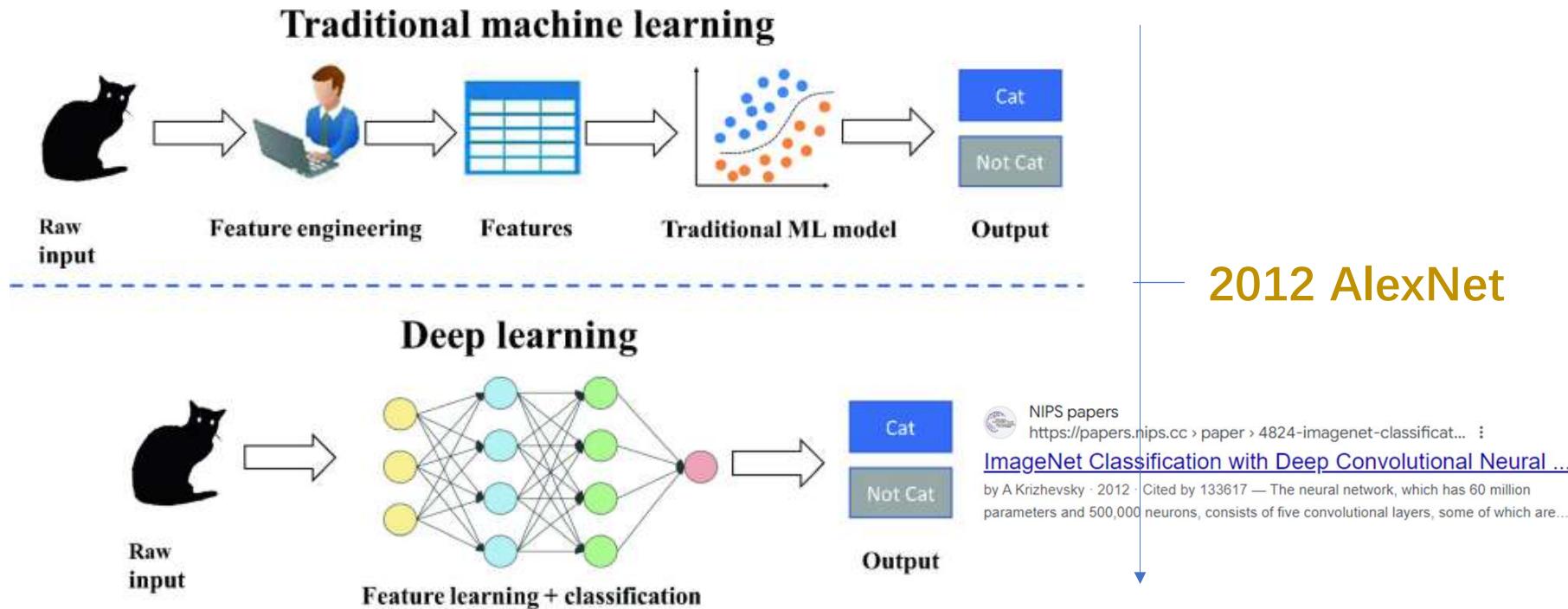


用前面学习的算法 / 工具  
能做什么事情？

# 传统图像处理

## Traditional Image Processing



## ChatGPT o1-preview ▾

What are important digital image processing tasks before deep learning

Before the advent and widespread adoption of deep learning techniques, digital image processing relied heavily on classical algorithms and methodologies to analyze and manipulate images. These tasks were fundamental in various applications like medical imaging, remote sensing, industrial inspection, and computer vision systems. Here are some of the most important digital image processing tasks from that era:

<https://chatgpt.com/share/66ee3a40-c6d0-8005-82e1-21e4d9a137fd>

What are important digital image processing tasks before deep learning

1. **Image Enhancement:**

- **Histogram Equalization:** Adjusting image intensities to improve contrast.
- **Noise Reduction:** Using filters like mean, median, Gaussian, or Wiener to reduce noise while preserving edges.
- **Sharpening:** Enhancing edges and fine details using techniques like unsharp masking.

2. **Image Restoration:**

- **Deblurring:** Correcting images blurred by motion or defocus using deconvolution methods.
- **Denoising:** Removing noise based on statistical models of the noise and the image.

3. **Edge Detection:**

- Identifying significant transitions in intensity, which correspond to object boundaries.
- Common algorithms included Sobel, Prewitt, Roberts, and Canny edge detectors.

What are important digital image processing tasks before deep learning

**13. Pattern Recognition and Classification:**

- Recognizing patterns using statistical methods, neural networks (pre-deep learning), and clustering algorithms.
- Applied in OCR (Optical Character Recognition) and biometric recognition systems.

**14. Motion Analysis and Tracking:**

- Estimating motion in sequences of images (optical flow).
- Tracking objects over time using Kalman filters or particle filters.

**15. 3D Reconstruction and Stereo Vision:**

- Reconstructing 3D structures from 2D images.
- Computing depth information from stereo image pairs.

please give some papers after 2000 for each term

## 1. Image Enhancement

### Histogram Equalization

- **Kim, M., & Chung, J. (2008).** "Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement." *IEEE Transactions on Consumer Electronics*, **54**(3), 1389–1397.
  - Introduces a method that enhances contrast while preserving the brightness of the image.
- **Wang, Q., & Ward, R. K. (2007).** "Fast image/video contrast enhancement based on weighted thresholded histogram equalization." *IEEE Transactions on Consumer Electronics*, **53**(2), 757–764.
  - Proposes a fast histogram equalization technique suitable for real-time applications.

### Noise Reduction

- **Buades, A., Coll, B., & Morel, J.-M. (2005).** "A non-local algorithm for image denoising." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 60–65.
  - Introduces the Non-Local Means algorithm, a significant advancement in image denoising.

please give some papers after 2000 for each term

## 5. Feature Extraction and Description

### SIFT

- **Lowe, D. G. (2004).** "Distinctive image features from scale-invariant keypoints." *International Journal of Computer Vision*, **60**(2), 91–110.
  - Official publication of the SIFT algorithm for feature detection and description.

### SURF

- **Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008).** "Speeded-Up Robust Features (SURF)." *Computer Vision and Image Understanding*, **110**(3), 346–359.
  - Details the SURF algorithm as a faster alternative to SIFT.

### ORB

- **Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011).** "ORB: An efficient alternative to SIFT or SURF." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2564–2571.
  - Introduces ORB, an efficient feature detector and descriptor for real-time applications.

利用已学习的算法 / 工具能否  
理解 / 实现这些papers ?

# What we have learned

image warping: change **domain** of image

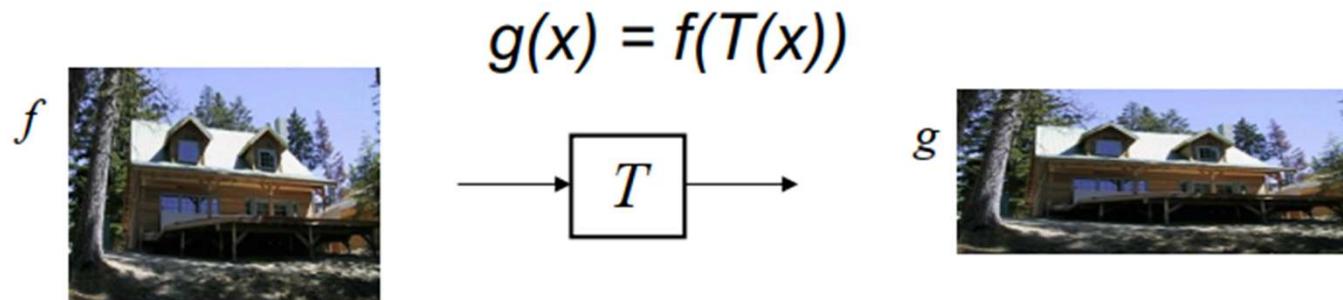
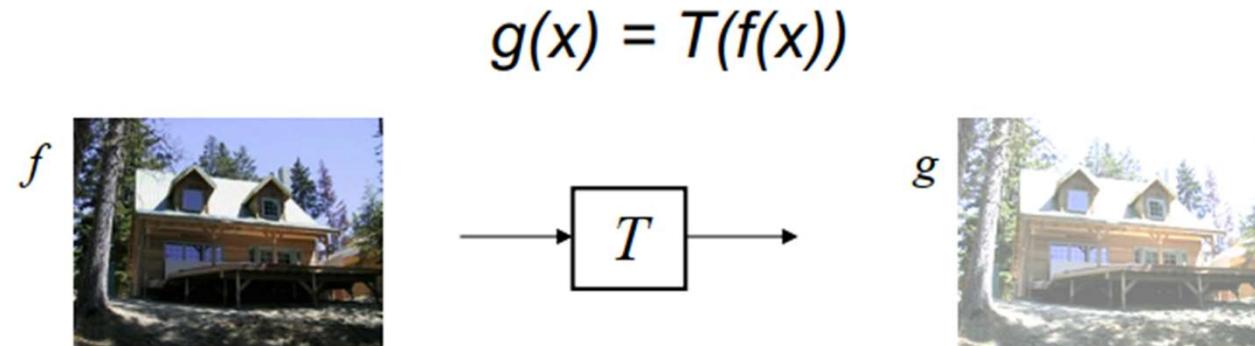


image filtering: change **range** of image



# What we have learned

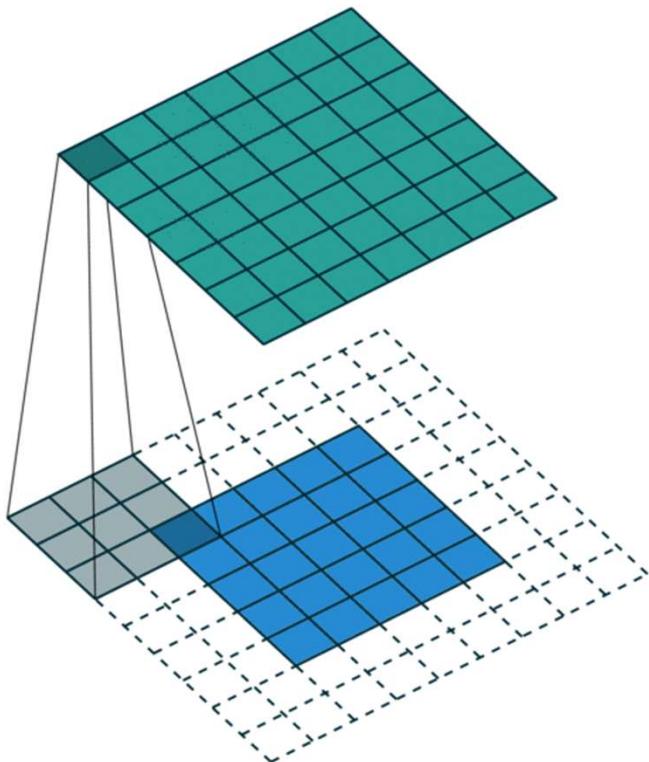
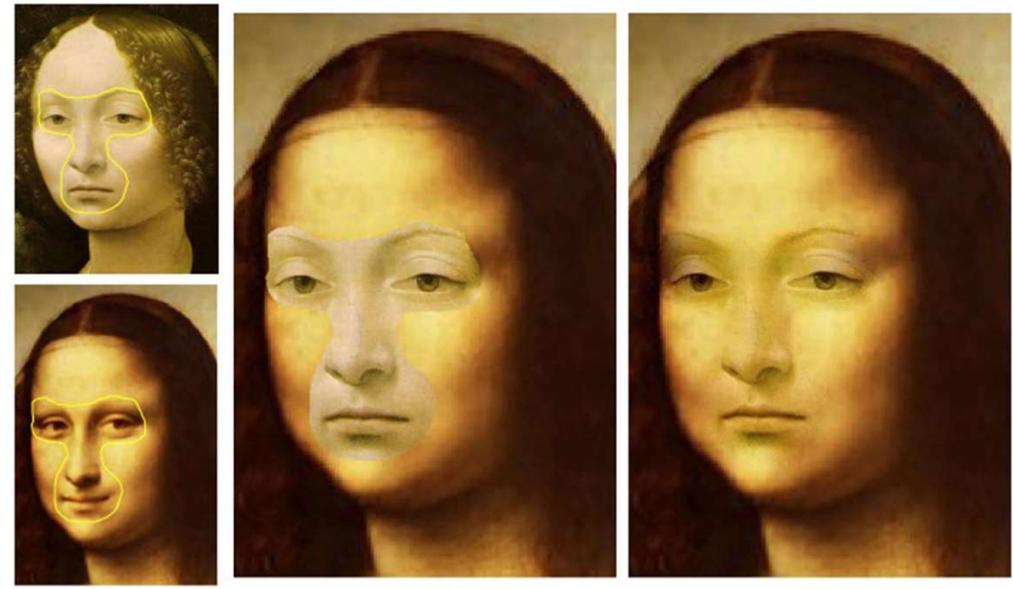


Image Convolution



Poisson Image Editing

# Task 1: Image Denoising



A Type of Image Enhancement

# Denoising代表性论文

A Non-Local Algorithm for Image Denoising.



Images

Books

Videos

Shopping

Maps

More

Tools



IEEE Xplore

<https://ieeexplore.ieee.org> > document

**A non-local algorithm for image denoising**

by A Buades · 2005 · Cited by 9478 — **A non-local algorithm for image denoising.** Abstract:  
We propose a new measure, the method noise, to evaluate and compare the performance of...

