# Project Report

Peiqi Wang

Thursday 4th June, 2020

# Contents

# 1 Abstract

We aim to improve upon low level image proecssing pipeline for the coded two-bucket camera. Specifically, we aim to jointly upsample, demultiplex, and denoise two-bucket images to produce full resolution images under different illumination conditions for downstream reconstruction tasks.

# 2 The Two-Bucket Camera

## 2.1 Notations

The coded two-bucket (C2B) camera is a pixel-wise coded exposure camera that outputs two images in a single exposure.[1] Each pixel in the sensor has two photo-collecting site, i.e. the two *buckets*, as well as a 1-bit writable memory controlling which bucket is actively collecting light. It was shown previously that C2B camera is capable of one-shot 3D reconstruction by solving a simpler image demosaicing and illumination demultiplexing problem instead of a difficult 3D reconstruction problem. We summarize the following notations relevant to discussion

| | Notation | Meaning |
|---|---|---|
| | F | number of video frames |
| | P | number of pixels |
| | S | number of sub-frames |
| | h,w | dimension of image |
| $P \times F \times S$ | $\mathbf{C}$ | code tensor |
| $P \times 1 \times S$ | $\widetilde{\mathbf{C}}$ | 1-frame code tensor that spatially multiplex $F$ frame tensor $\mathbf{C}$ |
| $F \times S$ | $\mathbf{C}^p$ | activity of bucket 0 pixel $p$ cross all frames and sub-frames |
| $F \times S$ | $\overline{\mathbf{C}}^p$ | activity of bucket 1 pixel $p$ cross all frames and sub-frames |
| $1 \times S$ | $\mathbf{c}_f^p$ | active bucket of pixel $p$ in the sub-frames of frame $f$ |
| $1 \times L$ | $\mathbf{l}_s$ | scene's illumination condition in sub-frame $s$ of every frame |
| $P \times S$ | $\mathbf{C}_f = [\mathbf{c}_1^p; \cdots; \mathbf{c}_F^p]$ | activity of bucket activity of all pixels across all sub-frames of $f$ |
| $S \times L$ | $\mathbf{L} = [\mathbf{l}_1; \cdots; \mathbf{l}_S]$ | time-varying illumination condition (same for all frames) |
| $2F \times S$ | $\mathbf{W}$ | optimal bucket multiplexing matrix |
| | $\mathbf{t}^p$ | transport vector at pixel $p$ |
| $F \times 1$ | $\mathbf{i}^p, \hat{\mathbf{i}}^p$ | measured two-bucket intensity at pixel $p$ in $F$ frames |
| $F \times 1$ | $r, \hat{r}$ | illumination ratios at pixel $p$ in $F$ frames |
| $F \times P$ | $\mathbf{I} = [\mathbf{i}^1 \cdots \mathbf{i}^P], \hat{\mathbf{I}}$ | two-bucket image sequence in $F$ frames |
| $P \times 2F$ | $\boldsymbol{I} = [\mathbf{I}^T \ \hat{\mathbf{I}}^T]$ | two-bucket image sequence |
| $P \times 2$ | $\mathbf{Y}$ | two-bucket illumination mosaic |
| $S \times 1$ | $\boldsymbol{i}^p$ | pixel intensity under $S$ illuminations at pixel $p$ |
| $P \times S$ | $\mathbf{X} = [\boldsymbol{i}^1 \cdots \boldsymbol{i}^P]^T$ | pixel intensity under $S$ illuminations |
| $2P \times 1$ | $\mathbf{y} = vec(\mathbf{Y})$ | vectorized two-bucket illumination mosaic |
| $SP \times 1$ | $\mathbf{x} = vec(\mathbf{X})$ | vectorized pixel intensity under $S$ illumiantions |
| $2P \times 2PF$ | $\mathbf{B}$ | subsampling linear map |
| $2P \times SP$ | $\mathbf{A} = \mathbf{B}(\mathbf{W} \otimes \mathbf{I}_P)$ | illumination multiplexing and subsampling linear map |

Illumination ratios are albedo *quasi-invariant*, a property which can be exploited for downstream processing

$$r = \frac{\mathbf{i}^p[f]}{\mathbf{i}^p[f] + \hat{\mathbf{i}}^p[f]} \qquad \hat{r} = \frac{\hat{\mathbf{i}}^p[f]}{\mathbf{i}^p[f] + \hat{\mathbf{i}}^p[f]}$$

