

Multiple View Image Denoising

by

Sundeep Vaddadi

A thesis submitted in partial fulfillment of the
requirements for the degree of

Master of Science
(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2009

APPROVED

By

Advisor Signature: _____

Advisor Title: Professor William Sethares

Date: _____

Abstract

A pinhole camera (large depth of field) image capture is essential in many computer vision applications such as Simultaneous Localization and Mapping, 3D reconstruction, video surveillance. For these applications obtaining a set of clean images (less noise, less blur) is important. In this thesis, we propose a new approach to acquiring pinhole images using many pinhole cameras. The cameras can be distributed spatially to monitor a common scene, or compactly assembled as a camera array. Each camera uses a small aperture and short exposure to ensure minimal optical defocus and motion blur. Under such camera settings, the incoming light is very weak and the images are extremely noisy. We cast pinhole imaging as a denoising problem and seek to restore all the pinhole images by jointly removing noise in different viewpoints. Our Multi-view denoising can be used as a prior to the applications mentioned above. Our algorithm takes noisy images taken from different viewpoints as input and groups similar patches in the input images using depth estimation. We model intensity-dependent noise in low-light conditions and use the principal component analysis and tensor analysis to remove such noise. The dimensionalities for both PCA and tensor analysis are automatically computed in a way that is adaptive to the complexity of image structures in the patches. Our method is based on a probabilistic formulation that marginalizes depth maps as hidden variables and therefore does not require perfect depth estimation. We validate our algorithm on both synthetic and real images with different content. Our algorithm compares favorably against several state-of-the-art denoising algorithms.

B.E in Electronics and Communications Engineering - 2007

Osmania University, College of Engineering, India

Research Publication

Li Zhang, Sundeep Vaddadi, Hailin Jin, and Shree Nayar “Multiple View Image Denoising”, In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2009

Fields of Study

Image processing, Computer vision

Acknowledgements

Through the course of my education in Wisconsin- Madison and as an intern in Qualcomm as part of my Masters I had the privilege to meet and work with some very smart and more importantly good people.

I would like to thank Asst. Professor Li Zhang for giving me the opportunity to work with him in his lab. This thesis and its accompanying paper would not have been possible without his guidance and contributions.

I would like to thank Professor William Sethares, my advisor, whose support in a very trying time made the completion of this work possible. He had the patience to listen to some of my very self conflicting statements and guide me.

I would like to thank Dr. John Hong, my supervisor and mentor in Qualcomm. I wonder how he listened to and understood my endless ‘ravings’. Yes that is the only word. He guided me through what I feel as one of the most important phases I will go through. I will be working with him in Qualcomm R&D for sometime to come in the future and it should be very exciting.

I would like to thank my parents and sister whose continuous support was and is essential to me. They might agree or disagree with my ideas but they always helped me in doing what I want. I will not attempt to put my thanks to them in words (well, I did).

Table of Contents

<i>Abstract</i>	<i>i</i>
<i>vita</i>	<i>ii</i>
<i>Acknowledgements</i>	<i>iii</i>
<i>Table of Contents</i>	<i>iv</i>
<i>List of Figures</i>	<i>v</i>
<i>List of Tables</i>	<i>ix</i>
<i>Introduction</i>	<i>1</i>
<i>Chapter 1 Literature Survey</i>	<i>4</i>
<i>1.1 Literature Overview</i>	
<i>1.2.1 Single image denoising</i>	
<i>1.2.2 Extension to video denoising</i>	
<i>1.2.3 Synthetic aperture denoising</i>	
<i>1.3 Multiple image denoising</i>	
<i>Chapter 2 Problem statement</i>	<i>8</i>
<i>2.1 Problem statement and solution overview</i>	

Chapter 3 Depth guided multiview image denoising11

3.1 Depth guided multi view image denoising

3.2 Joint multi view patch matching

3.3 Joint multi view patch denoising

3.3.1 Patch denoising using PCA

3.3.2 Finding dimensionality C

3.3.3 Patch denoising using tensor analysis

3.3.4 From denoised patches to denoised images

Chapter 4 Data set.....21

4.1 Datasets

4.1.1 Tsukuba image dataset - Middlebury University

4.1.2 Tarot image dataset - Stanford light archives

4.1.3 Evolve image dataset - Univ. Wisconsin -Madison

4.1.4 Text image dataset - Univ. Wisconsin - Madison

4.1.5 Line Art image dataset - Univ. Wisconsin - Madison

4.1.6 Usage of datasets

4.2.1 Input image data

4.2.2 Comparison to other denoising approaches

4.2.3 Intensity dependent variance

4.2.4 PCA vs. tensor analysis

4.2.5 How many views are enough?

4.2.6 Optimization vs. sampling depth maps

Chapter 5 Discussion.....37

Chapter 6 Additional results.....38

References.....46

List of Figures

2.1	Noise variance model	-----	9
3.2	Comparison between patch groupings	-----	12
3.3.2	Eigen values for collection of patches	-----	17
4.1.1	Tsukuba stereo data set	-----	21
4.1.2	Tarot data set	-----	22
4.1.3	Evolve data set	-----	23
4.1.4	Text data set	-----	24
4.1.5	Line art data set	-----	25
4.2.2	Comparison between denoising algorithms ¹	-----	28

4.2.2	Comparison between denoising algorithms ² -----	30
4.2.2	Depth estimation as an application -----	31
4.2.3	Fixed variance vs. Intensity dependent variance -----	32
4.2.4	PCA vs. tensor analysis -----	33
4.2.5	PSNR vs. No. of input views -----	34
4.2.6	Denoising using random sampled depth map -----	36
6	Additional results -----	38

List of Tables

4.2.2 Gaussian synthetic noise: Comparison of PSNR between denoising algorithms –29

4.2.2 Poisson synthetic noise: Comparison of PSNR between denoising algorithms ---29

4.2.2 Real image noise: Comparison of PSNR between denoising algorithms -----30

Introduction

Capturing a pinhole image (large depth-of-field) is important to many computer vision applications, such as 3D reconstruction, motion analysis, and video surveillance. For a dynamic scene, capturing pinhole images however is difficult: we have often to make a tradeoff between depth of field and motion blur. For example, if we use a large aperture and short exposure to avoid motion blur, the resulting images will have small depth-of-field; otherwise, if we use a small aperture and long exposure, the depth-of-field will be large, but at the expense of motion blur.

In this work, we propose a new approach to acquiring pinhole images using many pinhole cameras. The cameras can be distributed spatially to monitor a common scene, or compactly assembled as a camera array. Each camera uses a small aperture and short exposure to ensure minimal optical defocus and motion blur. Under such camera settings, the incoming light is very weak and the images are extremely noisy. We cast pinhole imaging as a denoising problem and seek to restore all the pinhole images by jointly removing noise in different viewpoints.

Using multi-view images for noise reduction has a unique advantage: pixel correspondence from one image to all other images is determined by its single depth map. This advantage contrasts with video denoising, where motion between frames in general has many more degrees of freedom. Although this observation is a common sense in 3D vision, we are the first to use it for finding similar image patches in multi view

denoising. Specifically, our denoising method is built upon the recent development in image denoising literature, where similar image patches are grouped together and “collaboratively” filtered to reduce noise. When considering whether a pair of patches in one image is similar or not, we simultaneously consider the similarity between corresponding patches in all other views using depth estimation. This depth guided patch matching improves patch grouping accuracy and substantially boosts denoising performance, as demonstrated later in this thesis. The main contributions of this thesis include:

- Depth guided denoising: Using depth estimation as a constraint, our method is able to group similar image patches in the presence of large noise and exploit data redundancy across views for noise removal.
- Removing signal dependent noise: In low light conditions, photon noise is manifest whose variance depends on its mean. We propose to use the principal component analysis and tensor analysis to remove such noise.
- Adaptive noise reduction: For both PCA and tensor analysis, we propose an effective scheme to automatically choose dimensionalities in a way that is adaptive to the complexity of image structures in the patches.
- Tolerance to depth imperfection: Our method is based on a probabilistic formulation that marginalizes depth maps as hidden variables and therefore does not require perfect depth estimation.

From an application perspective, our approach does not require any change in camera optics or image detectors. All it uses is a set of pinhole cameras, such as those

equipped on cell phones. Such flexibilities make our method applicable to places that can only take miniaturized cameras with simple optics, such as low-power video surveillance networks, portable camera arrays, and multi-camera laparoscopy. In all cases, the baselines between different cameras can be appreciable, making it possible to reconstruct the 3D scene structure from the denoised images, which can then be used in other applications, such as refocusing, new view synthesis, and 3D object detection and recognition.