Published in: 2005 IEEE Computer Society Conference on  
Computer Vision and Pattern Recognition (CVPR'05)

Date of Conference: 20-25  
June 2005  
DOI: [10.1109/CVPR.2005.38](https://doi.org/10.1109/CVPR.2005.38)

Publisher: IEEE

Date Added to IEEE Xplore:  
25 July 2005  
Conference Location: San  
Diego, CA, USA

Print ISBN: 0-7695-2372-2

*We propose a new measure, the method noise, to evaluate and compare the performance of digital image denoising methods. We first compute and analyze this method noise for a wide class of denoising algorithms, namely the local smoothing filters. Second, we propose a new algorithm, the non local means (NL-means), based on a non local averaging of all pixels in the image. Finally, we present some experiments comparing the NL-means algorithm and the local smoothing filters.*

# 论文解读 — Introduction

The goal of image denoising methods is to recover the original image from a noisy measurement,

$$v(i) = u(i) + n(i), \quad (1)$$

where  $v(i)$  is the observed value,  $u(i)$  is the “true” value and  $n(i)$  is the noise perturbation at a pixel  $i$ . The best simple way to model the effect of noise on a digital image is to add a gaussian white noise. In that case,  $n(i)$  are i.i.d. gaussian values with zero mean and variance  $\sigma^2$ .

Several methods have been proposed to remove the noise and recover the true image  $u$ . Even though they may be very different in tools it must be emphasized that a wide class share the same basic remark : **denoising is achieved by averaging**. This averaging may be performed locally: the Gaussian smoothing model (Gabor [7]), the anisotropic filtering (Perona-Malik [11], Alvarez et al. [1]) and the neighborhood filtering (Yaroslavsky [16], Smith et al. [14], Tomasi et al. [15]), by the calculus of variations: the Total Variation minimization (Rudin-Osher-Fatemi [13]), or in the frequency domain: the empirical Wiener filters (Yaroslavsky [16]) and wavelet thresholding methods (Coiffman-Donoho [5, 4]).

问题建模

# 为什么光滑可以降噪？



Let  $X$  and  $Y$  be independent random variables that are normally distributed (and therefore also jointly so), then their sum is also normally distributed. i.e., if

$$X \sim N(\mu_X, \sigma_X^2)$$

$$Y \sim N(\mu_Y, \sigma_Y^2)$$

$$Z = X + Y,$$

then

$$Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2).$$

Let  $X_1, X_2, \dots, X_n$  i.i.d.  $N(0, \sigma^2)$

Then  $\frac{X_1 + X_2 + \dots + X_n}{n} \sim N(0, \frac{\sigma^2}{n})$

# 论文解读 — Introduction

Formally we define a denoising method  $D_h$  as a decomposition

$$v = D_h v + n(D_h, v),$$

where  $v$  is the noisy image and  $h$  is a filtering parameter which usually depends on the standard deviation of the noise. Ideally,  $D_h v$  is smoother than  $v$  and  $n(D_h, v)$  looks like the realization of a white noise. The decomposition of an image between a smooth part and a non smooth or oscillatory part is a current subject of research (for example Osher et al. [10]). In [8], Y. Meyer studied the suitable functional spaces for this decomposition. The primary scope of this latter study is not denoising since the oscillatory part contains both noise and texture.

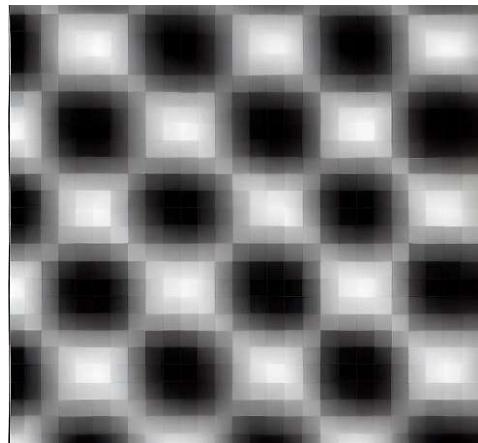
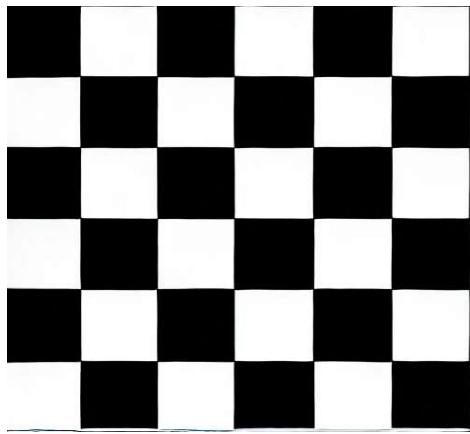
The denoising methods should not alter the original image  $u$ . Now, most denoising methods degrade or remove the fine details and texture of  $u$ . In order to better understand this removal, we shall introduce and analyze the *method noise*. The method noise is defined as the difference between the original (always slightly noisy) image  $u$  and its denoised version.

We also propose and analyze the *NL-means* algorithm, which is defined by the simple formula

$$NL[u](x) = \frac{1}{C(x)} \int_{\Omega} e^{-\frac{(G_a * |u(x+.) - u(y+.)|^2)(0)}{h^2}} u(y) dy,$$

where  $x \in \Omega$ ,  $C(x) = \int_{\Omega} e^{-\frac{(G_a * |u(x+.) - u(z+.)|^2)(0)}{h^2}} dz$  is a normalizing constant,  $G_a$  is a Gaussian kernel and  $h$  acts as a filtering parameter. This formula amounts to say that the denoised value at  $x$  is a mean of the values of all points whose gaussian neighborhood looks like the neighborhood of  $x$ . The main difference of the NL-means algorithm with respect to local filters or frequency domain filters is the systematic use of all possible self-predictions the image can provide, in the spirit of [6]. For a more detailed analysis on the NL-means algorithm and a more complete comparison, see [2].

# 论文解读 — Method



## 2.4. The neighborhood filtering

We call neighborhood filter any filter which restores a pixel by taking an average of the values of neighboring pixels with a similar grey level value. Yaroslavsky (1985) [16] averages pixels with a similar grey level value and belonging to the spatial neighborhood  $B_\rho(\mathbf{x})$ ,

$$YNF_{h,\rho} u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{B_\rho(\mathbf{x})} u(\mathbf{y}) e^{-\frac{|u(\mathbf{y}) - u(\mathbf{x})|^2}{h^2}} d\mathbf{y}, \quad (2)$$

where  $\mathbf{x} \in \Omega$ ,  $C(\mathbf{x}) = \int_{B_\rho(\mathbf{x})} e^{-\frac{|u(\mathbf{y}) - u(\mathbf{x})|^2}{h^2}} d\mathbf{y}$  is the normalization factor and  $h$  is a filtering parameter.

The problem with these filters is that comparing only grey level values in a single pixel is not so robust when these values are noisy. Neighborhood filters also create artificial shocks which can be justified by the computation of its method noise, see [3].

# 论文解读 — Method

## 3. NL-means algorithm

Given a discrete noisy image  $v = \{v(i) \mid i \in I\}$ , the estimated value  $NL[v](i)$ , for a pixel  $i$ , is computed as a weighted average of all the pixels in the image,

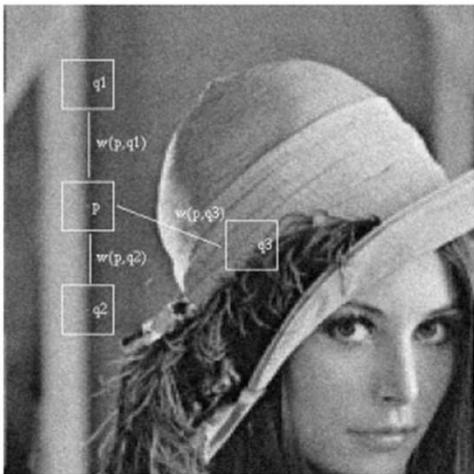
$$NL[v](i) = \sum_{j \in I} w(i, j)v(j),$$

where the family of weights  $\{w(i, j)\}_j$  depend on the similarity between the pixels  $i$  and  $j$ , and satisfy the usual conditions  $0 \leq w(i, j) \leq 1$  and  $\sum_j w(i, j) = 1$ .

The similarity between two pixels  $i$  and  $j$  depends on the similarity of the intensity gray level vectors  $v(\mathcal{N}_i)$  and  $v(\mathcal{N}_j)$ , where  $\mathcal{N}_k$  denotes a square neighborhood of fixed size and centered at a pixel  $k$ . This similarity is measured as a decreasing function of the weighted Euclidean distance,  $\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2$ , where  $a > 0$  is the standard deviation of the Gaussian kernel. The application of the Euclidean distance to the noisy neighborhoods raises the following equality

$$E\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2 = \|u(\mathcal{N}_i) - u(\mathcal{N}_j)\|_{2,a}^2 + 2\sigma^2.$$

This equality shows the robustness of the algorithm since in expectation the Euclidean distance conserves the order of similarity between pixels.



**Figure 1. Scheme of NL-means strategy.** Similar pixel neighborhoods give a large weight,  $w(p,q1)$  and  $w(p,q2)$ , while much different neighborhoods give a small weight  $w(p,q3)$ .

The pixels with a similar grey level neighborhood to  $v(\mathcal{N}_i)$  have larger weights in the average, see Figure 1. These weights are defined as,

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}},$$

where  $Z(i)$  is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}}$$

and the parameter  $h$  acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

The NL-means not only compares the grey level in a single point but the the geometrical configuration in a whole neighborhood. This fact allows a more robust comparison than neighborhood filters. Figure 1 illustrates this fact, the pixel  $q_3$  has the same grey level value of pixel  $p$ , but the neighborhoods are much different and therefore the weight  $w(p, q_3)$  is nearly zero.

# 大概如何实现

## 根据局部颜色分布计算pixel-dependent的 kernel权重

experimentation section we only compare the Yaroslavsky neighborhood filter.

The problem with these filters is that comparing only grey level values in a single pixel is not so robust when these values are noisy. Neighborhood filters also create artificial shocks which can be justified by the computation of its method noise, see [3].

### 3. NL-means algorithm

Given a discrete noisy image  $v = \{v(i) \mid i \in I\}$ , the estimated value  $NL[v](i)$ , for a pixel  $i$ , is computed as a weighted average of all the pixels in the image,

$$NL[v](i) = \sum_{j \in I} w(i, j)v(j),$$

expectation the Euclidean distance conserves the order of similarity between pixels.

The pixels with a similar grey level neighborhood to  $v(\mathcal{N}_i)$  have larger weights in the average, see Figure 1. These weights are defined as,

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}},$$

where  $Z(i)$  is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{\|v(\mathcal{N}_i) - v(\mathcal{N}_j)\|_{2,a}^2}{h^2}}$$

and the parameter  $h$  acts as a degree of filtering. It controls the decay of the exponential function and therefore the de-

# 结果展示



Figure 5. Denoising experience on a natural image. From left to right and from top to bottom: noisy image (standard deviation 20), Gauss filtering, anisotropic filtering, Total variation, Neighborhood filtering and NL-means algorithm. The removed details must be compared with the method noise experience, Figure 4.

# Summary

根据邻域相似性重新计算average权重  
使得相近区域计算平均来去噪  
从而保持原图像结构信息

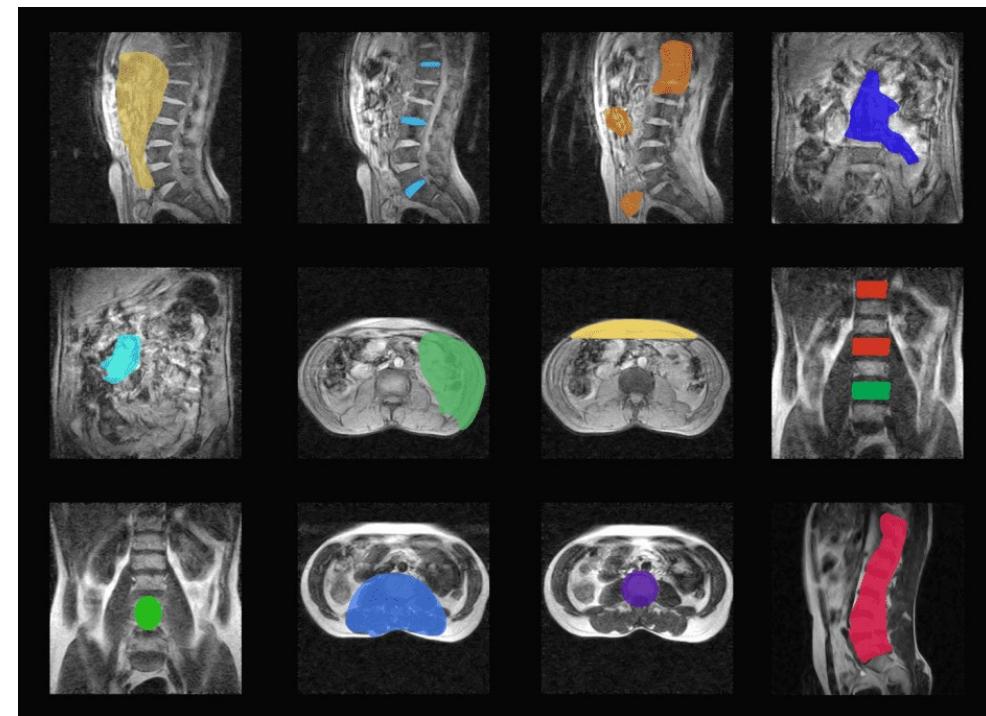


Easy to Implement  
With Kernels

# Task 2: Image Segmentation



# Task 2: Image Segmentation



# Segmentation 代表性论文

Contour Detection and Hierarchical Image Segmentation.