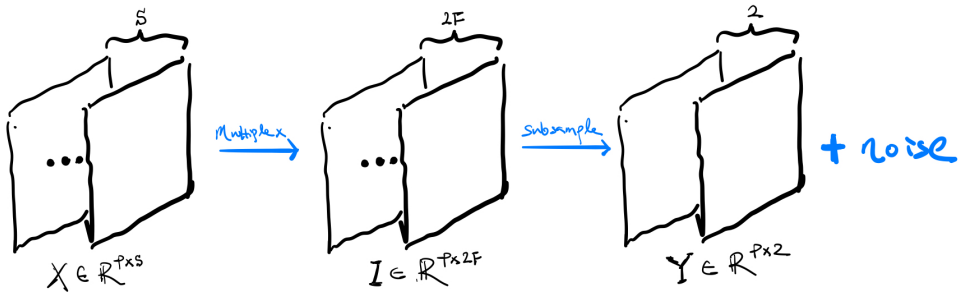


Figure 1: Image Formation Sketch

## 2.2 The Forward Model

**Subsampling Mapping** Let $\mathbf{S} \in \{1, 2, \cdots, F\}^P$ be a vector specifying how the one-frame code tensor $\widetilde{\mathbf{C}}$ is constructed, i.e. $\tilde{\mathbf{c}}_1^p := \mathbf{c}_{\mathbf{S}_p}^p$, for all pixels $p$. We can view $\mathbf{S}$ as a mask to construct a **S**ubsampling linear map that maps vectorized two-bucket image sequences $\boldsymbol{I}$ to the vectorized illumination mosaics $\mathbf{Y}$. In particular, let $\mathbf{B}' \in \mathbb{R}^{P \times PF}$ and $\mathbf{B} \in \mathbb{R}^{2P \times 2PF}$ be defined as follows

$$\mathbf{B}' = \begin{bmatrix} \mathbf{diag}\mathbb{1}_{\{1\}}(\mathbf{S}) & \mathbf{diag}\mathbb{1}_{\{2\}}(\mathbf{S}) & \cdots & \mathbf{diag}\mathbb{1}_{\{F\}}(\mathbf{S}) \end{bmatrix}$$
$$\mathbf{B} = \mathbf{I}_2 \otimes \mathbf{B}' = \begin{bmatrix} \mathbf{B}' & \mathbf{0} \\ \mathbf{0} & \mathbf{B}' \end{bmatrix}$$

Then we have the following relation between $\boldsymbol{I}$ and $\mathbf{Y}$,

$$vec\,(\mathbf{Y}) = \mathbf{B} vec\,(\boldsymbol{I}) \tag{1}$$

In essence, $\mathbf{B}$ is a linear operator that trade spatial resolution (measures $\frac{1}{F}$ of the pixels for each frame) for temporal resolution (one two-bucket shot instead of acquiring $F$ frames). We can think of a parallel in RGB color imaging, where bayer mosaic trade spatial resolution for spectral resolution. As an example when $F = 3$ and $P = 4$, the corresponding $\mathbf{S}$, when reshaped to dimension of a $2 \times 2$ image, and single image subsampling linear map $\mathbf{B}'$ are given by

$$\mathbf{S} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \qquad \mathbf{B}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

**Image Formation** Per-pixel image formation model is

$$\begin{bmatrix} \mathbf{i}^p \\ \hat{\mathbf{i}}^p \end{bmatrix} = \begin{bmatrix} \mathbf{C}^p \\ \overline{\mathbf{C}}^p \end{bmatrix} \begin{bmatrix} \mathbf{l}_1 \mathbf{t}^p \\ \vdots \\ \mathbf{l}_S \mathbf{t}^p \end{bmatrix} = \begin{bmatrix} \mathbf{C}^p \\ \overline{\mathbf{C}}^p \end{bmatrix} \boldsymbol{i}^p$$

If bucket activity is same for all pixels and we use the optimal bucket multiplexing matrix $\mathbf{W}$, we can write the above linear relationship compactly for all pixels as

$$\boldsymbol{I} = \mathbf{X}\mathbf{W}^T \tag{2}$$

As shown in Figure 1, illumination multiplexing and spatial subsampling can be combined to obtain a single linear function that maps images under $S$ different illuminations $\mathbf{X}$ to the two-bucket images $\mathbf{Y}$. From (1) and (2), there exists a linear relationship between $\mathbf{x}$ and $\mathbf{y}$,

$$\mathbf{y} = \mathbf{B} vec\,(\boldsymbol{I}) = \mathbf{B} vec\left(\mathbf{X}\mathbf{W}^T\right) = \mathbf{B}(\mathbf{W} \otimes \mathbf{I}_P) vec\,(\mathbf{X}) = \mathbf{A}\mathbf{x} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{2P \times SP}$ be a linear map that illumination multiplexes and subsamples $\mathbf{X}$,

$$\mathbf{A} = \mathbf{B}(\mathbf{W} \otimes \mathbf{I}_P)$$

and $\mathbf{I}_P \in \mathbb{R}^{P \times P}$ is identity.

