Robust "Sparkle Vision" via Sinkhorn Divergence

for image unscrambling

Nicolas Bolle

The Ohio State University

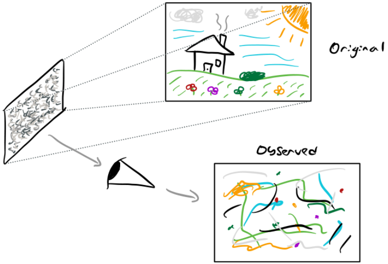
April 18, 2022

Outline

Image unscrambling

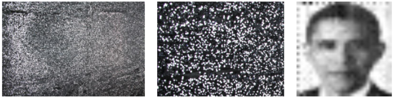
A problem in image recovery:

- A picture is reflected off of a chaotic surface (glitter) and recorded by a camera
- We can present calibration images to understand the distortion
- Can we recover the "unscrambling map" in a robust way?



Sparkle Vision

Solved[1]

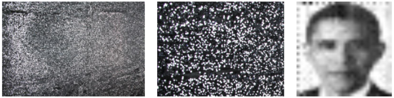


(a) Specular microfacets (b) Closer look (c) Reconstructed lighting

[1][Zhang et al., 2014]

Sparkle Vision

Solved[1]



(a) Specular microfacets (b) Closer look (c) Reconstructed lighting

Algorithm

- $X \in \mathbb{R}^{D \times N}$ the N calibration images with D pixels
- $Y \in \mathbb{R}^{D \times N}$ the scrambled images
- Find $A \in \mathbb{R}^{D \times D}$ the unscrambling map:

$$A := \min_{A} \|AY - X\|_F^2$$

- Standard solution (least squares): $A = XY^{-1}$ (Moore-Penrose)
- Robust to noise in the calibration images

[1] [Zhang et al., 2014]

Extensions

...but there is no follow-up work to address pixel-shift issues:

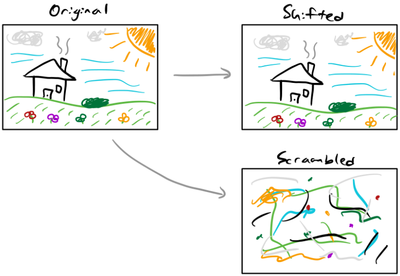
(a) 1 pixel (b) 0.8 pixel (c) 0.4 pixel (d) 0.2 pixel (e) No shift

Extensions

...but there is no follow-up work to address pixel-shift issues:

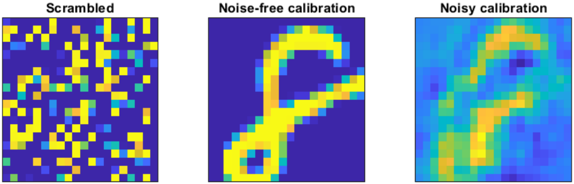
(a) 1 pixel (b) 0.8 pixel (c) 0.4 pixel (d) 0.2 pixel (e) No shift

So I pose a problem: can we make Sparkle Vision robust to misalignments during claibration?



Failure of Sparkle Vision

If the calibration images are shifted randomly by $N(0, I_2)$ pixels,



Decent, but not good

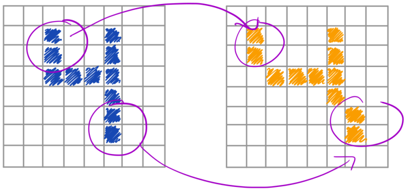
Outline

Enter Wasserstein

- General goal: undo global distortion with robustness to local distortion

Enter Wasserstein

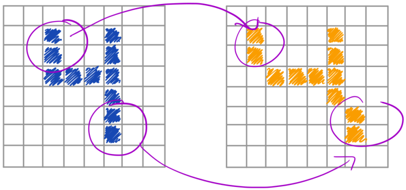
- General goal: undo global distortion with robustness to local distortion
- So, use mass transport:



Pixel-wise far away, close in the *Wasserstein metric*

Enter Wasserstein

- General goal: undo global distortion with robustness to local distortion
- So, use mass transport:

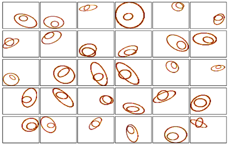


Pixel-wise far away, close in the *Wasserstein metric*

The Wasserstein metric captures local distortions

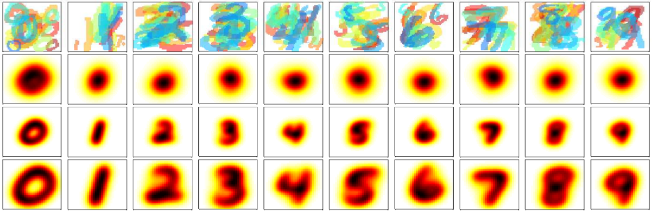
Other Wasserstein uses

Average case ellipses[2]:



$\rightarrow$

...or digits:



[2][Cuturi and Doucet, 2013]

Framework

So, redefine the objective:

$$\min_A \|AY - X\|_F^2 = \min_A \sum_{n=1}^{N} \|Ay_n - x_n\|_2^2$$

$$\downarrow$$

$$\min_A \sum_{n=1}^{N} W(Ay_n - x_n)$$

Optimizing this will be robust to local distortions: find A so that the unscrambled images are *near* the reference images.

Optimizing

Two issues:

- Wasserstein distance is slow to compute
- It is not smooth: optimal transport is too sensitive to small changes

[3] [Cuturi, 2013]
[4] [Luise et al., 2018]

Optimizing

Two issues:

- Wasserstein distance is slow to compute
- It is not smooth: optimal transport is too sensitive to small changes

Solution: Sinkhorn divergence[3]

$$W(a, b) \sim \min_T \langle T, C \rangle$$
$$S(a, b) \sim \min_T \langle T, C \rangle + [\text{entropy of } T]$$

[3] [Cuturi, 2013]

[4] [Luise et al., 2018]

Optimizing

Two issues:

- Wasserstein distance is slow to compute
- It is not smooth: optimal transport is too sensitive to small changes

Solution: Sinkhorn divergence[3]

$$W(a,b) \sim \min_T \langle T, C \rangle$$
$$S(a,b) \sim \min_T \langle T, C \rangle + [\text{entropy of } T]$$

There is a **fast** iteration to compute S, and it is **differentiable**[4].

[3] [Cuturi, 2013]
[4] [Luise et al., 2018]

Robust Sparkle Vision

- Objective:

$$\min_{A} \sum_{n=1}^{N} S(Ay_n - x_n)$$

  where $(x_n), (y_n)$ are the unscrambled/scrambled calibration images.
- Optimize by gradient descent
- Use A obtained to unscramble images

Robust Sparkle Vision

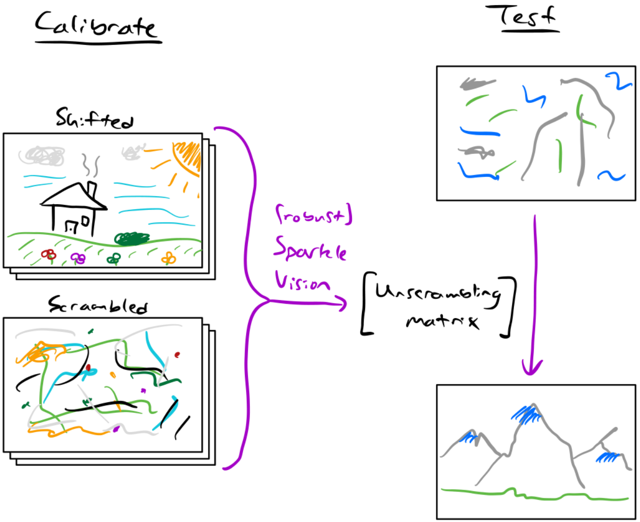
- Objective:

$$\min_A \sum_{n=1}^{N} S(Ay_n - x_n)$$

  where $(x_n), (y_n)$ are the unscrambled/scrambled calibration images.
- Optimize by gradient descent
- Use A obtained to unscramble images
- Details:
  - Sinkhorn parameters: λ, iterations
  - Adding noise to avoid division by zero in Sinkhorn
  - Normalizing A between descent steps
  - ℓ^1 regularization
  - Stochastic gradients, gradient step size