X

Published in: [IEEE Transactions on Pattern Analysis and Machine Intelligence](#) ( Volume: 33 , Issue: 5, May 2011)



IEEE Xplore

<https://ieeexplore.ieee.org> > document :

## Contour Detection and Hierarchical Image Segmentation

by P Arbeláez · 2010 · Cited by 6434 — This paper investigates two fundamental problems in

Page(s): 898 - 916

DOI: [10.1109/TPAMI.2010.161](https://doi.org/10.1109/TPAMI.2010.161)

Date of Publication: 26

August 2010

Publisher: IEEE

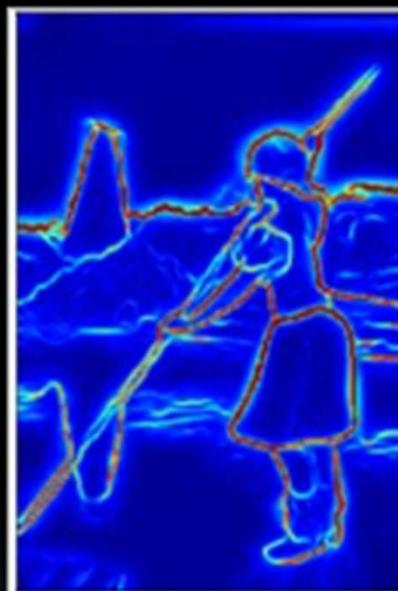
**Abstract**—This paper investigates two fundamental problems in computer vision: **contour detection and image segmentation**. We present state-of-the-art algorithms for both of these tasks. Our **contour detector combines multiple local cues into a globalization framework based on spectral clustering**. Our **segmentation algorithm consists of generic machinery for transforming the output of any contour detector into a hierarchical region tree**. In this manner, we reduce the problem of image segmentation to that of contour detection. Extensive experimental evaluation demonstrates that both our contour detection and segmentation methods significantly outperform competing algorithms. The automatically generated hierarchical segmentations can be interactively refined by user-specified annotations. Computation at multiple image resolutions provides a means of coupling our system to recognition applications.

# 问题解读

- From Contour to Segmentation



Original Image



Contour



Segmentation

# Introduction

## 1 INTRODUCTION

THIS paper presents a unified approach to contour detection and image segmentation. Contributions include:

- a high-performance contour detector, combining local and global image information,
- a method to transform any contour signal into a hierarchy of regions while preserving contour quality,
- extensive quantitative evaluation and the release of a new annotated data set.

Figs. 1 and 2 summarize our main results. The two figures represent the evaluation of multiple contour detection (Fig. 1) and image segmentation (Fig. 2) algorithms on the Berkeley Segmentation Data Set (BSDS300) [1], using the precision-recall framework introduced in [2]. This benchmark operates by comparing machine generated contours to human ground-truth data (Fig. 3) and allows evaluation of segmentations in the same framework by regarding region boundaries as contours.

Especially noteworthy in Fig. 1 is the contour detector *gPb*, which compares favorably with other leading techniques, providing equal or better precision for most choices of recall. In Fig. 2, *gPb-owt-ucm* provides universally better performance than alternative segmentation algorithms. We introduced the *gPb* and *gPb-owt-ucm* algorithms in [3] and

We begin with a review of the extensive literature on contour detection and image segmentation in Section 2.

Section 3 covers the development of the *gPb* contour detector. We couple multiscale local brightness, color, and texture cues to a powerful globalization framework using spectral clustering. The local cues, computed by applying oriented gradient operators at every location in the image, define an affinity matrix representing the similarity between pixels. From this matrix, we derive a generalized eigenproblem and solve for a fixed number of eigenvectors which encode contour information. Using a classifier to recombine this signal with the local cues, we obtain a large improvement over alternative globalization schemes built on top of similar cues.

To produce high-quality image segmentations, we link this contour detector with a generic grouping algorithm described in Section 4 and consisting of two steps. First, we introduce a new image transformation called the Oriented Watershed Transform for constructing a set of initial regions from an oriented contour signal. Second, using an agglomerative clustering procedure, we form these regions into a hierarchy which can be represented by an Ultrametric Contour Map, the real-valued image obtained by weighting each boundary by its scale of disappearance. We provide experiments on the BSDS300 as well as the BSDS500, a superset newly released here.

# Method — Contour

## 3 CONTOUR DETECTION

As a starting point for contour detection, we consider the work of Martin et al. [2], who define a function  $Pb(x, y, \theta)$  that predicts the posterior probability of a boundary with orientation  $\theta$  at each image pixel  $(x, y)$  by measuring the difference in local image brightness, color, and texture channels. In this section, we review these cues, introduce our own multiscale version of the  $Pb$  detector, and describe the new globalization method we run on top of this multiscale local detector.

### 3.1 Brightness, Color, and Texture Gradients

The basic building block of the  $Pb$  contour detector is the computation of an oriented gradient signal  $G(x, y, \theta)$  from an intensity image  $I$ . This computation proceeds by placing

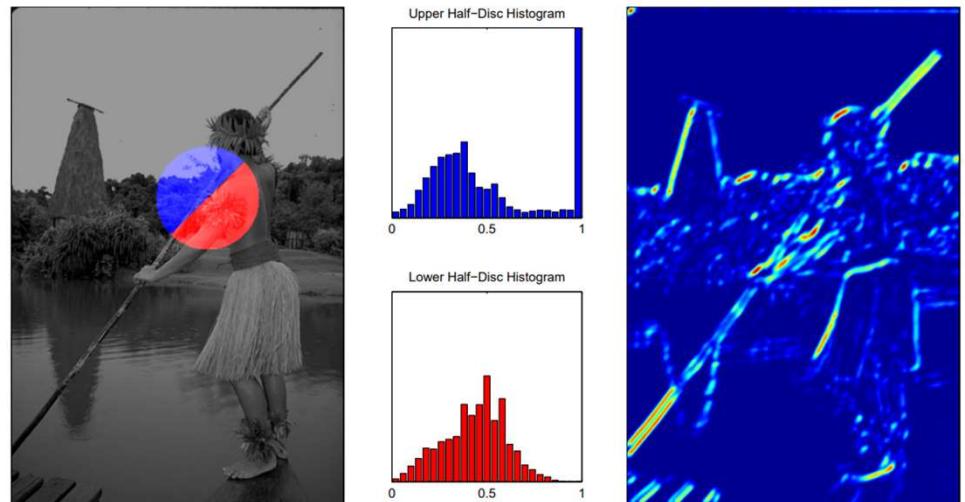


Fig. 4. **Oriented gradient of histograms.** Given an intensity image, consider a circular disc centered at each pixel and split by a diameter at angle  $\theta$ . We compute histograms of intensity values in each half-disc and output the  $\chi^2$  distance between them as the gradient magnitude. The blue and red distributions shown in the middle panel are the histograms of the pixel brightness values in the blue and red regions, respectively, in the left image. The right panel shows an example result for a disc of radius 5 pixels at orientation  $\theta = \frac{\pi}{4}$  after applying a second-order Savitzky-Golay smoothing filter to the raw histogram difference output. Note that the left panel displays a larger disc (radius 50 pixels) for illustrative purposes.

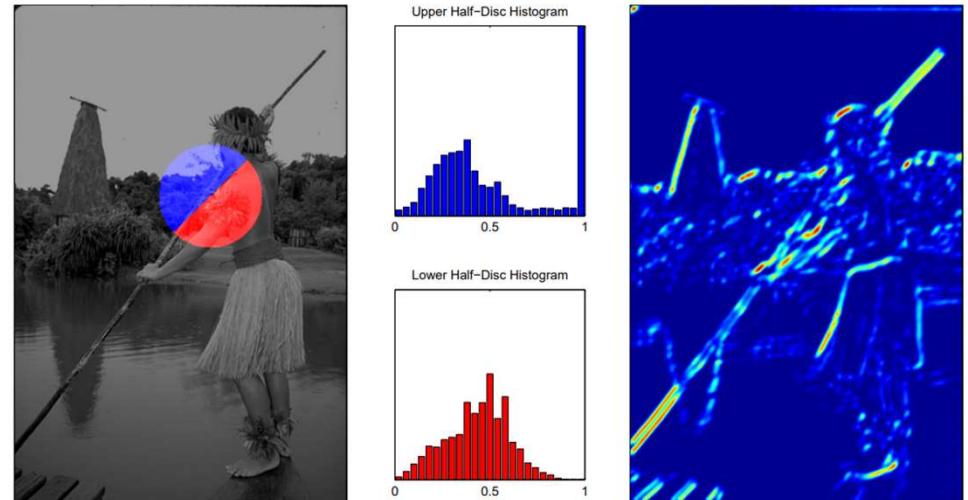
# Method — Contour

a circular disc at location  $(x, y)$  split into two half-discs by a diameter at angle  $\theta$ . For each half-disc, we histogram the intensity values of the pixels of  $I$  covered by it. The gradient magnitude  $G$  at location  $(x, y)$  is defined by the  $\chi^2$  distance between the two half-disc histograms  $g$  and  $h$ :

$$\chi^2(g, h) = \frac{1}{2} \sum_i \frac{(g(i) - h(i))^2}{g(i) + h(i)}. \quad (9)$$

We then apply second-order Savitzky-Golay filtering [63] to enhance local maxima and smooth out multiple detection peaks in the direction orthogonal to  $\theta$ . This is equivalent to fitting a cylindrical parabola, whose axis is orientated along direction  $\theta$ , to a local 2D window surrounding each pixel and replacing the response at the pixel with that estimated by the fit.

Fig. 4 shows an example. This computation is motivated by the intuition that contours correspond to image discontinuities and histograms provide a robust mechanism for modeling the content of an image region. A strong oriented gradient response means that a pixel is likely to lie on the boundary between two distinct regions.



**Fig. 4. Oriented gradient of histograms.** Given an intensity image, consider a circular disc centered at each pixel and split by a diameter at angle  $\theta$ . We compute histograms of intensity values in each half-disc and output the  $\chi^2$  distance between them as the gradient magnitude. The blue and red distributions shown in the middle panel are the histograms of the pixel brightness values in the blue and red regions, respectively, in the left image. The right panel shows an example result for a disc of radius 5 pixels at orientation  $\theta = \frac{\pi}{4}$  after applying a second-order Savitzky-Golay smoothing filter to the raw histogram difference output. Note that the left panel displays a larger disc (radius 50 pixels) for illustrative purposes.

# Method — 卡方距离

## 数学定义 [编辑]

若  $k$  个随机变量  $Z_1, \dots, Z_k$  是相互独立且符合标准正态分布的随机变量（数学期望为 0、方差为 1），则随机变量  $Z$  的平方和

