corresponding intensity edge (due to motion blur and atmospheric effects).

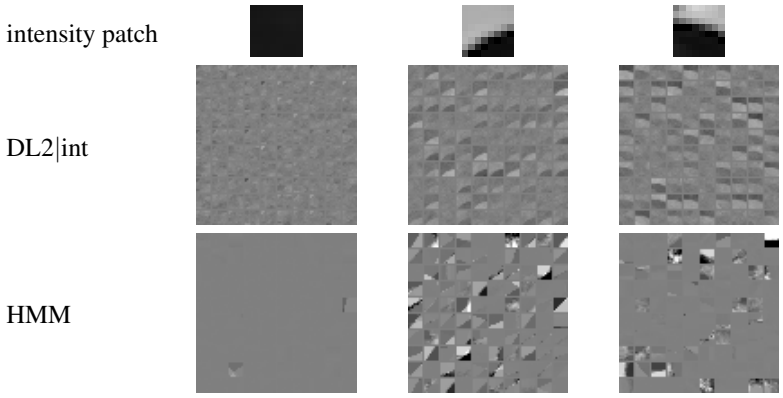


Figure 6: Disparity patches genereated conditionally given the intensity patches on the top. The DL2|int generates patches with edges that match exactly the intensity edge. The HMM can only approximate the edge form but can capture the distribution in its orientation and translation, and also the probability that the edge is missing.

2.1.2 Patch generation

A second way to evaluate the models is by using them to generate random data and testing for the visual similarity with ground truth data. We ommit DL1 from this test again since it does not allow for closed form generation of samples. Figure 4 shows ground truth 8×8 patches and patches generated from DL2. For better visibility we show all patches also with their DC (average value) subtracted. Looking at the ground truth disparity patches we can see that it is usually flat but occasionally contain a boundary edge. In comparison, patches generated from DL2 are a bit noisier and contain no structure.

In figure 5 we show the relationship between the disparity and intensity. The ground truth patches of disparity are shown together with the corresponding intensity patch. It can be seen that the relationship is not straightforward. First, in some cases both patches contain some structure which is not exactly correlated. Second, there are intensity edges without a corresponding disparity edge and there are disparity edge without a corresponding intensity edge. While the first direction can be attributed to many texture edges in intensity, the second direction which is perhaps more surprising is due to motion blur and atmospheric effects which are real effects that are deliberately modeled in the Sintel dataset[1].

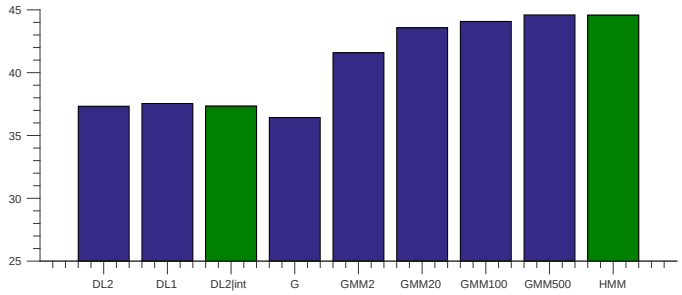
Figure 6 shows patches generated from DL2|int given 3 different patches of intensity. The generated patches usually match the intensity patch exactly, and sometimes do not contain a visible structure. The advantage of the patches generated with DL2|int over patches of DL2 is evident since it alows for spatial structure that is very similar to the ground truth patches, however it is not clear whether the dependence on the intensity is modeled correctly.

2.1.3 Patch restoration

A third way to evaluate density models is to use them in inference tasks and measure the quality of the results. Given ground truth patches we add noise using a known noise model and use Bayes

[1] we use Sintel's *final* pass of the intensity channel.

$\sigma = 5/255$



$\sigma = 15/255$

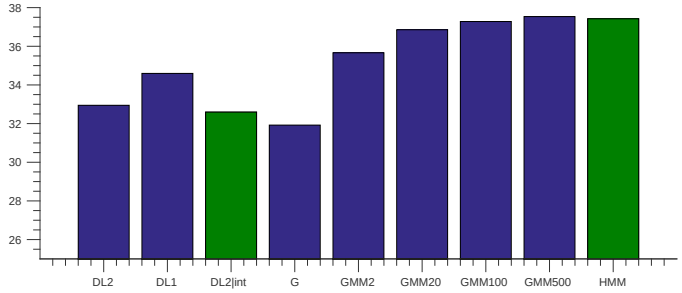


Figure 7: Patch denoising with different noise levels (average PSNR in dB). GMMs with enough components outperform all other models. Conditioning on the intensity does not lead to a significant improvement.