### 2.3 The Inverse Problem

The reconstruction pipeline is as follows

1. Use $\widetilde{\mathbf{C}}$ for bucket activities and capture the two-bucket image $\mathbf{Y}$
2. upsample the images to full resolution images $\boldsymbol{I}$
3. demultiplex $\boldsymbol{I}$ to obtain $S$ full resolution images $\mathbf{X}$ as a least squares solution to a (2)
4. use $\mathbf{X}$ to solve for disparity and albedo

Step 2 and 3 are critical to downstream reconstuctions. When $S = 3, S = 4$ and $\mathbf{S}$ being analogous to bayer mask, we can upsample the images using standard demosaicing algorithms. However, it is not immediately obvious to extend demosaicing methods to support arbitrary $\mathbf{S}$, or more specifically, for scenarios where the spatial subsampling scheme is not bayer and when number of frames is not 3. One approach which we consider later involves the following steps

1. Use $\widetilde{\mathbf{C}}$ for bucket activities and capture the two-bucket image $\mathbf{Y}$
2. Recover full resolution images $\mathbf{X}$ under $S$ illuminations from $\mathbf{Y}$ by solving a linear inverse problem
3. use $\mathbf{X}$ to solve for disparity and albedo

Jointly upsample and demultiplex enforces a prior knowledge of image formation. Instead of treating upsampling (recover $2F$ images $\boldsymbol{I}$ from 2 images $\mathbf{Y}$) and demultiplexing (recover $S$ images $\mathbf{X}$ from $2F$ images $\boldsymbol{I}$) as distinct steps, we aim to recover $\mathbf{X}$ directy from $\mathbf{Y}$, in a single step, by solving a linear inverse problem.

## 3 Methods

### 3.1 Optimization Problem

Assuming an isotropic Gaussian noise model ($\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ where $\mathbf{e} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$), and prior $p_{\mathsf{x}}(x) \propto \exp\left\{-(\lambda/\sigma^2)\rho(\mathbf{x})\right\}$ where $\rho : \mathbb{R}^{SP} \to \mathbb{R}$ is some regularizer for $\mathbf{x}$. Given noisy measurement $\mathbf{y}$, *max a posterior* estimate $\hat{\mathbf{x}}$ can be obtained by solving the following

$$\text{minimize } \frac{1}{2}\left\|\mathbf{A}\mathbf{x} - \mathbf{y}\right\|_2^2 + \lambda\rho(\mathbf{x}) \tag{4}$$

### 3.2 Solving Inverse Problem using RED

To integrate model-based optiimzation and recent advances in learning-based method, plug-and-play (PnP) prior enforces implicit prior by using state of art denoiser to solve the subproblem within iterative methods like ADMM and PGM [2]. RED derived an explicit form for the prior $\rho$ that give rise to PnP method [3]. Here we use RED to solve (4).

We first note that the illumination ratios are albedo quasi-invariant, and therefore smooth within object boundaries. Therefore, total variation regularization on illuination ratio images could be particularly effective. To avoid extra notations, we use $\mathbf{x}, \mathbf{y}$ as the corresponding illumination ratios that we want to reconstruct. Additionally, we adapt algorithm in [3] for imposing algorithm induced priors with state-of-the-art denoisers. In summary, we want to optimize the following constrained problem with a set of affine constraints,

$$\begin{aligned} \text{minimize} \quad & \left\|\mathbf{A}\mathbf{x}_1 - \mathbf{y}\right\|_2^2 + \frac{\lambda_2}{2}\mathbf{x}_2^T(\mathbf{x}_2 - \mathcal{D}(\mathbf{x}_2)) + \lambda_3\left\|\mathbf{x}_3\right\|_1 \\ \text{subject to} \quad & \mathbf{x}_1 - \mathbf{x}_2 = 0 \\ & \mathbf{G}\mathbf{x}_1 - \mathbf{x}_3 = 0 \end{aligned}$$

where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{SP}$, $\mathbf{x}_3 \in \mathbb{R}^{2SP}$. $\lambda_2, \lambda_3 > 0$ are weights to the regularizers. $\mathbf{G} \in \mathbb{R}^{2SP \times SP}$ is the discrete image gradient for $S$ images

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_S \otimes \mathbf{G}_x \\ \mathbf{I}_S \otimes \mathbf{G}_y \end{bmatrix}$$

where $\mathbf{G}_x, \mathbf{G}_y \in \mathbb{R}^{P \times P}$ are the discrete image gradients for a single image computed using forward difference. We can gather constraints into a single linear system

$$\mathbf{H}\mathbf{x} = 0 \qquad \text{where} \qquad \mathbf{H} = \begin{bmatrix} \mathbf{I}_{SP} & -\mathbf{I}_{SP} & 0 \\ \mathbf{G} & 0 & -\mathbf{I}_{SP} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}$$

and arrive at an equivalent optimization problem

$$\begin{aligned} \text{minimize} \quad & f_1(\mathbf{x}_1) + \lambda_2 f_2(\mathbf{x}_2) + \lambda_3 f_3(\mathbf{x}_3) \\ \text{subject to} \quad & (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \in \mathcal{C} \end{aligned} \tag{5}$$

where $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{4SP} \mid \mathbf{H}\mathbf{x} = 0\}$ and

$$\begin{aligned} f_1(\mathbf{x}_1) &= \|\mathbf{A}\mathbf{x}_1 - \mathbf{y}\|_2^2 \\ f_2(\mathbf{x}_2) &= \frac{1}{2}\mathbf{x}_2^T(\mathbf{x}_2 - \mathcal{D}(\mathbf{x}_2)) \\ f_3(\mathbf{x}_3) &= \|\mathbf{x}_3\|_1 \end{aligned}$$