$$X = \sum_{i=1}^k Z_i^2$$

被称为服从自由度为  $k$  的卡方分布，记作

$$\begin{aligned} X &\sim \chi^2(k) \\ X &\sim \chi_k^2 \end{aligned}$$

## 卡方距离 (Chi-square Measure)

卡方检验经常用来检验某一种观测分布是不是符合某一类典型的理论分布。

观察频数与期望频数越接近，两者之间的差异越小， $\chi^2$  值越小。

如果两个分布完全一致， $\chi^2$  值为 0，反之观察频数与期望频数差别越大，两者之间的差异越大， $\chi^2$  值越大。

大的  $\chi^2$  值表明观察频数远离期望频数

小的  $\chi^2$  值表明观察频数接近期望频数

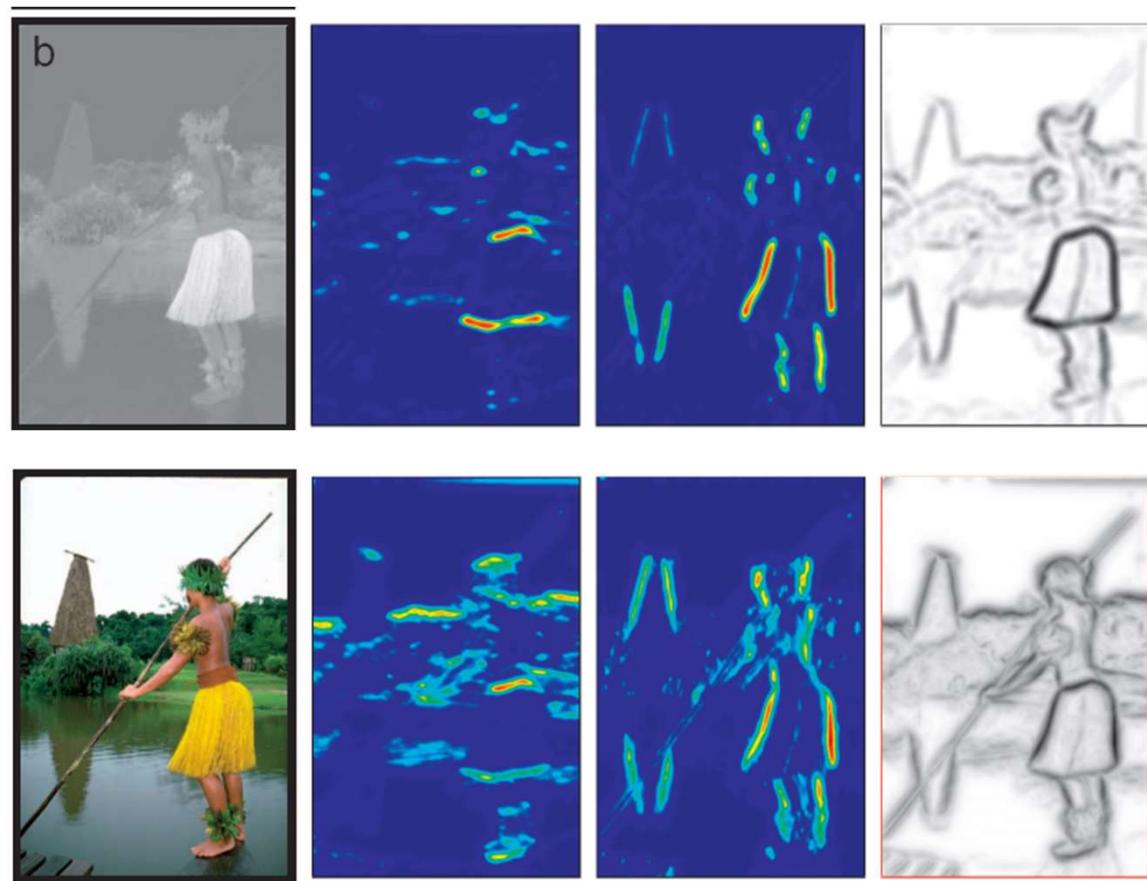
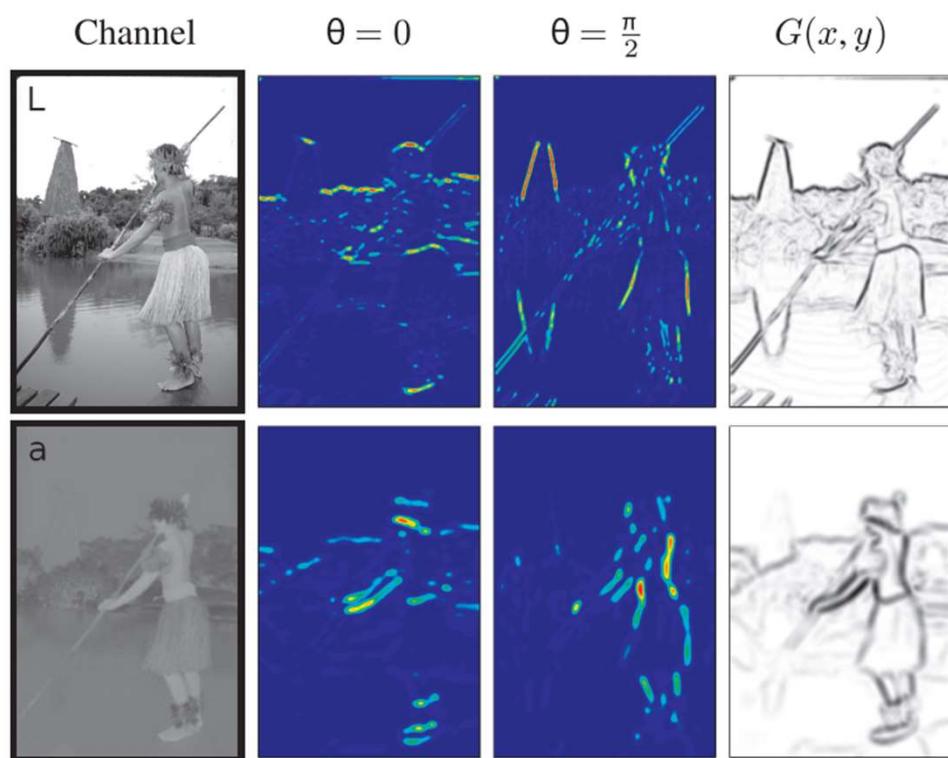
$\chi^2$  值是观察频数与期望频数之间距离的一种度量指标，也是假设成立与否的度量指标

计算公式如下：

$$\chi^2 = \sum_{i=1}^n \frac{(A_i - E_i)^2}{E_i} = \sum_{i=1}^k \frac{(A_i - np_i)^2}{kp_i}$$

其中， $A_i$  为  $A$  在水平  $i$  的观察频数， $E_i$  为  $E$  在水平  $i$  的期望频数， $k$  为总频数， $p_i$  为水平  $i$  的期望频率。水平  $i$  的期望频数  $E_i$  等于总频数  $k \times$  水平  $i$  的期望概率  $p_i$ 。当  $k$  比较大时， $\chi^2$  统计量近似服从  $n - 1$  个自由度的卡方分布。

# Method — Different Angles



$$mPb(x, y) = \max_{\theta} \{mPb(x, y, \theta)\}.$$

$mPb(x, y)$

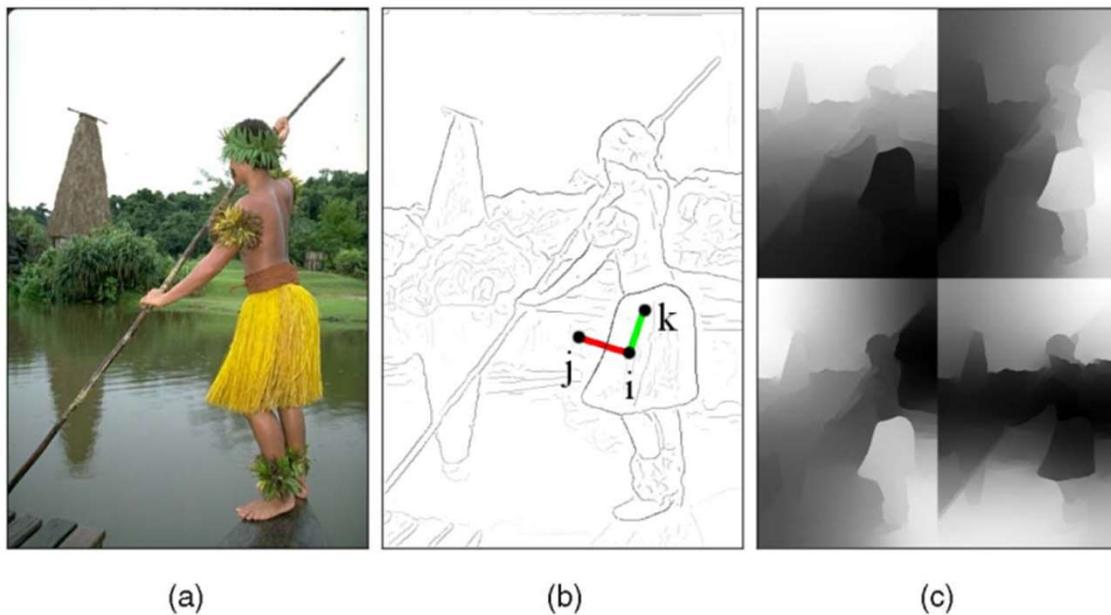
# Method — Clustering Region

Spectral clustering lies at the heart of our globalization machinery. The key element differentiating the algorithm described in this section from other approaches [45], [47] is the “soft” manner in which we use the eigenvectors obtained from spectral partitioning.

As input to the spectral clustering stage, we construct a sparse symmetric affinity matrix  $W$  using the *intervening contour* cue [49], [64], [65], the maximal value of  $mPb$  along a line connecting two pixels. We connect all pixels  $i$  and  $j$  within a fixed radius  $r$  with affinity:

$$W_{ij} = \exp\left(-\max_{p \in \overline{ij}}\{mPb(p)\}/\rho\right), \quad (12)$$

where  $\overline{ij}$  is the line segment connecting  $i$  and  $j$  and  $\rho$  is a constant. We set  $r = 5$  pixels and  $\rho = 0.1$ .



Pixels  $i$  and  $j$  have a low affinity as a strong boundary separates them whereas  $i$  and  $k$  have high affinity

# Spectral Clustering

## Spectral clustering

From Wikipedia, the free encyclopedia

In multivariate statistics, **spectral clustering** techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset.

In application to image segmentation, spectral clustering is known as [segmentation-based object categorization](#).

**Definitions** [edit]

Given an enumerated set of data points, the [similarity matrix](#) may be defined as a symmetric matrix  $A$ , where  $A_{ij} \geq 0$  represents a measure of the similarity between data points with indices  $i$  and  $j$ . The general approach to spectral clustering is to use a standard [clustering](#) method (there are many such methods,  $k$ -means is discussed [below](#)) on relevant [eigenvectors](#) of a Laplacian matrix of  $A$ . There are many different ways to define a Laplacian which have different mathematical interpretations, and so the clustering will also have different interpretations. The eigenvectors that are relevant are the ones that correspond to several smallest eigenvalues of the Laplacian except for the smallest eigenvalue which will have a value of 0. For computational efficiency, these eigenvectors are often computed as the eigenvectors corresponding to the largest several eigenvalues of a function of the Laplacian.

**Laplacian matrix** [edit]

Spectral clustering is well known to relate to partitioning of a mass-spring system, where each mass is associated with a data point and each spring stiffness corresponds to a weight of an edge describing a similarity of the two related data points, as in the [spring system](#). Specifically, the classical reference [1] explains that the eigenvalue problem describing transversal vibration modes of a mass-spring system is exactly the same as the eigenvalue problem for the graph [Laplacian matrix](#) defined as

$$L := D - A,$$

where  $D$  is the [diagonal matrix](#)

$$D_{ii} = \sum_j A_{ij},$$

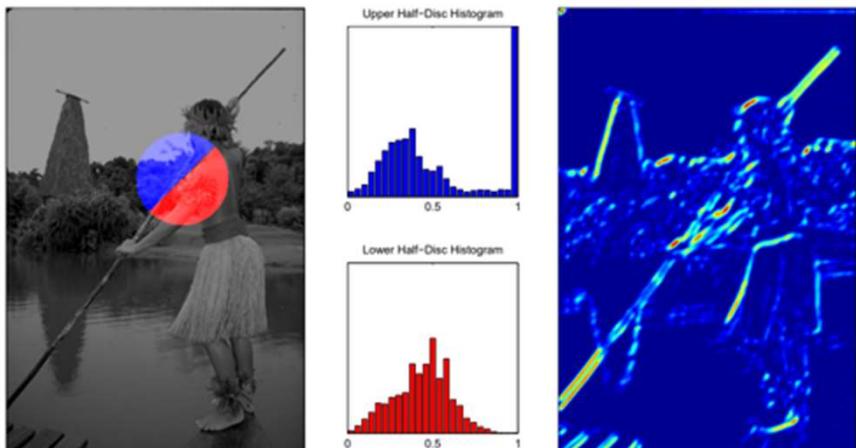
In order to introduce global information, we define  $D_{ii} = \sum_j W_{ij}$  and solve for the generalized eigenvectors  $\{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n\}$  of the system  $(D - W)\mathbf{v} = \lambda D\mathbf{v}$  (2) corresponding to the  $n + 1$  smallest eigenvalues  $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_n$ . Fig. 7 displays an example with four eigenvectors. In practice, we use  $n = 16$ .

At this point, the standard Normalized Cuts approach associates with each pixel a length  $n$  descriptor formed from entries of the  $n$  eigenvectors and uses a clustering algorithm such as K-means to create a hard partition of the image. Unfortunately, this can lead to an incorrect segmentation as large uniform regions in which the eigenvectors vary smoothly are broken up. Fig. 7 shows an example for which such gradual variation in the eigenvectors across the sky region results in an incorrect partition.

[A Tutorial on Spectral Clustering.](#)

# How to Implement

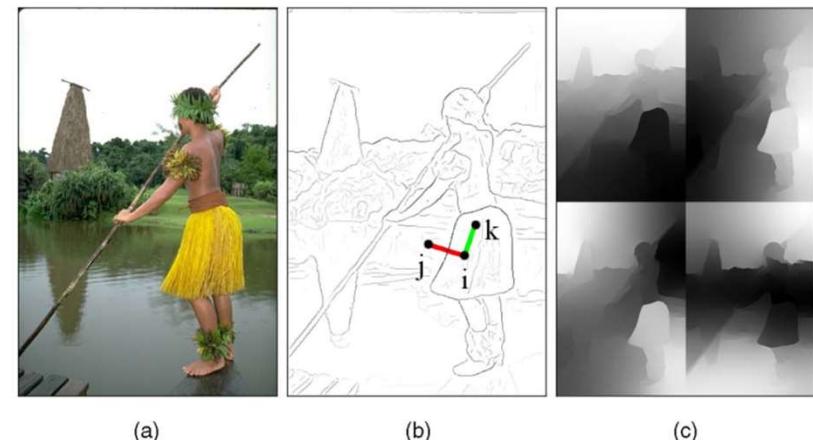
- Contour Detection:
  - Histogram
  - 计算卡方距离
  - Argmax angle