Least Squares (BLS) to estimate the clean patches again. We measure the quality of the estimation using the $PSNR = 10log_{10}(1/L)$, which is a function of the average squared loss over all restored patches:

$$L(\{\hat{d}\}) = \frac{1}{N}\sum_{i=1}^{N} \|\hat{d}_i - d_i\|_2^2$$

If the patches were generated from a known density model, then BLS inference with the true model would result in the optimal PSNR. Therefore we expect that BLS inference with models that are closer to the true density will result in a bigger PSNR.

Figure 7 shows the PSNR of BLS patch denoising using white Gaussian noise with 2 different standard deviations. Once again we cannot perform BLS inference using DL1 in closed form, instead we perform maximum a-posteriori (MAP) inference. We see that DL1 outperforms DL2 even though it is used with MAP inference which is sub-optimal. Figure 7 also shows that conditioning on the intensity does not lead to a significant improvement in patch denoising.

In figure 8, we show the results of patch inpainting where most of the patch is hidden and only 4 pixels in 2 corners are visible. This is equivalent to denoising with a noise model of very large variance in the hidden pixels. Here we see that conditioning on the intensity does lead to a significant improvement in the PSNR. The images on the bottom show some examples of the intensity, disparity, occluded disparity and restored disparity patches. We see that DL2|int does very well when there is a strong match between the disparity and intensity.

3 Learning density models

A natural question at this point is if we can use the available training set to learn better models of the disparity. Following the success in learning Gaussian Mixture Models (GMM) for natural image priors [8] and optical flow [13], we train a GMM model with a fixed mean and full covariance

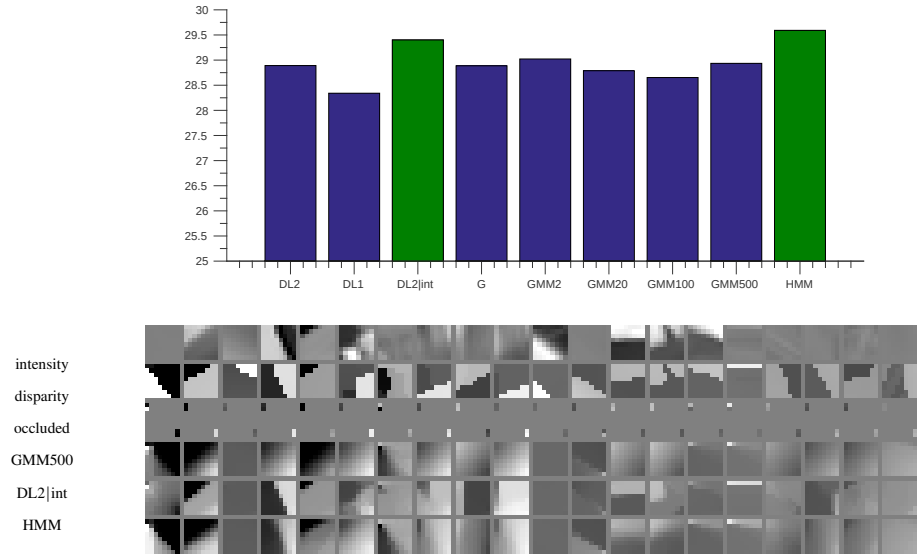


Figure 8: Patch inpainting: average PSNR in dB (top) and examples of restored patches (bottom). Conditioning on the intensity leads to a significant improvement. The HMM learned model outperforms all other models.

matrices over patches of 8×8 pixels:

$$Pr(d) = \sum_{k=1}^{K} \pi(k) \frac{1}{Z_k} e^{-\frac{1}{2}(d-d_0)^\top \Sigma_k^{-1}(d-d_0)} \tag{4}$$

We use the expectation maximization (EM) algorithm for training. The GMM has many parameters so we emphasize that the different evaluations are performed on a held-out test-set that was not used for training.

Figure 3 shows the log-likelihood on the test-set for a single Gaussian (G) and GMMs with a different number of components along with the hand-crafted models. We see that the Gaussian has a very similar log-likelihood to DL2, and that GMMs with enough components outperform other models.

Figure 4 shows patches that were randomly generated using the single Gaussian and the different GMMs. We see that (1) G has a very similar behavior as DL2, (2) GMM2 has mostly very flat patches and occasionally a noisy one, and (3) GMM100 and GMM500 capture the property that whenever a patch is not flat, it is likely to contain an edge with a certain orientation and translation. The patches generated by GMM500 appear very similar to the ground truth patches.