### 3.3 Optimization

As shown below, the scaled form ADMM for solving (5) is given by

$$\begin{aligned} \mathbf{x}_1^{k+1} &= \mathsf{prox}_{(1/\rho)f_1}(\mathbf{z}_1^k - \mathbf{u}_1^k) = (I + \frac{2}{\rho}A^TA)^{-1}(\mathbf{z}_1^k - \mathbf{u}_1^k + \frac{2}{\rho}A^Ty) \\ \mathbf{x}_2^{k+1} &= \mathsf{prox}_{(\lambda_2/\rho)f_2}(\mathbf{z}_2^k - \mathbf{u}_2^k) = \frac{1}{\lambda_2 + \rho}(\lambda_2\mathcal{D}(\mathbf{x}_2^k) + \rho(\mathbf{z}_2^k - \mathbf{u}_2^k)) \\ \mathbf{x}_3^{k+1} &= \mathsf{prox}_{(\lambda_3/\rho)f_3}(\mathbf{z}_3^k - \mathbf{u}_3^k) = \mathcal{S}_{\lambda_3/\rho}(\mathbf{z}_3^k - \mathbf{u}_3^k) \\ \mathbf{z}^{k+1} &= \mathsf{prox}_{(1/\rho)I_\mathcal{C}}(\mathbf{x}^{k+1} + \mathbf{u}^k) = (I - \mathbf{H}^\dagger\mathbf{H})(\mathbf{x}^{k+1} + \mathbf{u}^k) \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1} \end{aligned}$$

## 4 Details

### 4.1 Structured Light Stereo

Some relevant reiews are [4],[5] and slides. A *structured light stereometric system* is similar to a passive stereo system where one of the camera is replaced by a projector. A light source projects light a vertical plane of light that creates a narrow stripe on the scene. The intersection of an illumination plane of known spatial position (corresponds to a projector column) and a line of sight (corresponds to a camera pixel) determines a point. For dense reconstruction of the scene, many images must be taken. To speed up the scanning process, spatially modulated light projector has been suggested, in which multiple illumination planes or rays can be projected simultaneously as part of a single illumination pattern. Spatial-temporal modulation of illumination, i.e. sequentially projecting several patterns, can be used for reliable identification of light planes. To enable acquisition of dynamic scenes, the number of projected patterns used should be as small as possible. Intuitively, the projected pattern impose illusion of texture on the object, increasing the number of correspondences, which enables reconstruction. Structured light stereo is equivalent to (1) solving the correspondence problem and then (2) computing stereo using triangulation.

### 4.1.1 The Correspondence Problem

The correspondence problem can be stated simply

*For each point in the left image, find the corresponding point in the right image*

We first note that search space for the corresponding point can be restricted to pixels lying on the epipolar line. Epipolar plane is plane formed from points $p, o_1, o_2$ and epipolar line is intersection of epipolar plane with the image plane. For structured light stereo systems, the art of designing robust, fast, reliable coding

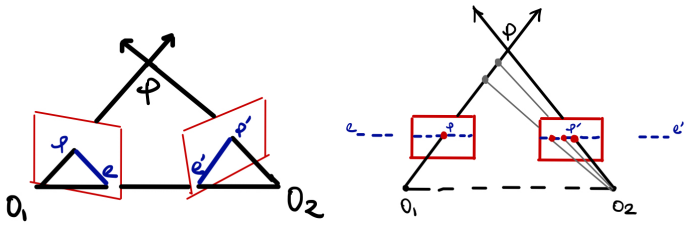


Figure 2: non parallel (left) and parallel (right) camera setup. $pe$ and $p'e'$ are the epipolar lines

schemes serves to solve the correspondence problem.

### 4.1.2 Triangulation

Given correspondence between projector and camera images, triangulation refers to the process of computing the distance of object relative to camera. Since a 3D point can be obtained by intersecting a ray (pixel of camera image) with a projector plane (a single code), it is necessary to encode a single axis in projected image to ensure unique reconstruction. *disparity* is the displacement between points of a conjugate pair

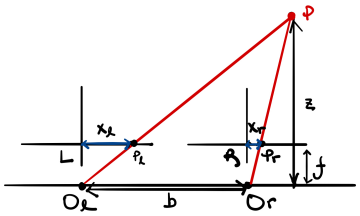


Figure 3: parallel-calibrated binocular stereo

$p_l, p_r$ (points in different images of projection of same point in the scene) when the two images are superimposed. In the context of parallel calibrated cameras, disparity is inversely proportional to depth, when baseline $b$ and focal length $f$ are known

$$\frac{b}{z} = \frac{b + x_l - x_r}{z - f} \quad \text{implies} \quad z = \frac{bf}{x_l - x_r}$$

## 4.2 Structured Light Coding

The goal of structured light coding is to find coding schemes (maps pattern intensity to indices of the projector light planes) that enables robust, reliable algorithms for finding correspondence.