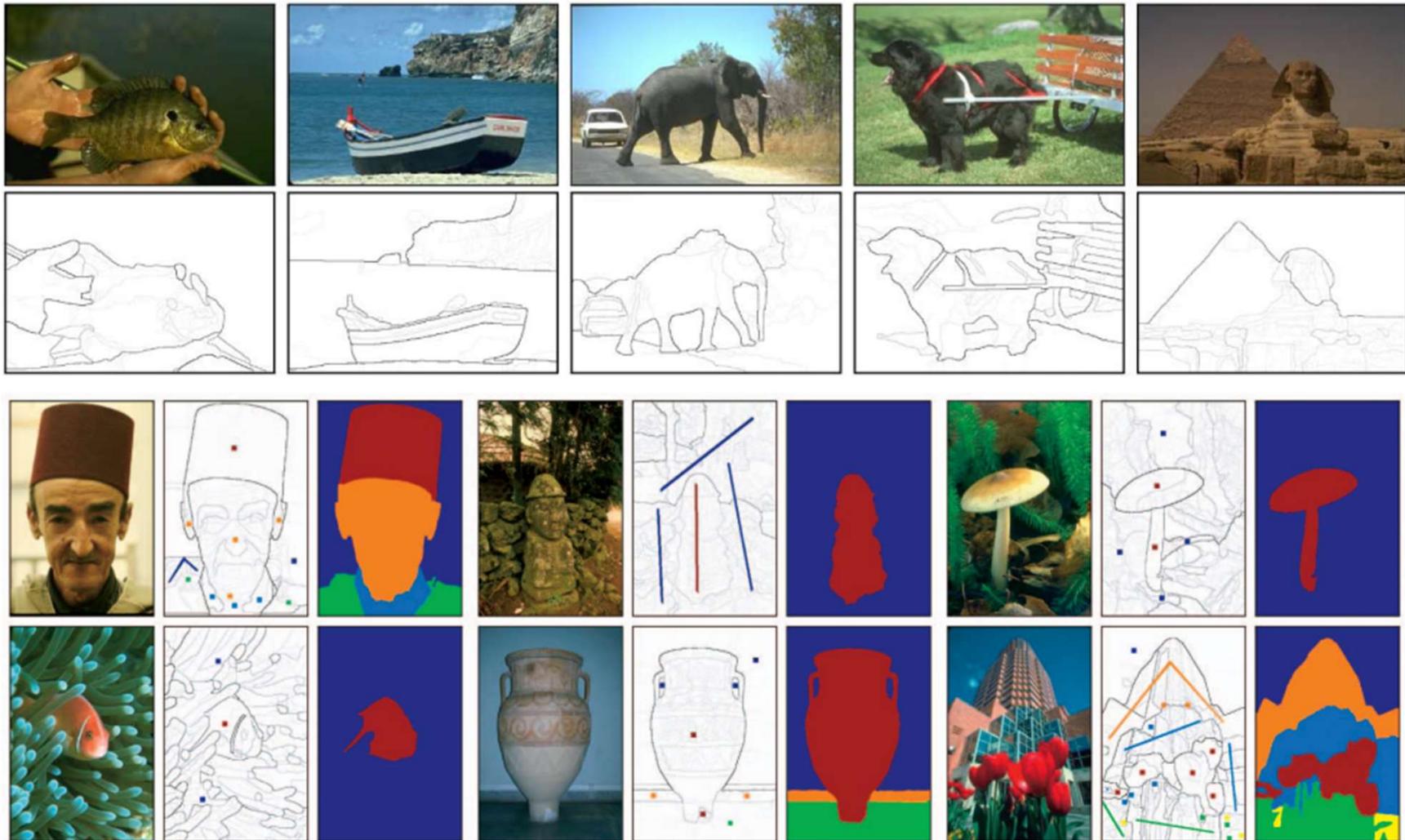
- Clustering Segmentation:
    - 计算加权邻接矩阵
    - 计算特征向量
- numpy.linalg.eig #

`linalg.eig(a)`

Compute the eigenvalues and right eigenvectors of a square array.

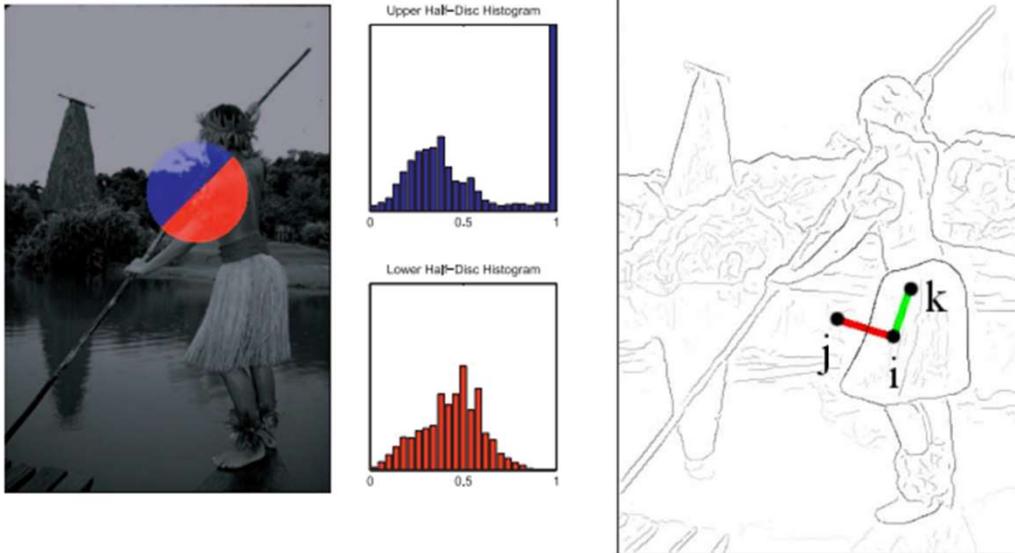


# Result



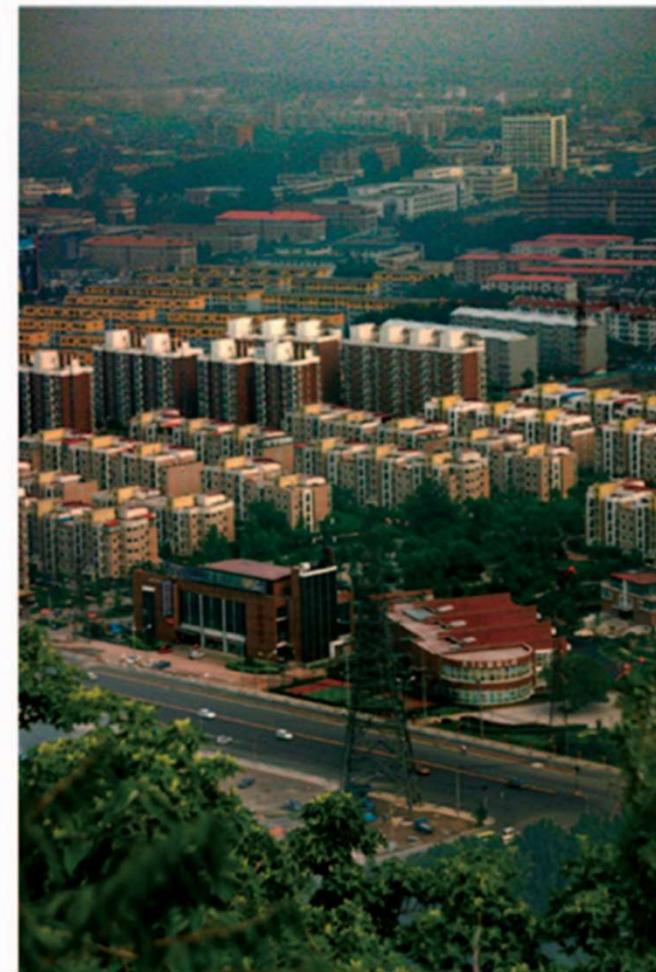
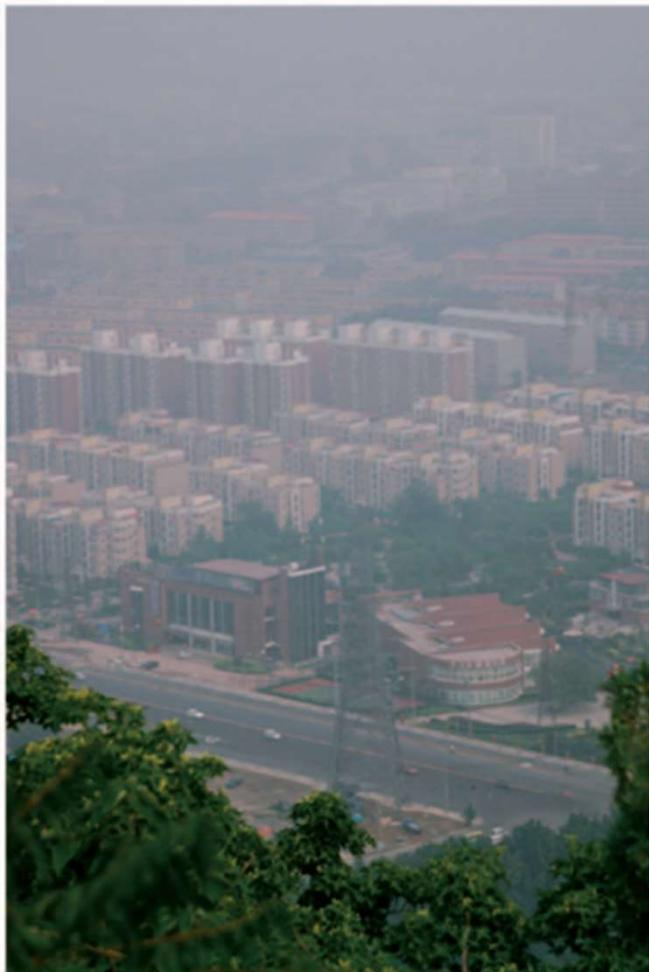
# Summary

根据角度区分区域的卡方距离得到contour  
根据跨越contour的概率计算邻近矩阵  
谱聚类得到Segmentation



Easy to Implement  
With Histogram and eig

# Task 3: Haze Removal



# Haze Removal代表性论文



single image haze removal using dark channel



All

Videos

Images

Shopping

Books

Maps

More

Tools



CUHK MMLab

<http://mmlab.ie.cuhk.edu.hk> > Haze



## Single Image Haze Removal Using Dark Channel Prior

by K He · 2010 · Cited by 9412 — Abstract—In this paper, we propose a **simple** but effective **image prior—dark channel prior** to remove haze from a single input image. The **dark...**

13 pages

## Single image haze removal using dark channel prior

Publisher: IEEE

Cite This



Kaiming He; Jian Sun; Xiaou Tang All Authors

563

Cites in  
Papers

19

Cites in  
Patents

18548

Full  
Text  
Views



### Abstract

**Abstract:**  
In this paper, we propose a simple but effective image prior - dark channel prior to remove haze from a single input image. The dark channel prior is a kind of statistics of the haze-free outdoor images. It is based on a key observation - most local patches in haze-free outdoor images contain some pixels which have very low intensities in at least one color channel. Using this prior with the haze imaging model, we can directly estimate the thickness of the haze and recover a high quality haze-free image. Results on a variety of outdoor haze images demonstrate the power of the proposed prior. Moreover, a high quality depth map can also be obtained as a by-product of haze removal.

### 1. Introduction

### 2. Background

### 3. Dark Channel Prior

### 4. Haze Removal Using Dark Channel Prior

### 5. Experimental Results

Show Full  
Outline ▾

### Published in:

2009 IEEE Conference on Computer Vision and  
Pattern Recognition

Date of Conference: 20-25

June 2009

DOI:

10.1109/CVPR.2009.5206515

Date Added to IEEE Xplore: Publisher: IEEE

18 August 2009

# Kaiming He

TITLE	CITED BY	YEAR
Deep Residual Learning for Image Recognition K He, X Zhang, S Ren, J Sun Computer Vision and Pattern Recognition (CVPR), 2016	241580	2016
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks S Ren, K He, R Girshick, J Sun Neural Information Processing Systems (NIPS), 2015	80318	2015
Mask R-CNN K He, G Gkioxari, P Dollár, R Girshick International Conference on Computer Vision (ICCV), 2017	38836	2017
Focal Loss for Dense Object Detection TY Lin, P Goyal, R Girshick, K He, P Dollár International Conference on Computer Vision (ICCV), 2017	33106	2017
Feature Pyramid Networks for Object Detection TY Lin, P Dollár, R Girshick, K He, B Hariharan, S Belongie Computer Vision and Pattern Recognition (CVPR), 2017	28628	2017
Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification K He, X Zhang, S Ren, J Sun International Conference on Computer Vision (ICCV), 2015	25409	2015
Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition K He, X Zhang, S Ren, J Sun European Conference on Computer Vision (ECCV), 2014	16113	2014
Learning a Deep Convolutional Network for Image Super-Resolution C Dong, CC Loy, K He, X Tang European Conference on Computer Vision (ECCV), 2014	15019 *	2014
Aggregated Residual Transformations for Deep Neural Networks S Xie, R Girshick, P Dollár, Z Tu, K He Computer Vision and Pattern Recognition (CVPR), 2017	13205	2017
Momentum Contrast for Unsupervised Visual Representation Learning K He, H Fan, Y Wu, S Xie, R Girshick Computer Vision and Pattern Recognition (CVPR), 2020	12899	2020
Identity Mappings in Deep Residual Networks K He, X Zhang, S Ren, J Sun European Conference on Computer Vision (ECCV), 2016	12703	2016
Non-local Neural Networks X Wang, R Girshick, A Gupta, K He Computer Vision and Pattern Recognition (CVPR), 2018	11383	2018
Single Image Haze Removal using Dark Channel Prior K He, J Sun, X Tang Computer Vision and Pattern Recognition (CVPR), 2009	9460	2009
Guided Image Filtering K He, J Sun, X Tang European Conference on Computer Vision (ECCV), 2010	8286 *	2010



## Kaiming He 何恺明

Associate Professor, EECS, MIT

Office: 45-701H, 51 Vassar St, Cambridge, MA 02139

kaiming@mit.edu

I am an Associate Professor of the [Department of Electrical Engineering and Computer Science \(EECS\)](#) at [Massachusetts Institute of Technology \(MIT\)](#), as of Feb. 2024. My research covers a wide range of topics in computer vision and deep learning. Through the lens of computer vision problems, I aim to develop generalizable methods applicable to various domains. My research currently focuses on building computer models that can learn representations and develop intelligence from and for the complex world. The long-term goal of my research is to augment human intelligence with more capable artificial intelligence.

I am best known for my work on [Deep Residual Networks \(ResNets\)](#), with the residual connections therein now being used everywhere in modern deep learning models, including Transformers (e.g., GPT, ChatGPT), AlphaGo Zero, AlphaFold, Diffusion Models, and more. I am also known for my work on visual object detection and segmentation (e.g., [Faster R-CNN](#), [Mask R-CNN](#)) and visual self-supervised learning (e.g., [MoCo](#), [MAE](#)). I am a recipient of several prestigious awards, including the PAMI Young Researcher Award in 2018, the Best Paper Award in CVPR 2009, CVPR 2016, ICCV 2017, the Best Student Paper Award in ICCV 2017, the Best Paper Honorable Mention in ECCV 2018, CVPR 2021, and the Everingham Prize in ICCV 2021. My publications have over [500,000](#) citations (as of Nov. 2023) with an increase of over 100,000 per year.