1.1 Literature overview

In the last decade, great progress has been made in image denoising, for example [17, 2, 7, 14, 6, 19, 8], just to name a few. We refer the readers to the previous work sections in [2, 6] for excellent reviews of the literature. Among these methods, several produce very impressive results, such as non-local mean [2], BM3D [6], and SA-DCT [8]. All these methods are built upon the same observation that local image patches are often repetitive within an image. Similar patches in an image are grouped together and “collaboratively” filtered to remove noise. While these methods have different algorithmic details, their performance is comparable. *Although there is no theoretic proof, we conjecture that the performance limit of single-image denoising has probably been reached.*

One approach to break this limit is to use more input images, such as video denoising [1, 5, 3]. To exploit redundant data in a video, similar patches need to be matched over time for noise removal. Another way of leveraging more input images is to reconstruct a clean image from noisy measurements from multiple viewpoints, proposed by Vaish et al. [20]. Only image redundancy across viewpoints is exploited in [20], and patch similarity within individual images is however neglected. In [10], Heo et al. proposed to combine NL-mean denoising with binocular stereo matching, therefore

exploiting data redundancy both across views and within each image. Heo et al.'s main idea is to apply NL-mean to both left and right images and then use the estimated depth to average the two denoised images. Note that, when applying NL-mean, their method matches patches in each image independently; such an approach is fragile in the presence of large image noise. Indeed, their method has only been evaluated using image patches with a noise standard deviation of 20. Our method matches patches simultaneously among all input images using depth as a constraint which is more robust to noise (standard deviation of 50), as shown in this thesis. A comparison between patch matching using a single image versus multiple images is shown in fig2

1.2

We discuss in some detail here some image denoising methods

1.2.1 Single image denoising

In [6] a single image denoising method is introduced where denoising is done in the enhanced sparse representation in a transform domain. In [6] it is noted that the transform-domain denoising methods typically assume that the true signal can be well approximated by a linear combination of few basis elements. "That is, the signal is sparsely represented in the transform domain. Hence, by preserving the few high-magnitude transform coefficients that convey mostly the true-signal energy and discarding the rest which are mainly due to noise, the true signal can be effectively estimated". An enhancement of sparsity is done by grouping together similar 2D patches

to form a 3D array which is called a ‘group’. Now collaborative filtering (as described below) is done on the transform domain of the ‘group’ to attenuate noise.

In collaborative filtering, for a group of $d+1$ dimension a $d+1$ dimensional linear transform is applied. Applying a $d+1$ dimensional transform rather than a d dimensional transform has the effect of exploiting the redundancy that the grouped patches are similar and the patches can be represented by lesser coefficients achieving an enhanced sparse representation

Once sparse representation in transform domain is obtained, shrinkage is done using soft/hard thresholding or wiener filtering the transform coefficients to attenuate noise.

An inversion of the linear transform is done to obtain the estimate of the noiseless patches.

1.2.2 Extension to video denoising

In [5], an extension to [6] is made in which a noisy video is processed in a block wise manner. For each block a ‘group’ is formed by spatio temporal predictive search block matching. Each ‘group’ is filtered and patches are reconstructed as in [6].

1.2.3 Synthetic aperture denoising

In [20], research is done into estimating cost functions which are robust to occlusions using synthetic apertures. The idea behind synthetic aperture being that if there are enough cameras which span a baseline wider than the occluders in scene, we can capture enough light rays that go around the foreground occluders and are incident on the

partially occluded background objects. If we treat noise as occlusion (foreground), it is possible to attempt denoising stereo images using the algorithm of [20].

1.3 Multiple image denoising

Using multiple input images to improve the accuracy of patch matching is the key idea in multi-baseline stereo [16] for depth estimation. In this thesis, we use depth estimation as a constraint to group similar patches in multi-view images for denoising. Patch repetitiveness is also the cornerstone of Epitome analysis [4], which can be used for compression and super-resolution, in addition to denoising. There has been no evaluation between epitome-based denoising and state-of-the-art denoising methods.

Recently, light field cameras [15] have been proposed to achieve large depth of field and high signal-to-noise ratio, at the expense of reduced image resolution. Such an approach requires modifications to existing camera construction, while our method uses only off-the-shelf cameras.

2.1 Problem Statement and Solution Overview

Let $I = \{I_m\}$ be a set of images taken from M different viewpoints at the same time instant. We model each image as a sum of its underlying noiseless image, G_m and zero mean noise n_m

$$I_m = G_m + n_m \quad (1)$$

Our goal is to recover $G = \{G_m\}_{m=1}^I$. There are five major sources of image noise [9]: fixed pattern noise, dark current noise, shot noise, amplifier noise, and quantization noise. Since fixed pattern noise and dark current noise can be precalibrated and quantization noise is usually much smaller than other noise, we focus on amplifier noise and shot noise and model the noise variance as

$$\sigma^2 = \tau^2 + \kappa G, \quad (2)$$

where τ^2 represents the amplifier noise whose variance is independent of intensity and κG represents the shot noise whose variance is proportional to intensity G . We verified this noise variance model on two camera models, Nikon D50 and Point- Grey Dragonfly Express¹, shown in Figure 1.

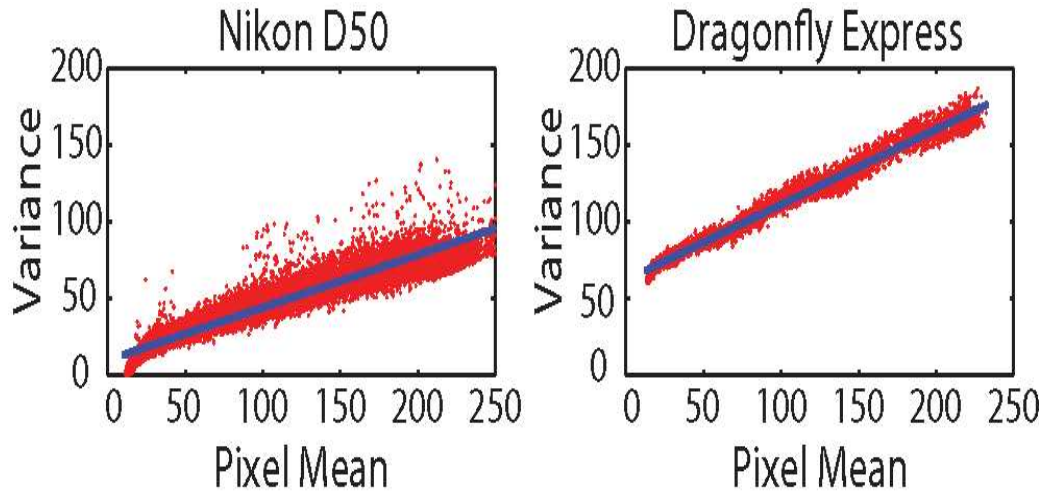


Figure 1. The relationship between pixel mean (horizontal-axis) and variance (vertical-axis) for Nikon D50 (left) and PointGrey Dragonfly Express (right). Each red point represents the (mean, variance) of a pixel estimated from multiple images of a static scene. The blue lines are the linear regression of the red points. This figure validates our noise variance model of Eq. (2).

To jointly reduce noise in all input images, we seek to exploit patch correspondences across different viewpoints using depth estimation. Let Z_m be the depth map for image I_m , which determines the pixel correspondence between I_m and all other images in I . Let $Z = \{Z_m\}_1^m$ be the set of depth maps for all the input images, which is unknown and needs to be estimated from I . It is well understood in stereo vision that depth estimation is often ambiguous in practice [12] and the true depth map is challenging to compute. With this fact in mind, we formulate the multi-view denoising problem by taking into account a family of likely depth solutions. Specifically, we consider the conditional probability of noiseless images G given noisy input images I , marginalizing over all possible depth hypothesis:

¹ The noise model in Eq (2) assumes linear camera response

$$P(G | I) = \int_Z P(G, Z | I) = \int_Z P(G | Z, I)P(Z | I) \quad (3)$$

To estimate G , we have two choices: Maximum Likelihood (ML) solution $\bar{G} = \arg_G \max P(G | I)$ and conditional mean solution $\bar{G} = E_{p(G|I)}(G | I)$. For either choice, an analytical solution is hard to find. We note however that the last term in Eq. (3), $P(Z|I)$, is the probability of depth maps Z given input images I and many formulations have been proposed for it in stereo matching literature [18]. We can sample depth maps based on $P(Z | I)$ and then approximate $P(G|I)$ using the sampled depth maps. For example, given a sample of depth maps, Z_i , we can use it to jointly denoise all the input images, therefore, generating a sample of noiseless images G_i . After a sequence of depth samples, we can compute a weighted average of all G_i as the approximate conditional mean solution.

$$\bar{G} = \int_{G,Z} GP(G, Z | I) \approx \int_{G_i Z_i} G_i P(G_i | Z_i, I) P(Z_i | I) \quad (4)$$

A special case of Eq. (4) is to take only a single sample for Z , e.g., the ML solution of $P(Z | I)$, and then use it to denoise the input images. We report our results using both ML depth estimation and random depth sampling.

Computing depth maps from a set of input images is not a contribution of this thesis; we use the simple window matching to compute or sample depth maps [16]. Next, we present our multi-view denoising using the computed depth maps.

Chapter 3

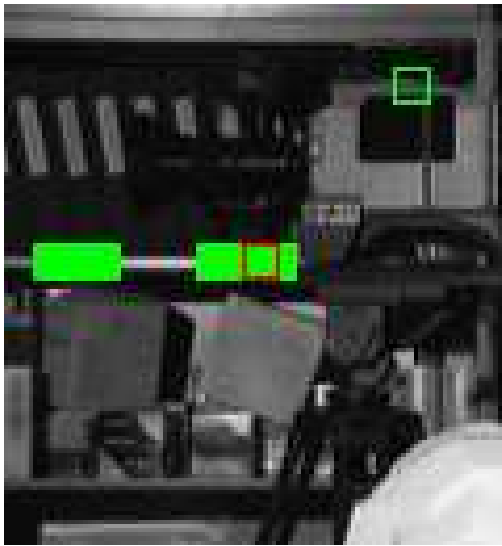
3.1 Depth-Guided Multi-View Image Denoising

In this chapter, we present a novel method for denoising multi-view images given depth map estimation. Leveraging multi-view data, our method addresses two key challenges in single image denoising. First, multi-view images provide more measurements for noise cancelation, thereby enabling denoising patches that are non-repetitive within a single image. Second, we use depth-induced constraints among different views during patch matching, thereby improving the patch grouping accuracy in the presence of large noise.