### 4.2.1 Horn & Kiryati

This paper generalizes Gray code to n-ary code in order to reduce the number of patterns that needs to be projected, ($L^K$ instead of $2^K$ code words) [6]. The authors draw inspirations from communication theory, where the projector projects unique temporal codes, received at each image plane through a noisy channel and subsequently decoded. Let there be $K$ patterns and $L$ code words (distinct planes of light), we want

to encode the indices of vertical light planes $x \in [L]$ using some encoding scheme $f : [L] \to \mathbb{R}^K$ such that the nearest neighbor decoding $\hat{x}(y) = \arg\min_{x \in [L]} (f(x) - y)^2$ of a normalized noisy observation $y \in \mathbb{R}^K$ minimizes the probability of depth estimation error. Given the forwarding model,

$$\mathsf{y} = f(\mathsf{x}) + \mathsf{n} \qquad \text{where} \qquad \mathsf{x} \sim \text{Cat}(1/L) \quad \mathsf{n} \sim p_{\mathsf{n}}$$

implying $\mathsf{y}|\mathsf{x} = x \sim p_{\mathsf{n}}(y - f(x))$. We want to minimize the probability of depth estimation error, which roughly proportional to difference between true index $x$ of the plane of light and the estimated index $\hat{x}$,

$$\text{minimize}_f \quad \left[ \mathbb{E}_{\mathsf{x},\mathsf{y}} \left[ (x - \hat{x}(y))^2 \right] = \sum_{x=1}^{L} p_{\mathsf{x}}(x) \int p_{\mathsf{y}|\mathsf{x}}(y|x)(x - \hat{x}(y))^2 dy \propto \sum_{x=1}^{L} \int (x - \hat{x}(y))^2 p_{\mathsf{n}}(y - f(x)) dy \right]$$

This optimization problem is hard. The paper suggest the use of space filling curves as the encoding function and established that Gray code is a special limiting case of the space filling curve. Note here we assume there is no *mutual illumination*, i.e. there is no interval reflection and so the projected codes $f(x)$ is proportional to observation $y$.

### 4.2.2 Phase Shifting

Phase shifting is a particular coding method whereby the projector column coordinates are encoded as the (absolute) phase of a spatial sinusoidal pattern. Note we want to encode column coordinates because we only need to search along the horizontal epipolar lines. Let $N$ be number of columns to be encoded. When the scene is projected with a cosine pattern of period $T$ (measured in number of pixels) and therefore frequency $f = \frac{1}{T}$, an idealized image formation model for any pixel is

$$I = I_0 + A\cos(\Phi) = I_0 + A\cos(\phi) \tag{6}$$

where $I_0$ is the pixel intensity resulting from ambient illumination; $I$ is the pixel intensity measured; $A$ is amplitude (albedo/reflectance) of the signal corresponding to intensity of scene assuming unit intensity for projector patterns; $\Phi = 2\pi f x = 2\pi n + \phi \in [0, 2\pi f N]$ for some number of period $n \in \mathbb{N}$ is the absolute phase, $\phi \in [0, 2\pi]$ is the relative phase. When measured w.r.t. number of pixels, $x$ is the corresponding absolute phase and $\tilde{x}$ the corresponding relative phase, satisfying

$$x = Tn + \tilde{x} \qquad \text{or} \qquad x \equiv \tilde{x} \mod T \tag{7}$$

where $\tilde{x} = \frac{T\phi}{2\pi}$. In (6), $I_0, A, \phi$ are unknown and so $\phi$ cannot be determined. To solve for $\phi$, phase shifting method projects $K$ sinusoidal patterns of same frequency, each shifted by $\varphi_k = \frac{2\pi(k-1)}{K}$ for $k = 1, \cdots, K$. Thereby obtaining a system of $K$ equations in 3 unknowns.

$$I_k = I_0 + A\cos(\phi + \varphi_k) \qquad \text{for} \qquad k = 1, \cdots, K \tag{8}$$

Although $K = 3$ suffices, larger values of $K$ makes determination of relative phase $\phi$ more robust to noise. We determine $\phi$ using least squares

$$\text{minimize}_\phi \quad \left[ \epsilon(\phi, I_0, A) := \sum_{k=1}^{K} \left[ I_k - (I_0 + A\cos(\phi + \varphi_k)) \right]^2 \right] \tag{9}$$

Similar to appendix in [7] and results shown in [8, 7], we can show the following

$$\begin{aligned} 0 = \frac{\partial \epsilon}{\partial \phi} &= 2A \sum_{k=1}^{K} I_i \sin(\phi + \varphi_k) \propto \cos(\phi) \sum_{k=1}^{K} I_k \sin(\varphi_k) + \sin(\phi) \sum_{k=1}^{K} I_k \cos(\varphi_k) \\ \phi &= \tan^{-1} \left[ -\frac{\sum_{k=1}^{K} I_k \sin(\varphi_k)}{\sum_{k=1}^{K} I_k \cos(\varphi_k)} \right] \end{aligned} \tag{10}$$