# 背后的小故事

这个段子的主角就是现今计算机视觉领域的超级大牛何恺明<sup>+</sup> (Kaiming He - FAIR) , 想必在圈儿内的人都听过吧。当时是T老板来我校招生，介绍了Kaiming读phd<sup>+</sup>期间的一篇paper以及围绕这篇paper的轶事。。

事情大概是这样，Kaiming本身就是光环学生（高考状元），本科毕业后在T教授的强力忽悠下去了那里读phd，可谓是被寄予厚望。和Kaiming同期进来的一批学生（我们后文简称kaiming的peer）当年也都是各大高校top3（现在大家可以试试随便Google一下，你会发现他们都已经成为CV<sup>+</sup>圈儿内有头有脸的人物）。。。但是他们的光环效应<sup>+</sup>肯定是没有状元强的。

背景交代完毕。。那么现在phd开始了，大家处在同一个起跑线上：事实证明和kaiming一起进来的同学们各个都是paper machine。那时还是2010年之前，大陆高校（包括香港）想中篇顶级cv会还是颇有难度的，然而kaiming的peer们cvpr<sup>+</sup>, iccv, eccv什么的已经开始刷的飞起，一篇两篇三四篇，poster oral六七篇。。一年两年过去了，和kaiming进来的一批大牛们基本各个都是手握至少2-3篇cvpr（或同等级），然后第三年申phd的拿到各种四大的phd offer。

然而，作为最被寄予厚望的种子选手和光环选手kaiming还是一无所有。众所周知香港大部分高校phd也就4年，一眨眼几年过去了，也该想thesis了，但是两手空空，捉急啊...而且，Kaiming的phd大部分时间是在msra度过，那时候的msra是圣地啊，公认的大陆少有的几个能发paper的地方，那时候能去msra当intern的各个都是大牛。在这种peer环境下，光环学生，高考状元，这么久了还是没paper，而且据T教授说，甚至是投都没有投过！！各位观众可以做一下换位思考，想想如果是你，你会有多焦虑。据T教授亲口说，当时的Kaiming已经在考虑随便拿个学位回母校教书（对你没听错，是回高中母校当高中老师！！），你们感受一下。。。

然而。天才就是天才，神牛就是神牛。我记得应该是2009年吧，一篇dark channel prior的cvpr paper横空出世，当时T教授评价这篇paper是“拨云见雾，神来之笔”。

没错，就是Kaiming的第一篇paper就成为了cvpr best paper，亚洲的第一篇cvpr best paper，我建议各位即使不是cv领域的人也可以去翻来读一下这篇paper，你就明白本科学个物理和数学思考问题和cs的人有啥不同。。。随之而来的当然就是新的光环，等等。

自那以后呢，他就成了现在大家都膜拜的Kaiming了，发paper有如滔滔洪水一般一发不可收。。。

回到这个问题上，讲这个段子呢（虽然我不知道真假，因为不知道T教授在讲的时候是不是添油加醋了），是想告诉大家三个道理：

1. phd第三年还没有发出paper, 不要着急！要有耐心。
2. 说不定第四年你就发出来了，而且还可能像Kaiming一样中best paper。
3. 就算第四年中不了，你也可以回高中母校当高中老师嘛！

<https://www.zhihu.com/question/58843207/answer/160184256>

# 数学模型

**Abstract**—In this paper, we propose a simple but effective image prior—dark channel prior to remove haze from a single input image. The dark channel prior is a kind of statistics of outdoor haze-free images. It is based on a key observation—most local patches in outdoor haze-free images contain some pixels whose intensity is very low in at least one color channel. Using this prior with the haze imaging model, we can directly estimate the thickness of the haze and recover a high-quality haze-free image. Results on a variety of hazy images demonstrate the power of the proposed prior. Moreover, a high-quality depth map can also be obtained as a byproduct of haze removal.

## 1 INTRODUCTION

IMAGES of outdoor scenes are usually degraded by the turbid medium (e.g., particles and water droplets) in the atmosphere. Haze, fog, and smoke are such phenomena due to atmospheric absorption and scattering. The irradiance received by the camera from the scene point is attenuated along the line of sight. Furthermore, the incoming light is blended with the *airlight* [1]—ambient light reflected into the line of sight by atmospheric particles. The degraded images lose contrast and color fidelity, as shown in Fig. 1a. Since the amount of scattering depends on the distance of the scene points from the camera, the degradation is spatially variant.

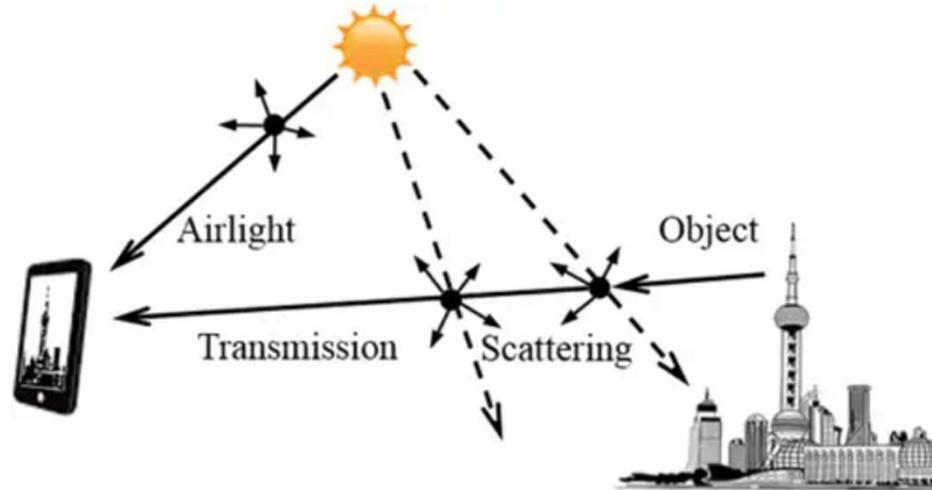
In this paper, we propose a novel prior—*dark channel prior*—for single image haze removal. The dark channel prior is based on the statistics of outdoor haze-free images. We find that, in most of the local regions which do not cover the sky, some pixels (called *dark pixels*) very often have very low intensity in at least one color (RGB) channel. In hazy images, the intensity of these dark pixels in that channel is mainly contributed by the airlight. Therefore, these dark pixels can directly provide an accurate estimation of the haze transmission. Combining a haze imaging model and a soft matting interpolation method, we can recover a high-quality haze-free image and produce a good depth map.

# 数学模型

In computer vision and computer graphics, the model widely used to describe the formation of a hazy image is [2], [5], [10], [11]:

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x})), \quad (1)$$

where  $\mathbf{I}$  is the observed intensity,  $\mathbf{J}$  is the scene radiance,  $\mathbf{A}$  is the global atmospheric light, and  $t$  is the medium transmission describing the portion of the light that is not scattered and reaches the camera. The goal of haze removal is to recover  $\mathbf{J}$ ,  $\mathbf{A}$ , and  $t$  from  $\mathbf{I}$ . For an  $N$ -pixel color image  $\mathbf{I}$ , there are  $3N$  constraints and  $4N + 3$  unknowns. This makes the problem of haze removal inherently ambiguous.



When the atmosphere is homogenous, the transmission  $t$  can be expressed as

$$t(\mathbf{x}) = e^{-\beta d(\mathbf{x})}, \quad (2)$$

where  $\beta$  is the scattering coefficient of the atmosphere and  $d$  is the scene depth. This equation indicates that the scene radiance is attenuated exponentially with the depth. If we can recover the transmission, we can also recover the depth up to an unknown scale.

# Dark Channel Prior

To formally describe this observation, we first define the concept of a *dark channel*. For an arbitrary image  $\mathbf{J}$ , its dark channel  $J^{\text{dark}}$  is given by

$$J^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_{c \in \{r,g,b\}} J^c(\mathbf{y}) \right), \quad (5)$$

where  $J^c$  is a color channel of  $\mathbf{J}$  and  $\Omega(\mathbf{x})$  is a local patch centered at  $\mathbf{x}$ . A dark channel is the outcome of two minimum operators:  $\min_{c \in \{r,g,b\}}$  is performed on each pixel (Fig. 3b), and  $\min_{\mathbf{y} \in \Omega(\mathbf{x})}$  is a minimum filter (Fig. 3c). The minimum operators are commutative.

Using the concept of a dark channel, our observation says that if  $\mathbf{J}$  is an outdoor haze-free image, except for the sky region, the intensity of  $\mathbf{J}$ 's dark channel is low and tends to be zero:

$$J^{\text{dark}} \rightarrow 0. \quad (6)$$

We call this observation *dark channel prior*.

The low intensity in the dark channel is mainly due to three factors: a) shadows, e.g., the shadows of cars, buildings, and the inside of windows in cityscape images, or the shadows of leaves, trees, and rocks in landscape images; b) colorful objects or surfaces, e.g., any object with low reflectance in any color channel (for example, green grass/tree/plant, red or yellow flower/leaf, and blue water surface) will result in low values in the dark channel; c) dark objects or surfaces, e.g., dark tree trunks and stones. As the natural outdoor images are usually colorful and full of shadows, the dark channels of these images are really dark!

# Dark Channel Prior

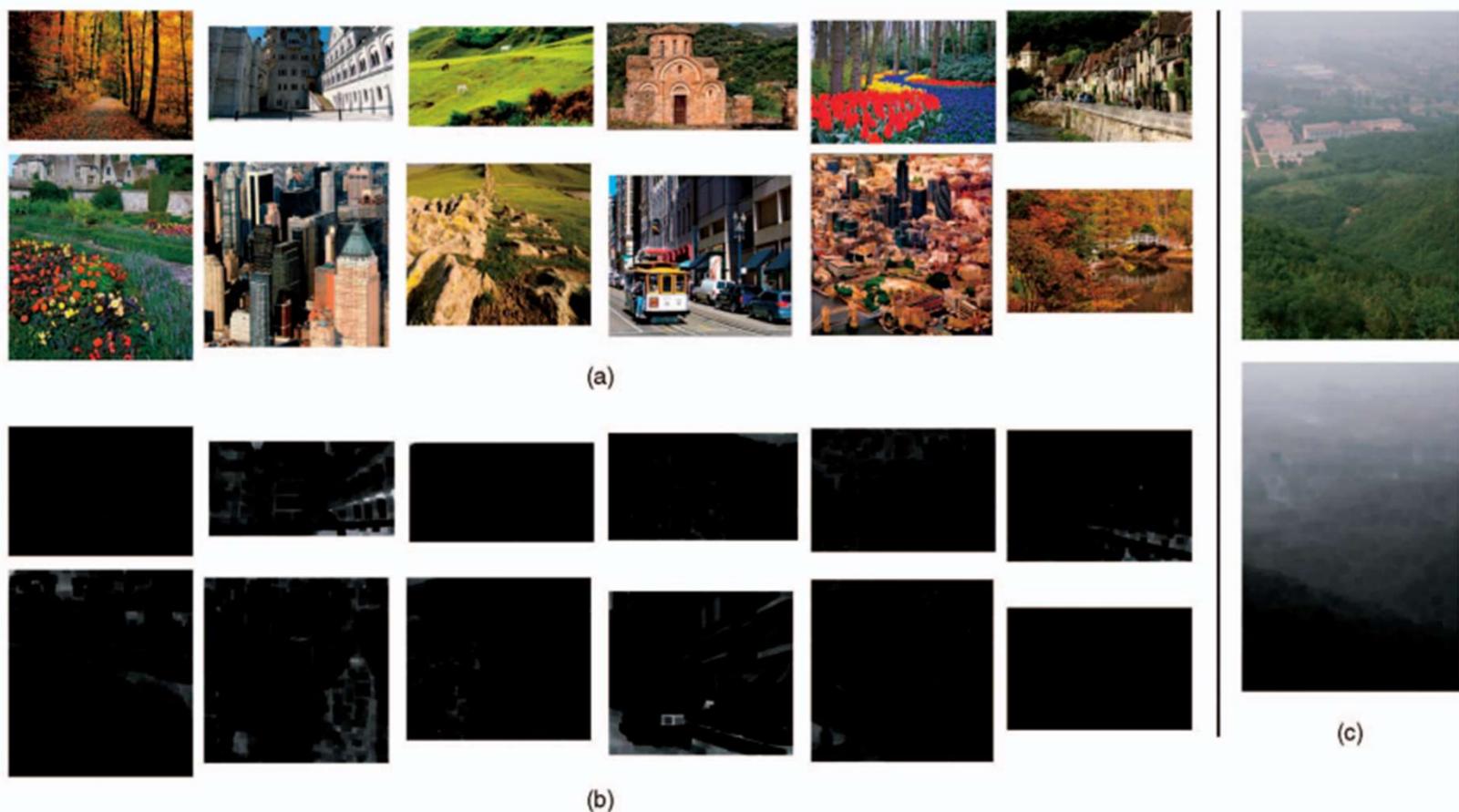


Fig. 4. (a) Example images in our haze-free image database. (b) The corresponding dark channels. (c) A hazy image and its dark channel.

# Haze Removal using Dark Channel Prior

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x})), \quad (1)$$

We assume that the atmospheric light  $\mathbf{A}$  is given. An automatic method to estimate  $\mathbf{A}$  is proposed in Section 4.3. We first normalize the haze imaging equation (1) by  $A$ :

$$\frac{I^c(\mathbf{x})}{A^c} = t(\mathbf{x}) \frac{J^c(\mathbf{x})}{A^c} + 1 - t(\mathbf{x}). \quad (7)$$

Note that we normalize each color channel independently.