Outline

Overview


Calibrate
Test
Shifted
Scrambled
[robust]
Sparkle
Vision
[Unscrambling
matrix]

Overview

Problem:

- X: ~ 1000 images for calibration
- Y: scrambled versions of the pictures
- There is some A with $AY = X$
- Images X are shifted randomly by $N(0, I_2)$ pixels *after scrambling*

Overview

Problem:

- X: ~ 1000 images for calibration
- Y: scrambled versions of the pictures
- There is some A with $AY = X$
- Images X are shifted randomly by $N(0, I_2)$ pixels *after scrambling*
- Recover A from X, Y, and use it to recover scrambled test images

Overview

Problem:

- X: ~ 1000 images for calibration
- Y: scrambled versions of the pictures
- There is some A with $AY = X$
- Images X are shifted randomly by $N(0, I_2)$ pixels *after scrambling*
- Recover A from X, Y, and use it to recover scrambled test images

Solution:

- Run Sparkle Vision and robust Sparkle Vision on X, Y to recover A
  - 100 iterations of single $k = 100$ stochastic gradients
  - $\lambda = 10$ for Sinkhorn, cost matrix entries capped at a value of 10
  - Images normalized to total value of 1, 0.1 units of noise added to avoid divison by zero in Sinkhorn
  - Step size $\eta = 0.1$, ℓ^1 regularization $\rho = 0.001$ for A

Overview

Problem:

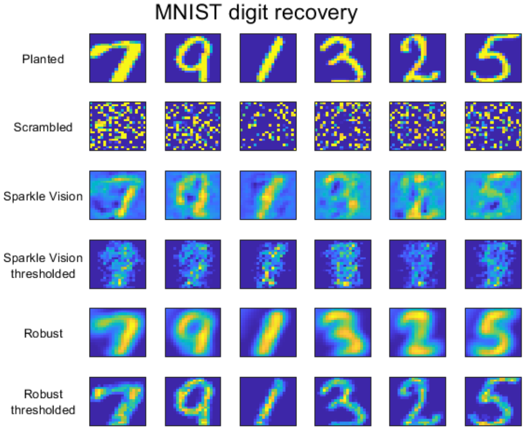
- X: ~ 1000 images for calibration
- Y: scrambled versions of the pictures
- There is some A with $AY = X$
- Images X are shifted randomly by $N(0, I_2)$ pixels *after scrambling*
- Recover A from X, Y, and use it to recover scrambled test images

Solution:

- Run Sparkle Vision and robust Sparkle Vision on X, Y to recover A
  - 100 iterations of single $k = 100$ stochastic gradients
  - $\lambda = 10$ for Sinkhorn, cost matrix entries capped at a value of 10
  - Images normalized to total value of 1, 0.1 units of noise added to avoid divison by zero in Sinkhorn
  - Step size $\eta = 0.1$, ℓ^1 regularization $\rho = 0.001$ for A
- Threshold the recovered A to keep its largest entries
- Use the recovered A to unscramble test images

MNIST

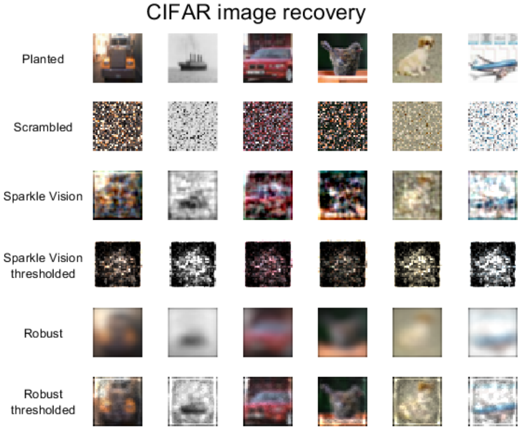
20×20 grayscale images, 1000 calibration images