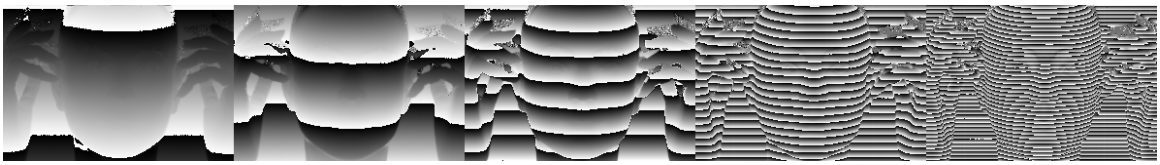

Figure 4: Relative phase $\phi$ for spatial sinusoids with period 1, 2, 5, 17, 31 solved via (10)

**phase unwrapping** aims to recover absolute phase $x$ from relative phase $\tilde{x}$, which is non-trivial unless $T \geq N$. One particular choice of phase unwrapping method relies results in number theory [9]. The idea is to project patterns whose periods $T_1, \cdots, T_F$ are relative co-prime, each shifted by $K$ times such that relative phase $\tilde{x}_1, \cdots, \tilde{x}_F$ can be solved using (9) or (12), from which we can use the Chinese Reminder Theorem to solve a system of congruences

$$\begin{aligned} x &= \tilde{x}_1 \mod T_1 \\ &\vdots \\ x &= \tilde{x}_F \mod T_F \end{aligned} \tag{11}$$

Intuitively, projecting $F$ spatial sinusoids with period $T_1, \cdots, T_F$ with different frequency emulates the projection of a low frequency spatial sinusoid with period $T = T_1 \times \cdots \times T_F$, shown in Figure (5). As specified in [10] and experimented in Figure (6), phase unwrapping is unstable when the relatives phases $\tilde{x}_1, \cdots, \tilde{x}_F$ are noisy. Simple application of medium filtering of the relative phase does not seem to help with phase unwrapping. Wavelet denoising using `wdenoise` seems to help to a limited extend, as shown in Figure (7)

Figure 5: Two sinusoids with $T_1 = 17, T_2 = 31$ emulates a low frequency sinusoid with $T = 527$

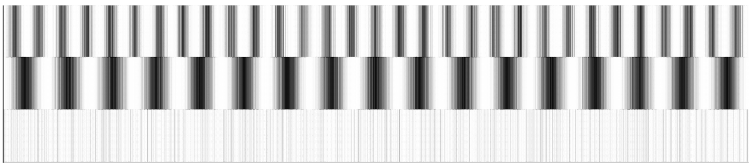

Figure 6: When relative phase $\tilde{x}_1, \tilde{x}_2$ are corrupted with i.i.d. additive Gaussian noise $\mathcal{N}(\mathbf{0}, 0.1 \cdot \mathsf{range}(\tilde{x}) \cdot \mathbf{I})$, phase unwrapping is unable to recover the absolute phase $x$ robustly.

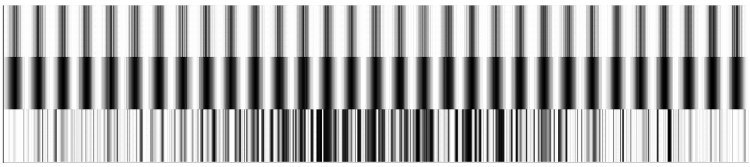

Figure 7: $\tilde{x}_1$ and wavelet denoised $\tilde{x}_1$ ($x_2, \tilde{x}_2$ not shown) helps with phase unwrapping to some extend

#### 4.2.3 Micro Phase Shifting

An alternative formulation as mentioned in [10], we can write (8) as a linear system of equations

$$I_k = I_0 + A\cos(\phi)\cos(\varphi_k) - A\sin(\phi)\sin(\varphi_k) = \begin{bmatrix} 1 & \cos(\varphi_k) & -\sin(\varphi_k) \end{bmatrix} \begin{bmatrix} I_0 \\ A\cos(\phi) \\ A\sin(\phi) \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} I_1 \\ \vdots \\ I_K \end{bmatrix}}_{\mathbf{I}} = \underbrace{\begin{bmatrix} 1 & \cos(\varphi_1) & -\sin(\varphi_1) \\ \vdots & \vdots & \vdots \\ 1 & \cos(\varphi_K) & -\sin(\varphi_K) \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} I_0 \\ A\cos(\phi) \\ A\sin(\phi) \end{bmatrix}}_{\mathbf{u}}$$

and solve for $\mathbf{u}$ using least squares,

$$\text{minimize}_{\mathbf{u}} \quad \|\mathbf{I} - \mathbf{M}\mathbf{u}\|_2^2 \tag{12}$$

The relative phase is then given by

$$\phi = \cos^{-1}\left(\frac{\mathbf{u}_2}{A}\right) \qquad \text{where} \qquad A = \sqrt{\mathbf{u}_2^2 + \mathbf{u}_3^2} \quad \text{and} \quad \mathbf{u} = M^{\dagger}\mathbf{I}$$

Note the optimization problem formulated in (9,12) are not equivalent.