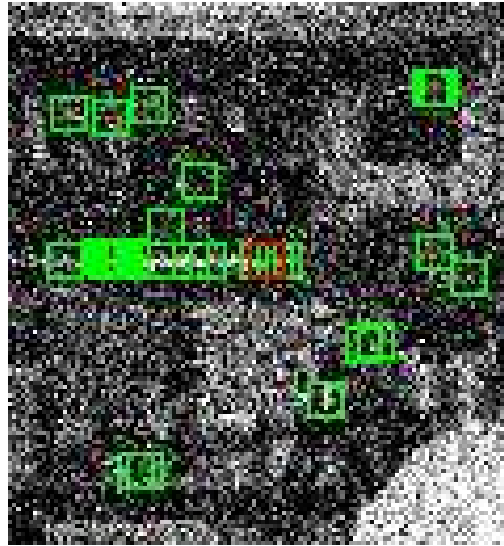
3.2 Joint Multi-View Patch Matching

Given multiple images, we choose one of them as a reference image, I_1 for example.² Consider one image patch b_p (say 8×8) centered at pixel p in the reference image. We call this patch a reference patch. To denoise this reference patch, we search for patches that are similar to b_p in all the input images, including the reference image itself.

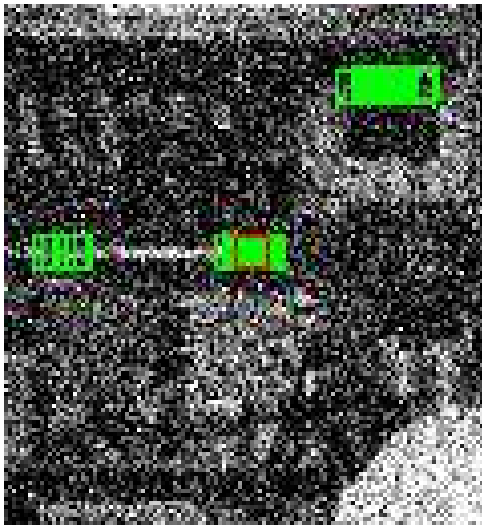
One way to achieve this goal is to compare the reference patch to all other patches, using a distance metric such as L2 norm. Such an approach however is susceptible to large image noise, as shown in Figure 2(a, b). The inaccuracy in patch grouping lowers the performance of image denoising.



2(a)



2(b)



2(c)

Figure 2. Comparison between patch grouping using a single image versus multiple images. (a) The green patches are the closest $K = 35$ patches to the reference patch (shown in red) in a clean image. (b) After the image is corrupted by noise with standard deviation = 65, for the same reference patch, the closest $K = 35$ patches are scattered around, and do not correspond to the closest patches that would be found in the clean image. (c) Using 25 noisy images taken from multiple viewpoints (only one shown here), the $K = 35$ closest patches to the same reference patch better resemble those in (a). In

short, depth-guided multi-view patch matching improves patch grouping accuracy in the presence of large noise.

In practice, perfect patch matching is unobtainable as the noiseless image is unknown. However, we can improve the accuracy of patch matching using multi-view images as follows. When deciding whether a patch A_1 is similar to a patch B_1 in the reference image I_1 , we find their corresponding patches A_2 and B_2 respectively in the second image I_2 using the depth map. If A_1 is similar to B_1 , A_2 should also be similar to B_2 . Additionally, if we have more views, we have more measurements to verify whether A_1 and B_1 in the reference view are indeed similar.

Specifically, we compute the similarity measure between patches b_p and b_q at locations p and q in the reference view as follows. We sum up the distances between patch pairs in all views that correspond to b_p and b_q :

$$\phi(b_p, b_q) = \sum_{m=1}^M \| b_{W_m(p)} - b_{W_m(q)} \|^2, \quad (5)$$

where $W_m(p)$ is the warped location of pixel p from the reference image I_1 to image I_m using its depth, and $b_{W_m(p)}$ is the patch centered at $W_m(p)$ in image I_m . Under this notation, $W_1(\cdot)$ is the identity warp for the reference image I_1 itself.

Using $\phi(b_p, b_q)$ in Eq. (5) as a metric, we can select K most similar patch locations for each patch in the reference image. After warping these K locations to all other $M - 1$ views, we collect a group of KM similar patches for noise reduction. Figure 2(c) shows an example of this method.

3.3 Joint Multi-View Patch Denoising

Given the KM patches $\{\mathbf{b}_q\}_{q=1}^{KM}$ with similar underlying image structures, we now seek to remove their noise. Let the size of each patch be $S \times S = D$. We treat each patch \mathbf{b}_q as a D -dimensional vector³. We explore two methods for denoising: PCA and tensor analysis.

3.3.1 Patch Denoising using PCA

Since all the patches in the set have similar underlying image structures, we assume that their noiseless patches lie in a low dimensional subspace, centered at \mathbf{u}_0 and spanned by bases $\mathbf{u}_{d=1}^C$.

Let $\bar{\mathbf{b}}_q$ be the noiseless patch for \mathbf{b}_q .

$$\bar{\mathbf{b}}_q = \mathbf{u}_0 + \sum_{d=1}^C \mathbf{u}_d f_{q,d} \quad (6)$$

where $f_{q,d}$ is the coefficient of patch $\bar{\mathbf{b}}_q$ along basis \mathbf{u}_d . We estimate the subspace by minimizing the difference between noisy patches and denoised patches:

$$\sum_{q=1}^{KM} \|\mathbf{b}_q - \bar{\mathbf{b}}_q\|_{\sigma}^2 \quad (7)$$

Where $\|x\|_{\sigma}^2 = \sum_{i=1}^D \frac{x_i^2}{\sigma_i^2}$ is an element-wise variance-normalized L2 norm that accounts for

the intensity dependent noise.

We approximate σ_i^2 as $\sigma_i^2 = T^2 + k[u_0]_i$ where $[u_0]_i$ is the intensity for pixel i in the patch u_0 .

In Eq. (7), if all σ_i are the same, the subspace can be directly computed using SVD. In

the presence of varying σ_i , we first compute the mean patch $u_0 = \frac{1}{KM} \sum_{q=1}^{KM} b_q$ and subtract

it from the input patches to obtain $b'_q = b_q - u_0$. Afterward, for each b'_q , we multiply its

i 'th element by $\frac{1}{\sigma_i}$.

Then we apply SVD on the matrix $B' = [b'_1, b'_2, \dots, b'_{KM}]$ to compute C bases $\{u'_d\}_{d=1}^C$. Note

that these bases are for $\{b'_q\}$ but not for $\{b_q\}$. Finally, we obtain the subspace bases

$\{u_d\}_{d=1}^C$ by multiplying each u'_d with $[\sigma_1, \sigma_2, \dots, \sigma_D]$ element-wise. Once we have the

subspace bases, the denoised patch b_q is computed using Eq. (6). We remark that our

minimization of Eq. (7) is provably optimal if we approximately evaluate per pixel noise

variance using the mean image patch. Without this approximation, the optimization is

nonlinear and needs an iterative solution.

3.3.2 Finding Dimensionality C

In practice, we need to determine the subspace dimension C . One choice is to use a fixed value, e.g., $C = 1$. However, different groups of patches require different numbers of principal components for best reconstruction, as shown in Figure 3.

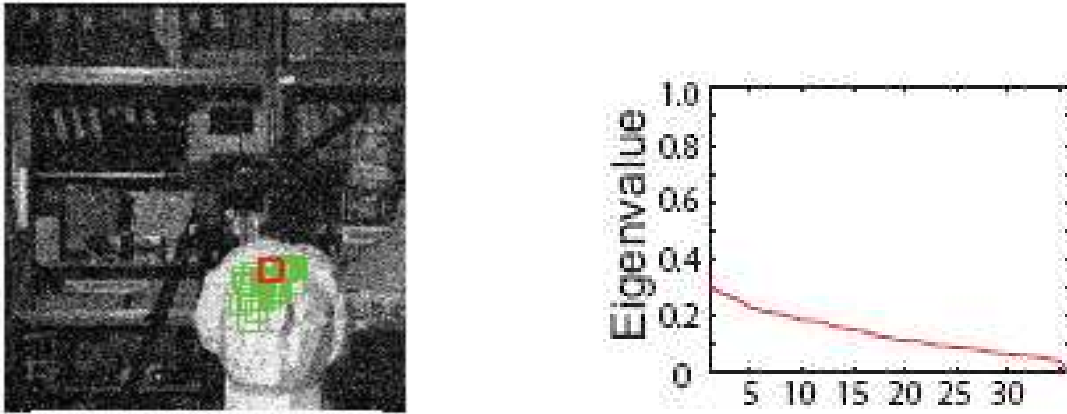


Figure 3(a)

²We use the notion of reference image to facilitate algorithm description. Our method is able to symmetrically denoise all input images.