We further assume that the transmission in a local patch  $\Omega(\mathbf{x})$  is constant. We denote this transmission as  $\tilde{t}(\mathbf{x})$ . Then, we calculate the dark channel on both sides of (7). Equivalently, we put the minimum operators on both sides:

$$\begin{aligned} \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{I^c(\mathbf{y})}{A^c} \right) &= \tilde{t}(\mathbf{x}) \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{J^c(\mathbf{y})}{A^c} \right) \\ &+ 1 - \tilde{t}(\mathbf{x}). \end{aligned} \quad (8)$$

Since  $\tilde{t}(\mathbf{x})$  is a constant in the patch, it can be put on the outside of the min operators.

As the scene radiance  $\mathbf{J}$  is a haze-free image, the dark channel of  $J$  is close to zero due to the dark channel prior:

$$J^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c J^c(\mathbf{y}) \right) = 0. \quad (9)$$

As  $A^c$  is always positive, this leads to

$$\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{J^c(\mathbf{y})}{A^c} \right) = 0. \quad (10)$$

Putting (10) into (8), we can eliminate the multiplicative term and estimate the transmission  $\tilde{t}$  simply by

$$\tilde{t}(\mathbf{x}) = 1 - \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{I^c(\mathbf{y})}{A^c} \right). \quad (11)$$

In fact,  $\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{I^c(\mathbf{y})}{A^c} \right)$  is the dark channel of the normalized hazy image  $\frac{I^c(\mathbf{y})}{A^c}$ . It directly provides the estimation of the transmission.

# Haze Removal using Dark Channel Prior

## 4.4 Recovering the Scene Radiance

With the atmospheric light and the transmission map, we can recover the scene radiance according to (1). But the direct attenuation term  $J(x)t(x)$  can be very close to zero when the transmission  $t(x)$  is close to zero. The directly recovered scene radiance  $J$  is prone to noise. Therefore, we restrict the transmission  $t(x)$  by a lower bound  $t_0$ , i.e., we preserve a small amount of haze in very dense haze regions. The final scene radiance  $J(x)$  is recovered by

$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A. \quad (22)$$

A typical value of  $t_0$  is 0.1. Since the scene radiance is usually not as bright as the atmospheric light, the image after haze removal looks dim. So we increase the exposure of  $J(x)$  for display. Some final recovered images are shown in Fig. 6e.

# How to Implement

Very Easy to implement  
with Matrix Operation!

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x})), \quad (1)$$

We assume that the atmospheric light  $\mathbf{A}$  is given. An automatic method to estimate  $\mathbf{A}$  is proposed in Section 4.3. We first normalize the haze imaging equation (1) by  $A$ :

$$\frac{I^c(\mathbf{x})}{A^c} = t(\mathbf{x}) \frac{J^c(\mathbf{x})}{A^c} + 1 - t(\mathbf{x}). \quad (7)$$

Note that we normalize each color channel independently.

We further assume that the transmission in a local patch  $\Omega(\mathbf{x})$  is constant. We denote this transmission as  $\tilde{t}(\mathbf{x})$ . Then, we calculate the dark channel on both sides of (7). Equivalently, we put the minimum operators on both sides:

$$\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{I^c(\mathbf{y})}{A^c} \right) = \tilde{t}(\mathbf{x}) \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{J^c(\mathbf{y})}{A^c} \right) + 1 - \tilde{t}(\mathbf{x}). \quad (8)$$

Since  $\tilde{t}(\mathbf{x})$  is a constant in the patch, it can be put on the outside of the min operators.

As the scene radiance  $\mathbf{J}$  is a haze-free image, the dark channel of  $J$  is close to zero due to the dark channel prior:

$$J^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c J^c(\mathbf{y}) \right) = 0. \quad (9)$$

As  $A^c$  is always positive, this leads to

$$\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{J^c(\mathbf{y})}{A^c} \right) = 0. \quad (10)$$

Putting (10) into (8), we can eliminate the multiplicative term and estimate the transmission  $\tilde{t}$  simply by

$$\tilde{t}(\mathbf{x}) = 1 - \min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{I^c(\mathbf{y})}{A^c} \right). \quad (11)$$

In fact,  $\min_{\mathbf{y} \in \Omega(\mathbf{x})} \left( \min_c \frac{I^c(\mathbf{y})}{A^c} \right)$  is the dark channel of the normalized hazy image  $\frac{I^c(\mathbf{y})}{A^c}$ . It directly provides the estimation of the transmission.

## 4.4 Recovering the Scene Radiance

With the atmospheric light and the transmission map, we can recover the scene radiance according to (1). But the direct attenuation term  $\mathbf{J}(\mathbf{x})t(\mathbf{x})$  can be very close to zero when the transmission  $t(\mathbf{x})$  is close to zero. The directly recovered scene radiance  $\mathbf{J}$  is prone to noise. Therefore, we restrict the transmission  $t(\mathbf{x})$  by a lower bound  $t_0$ , i.e., we preserve a small amount of haze in very dense haze regions. The final scene radiance  $\mathbf{J}(\mathbf{x})$  is recovered by

$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - \mathbf{A}}{\max(t(\mathbf{x}), t_0)} + \mathbf{A}. \quad (22)$$

A typical value of  $t_0$  is 0.1. Since the scene radiance is usually not as bright as the atmospheric light, the image after haze removal looks dim. So we increase the exposure of  $\mathbf{J}(\mathbf{x})$  for display. Some final recovered images are shown in Fig. 6e.

# Results

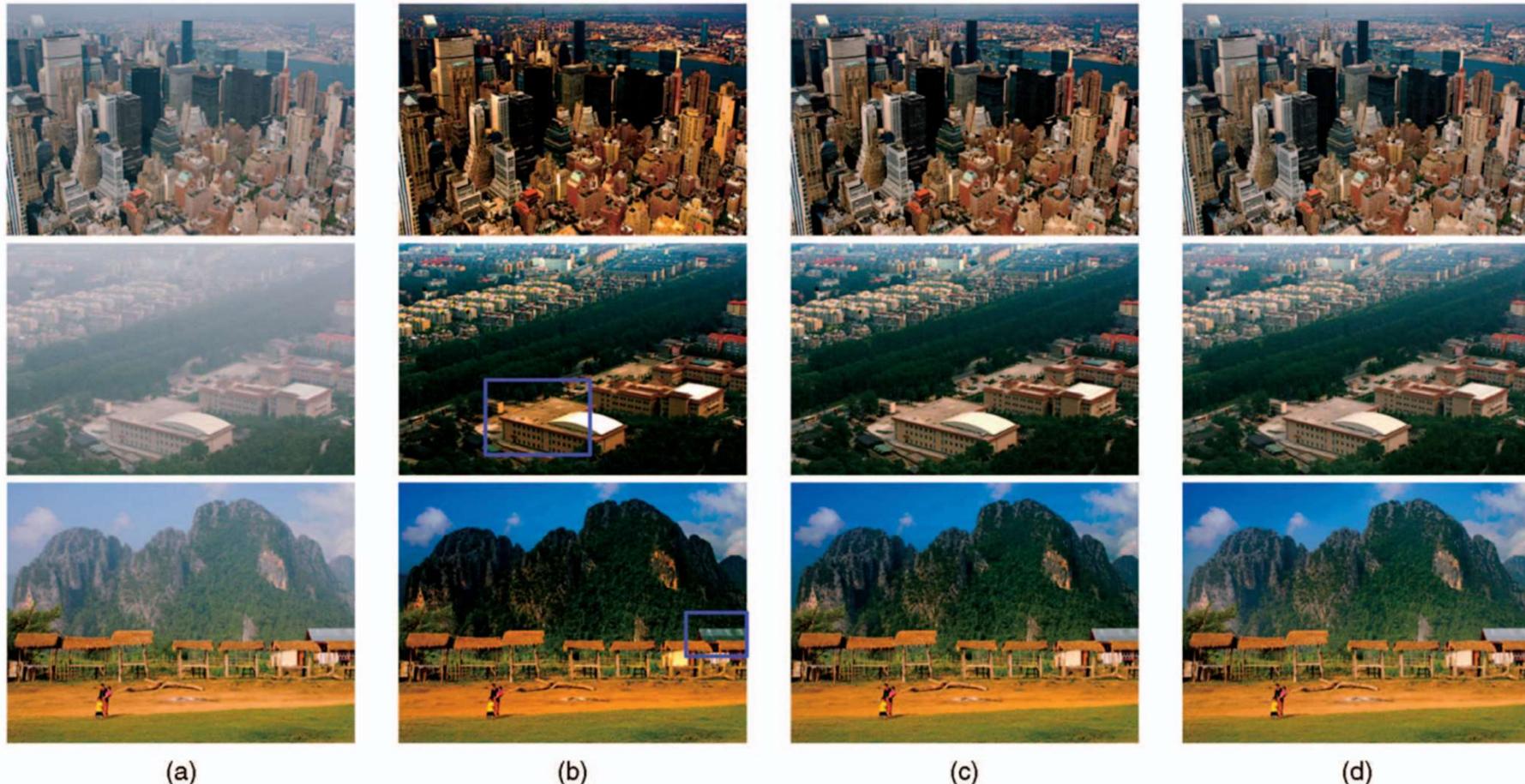


Fig. 11. Recovering images using different patch sizes (after soft matting). (a) Input hazy images. (b) Using  $3 \times 3$  patches. (c) Using  $15 \times 15$  patches. (d) Using  $30 \times 30$  patches.

# Task 4: Image Stitching



# 代表性论文

Google search results for "Automatic Panoramic Image Stitching using Invariant". The top result is a Scholarly article from Springer, titled "Automatic panoramic image stitching using invariant ...". Below it is a snippet from the Springer website.

Scholarly articles for Automatic Panoramic Image Stitching using Invariant  
Automatic panoramic image stitching using invariant ... - Brown - Cited by 3648

Springer  
<https://link.springer.com/article/10.1007/s11263-006-0011-2>

Automatic Panoramic Image Stitching using Invariant ...  
by M Brown · 2007 · Cited by 3648 — In this work, we formulate stitching as a multi-image matching problem, and use invariant local features to find matches between all of the ...

## Abstract

*This paper concerns the problem of fully automated panoramic image stitching. Though the 1D problem (single axis of rotation) is well studied, 2D or multi-row stitching is more difficult. Previous approaches have used human input or restrictions on the image sequence in order to establish matching images. In this work, we formulate stitching as a multi-image matching problem, and use invariant local features to find matches between all of the images. Because of this our method is insensitive to the ordering, orientation, scale and illumination of the input images. It is also insen-*

Home > International Journal of Computer Vision > Article

# Automatic Panoramic Image Stitching using Invariant Features

Published: 19 December 2006

Volume 74, pages 59–73, (2007) [Cite this article](#)

*sitive to noise images that are not part of a panorama, and can recognise multiple panoramas in an unordered image dataset. In addition to providing more detail, this paper extends our previous work in the area [BL03] by introducing gain compensation and automatic straightening steps.*

# 数学模型 — Camera Space Registration

Assuming that the camera rotates about its optical centre, the group of transformations the images may undergo is a special group of homographies. We parameterise each camera by a rotation vector  $\theta = [\theta_1, \theta_2, \theta_3]$  and focal length  $f$ . This gives pairwise homographies  $\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j$  where

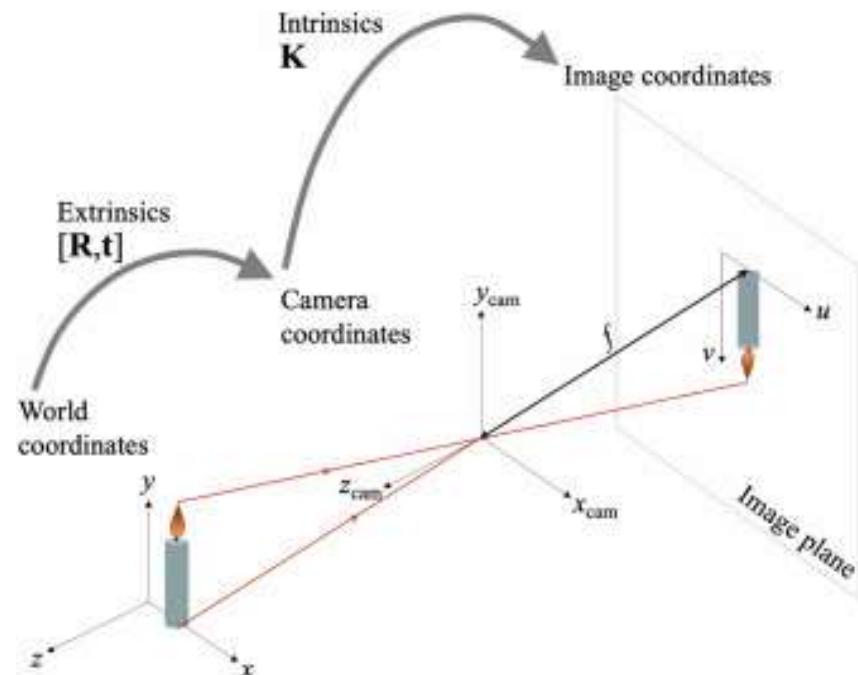
$$\mathbf{H}_{ij} = \mathbf{K}_i \mathbf{R}_i \mathbf{R}_j^T \mathbf{K}_j^{-1} \quad (1)$$

and  $\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j$  are the homogeneous image positions ( $\tilde{\mathbf{u}}_i = s_i[\mathbf{u}_i, 1]$ , where  $\mathbf{u}_i$  is the 2-dimensional image position). The 4 parameter camera model is defined by

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

and (using the exponential representation for rotations)

$$\mathbf{R}_i = e^{[\theta_i]_\times}, \quad [\theta_i]_\times = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}. \quad (3)$$