#### 4.2.4 ZNCC Decoding

Let $L$ be number of vertical light planes to be encoded, $f : [L] \to \mathbb{R}^K$ is some encoding scheme that encodes index $x$ to a code $f(x)$, $y$ is per pixel measurement. Similar to nearest neighbor decoder used in [6]

$$\hat{x}(y) = \arg\min_{x \in [L]} \left(f(x) - y\right)^2 \tag{13}$$

[11] defines a decoding function that is optimal in a sense in small noise regimen,

$$\hat{x}(y) = \arg\max_{x \in [L]} \mathsf{ZNCC}\left(y, f(x)\right) \tag{14}$$

Note the quality of decoding depends heavily on the noise level. Additionally, number of shifts is crucial for good reconstruction of absolute phase, see Figure 8

#### 4.2.5 Geometric Perspective & Hamiltonian Code

When trying to find optimal encoding scheme that minimizes bayesian decoding/correspondence error, a surrogate objective is the length of the coding curve. [12] argues that an optimal coding scheme should have a coding curve that is long, non-intersecting, and distance preserving (points far along the curve should also be far in euclidean distance). Sinusoidal codes has intersection with itself, i.e. multiple column indices are coded to the same location in measurement space implying there is $2\pi$ phase ambiguity when trying to compute its inverse decoding function. The distance preserving property is especially important in noisy regime, where non-distance preserving coding scheme would potentially yield large error as the decoding function maps (w.r.t. Euclidean distance) noisy measurement to indices far away from true value. For example, A La Carte code is not distance preserving as shown in Figure 10.

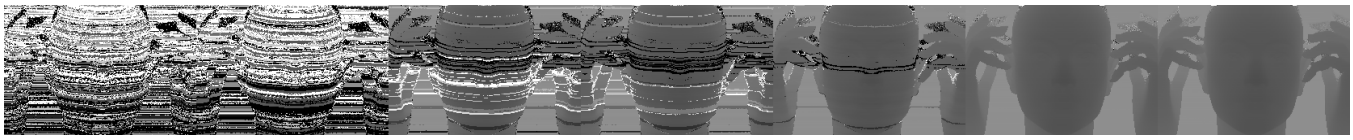

Figure 8: Here we have spatial sinusoids with period 1,2,5,17,31, each shifted 30 times projected onto a static scene. Resulting in $5 \cdot 30 = 150$ measurement, each stacked from 250 noisy images. We see ZNCC decoding taking 5,7,9,11,13,50,149 randomly chosen measurements respectively yield progressively better disparity reconstruction

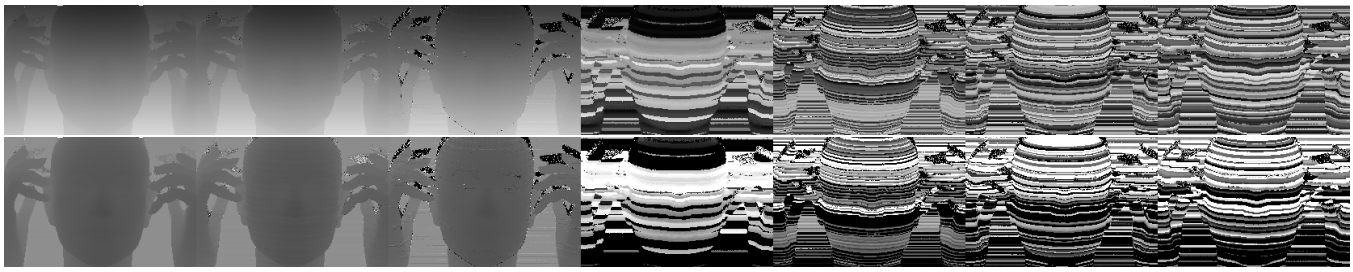

Figure 9: ZNCC decoded phase and disparity for (from left to right) ground truth shown in Figure 8, Hamiltonian, MPS, Optimized-MDE, Optimized-Top0, Optimized-Top1, Optimized-Top2. Apart from ground truth, reconstruction takes intensity of scene under 7 projector patterns, each stacked from 250 noisy measurements

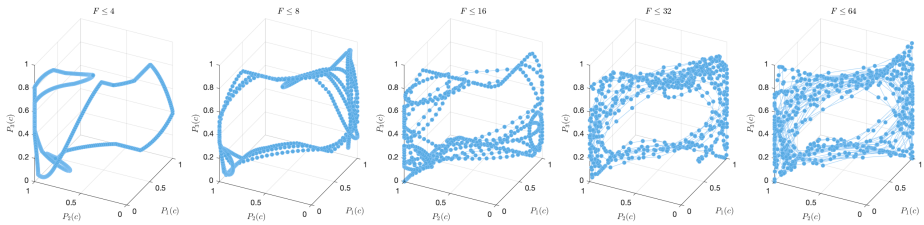

Figure 10: A La Carte codes for 3 patterns with varying max spatial frequency [11] is not distance preserving, i.e. points far along the curve are close in measurement space in the Euclidean sense. So we'd expect it to have sub-optimal performance when measurements are noisy