³For color images, we treat each patch as a 3D-dimensional vector

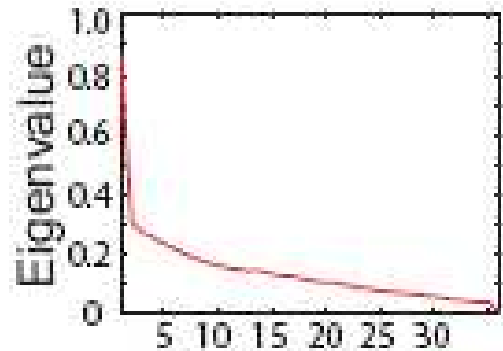


Figure 3(b)

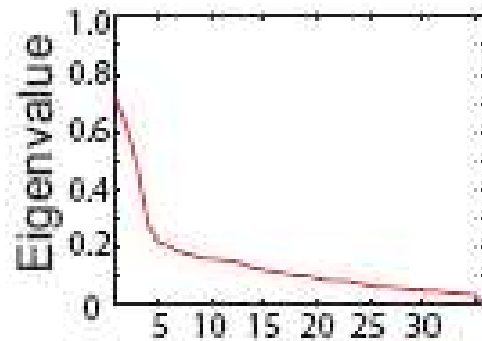
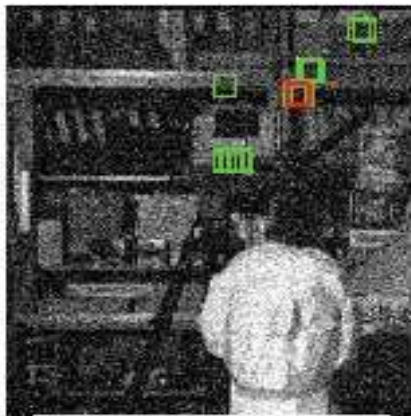


Figure 3(c)

Figure 3. Illustration of eigenvalues for different collections of patches. (a) we show a reference patch in red and its 35 most similar patches in green. The eigenvalues for this set of patches is shown in the inset. Since this collection of patches is on the bust, the residual patches after mean removal approximately correspond to noise. The eigenvalues therefore correspond to noise power along the components. (b) The reference patch contains an intensity edge. The top component captures the subtle variation of these edge patches. The rest of the small eigenvalues correspond to noise power along these components. (c) The reference patch has an irregular intensity pattern and more components are needed to capture patch variation within the collection. We propose an automatic way of determining the number eigenvectors for patch denoising

In general, too many components tend to introduce noise in the results and too few components tend to over-smooth the results. We propose a new way of choosing the

dimension of the subspace for each patch stack that is adaptive to the underlying image structure. Our basic idea is that if we choose the right dimension for the subspace, the average squared residuals between noisy patches and denoised patches should be very close to the noise variance. Recall that we have KM patches, each having D pixels, and the average residual errors (scaled by variance) is

$$\frac{1}{KMD} \sum_{q=1}^{KM} \|b_q - \bar{b}_q\|_{\sigma}^2 = \frac{1}{KMD} \sum_{d=C+1}^D \lambda_d^2 \quad (8)$$

where $\{\lambda_d\}_{d=1}^D$ are the singular values of the matrix B . We therefore look for C such that Eq. (8) is closest to 1. Since Eq. (8) monotonically increases as C decreases from D to 1, a binary search can be used to quickly find an optimal C.

3.3.3 Patch Denoising using Tensor Analysis

Inspired by its successful application in modeling textures [21], we have also explored using tensor analysis for patch denoising. Specifically, rather than stacking patches from different images in a single matrix, we put patches from the same image in a stack, and view the patches from multiple images as a multi dimensional array. Let this array be B and $[B]_{i,k,m}$ be the intensity of pixel ‘i’ in the patch ‘k’ in the view ‘m’.

Since all the patches are similar, we assume that the underlying noiseless patch array

\bar{B} lie in a multi linear subspace centered at u_0 and spanned by bases

$\{u_i\}_{i=1}^C \otimes \{v_k\}_{k=1}^J \otimes \{w_m\}_{m=1}^L$ where u_i , v_k , and w_m are the basis vectors for the three

modes: i , k , and m , which is written as

$$\bar{B} = u_0 \oplus_1 \sum_{i=1}^C \sum_{k=1}^J \sum_{m=1}^L f_{i,k,m} u_i \otimes v_k \otimes w_m, \quad (9)$$

Where $u_0 \oplus_1 \chi$ means “adding u_0 to each mode-1 vector of the tensor χ ”, and $f_{i,k,m}$ is the tensor coefficient. Similar to the PCA denoising, we estimate this multi-linear subspace by minimizing the difference between noisy patches and denoised patches as in Eq. (7), with the exception that

\bar{b}_q is computed using Eq. (9), rather than Eq. (6). We follow the same procedure as PCA denoising to estimate the multilinear subspace, except that we replace matrix SVD with tensor SVD [13]. We also apply the same method of choosing subspace dimension in PCA denoising to each mode of the tensor SVD separately to determine C , J , and L . We compare PCA and tensor denoising.

3.3.4 From Denoised Patches to Denoised Images

By applying the patch grouping and patch denoising to each patch in input images, we can denoise all of them. We now use the denoised patches to form denoised images. Each pixel is often covered by several denoised patches. To determine the value of a particular pixel in a denoised image, we take a weighted average of denoised patches at this pixel. The weight reflects our belief in the likelihood that the denoised patch resembles the true underlying noiseless image. Since the denoised patch is computed from the PCA or tensor analysis, a lower dimension of the subspace suggests less image structure variation

in the patch collection and the noise has a better chance to be canceled out. Therefore, we have experimented with using $1/C$ as patch weight for the PCA denoising and $1/CJL$ for tensor denoising. We have also experimented with other weighting choices, such as $\exp(-\|b - \bar{b}\|^2)$ which favors denoised patches that are closer to the original patches. We found these different choices have comparable performance.

It is worth noting that the depth estimation we use does not model occlusion explicitly. As a result, for a reference patch near an occlusion boundary, its patch collection may include patches from other views that have considerably different intensity pattern. Using a weighting scheme that favors more compact patch collections helps to reduce artifacts near occlusion boundaries.

Chapter 4

4.1 Dataset

We have used available stereo image datasets and also created our own datasets.

4.1.1

Tsukuba stereo image dataset from *Middlebury University*

[<http://vision.middlebury.edu/stereo/data/scenes2001/>]. This dataset consists of 25

images taken by a 5 by 5 camera array. A sample image of the scene as seen by the 5 by

5 camera array is shown here.



Figure 4

4.1.2

Tarot image dataset from *Stanford light archives*

[<http://lightfield.stanford.edu/lfs.html>]. This dataset consists of 289 views on a 17x17 grid. We sampled 25 images from this set. A sample image is shown here.



Figure 5

4.1.3

Evolve image dataset taken in *University of Wisconsin Madison*, Computer Vision and Graphics lab. The dataset is taken by a point gray research 5 by 5 camera array. A sample image is shown here.

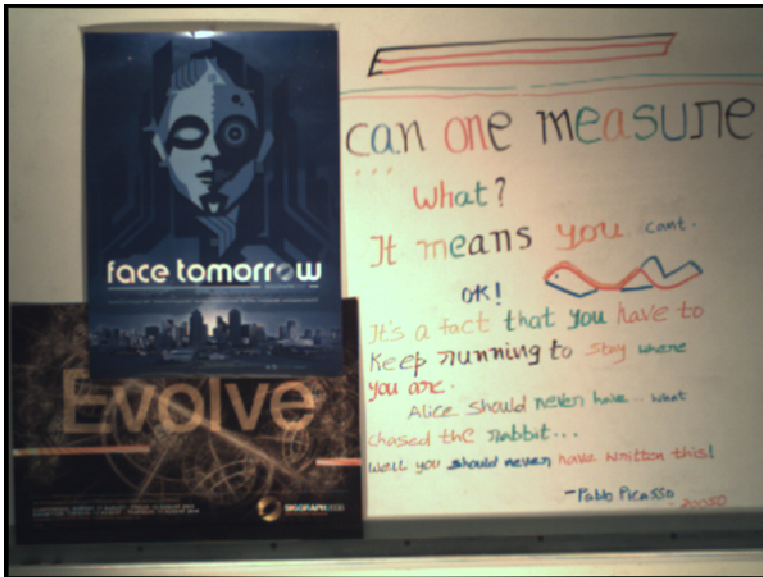


Figure 6

4.1.4

Text board dataset taken in *University of Wisconsin Madison*, Computer vision and Graphics lab. 25 images were taken by a point gray dragon fly express camera by moving it on a linear rail. The camera was kept in its highest gain to produce real time noise. A sample image is shown here.

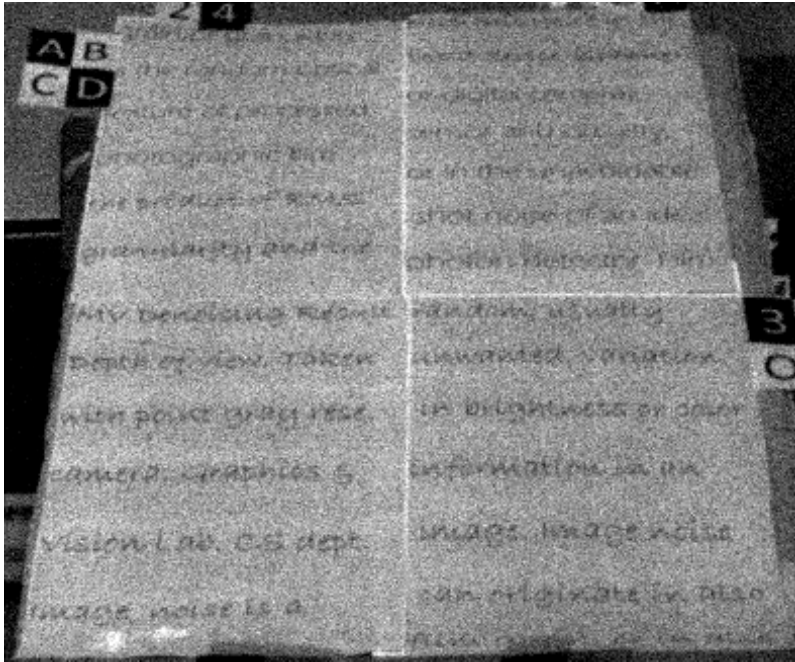


Figure 7

4.1.5

Line art image dataset taken in *University of Wisconsin Madison*, Computer vision and Graphics lab. 25 images were taken by a point gray dragon fly express camera by moving it on a linear rail. The camera was kept in its highest gain to produce real time noise.