Figure 7 and Figure 8 show that also in terms of patch restoration, a GMM with enough components outperforms any independent model (which does not depend on intensity), however even a GMM with 500 components is outperformed by DL2|int when the dependence on intensity is critical, like in inpainting. The bottom image in figure 8 shows that it is hopeless to expect an independent model to recover some of the patches given only 4 visible pixels. In the next section we describe a learned conditional model, but first we elaborate on the GMM.

The GMM is a model with a single discrete hidden variable which is the index of the Gaussian component. This hidden component has a prior distribution which is the mixing-weights. The division of the 64 dimensional space of disparity patches into different components can be seen as a way to concentrate the density around different subspaces. Figure 9 shows how the space is divided as we train GMMs with more components: The first line shows what a single Gaussian learns. On the left we show the leading 5 eignevectors of the covariance matrix and on the right we show patches generated from the Gaussian. As we've seen before the behavior is very similar to DL2 which is also a Gaussian model. The second and third line show the leading eignevectors of the covariance and generated samples from the 2 components of GMM2. We see that there is an explicit

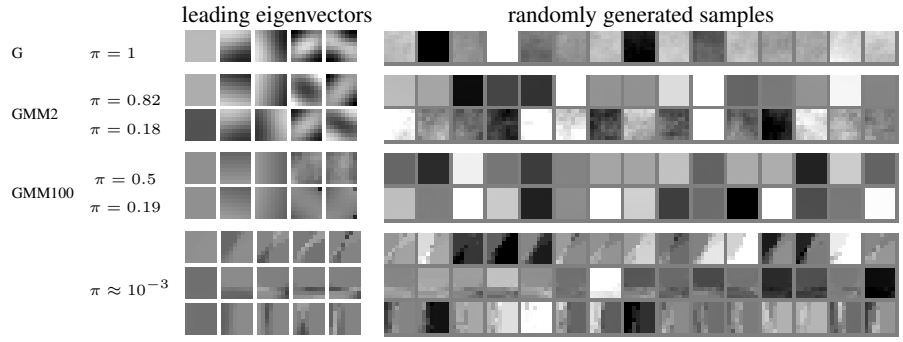


Figure 9: Leading eigen-vectors and generated samples from the single Gaussian, from the 2 components of GMM2 and from some of the components of GMM100. As more components are used, the GMM learns to explicitly model flat patches and edges with different orientations and translations.

division between very flat patches that occur in probability 0.82 (as shown by the mixing weight on the left), and noisy patches in probability 0.18. When we train GMMs with more components we see the explicit assignment of every component to either a flat patch or to a patch with an edge in a certain orientation and translation. We show here only a subset of 5 components.

3.1 Learning the dependence on intensity

In order to capture a possible dependence on intensity, we train on top of the GMM500 another model called an HMM as was done in [13]. The HMM is built of 2 GMMs: the first is a GMM over the intensity like in [8], and the second one is a GMM over the disparity but instead of having independent mixing weights (i.e. a prior on the component), the disparity component depends on the intensity component through a transition matrix. The HMM is equivalent to having a GMM model over the disparity with mixing weights that change according to the intensity. Since the intensity GMM also assigns different components to different orientations and translations of edges, this allows the occurrence of intensity edges to give a higher prior for disparity edges in the same orientation and translation.

Looking at the generated samples in figure 6 we see that this is exactly what the HMM does. Given an intensity edge, disparity edge components with similar orientation and translation become more likely. Note that this intensity dependent prior is 'soft' and allows also flat patches and edges in very different orientation and translation to occur but in a lower probability. If we compare the HMM samples to the DL2|int samples we see that DL2|int has the advantage of being able to exactly match the intensity edge however it lacks the power of the HMM to model the non-negligible probability of similar orientations and translation of edges as the ground truth data also exhibits in figure 5.

In terms of log-likelihood and patch restoration, the HMM model is superior to all other models in all the different evaluations. It has similar results to the GMM500 in log-likelihood (figure 3), and patch denoising (figure 7), and outperforms it when the dependence on intensity is needed for inpatining (figure 8). For inpainting it also outperforms the hand-crafted conditional model DL2|int.