Sparkle Vision $\ll 1$ second, robust Sparkle Vision 6 seconds

CIFAR-10

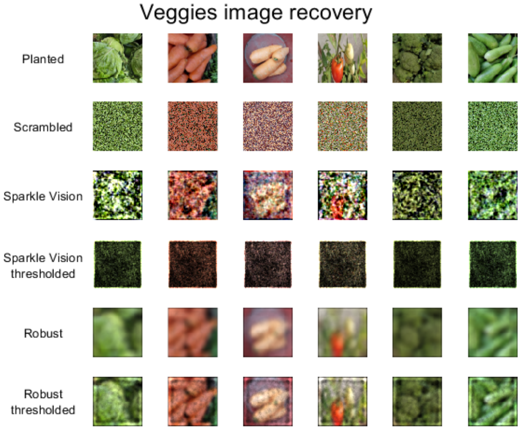
32×32 color images, 1000 calibration images



Sparkle Vision < 1 second, robust Sparkle Vision 40 seconds

Vegetables

56 $\times$ 56 color images, 2000 calibration images



Sparkle Vision 12 seconds, robust Sparkle Vision 6 minutes

Outline

Closing thoughts

- Robust Sparkle Vision is a natural extension of Sparkle Vision
- It outperforms Sparkle Vision by wide margins with good data
- Runtimes are long, but not completely insane
- Could put more thought into de-blurring the robust Sparkle Vision recovered image: de-convolution and other image processing techniques
- The vegetable example with 112×112 images runs into numerical issues: implement stabilized Sinkhorn algorithms
- Consider Sinkhorn for detecting shifts in the scrambled image
- Publication generator: add Sinkhorn to existing algorithms for robustness

`https://github.com/nicolas-bolle/robust_sparkle_vision`

References I

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation distances.

Cuturi, M. and Doucet, A. (2013).
Fast computation of wasserstein barycenters.

Luise, G., Rudi, A., Pontil, M., and Ciliberto, C. (2018).
Differential properties of sinkhorn approximation for learning with wasserstein distance.

Sinkhorn, R. (1967).
Diagonal equivalence to matrices with prescribed row and column sums.
The American Mathematical Monthly, 74(4):402–405.

Zhang, Z., Isola, P., and Adelson, E. H. (2014).
Sparkle vision: Seeing the world through random specular microfacets.

Appendix: Sinkhorn basics

Let $a, b \in \mathbb{R}^n_{>0}$ with $\|a\|_1 = \|b\|_1$, $C \in \mathbb{R}^{n\times n}_{\geq 0}$, $\lambda > 0$. The Sinkhorn divergence is

$$
\begin{aligned}
S(a,b) &= \langle T^*, C\rangle \\
T^* &= \operatorname{argmin}_{T\in\mathbb{R}^{n\times n}} \langle T, C\rangle + \frac{1}{\lambda}\sum_{i,j=1}^{n} T_{ij}\log T_{ij} \\
&\text{subject to} \quad \sum_{j=1}^{n} T_{ij} = a \quad \sum_{i=1}^{n} T_{ij} = b
\end{aligned}
$$

Standard convex duality theory tell us that we can write

$$T^* = \operatorname{diag}(u) e^{-\lambda C} \operatorname{diag}(v)$$

(element-wise exponential) for some $u, v \in \mathbb{R}^n_{>0}$ relating to dual variables. It's known[5] that alternate row/column scaling of $e^{-\lambda C}$ to match a/b marginals converges to a unique matrix, which must be T^*.

[5] [Sinkhorn, 1967]

Appendix: Sinkhorn gradients

By standard theory, dual variables give subgradients so

$$\widetilde{S}(a, b) = \min_{T \in \mathbb{R}^{n \times n}} \langle T, C \rangle + \frac{1}{\lambda} \sum_{i,j=1}^{n} T_{ij} \log T_{ij}$$

$$\text{subject to} \quad \sum_{j=1}^{n} T_{ij} = a \quad \sum_{i=1}^{n} T_{ij} = b$$

is easy to differentiate. $\frac{\partial \widetilde{S}}{\partial a} = \frac{\log(u)}{\lambda}$ where

$$T^* = \text{diag}(u) e^{-\lambda C} \text{diag}(v)$$

This is the *regularized* gradient, which I used for stability.
You can also differentiate the $S(a, b) = \langle T^*, C \rangle$ formulation[6].

[6] [Luise et al., 2018]