A sample image is shown here.

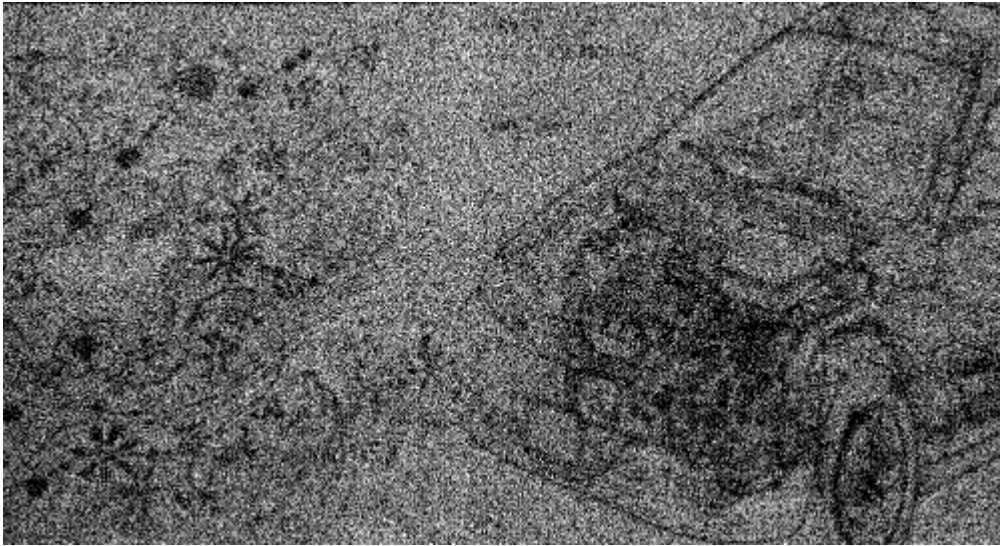


Figure 8

4.1.6

The datasets were used in the following manner.

- Gaussian synthetic noise was added to evolve dataset to compare various denoising algorithms results on Gaussian noise.
- Poisson synthetic noise was added to Tsukuba (Ohta) and tarot dataset to compare various denoising algorithms results on Poisson noise.
- Text and line art datasets were created using highest gain settings in the input camera to produce real noise. These datasets were used to compare various denoising algorithms results on real noise.

4.2

Experimental Results

We have implemented our method in C++ and evaluated it on different images. By default, we use $M = 25$ views, set patch size to be $D = 8$ by 8 pixels, and choose a reference patch for every 4 pixels. For each reference patch, we choose $K = 28$ most similar patch locations.

4.2.1 Input Image Data

Our experiments include images with synthetic noise as well as real noise. To create an noisy image I from a clean image G , we use $I = k \cdot \text{poissrnd}(\frac{G}{K})$ in Matlab, where ‘ k ’ is a scalar parameter. $\text{poissrnd}(x)$ generates a Poisson random number with mean x and variance x . This operation simulates the process that incoming light is darkened by a factor of $\frac{1}{k}$, recorded by a photoreceptor, and then amplified by a factor of

k . At the end, I has mean G and variance kG . We generated synthetic noisy images using two image sets:

Ohta and Tarot Card. Ohta images are from the Middlebury Stereo website⁴, and were taken from a grid of 5x5 viewpoints. For each image, we added Poisson noise with $k = 38$. Tarot Card images are from Stanford Light Field Archive.⁵The original data set has

17x17 images, and we used a subset of 5x5 and added noise with $k = 18$. We also captured a sequence of 25 noisy images for a board with texts and line arts using PointGrey Dragonfly Express at the highest gain. The camera moves about 5mm between neighboring images and the scene is approximately 1.5 meters away from the camera. We use Voodoo software⁶ to calibrate the 3D camera path.

4.2.2 Comparison to Other Denoising Approaches

We first applied our denoising method to the Ohta images and the Tarot Card images. Figure 9 shows our results, compared with the results generated by BM3D, one of the state-of-the-art single image denoising methods [6], and a multi-view image reconstruction method [20]. Our method substantially outperforms the other two methods both visually and quantitatively. Additional results are shown in Chapter 6. Note that the improvement of our approach over single image denoising is substantial, much more dramatic than the performance difference between various state-of-the-art denoising methods evaluated on a single image. We attribute this performance gain to more accurate patch grouping and more data for noise cancelation. We believe our work can inspire the image denoising community to design algorithms that can be conveniently extended to leverage multi-view images for significant performance gain. We also applied our method to the text board images. Figure 10 shows our results, compared with the results by BM3D, and its video denoising extension, VBM3D [5]. Our method has a clear advantage over these two. This comparison suggests that if patches cannot be

accurately grouped over time, additional image measurements may not contribute significantly to the denoising performance. As an application of our denoising algorithm, we have tested depth estimation on both noisy and denoised images and report the results in Figure 11. The benefit of using denoised images for depth estimation is clear. Table 1, 2, 3 shows the comparison between the PSNR obtained by our algorithm and [5], [6], [20].

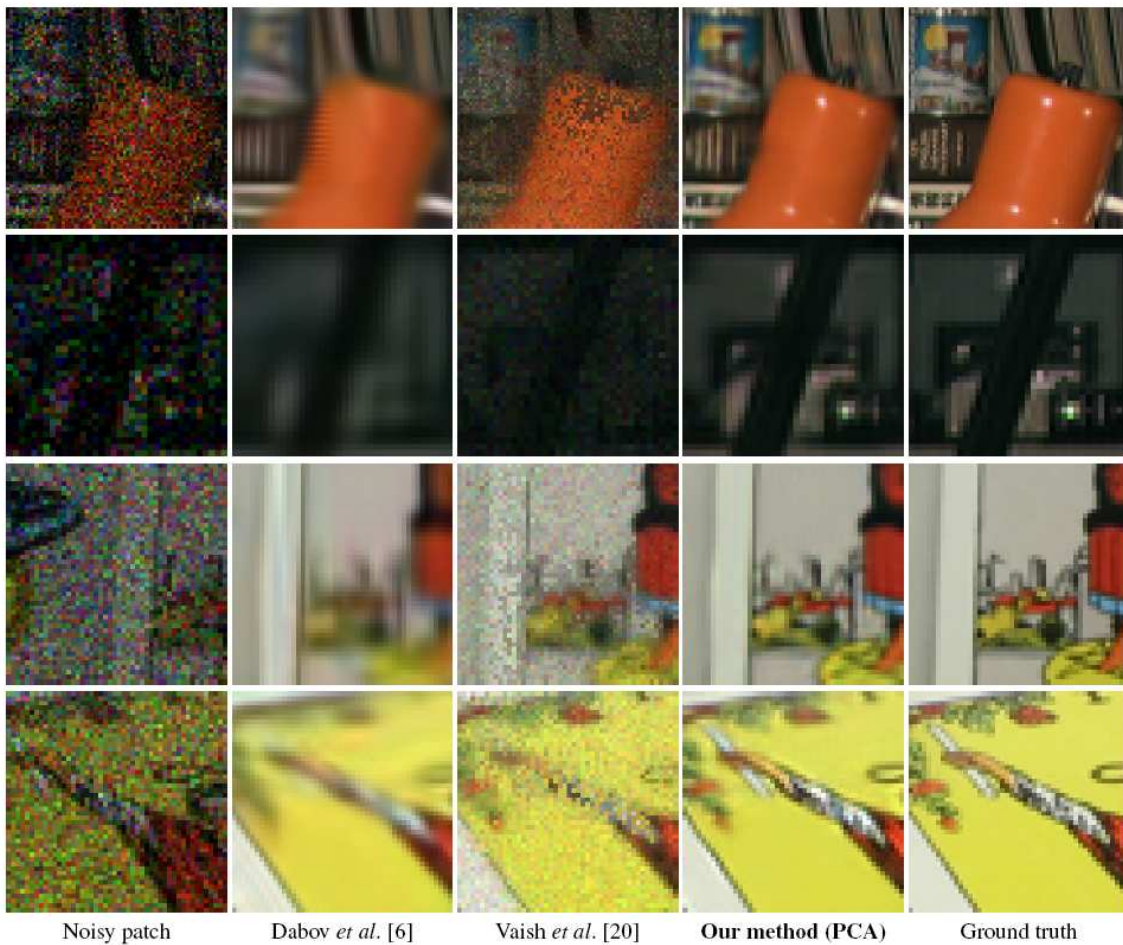


Figure 9. Comparison between our 25-view denoising and a state-of-the-art single image denoising (Dabov *et al.* [6] applied on one image) and an existing multi-view denoising (Vaish *et al.* [20] applied on all 25 images).