# Image Matching with Features

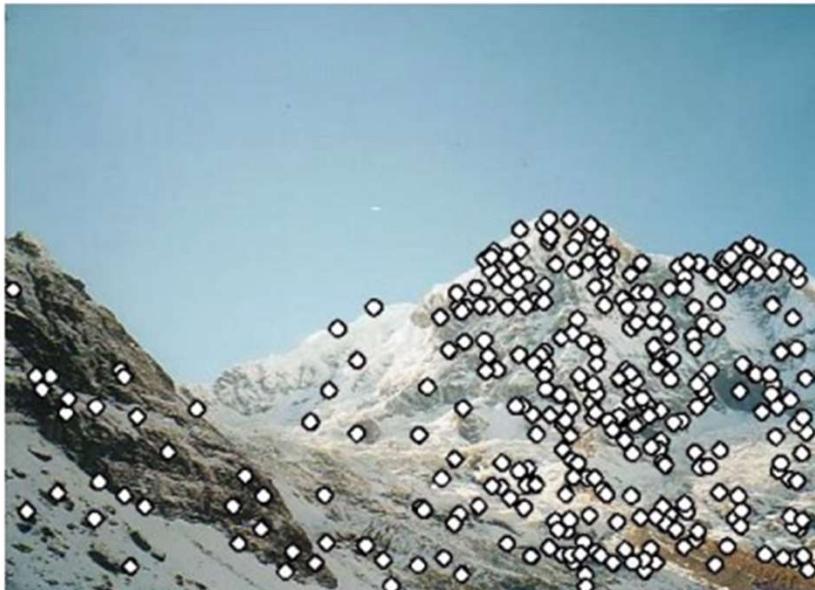


(a) Image 1



(b) Image 2

# Image Feature Points



(c) SIFT matches 1



(d) SIFT matches 2



UBC Computer Science Department

<https://www.cs.ubc.ca/~lowe/papers/ijcv04> [PDF](#)

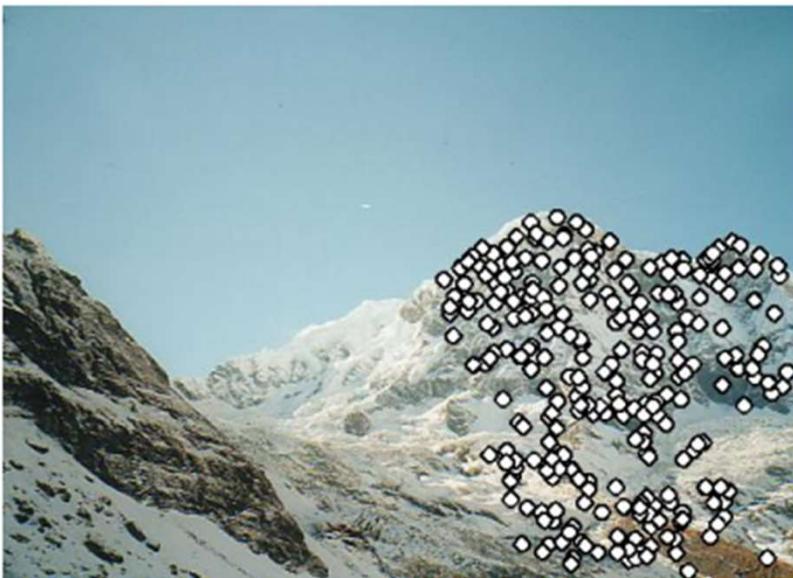
:

[Distinctive Image Features from Scale-Invariant Keypoints](#)

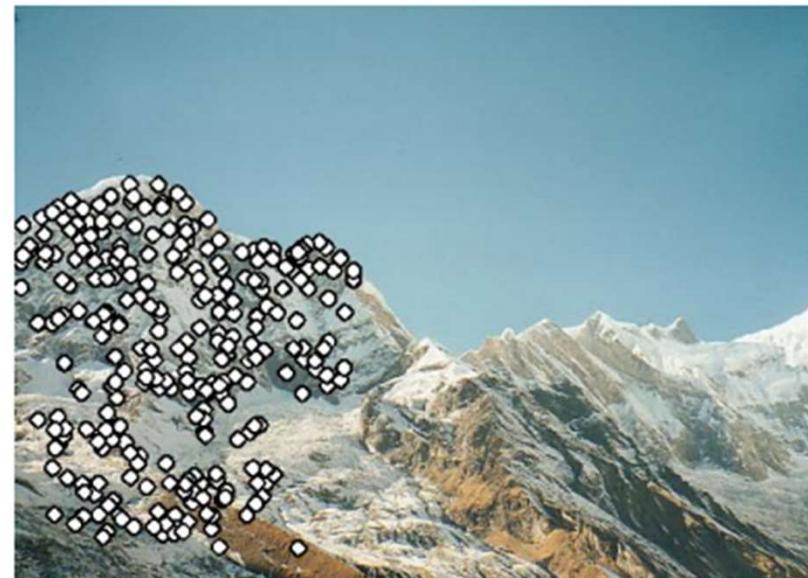
by DG Lowe · 2004 · Cited by 76110 — This [paper](#) presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between ...

28 pages

# Image Feature Points



(e) RANSAC inliers 1



(f) RANSAC inliers 2

# Image Matching — Bundle Adjustment

The objective function we use is a robustified sum squared projection error. That is, each feature is projected into all the images in which it matches, and the sum of squared image distances is minimised with respect to the camera parameters<sup>1</sup>. Given a correspondence  $\mathbf{u}_i^k \leftrightarrow \mathbf{u}_j^l$  ( $\mathbf{u}_i^k$  denotes the position of the  $k$ th feature in image  $i$ ), the residual is

$$\mathbf{r}_{ij}^k = \mathbf{u}_i^k - \mathbf{p}_{ij}^k \quad (14)$$

where  $\mathbf{p}_{ij}^k$  is the projection from image  $j$  to image  $i$  of the point corresponding to  $\mathbf{u}_i^k$

$$\tilde{\mathbf{p}}_{ij}^k = \mathbf{K}_i \mathbf{R}_i \mathbf{R}_j^T \mathbf{K}_j^{-1} \tilde{\mathbf{u}}_j^l. \quad (15)$$

The error function is the sum over all images of the robustified residual errors

$$e = \sum_{i=1}^n \sum_{j \in \mathcal{I}(i)} \sum_{k \in \mathcal{F}(i,j)} h(\mathbf{r}_{ij}^k) \quad (16)$$

where  $n$  is the number of images,  $\mathcal{I}(i)$  is the set of images matching to image  $i$ ,  $\mathcal{F}(i,j)$  is the set of feature matches between images  $i$  and  $j$ . We use a Huber robust error function [Hub81]

$$h(\mathbf{x}) = \begin{cases} |\mathbf{x}|^2, & \text{if } |\mathbf{x}| < \sigma \\ 2\sigma|\mathbf{x}| - \sigma^2, & \text{if } |\mathbf{x}| \geq \sigma \end{cases}. \quad (17)$$

# Solve Bundle Adjustment

This is a non-linear least squares problem which we solve using the Levenberg-Marquardt algorithm. Each iteration step is of the form

$$\Phi = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C}_p^{-1})^{-1} \mathbf{J}^T \mathbf{r} \quad (18)$$

where  $\Phi$  are all the parameters,  $\mathbf{r}$  the residuals and  $\mathbf{J} = \partial \mathbf{r} / \partial \Phi$ . We encode our prior belief about the parameter changes in the (diagonal) covariance matrix  $\mathbf{C}_p$

# Levenberg–Marquardt algorithm

## The problem [\[edit\]](#)

The primary application of the Levenberg–Marquardt algorithm is in the least-squares curve fitting problem: given a set of  $m$  empirical pairs  $(x_i, y_i)$  of independent and dependent variables, find the parameters  $\beta$  of the model curve  $f(x, \beta)$  so that the sum of the squares of the deviations  $S(\beta)$  is minimized:

$$\hat{\beta} \in \operatorname{argmin}_{\beta} S(\beta) \equiv \operatorname{argmin}_{\beta} \sum_{i=1}^m [y_i - f(x_i, \beta)]^2, \text{ which is assumed to be non-empty.}$$

In each iteration step, the parameter vector  $\beta$  is replaced by a new estimate  $\beta + \delta$ . To determine  $\delta$ , the function  $f(x_i, \beta + \delta)$  is approximated by its [linearization](#):

$$f(x_i, \beta + \delta) \approx f(x_i, \beta) + \mathbf{J}_i \delta,$$

where

$$\mathbf{J}_i = \frac{\partial f(x_i, \beta)}{\partial \beta}$$

# Levenberg–Marquardt algorithm

The sum  $S(\boldsymbol{\beta})$  of square deviations has its minimum at a zero gradient with respect to  $\boldsymbol{\beta}$ . The above first-order approximation of  $f(x_i, \boldsymbol{\beta} + \boldsymbol{\delta})$  gives

$$S(\boldsymbol{\beta} + \boldsymbol{\delta}) \approx \sum_{i=1}^m [y_i - f(x_i, \boldsymbol{\beta}) - \mathbf{J}_i \boldsymbol{\delta}]^2,$$

or in vector notation,

$$\begin{aligned} S(\boldsymbol{\beta} + \boldsymbol{\delta}) &\approx \|\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}\|^2 \\ &= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}] \\ &= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T \mathbf{J}\boldsymbol{\delta} - (\mathbf{J}\boldsymbol{\delta})^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] + \boldsymbol{\delta}^T \mathbf{J}^T \mathbf{J}\boldsymbol{\delta} \\ &= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - 2[\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T \mathbf{J}\boldsymbol{\delta} + \boldsymbol{\delta}^T \mathbf{J}^T \mathbf{J}\boldsymbol{\delta}. \end{aligned}$$

Taking the derivative of this approximation of  $S(\boldsymbol{\beta} + \boldsymbol{\delta})$  with respect to  $\boldsymbol{\delta}$  and setting the result to zero gives

$$(\mathbf{J}^T \mathbf{J}) \boldsymbol{\delta} = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})],$$

Levenberg's contribution is to replace this equation by a "damped version":

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \boldsymbol{\delta} = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})],$$

# Image Stitching



(a) Half of the images registered



# How to Implement



(g) Images aligned according to a homography

$$\Phi = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C}_p^{-1})^{-1} \mathbf{J}^T \mathbf{r} \quad (18)$$

- Feature Extraction:
  - SIFT
  - Image Convolution
- Image Registration:
  - Bundle Adjustment
  - Matrix Inverse

`numpy.linalg.inv` #

`linalg.inv(a)`

Compute the inverse of a matrix.

# Other Tasks and Easy Implement



This is an image for inpainting



PSNR = 28.20 (28.35, 28.43, 27.83)



This is an image of Lena for inpainting



PSNR = 25.57 (25.86, 25.18, 25.70)

Suppose  $\Psi_p$  is a patch on the fill-front  $\partial\Omega$ , its neighboring patch  $\Psi_{p_j}$  is defined as the patch that is in the known region and with the center  $p_j$  in the neighborhood of pixel  $p$ , i.e.,  $p_j$  belongs to the set

$$N_s(p) = \{p_j : p_j \in N(p) \text{ and } \Psi_{p_j} \subset \bar{\Omega}\}. \quad (1)$$

$N(p)$  is a neighborhood window centered at  $p$ , which is set to be larger than the size of patch  $\Psi_p$ . Suppose  $P$  is a matrix to extract the missing pixels of  $\Psi_p$ , and  $\bar{P}$  extracts the already known pixels of  $\Psi_p$ , then the similarity between  $\Psi_p$  and  $\Psi_{p_j}$  is defined as

$$w_{p,p_j} = \frac{1}{Z(p)} \exp \left( -\frac{d(\bar{P}\Psi_p, \bar{P}\Psi_{p_j})}{\sigma^2} \right) \quad (2)$$

where  $d(\cdot, \cdot)$  measures the mean squared distance,  $Z(p)$  is a normalization constant such that  $\sum_{p_j \in N_s(p)} w_{p,p_j} = 1$ , and  $\sigma$  is set to 5.0 in our implementation.

[Image Inpainting by Patch Propagation Using Patch Sparsity. TIP 2010.](#)

# Other Tasks and Easy Implement



Fig. 1. Football sequence, tracking player number 75. The frames 30, 75, 105, 140, and 150 are shown.

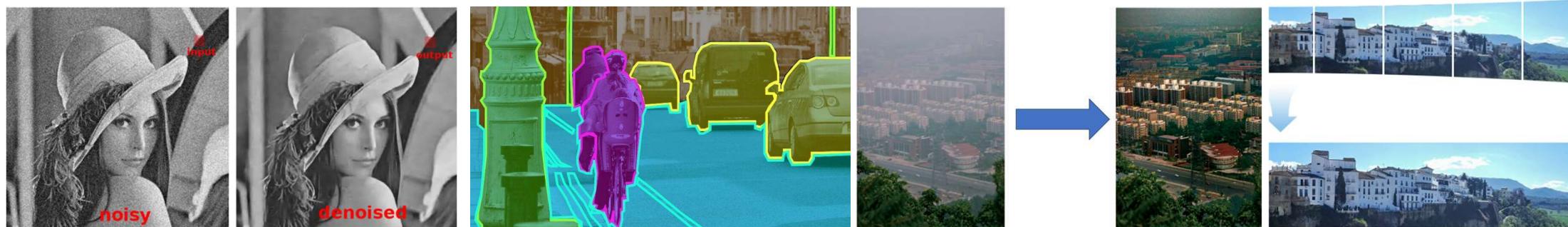
[Kernel-based object tracking. TPAMI 2003.](#)

# Course Summary

阅读、理解、实现paper是一件**容易的事情**

仅通过三节课即可实现**几乎全部**2012年之前的DIP Papers

前沿的研究进展理解和实现也是类似的过程





中国科学技术大学

University of Science and Technology of China

谢谢观看！