4 Disparity estimation in full images

Given the superior performance on patches, we would like to use the learned models to perform disparity estimation in full images. As long as the degredation in disparity is local and contains noise and small holes, a simple approach is to perform patch restoration on all overlapping patches in the image and average the results over overlapping pixels. However, when there are big holes as in the dataset used in [7], global inference is needed. While the hand-crafted models DL2, DL1 and DL2|int can be extended to a full image model, for the GMMs it is not feasible. The reason is that extending a mixture model over patches to an image with thousands or milions of patches would require to go over all the combination of mixture components. Moreover, since the model was learned over patches it cannot capture the depndence between neighboring (or even overlapping)

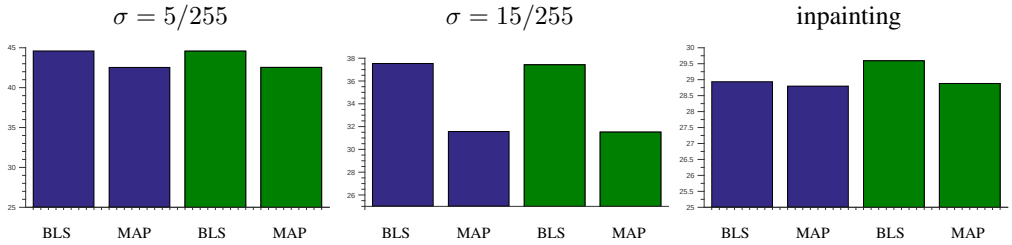


Figure 10: BLS vs MAP inference for the GMM500 (blue) and HMM (green) models. MAP inference is inferior in both patch denoising and inpainting.

combined-BLS	colorization	LRC	JBF	NLM	SGF	SHF	GIF
40.2	36.7	39.3	37.9	37.2	33.9	36.5	37.0

Table 1: Average PSNR (dB) of combined-BLS, colorization and the methods compared in [7].

patches. One option is to treat all patches as independent and perform global MAP inference. This is shown to work succefully in the EPLL framework of [8]. Another implementation of global MAP inference can be done using the EM-MAP method [14]. This is performed iteratively by building a sparse inverse covariance matrix over the whole image and inverting it in each iteration.

However, one drawback of these methods is that even if the optimization succeeds, the MAP solution is not guarenteed to have good performance even for good density models. In fact, if we evaluate the result of MAP inference over patches we see that it is significantly inferior to BLS inference (see [15] for a similar result in image restoration). Figure 10 shows that the performance drops for both denoising and inpaintining once we turn to MAP inference. For inpainting we see that the gap between the HMM and the GMM, which was due to the dependence on intensity, disappears. The performance of HMM-MAP is also worse than the performance of DL2|int (for which MAP and BLS inference are the same).

Therefore, in order to restore a given disparity image that contains noise and holes, we do the following 2 steps:

1. We perform BLS inference using the HMM over all overlapping patches in the image and average the results over overlapping pixels.

2. Using the resulting image, we perform global BLS inference on the large holes using the DL2|int model.

We run this procedure, on the online availabe dataset used by Lu et al [7] which consists of 30 images from Middlebury [16] and 9 images from the RGBZ dataset [17]. The noisy intensity image is denoised using EPLL [8]. We compare our proposed method, termed *combined-BLS*, to the global *colorization* method used by [1] (equivalent to performing only global inference with DL2|int), and to the methods that were compared in [7]. These methods include the Joint Bilateral Filter (JBF) [5] and the LRC method of Lu et al. that assumes that concatenated vectors of disparity patches and corresponding color patches lie in a low rank subspace. Our proposed method acheives an improvement in average PSNR of almost 1dB over the state-of-the-art results of LRC.

Table 1 shows the average PSNR of the methods: combined-BLS (our method), colorization, and the different methods that were compared in [7]. Figure 11 shows examples of our results compared to LRC and the colorization method. We emphasize that even though the models were trained on the synthetic data of Sintel, we acheive a significant improvement on the Middlebury+RGBZ dataset of real images.