Gaussian synthetic noise: Table 1

PSNR	Noisy image	Dabov et al. [6]	Vaish et al. [20]	Our algorithm
CROC	14.1449	22.992	21.207	28.056
EVOLVE	16.0809	31.154	21.062	33.781

Poisson synthetic noise: Table 2

PSNR	Noisy image	Dabov et al. [6]	Vaish et al. [20]	Our algorithm
OHTA	14.1471	26.476	19.837	30.659
TAROT	13.9605	22.681	19.307	27.383
EUCA	13.6957	24.758	17.124	27.699

Real image Noise: Table 3

PSNR	Noisy	Dabov et	Dabov et	Our
------	-------	----------	----------	-----

	image	al. [6]	al. [5]	algorithm
TEXT	27.7509	36.197	36.026	41.736
LINE ART	28.6831	39.9480	39.3697	44.1641

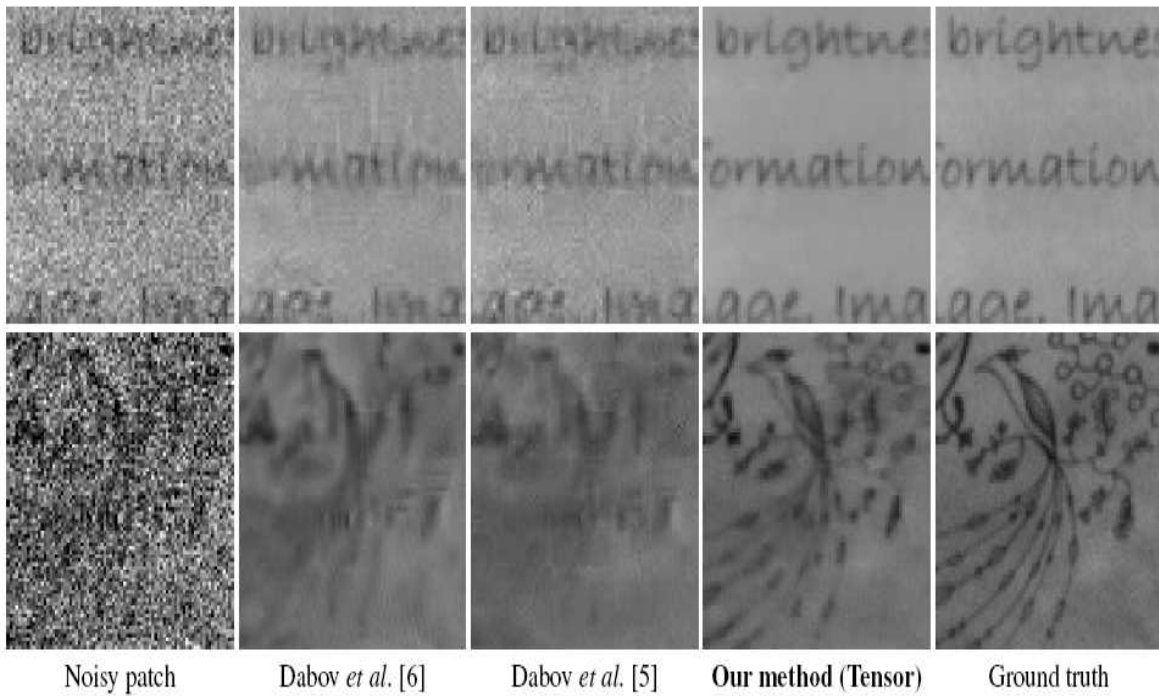


Figure 10. Comparison between our 25-view denoising and a state-of-the-art single image denoising (Dabov et al. [6] applied on one image) and its extension to video denoising (Dabov et al. [5] applied on all 25 images as a video sequence).

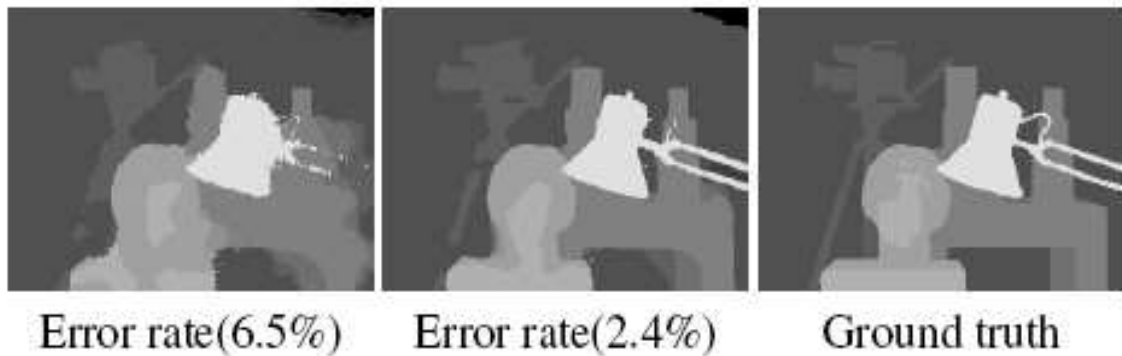


Figure 11. An application of our multi-view denoising algorithm for depth estimation. Left: the depth map estimated from the 25 noisy Ohta images. Middle: the depth map estimated from the 25 denoised images produced by our denoising algorithm. In both cases, the depth maps are estimated using graph cuts [11]. Right: the ground truth depth map. The error rate is defined as the percentage of the pixels with over one disparity difference compared to the ground truth. The benefit of using our algorithm for depth estimation is clear.

4.2.3 Intensity Dependent Variance

We have evaluated the effectiveness of modeling intensity dependence variance for images with Poisson noise. Using intensity-dependence variance reduces noise in all regions without losing details in dark regions, as shown in Figure 12.

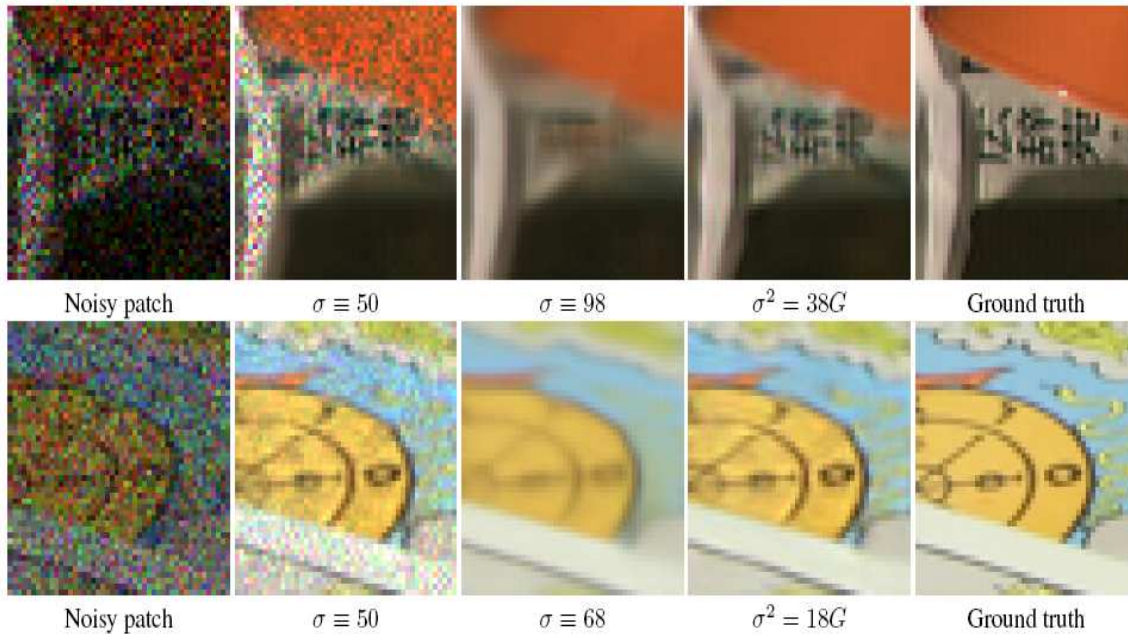


Figure 12. Comparison of multi-view denoising for images with Poisson noise using a fixed variance versus an intensity-dependent variance. Using a small and fixed noise variance keeps image details but is unable to reduce large noise over bright regions. Using a large and fixed noise variance reduces noise in all regions but over-smooths the details in the dark regions. Using intensity-dependent variance reduces noise in all regions without losing details in dark regions.

4.2.4 PCA versus Tensor Analysis

We have compared denoising results using PCA versus Tensor analysis. Both have comparable performance, as shown in Figure 13. Tensor denoising yields smoother results because it tends to treat appearance variation across viewpoints due to occlusion or reflectivity as noise, while PCA is more flexible to preserve these variations.

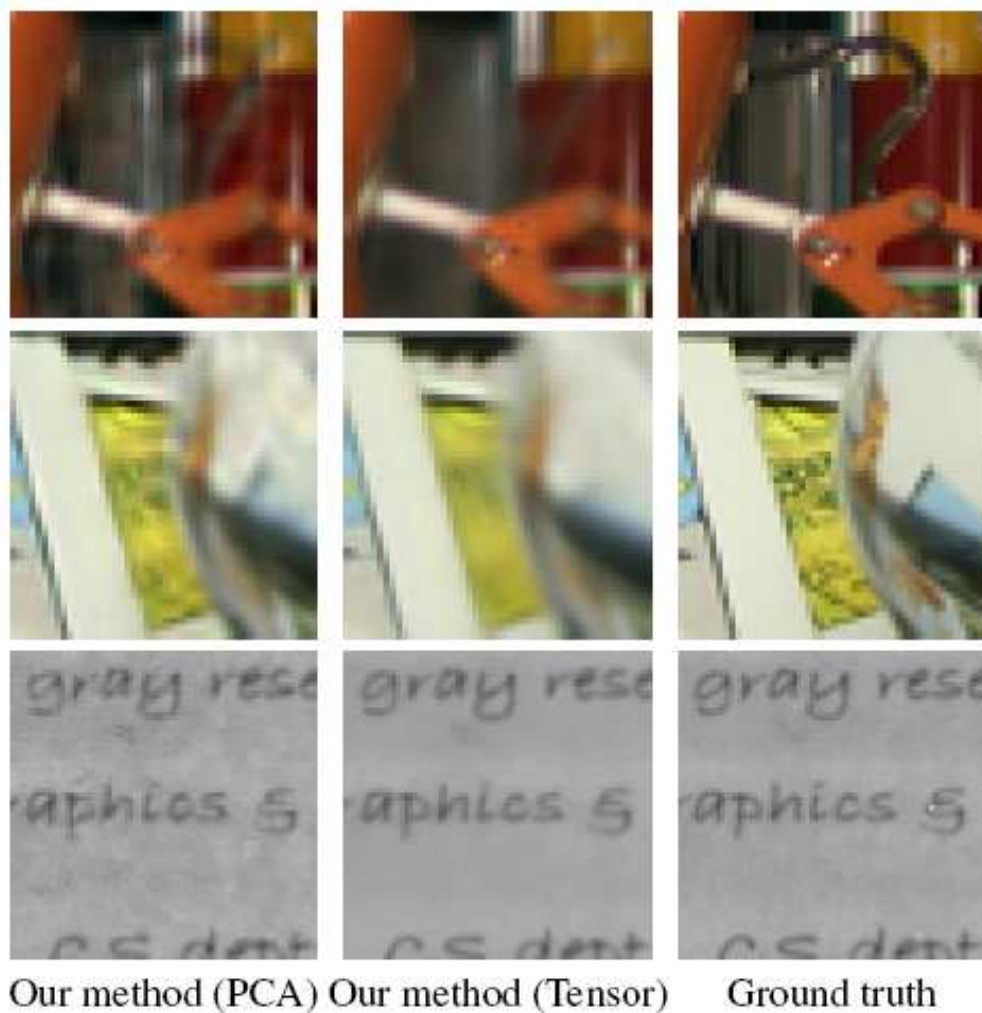


Figure 13: Comparison between patch denoising using PCA versus tensor analysis.

Both have comparable performances.

4.2.5 How Many Views Are Enough?

Figure 14 illustrates the performance of our approach as a function of the number of input views. The performance is measured in terms of peak signal-to-noise ratio (PSNR). It steadily improves as the number of views increases till 15-20, after which it flattens. The cause for this phenomenon remains an open question for future study. One speculation is that, as our denoising performance depends on how precisely we group similar patches together, the only way to achieve a possible ideal noiseless image would be to have **1**. A perfect depth map **2**. Smaller patch sizes (ideally 1 pixel) to group together **3**. Group patches only from different views but not within an image itself. The 3rd points out that though structures are repeated, finding similar structures within an image is not ideal, i.e. they are similar. But only grouping across views makes sure that we will have same patches for a reference patch.

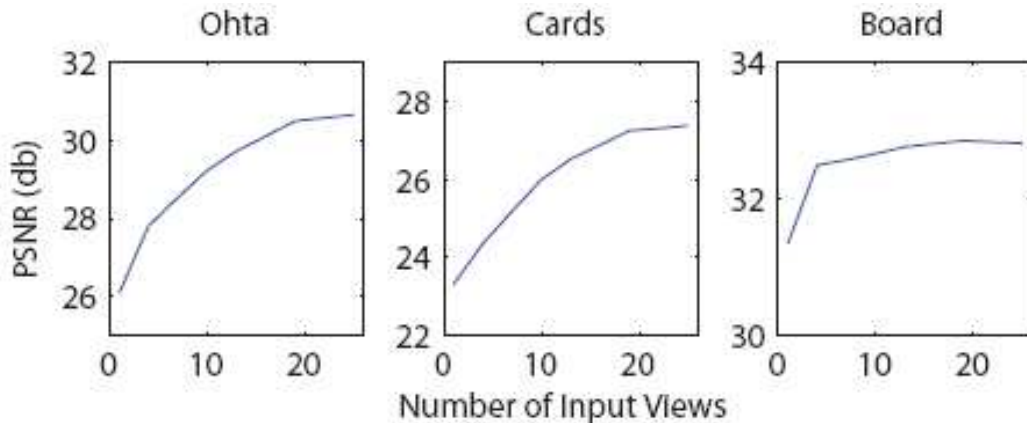


Figure 14

4.2.6 *Optimizing versus Sampling Depth Maps*

All our results presented so far use only a single depth map, which is computed with a window-based winner-take-all approach [16]. We have also experimented with using randomly sampled depth maps for denoising. When sampling for a depth map, we assume that depth for each pixel is independent and use the window-based stereo matching cost to compute depth distribution. Specifically, the probability that pixel p has depth

$$P(z) \propto \exp\left(-\sum_{m=2}^M \|b_p - b_{W_m(p)}\|_{\sigma}^2\right) \quad (10)$$

where $W_m(p)$ is the warped location of pixel p from image 1 to image m using depth z , and $b_{W_m(p)}$ is the patch centered at $W_m(p)$ in image m . Figure 15 shows denoising results using a sampled depth map. These results are very comparable to our results in Figure 13, which use window-based winner-take-all depth estimation. The similarity between Figure 15 and 13 is because depth distribution $P(z)$ is highly peaked for patches that have a unique intensity pattern and is more spread out for ambiguous patches.

Taking a solid white image region as an extreme example, depth sampling has large uncertainty, but any depth value works equally well for grouping patches in this region for denoising. We have also tried averaging denoising results using multiple sampled depth maps, and found that the results do not differ much from those obtained using a single depth map. This experiment suggests that imperfect depth estimation can be good enough for multi-view image denoising. How to rigorously aggregate different denoising results using sampled depth maps remains a theoretic question for future study



Figure 15

Figure 15. Our multi-view image denoising using depth maps randomly sampled from the depth distribution of Eq. (10). These results are very comparable to those in the left column of Figure 13. This experiment suggests that imperfect depth maps due to matching ambiguity can be used to generate good denoising results.

Chapter 5

Discussion and Future Work

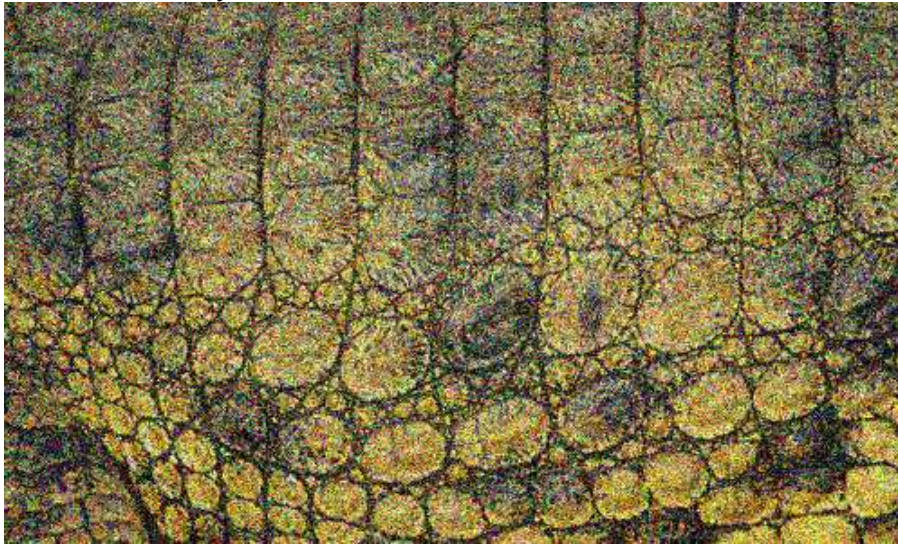
In this paper, we cast multi-view pinhole imaging as a multi-view denoising problem and seek to restore all the pinhole images by jointly removing noise in different viewpoints.

We believe our work opens several interesting venues for future work. First, our current method does not model occlusion between different views. This has not generated very objectionable artifacts in the results, due to the weighting scheme in Section 4.3.

However, in the Ohta example, we do see a small amount of color bleeding of the red lamp arm into the background poster board. We believe that adopting robust PCA or tensor analysis to patch denoising will address this issue. Second, our current implementation does not consider patch deformation when matching patches across views. An affine transformation model with sub pixel matching will improve the performance of our algorithm. Third, it is intriguing to note that our performance curve flattens after 15-20 views. We would like to design algorithms that always benefit from more input views.