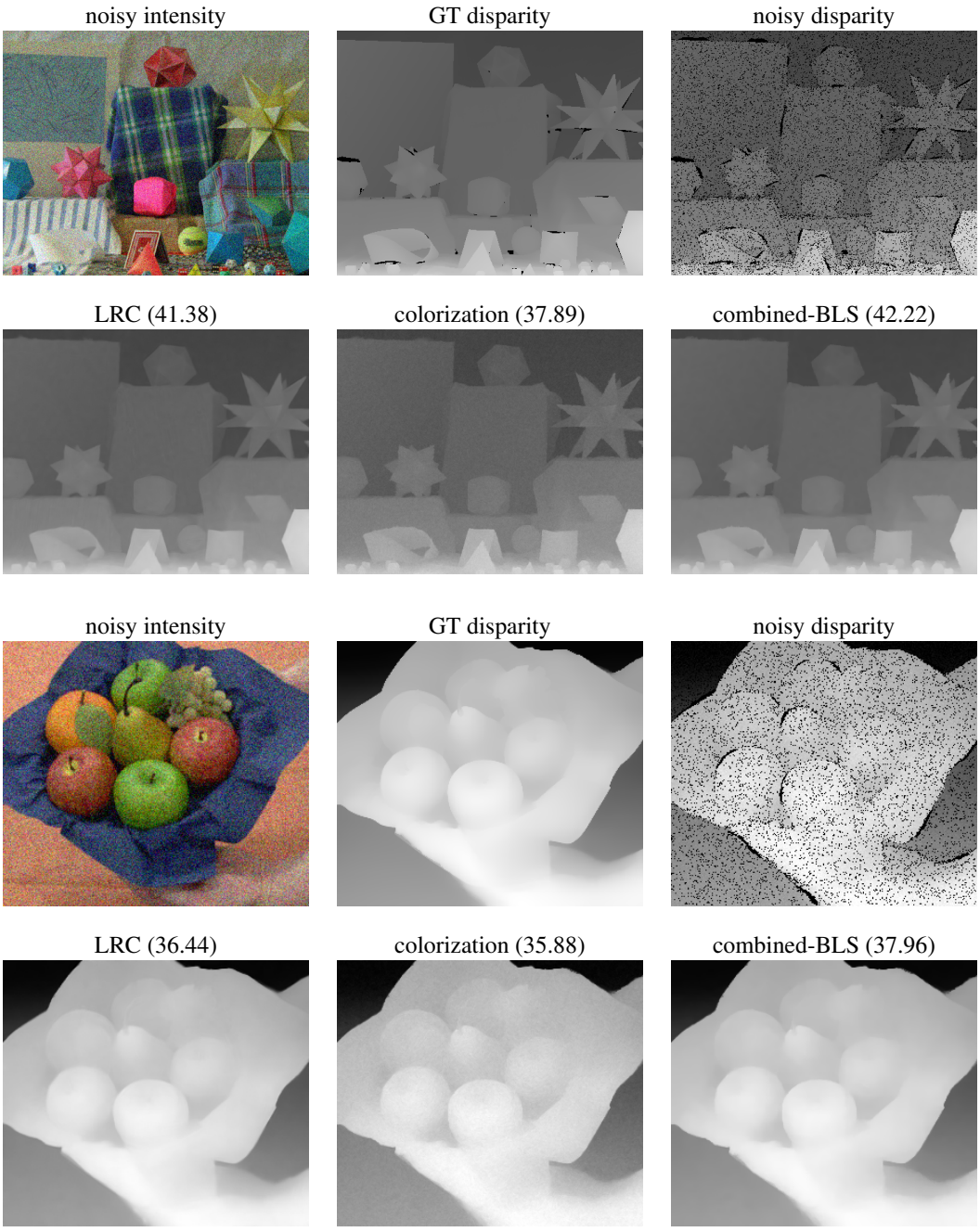


Figure 11: Examples of disparity images enhanced with LRC, colorization and combined-BLS. PSNR values are in dB.

5 Discussion

An advantage of using learning based approaches for vision is that we can compare what is learned to assumptions commonly made by Computer Vision researchers. The majority of previous approaches to improving D given RGB used the assumption that depth edges are correlated with intensity edges and assumed very little additional structure on the depth. In this paper we have shown that a generative model that is learned from ground truth RGBD patches indeed finds a correlation between depth edges and intensity edges but this correlation is relatively weak. At the same time, the generative model learns very strong structural constraints on the depth: that depth patches are usually either flat or edges. By using a learned model that combines both the depth structure and the correlation with intensity we were able to outperform the state-of-the-art in improving the quality of the depth channel given RGB. Even though our training was performed on synthetic images, we gained a significant advantage (about 1dB on average) in restoring real images.

References

[1] Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV. (2012)

[2] Liu, J., Gong, X., Liu, J.: Guided inpainting and filtering for kinect depth maps. In: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE (2012) 2055–2058

[3] Liu, S., Wang, Y., Wang, J., Wang, H., Zhang, J., Pan, C.: Kinect depth restoration via energy minimization with tv 21 regularization. In: Image Processing (ICIP), 2013 20th IEEE International Conference on, IEEE (2013) 724–724

[4] Qi, F., Han, J., Wang, P., Shi, G., Li, F.: Structure guided fusion for depth map inpainting. Pattern Recognition Letters **34**(1) (2013) 70–76

[5] Richardt, C., Stoll, C., Dodgson, N.A., Seidel, H.P., Theobalt, C.: Coherent spatiotemporal filtering, upsampling and rendering of rgbz videos. In: Computer Graphics Forum. Volume 31., Wiley Online Library (2012) 247–256

[6] Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM Transactions on Graphics (TOG). Volume 23., ACM (2004) 689–694

[7] Lu, S., Ren, X., Liu, F.: Depth enhancement via low-rank matrix completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3390–3397

[8] Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 479–486

[9] Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 2774–2781

[10] Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. arXiv preprint arXiv:1211.1544 (2012)