Chapter 6

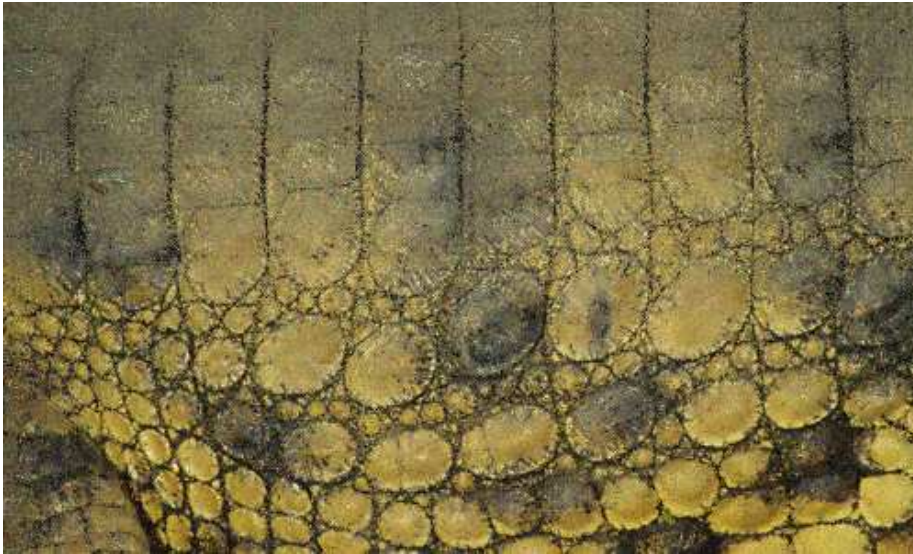
Additional Results *Gaussian Synthetic Noise*



One of 25 noisy input images, PSNR=14.1449



Single image denoising [[Dabov et al.](#)], PSNR=22.992



25-view stereo denoising [[Vaish et al.](#)], PSNR=21.207



Our 25-view denoising, PSNR=28.056



Ground Truth

Poisson Synthetic Noise

$$\text{NoisyImage} = K * \text{poissrnd}(\text{CleanImage}/K)$$



One of 25 noisy input images, PSNR=14.1471



Single image denoising [[Dabov et al.](#)], PSNR=26.476



25-view stereo denoising [[Vaish et al.](#)], PSNR=19.837

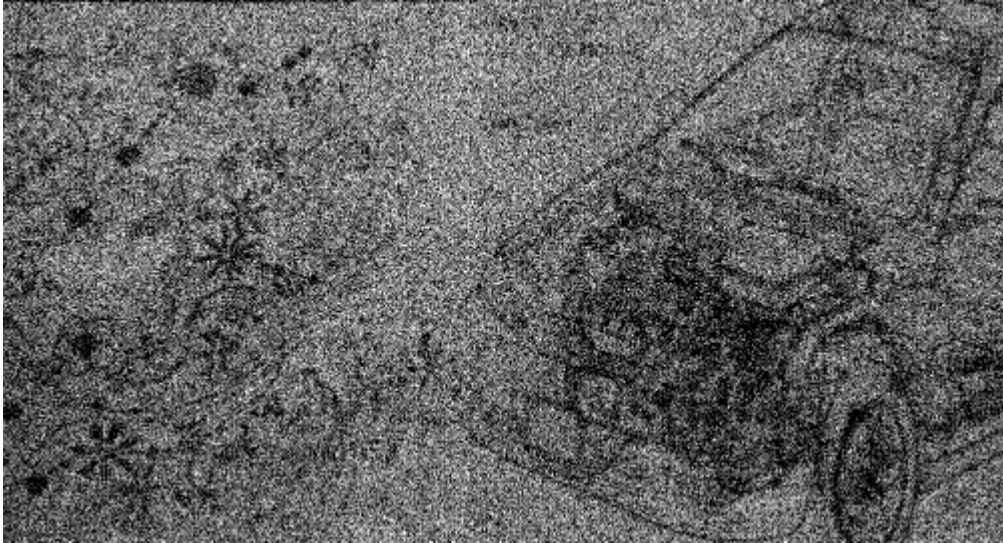


Our 25-view denoising (PCA), PSNR: 30.659



Ground Truth

Real Image Noise



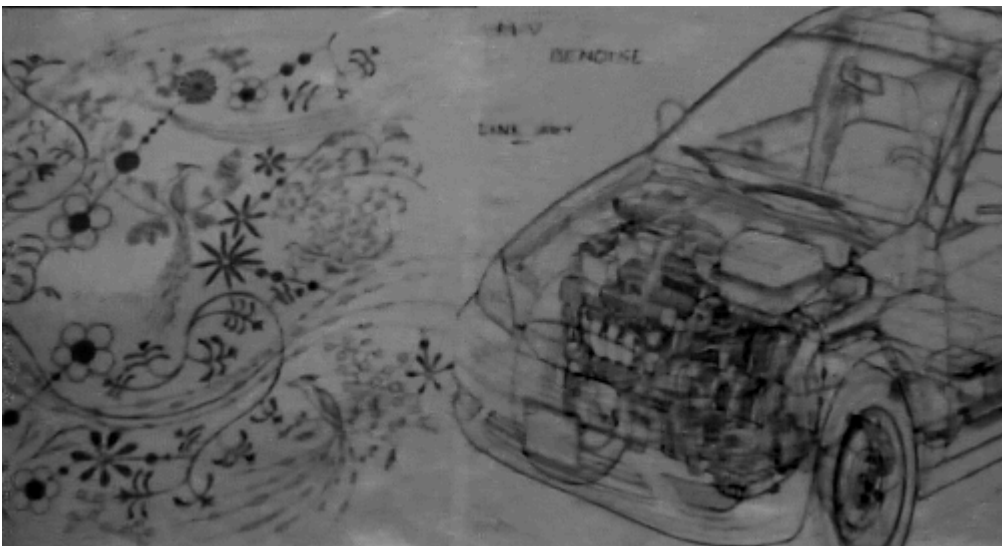
One of 25 noisy input images, PSNR=28.6831



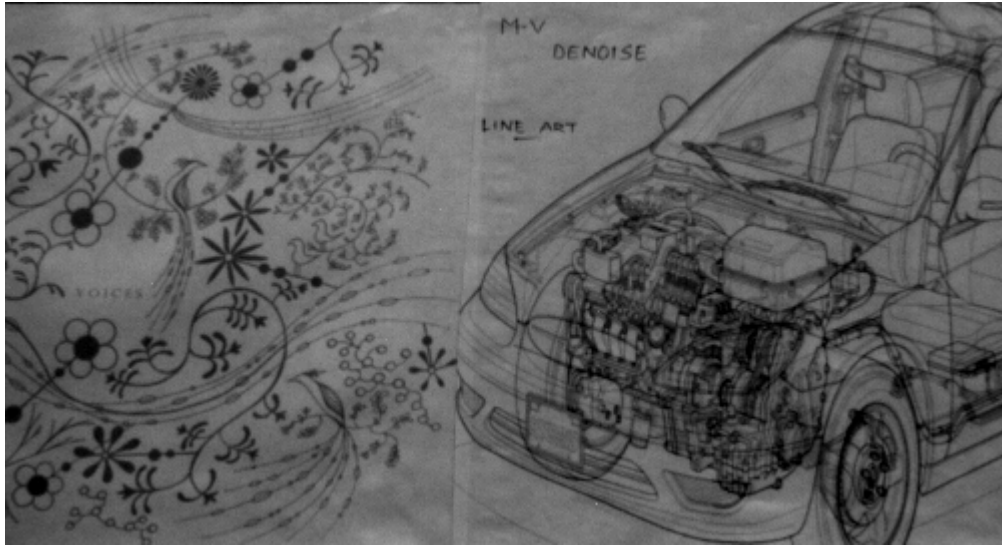
Single image denoising [[Dabov et al.](#)], PSNR=39.9480



Video denoising on 25 images [[Dabov et al.](#)], PSNR=39.3697



Our 25-view denoising, PSNR=44.1641



Ground truth



References

- [1] E. P. Bennett and L. McMillan. Video enhancement using perpixel virtual exposures. In SIGGRAPH, 2005.
- [2] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Simulation*, 4, 2005.
- [3] J. Chen and C.-K. Tang. Spatio-temporal markov random field for video denoising. In CVPR, 2007.
- [4] V. Cheung, B. J. Frey, and N. Jojic. Video epitomes. *IJCV*, 76(2), 2008.
- [5] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. In Proc. 15th European Signal Processing Conference, 2007.
- [6] K. Dabov, R. Foi, V. Katkovnik, K. Egiazarian, and S. Member. Image denoising by sparse 3d transform-domain collaborative filtering. *TIP*, 16:2007, 2007.
- [7] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In CVPR, 2006.
- [8] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shapeadaptive dct for high-quality denoising and deblocking of grayscale and color images. *TIP*, 16(5):1395–1411, 2007.
- [9] G. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *TPAMI*, 16(3):267–276, 1994.

- [10] Y. S. Heo, K. M. Lee, and S. U. Lee. Simultaneous depth reconstruction and restoration of noisy stereo images using non-local pixel distribution. In CVPR, pages 1–8, 2007.
- [11] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In ECCV, 2002.
- [12] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. IJCV, 38(3):199–218, 2000.
- [13] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. On the best rank-1 and rank- (r_1, r_2, \dots, r_m) approximation of higher-order tensors. SIAM J. Matrix Analysis and Applications, 21(4):1324–1342, 2000.
- [14] S. Lyu and E. P. Simoncelli. Statistical modeling of images with fields of gaussian scale mixtures. In NIPS, 2006.
- [15] R. Ng. Fourier slice photography. ACM Trans. Graph., 24(3), 2005.
- [16] M. Okutomi and T. Kanade. A multiple-baseline stereo. TPAMI, 15(4):353–363, 1993.
- [17] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In CVPR, pages 860–867, 2005.
- [18] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV, 47(1-3), 2002.
- [19] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning gaussian conditional random fields for low-level vision. In CVPR, pages 1–8, 2007.

[20] V. Vaish, M. Levoy, R. Szeliski, C. L. Zitnick, and S. B. Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In CVPR, pages 2331–2338, 2006.

[21] M. A. O. Vasilescu and D. Terzopoulos. Tensortextures: multilinear image-based rendering. ACM Trans. Graph., 23(3), 2004

[22] <http://www.cs.cmu.edu/~elaw/papers/pca.pdf>

[23] <http://en.wikipedia.org/wiki/Tensor>