[11] Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), ed.: European Conf. on Computer Vision (ECCV). Part IV, LNCS 7577, Springer-Verlag (October 2012) 611–625

[12] Huang, J., Lee, A.B., Mumford, D.: Statistics of range images. In: 2000 Conference on Computer Vision and Pattern Recognition (CVPR 2000), 13-15 June 2000, Hilton Head, SC, USA. (2000) 1324–1331

[13] Rosenbaum, D., Zoran, D., Weiss, Y.: Learning the local statistics of optical flow. In: Advances in Neural Information Processing Systems. (2013) 2373–2381

[14] Levi, E.: Using natural image priors-maximizing or sampling? PhD thesis, The Hebrew University of Jerusalem (2009)

[15] Schmidt, U., Gao, Q., Roth, S.: A generative perspective on mrfs in low-level vision. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 1751–1758

[16] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92**(1) (2011) 1–31

[17] Richardt, C., Stoll, C., Dodgson, N.A., Seidel, H.P., Theobalt, C.: Coherent spatiotemporal filtering, upsampling and rendering of rgbz videos. Computer Graphics Forum **31**(2pt1) (2012) 247–256

Chapter 3

Discussion

The work presented in this dissertation, demonstrates different ways in which machine learning can be used in low level vision problems. The basic approach we use is the generative approach, modeling the generation of the data and the observation. However, we show that by also learning the inference, determining the way in which the generative models will be used at test time, we can benefit from advantages usually associated with a direct discriminative approach.

In the work presented in 2.3, we show that typically what makes the inference hard is the need to predict the state of the hidden variables in generative models. This idea has already been present in the Helmholtz machine by Dayan et al. [10], which is a hierarchical feed-forward generative model coupled with a "backwards" inference network. This generative model can easily be used for sampling data from the modeled distribution, however, since it consists of several layers of hidden states, it is not practical for inference - computing the posterior probability of the hidden variables requires the integration over all hidden states. The inference model is used to overcome this difficulty by approximating the posterior of hidden states.

The original paper from 1995 showed how the generative and inference models can be trained together using the Wake-Sleep algorithm. Later however, it was shown that this training method is not good enough and results in models that are worse than simple tractable models [16], and this direction was largely abandoned. In recent years, there has been a resurgence of the Helmholtz machine idea using new training methods, first with a variational inference approach [25, 37], and later by making some modifications to the original Wake-Sleep method [4].

However, in all of the above examples, the inference model is used only to allow an efficient training of a complex generative model. Perhaps a more fundamental aspect of learning the inference comes from the basic trade-offs of using prior knowledge in both the generative and discriminative directions. When approaching a new problem there can be different kinds of prior knowledge. In the generative direction: the noise model can be given in advance, some knowledge about the causal relationships of variables etc.; but there is also prior knowledge in the inference direction, regarding how the inference will be carried out, e.g. running time and hardware constraints. Making the choice to either use a generative approach or a discriminative approach means that some of the prior knowledge is thrown away.

As an illustration, think of the problem of estimating x when the observed data is known to be generated by $y = Ax + \xi$ where both A and the statistics of the noise ξ are known, and in addition, there is a constraint that the inference should be a linear predictor. Given a training set of x, y pairs, what is the

best approach to train the linear predictor? If we just search for the best linear predictor on the training data, then the knowledge of the noise process is not used, but if we just fit a model to x it is not clear how to use it to come up with a good linear predictor.

In our work we give a simple example of how this trade-off can be solved for image restoration with mixture model priors, where predicting the hidden state boils down to choosing which mixture component to use, also termed "gating". However I think that the idea of learning both the generative and the inference directions can be effectively applied in different ways to more complex models and to many additional problems.

3.1 Future work

Learning global inference with local generative models

A major factor in the difficulty to find good models for natural images and low level vision comes from the large dimension of data. The EPLL method [48] shows that sometimes good local models of small patches can be more useful than approximate global models of the whole image. In the EPLL method the local models are combined when performing global inference over the whole image, using a very simple and naive approximation - that all patches are independent. This assumption clearly doesn't hold since neighboring patches have a strong dependence, and even more so, overlapping patches that share many pixels.

Provided that learning good models for bigger patches and whole images still isn't successful, one way to improve the inference is to learn the best way to combine the patch models at inference time. Instead of assuming that models are independent, or trying to model the dependency, it might be more useful to directly learn how to perform global inference with the local models. For example, the work in 2.3 could be extended by learning gating networks that depend on the whole image or at least larger patches, i.e. when using mixture models such as GMMs over patches, predicting the posterior of the mixture components could depend on the whole image rather than the patch alone. This will capture the dependency between neighboring patches, for example when a patch is located on a long edge that crosses the image, the posterior probability that the patch also contains an edge will increase.

Learning the inference for optical flow estimation

In the work presented in sections 2.1 and 2.2, we show how to improve the energy function for optical flow estimation. Using the ground-truth data of the MPI-Sintel dataset [8] we learn prior models and noise models and show that they are better than existing models used in common optical flow estimation methods. The models that we learn and evaluate are on small patches, and while for small patches we see their superiority over other models in terms of likelihood and different inference tasks, when trying to use these models for full optical flow estimation over whole images we don't see an improvement relative to common methods.

Two possible reasons for that are: (1) Unlike common image restoration problems, in optical flow the "noise" model $Pr(I_2|I_1, v)$, that generates the observed second frame given the first frame and the flow field, is very non-local. Objects can move from one side of the image to the other and so using only local models that look at small patches might be too naive. (2) Optical flow energy functions are hard to optimize and require many optimization heuristics such as coarse-to-fine iterations, Newton steps

which include the linearization of the warping function etc. The hope that improving the energy function will lead to better results stems from the assumption that all the relevant knowledge for optical flow estimation is encoded in the energy function and the rest is just optimization. This assumption might simply be not true, and possibly optical flow methods succeed because their optimization heuristics are specially tailored for optical flow and in a way encode additional assumptions about the solution than the energy function.

A possible way to extend the work in sections 2.1 and 2.2 is to learn the inference, i.e how to use the learned models for optical flow estimation of whole images. Specifically by "unrolling" an inference process on the learned models which includes coarse-to-fine iterations, linearization of the warping and gating the components in the learned mixture models, the parameters of those operations can be learned discriminatively by minimizing the loss of the inference output compared to the ground-truth.

Bibliography

[1] Guillem Alenyà, Sergi Foix, and Carme Torras. Using tof and rgbd cameras for 3d robot perception and manipulation in human environments. *Intelligent Service Robotics*, 7(4):211–220, 2014.

[2] Anthony J Bell and Terrence J Sejnowski. The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

[3] Michael J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.

[4] Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. *arXiv preprint arXiv:1406.2751*, 2014.

[5] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[6] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005.

[7] Harold Christopher Burger, Christian J Schuler, and Stefan Harmeling. Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds. *arXiv preprint arXiv:1211.1544*, 2012.

[8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[9] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

[10] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[12] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[13] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.

[14] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 787–794. ACM, 2006.

[15] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.

[16] Brendan J Frey, Geoffrey E Hinton, and Peter Dayan. Does the wake-sleep algorithm produce good density estimators? In *Advances in neural information processing systems*, pages 661–670. MORGAN KAUFMANN PUBLISHERS, 1996.

[17] David Gadot and Lior Wolf. Patchbatch: a batch augmented loss for optical flow. *arXiv preprint arXiv:1512.01815*, 2015.

[18] Donald Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995.

[19] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[20] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

[21] Yacov Hel-Or and Doron Shaked. A discriminative approach for wavelet denoising. *Image Processing, IEEE Transactions on*, 17(4):443–457, 2008.

[22] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.

[23] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009.

[24] Yan Karklin and Michael S Lewicki. Emergence of complex cell properties by learning to generalize in natural scenes. *Nature*, 457(7225):83–86, 2009.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, volume 22, pages 1–9, 2009.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[28] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007.

[29] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 689–694. ACM, 2004.

[30] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.

[31] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[32] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[33] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[34] Bruno A Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[35] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image processing*, 12(11):1338–1351, 2003.

[36] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *Computer Vision and Pattern Recognition*, 2015.

[37] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[38] William Hadley Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.

[39] John P Rossi. Digital techniques for reducing television noise. *Smpte Journal*, 87(3):134–140, 1978.

[40] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 860–867. IEEE, 2005.

[41] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2774–2781. IEEE, 2014.

[42] Eero P Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 1, pages 673–678. IEEE, 1997.

[43] Lucas Theis, Sebastian Gerwinn, Fabian Sinz, and Matthias Bethge. In all likelihood, deep belief is not enough. *Journal of Machine Learning Research*, 12(Nov):3071–3096, 2011.

[44] Benigno Uria, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, pages 2175–2183, 2013.

[45] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[46] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015.

[47] Song Chun Zhu and David Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, 1997.

[48] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.

תקציר

בבעיות של ראייה ראשונית כמו שחזור תמונה או שיערוך תנועה בין תמונות בסרט, הפלט הרצוי הוא בעצמו תמונה המכילה מידע רב-ממדי בעל מבנה דו-ממדי. רוב הבעיות מהסוג הזה אינן ניתנות לפתרון ללא הנחות נוספות על הפלט. הממד הגבוה והמבנה העשיר שמצוי בתמונות טבעיות מקשים על פיתוח ידני של שיטות פתרון ומזמינים גישות מבוססות מידע, בהם ההנחות נלמדות ישירות מנתונים. השימוש בלמידה חישובית הוא לפיכך טבעי לבעיות אלה, והוא מאפשר ללמוד ולנצל את המבנה המצוי בתמונות באופן אוטומטי. אולם עד כה, השימוש בלמידה חישובית בבעיות של ראייה ראשונית עדין לא הוביל אל השיפורים הצפויים. למרות שעבור בעיות קלות יותר כמו שחזור תמונה התוצאות עולות על אלו של שיטות מהונדסות ידנית, עבור בעיות קשות יותר כמו שערוך תנועה, הביצועים של גישות מבוססות מידע כמו גם אלה של הגישות המהונדסות עדיין לא מספקים.

ניתן להשתמש בלמידה חישובית בדרכים שונות. בגישה הגנרטיבית, ההנחות על המבנה של המידע מנוסחות כמודלים הסתברותיים שנלמדים מנתונים. לאחר מכן השערוך מתבצע כבעיית הסקה בשימוש בחוק בייס. בגישה הדיסקרימינטיבית, לומדים ישירות את המשערך לבעיה מסוימת, וכך נמנע הצורך בביצוע תהליך של הסקה בזמן מבחן. היתרון בגישה הגנרטיבית הוא שלפעמים יותר טבעי לנסח בה הנחות שונות על המידע, מה שמוביל לאימון מהיר יותר ואפשרות לשימוש מודולרי במשערך על ידי שינוי של מרכיבים שונים שלו בזמן מבחן. היתרון בגישה הדיסקרימינטיבית הוא שלרוב היא מובילה למשערך מהיר יותר. זה קורה משום שבעוד שתהליך הסקה עם מודל גנרטיבי לרוב מכיל בעיית אופטימיזציה קשה בזמן מבחן, בגישה הדיסקרימינטיבית, כל האופטימיזציה מתבצעת מראש בזמן האימון על ידי מציאת המשערך הטוב ביותר בהינתן ארכיטקטורה ואילוצים שונים על זמן הריצה.

בעבודה שמוצגת כאן, אנו מדגימים את הדרכים השונות בהן ניתן להשתמש בלמידה חישובית עבור בעיות של ראייה ראשונית. בהסתכלות כללית יותר על הגישה הגנרטיבית, ניתן לחלק אותה לשלושה מרכיבים: (1) מודל אפריורי שמנסח את ההנחות המקדימות על המידע החבוי אותו רוצים לשערך, (2) מודל הנראות שמנסח את ההנחות על האופן בו נוצר הקלט הנראה בהינתן המידע החבוי, ו-(3) תהליך ההסקה אשר משתמש במודל האפריורי ובמודל הנראות כדי לשערך את המידע החבוי.

התוצאות שלנו מוצגות בארבעה מאמרים. במאמר הראשון והשני אנו מראים איך ניתן לחלץ הנחות שונות מתוך שיטות מהונדסות ידנית של שערוך תנועה ולנסח אותן כמודלים אפריוריים ומודלי נראות. לאחר מכן אנו בוחנים את המודלים השונים ומראים כיצד ניתן ללמוד מודלים טובים יותר ישירות ממידע אמת. במאמר השלישי אנו מראים איך בהנתן מודל אפריורי, ניתן ללמוד ישירות מנתונים את תהליך ההסקה עבור שחזור תמונה. אנו מראים שהשיטה מובילה למשערך שמשלב את היתרונות של הגישות הדיסקרימינטיבית והגנרטיבית, בכך שהוא גם מהיר בזמן מבחן וגם משמר את תכונת המודולריות שמאפשרת שימוש עבור בעיות שחזור שונות כמו ניקוי רעשים והסרת טשטוש, ללא אימון מחודש. המאמר הרביעי עוסק בבעיית שיפור מפת עומקים במצלמות RGBD. אנו מראים שעל ידי בחינה ולמידה של מודלים בשימוש במידע אמת, ניתן להשיג תוצאות טובות יותר מהשיטות הקיימות עבור שיפור מפת עומקים.

עבודה זו נעשתה תחת הדרכתו של
פרופ' יאיר וייס

למידה של מודלים גנרטיביים ותהליך ההסקה בראייה ראשונית

חיבור לשם קבלת תואר דוקטור לפילוסופיה

מאת

דן רוזנבאום

הוגש לסנט האוניברסיטה העברית בירושלים

אוגוסט 2016