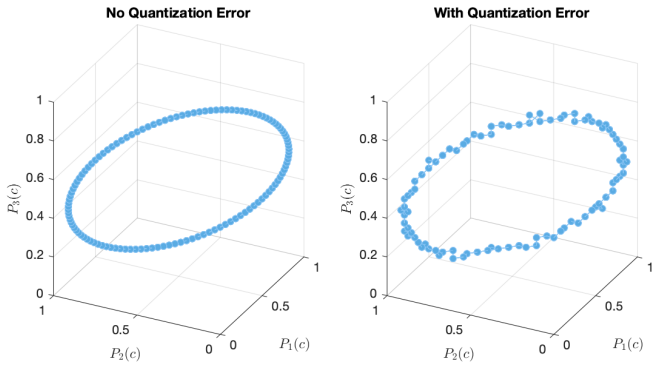


Figure 11: Projector quantization error translates to jitters in the coding curve for sinusoidal patterns

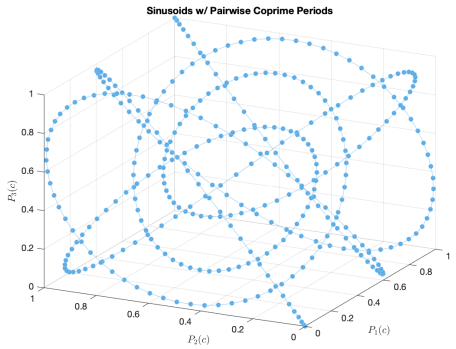


Figure 12: Sinusoids with pairwise co-prime periods (5, 11, 13) is non-intersecting and therefore emits unique decoder using the Chinese Reminder Theorem. However, certain points along the curve does not satisfy the distance preserving property, implying the code is not optimal when measurements are noisy. For sinusoidal codes, non-intersection and distance preserving is a conflicting objective so it seems.

## 4.3 Alternating Direction Method of Multipliers

### 4.3.1 Formulation

Consider the following equality constrained convex optimization problem

$$\text{minimize } f(x) + g(z) \tag{15}$$
$$\text{subject to } Ax + Bz = c \tag{16}$$

assuming $f, g$ are both convex. The augmented lagrangian $L_\rho$ makes the objective better behaved,

$$L_\rho(x, z, y) = f(x) + g(z) + y^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \tag{17}$$

where $y$ is the dual variable, and $\rho > 0$ is the penalty parameter. The method of multipliers is simply doing gradient ascent (using $\rho$ as stepsize) on the dual objective computed from the augmented lagrangian

$$(x^{k+1}, z^{k+1}) = \arg\min_{x,z} L_\rho(x, z, y^k) \tag{18}$$
$$y^{k+1} = y^k + \rho \nabla L_\rho(x^{k+1}, z^{k+1}, y) = y^k + \rho \left( Ax^{k+1} + Bz^{k+1} - c \right) \tag{19}$$

When regular lagrangian is used and when objective is separable, the minimization step (18) is also separable with respect to the primal variables $x, z$. The addition of Euclidean norm on the residual of primal equlity constraints makes the optimization coupled. ADMM is method of multiplier where a single Gauss-Seidel pass over the primal variable is used instead of jointly optimizing for $x, z$.

$$\begin{aligned} x^{k+1} &= \arg\min_x L_\rho(x, z^k, y^k) \\ z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho \left( Ax^{k+1} + Bz^{k+1} - c \right) \end{aligned} \tag{20}$$

We can gather the residual term $r = Ax + Bz - c$ and completing the squares and get

$$L_\rho(x, z, y) = f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c + u\|_2^2 \tag{21}$$

where $u = \frac{1}{\rho} y$ is scaled dual variable. The scaled form ADMM is then

$$\begin{aligned} x^{k+1} &= \arg\min_x \left[ f(x) + \frac{\rho}{2} \left\| Ax + Bz^k - c + u^k \right\|_2^2 \right] \\ z^{k+1} &= \arg\min_z \left[ g(z) + \frac{\rho}{2} \left\| Ax^{k+1} + Bz - c + u^k \right\|_2^2 \right] \\ u^{k+1} &= u^k + Ax^{k+1} + Bz^{k+1} - c \end{aligned} \tag{22}$$

#### 4.3.2 LASSO Problem

The lasso problem is

$$\text{minimize} \quad \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{23}$$

We can apply variable splitting and obtain an equivalent constrained optimization problem

$$\text{minimize} \quad \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \tag{24}$$

$$\text{subject to} \quad x - z = 0 \tag{25}$$

Specialize (22) to lasso to obtain iterates of the form

$$x^{k+1} = \left(A^T A + \rho I\right)^{-1} \left(A^T b + \rho(z^k - u^k)\right) \tag{26}$$

$$z^{k+1} = \mathcal{S}_{\lambda/\rho} \left(x^{k+1} + u^k\right) \tag{27}$$

$$u^{k+1} = u^k + x^{k+1} - z^{k+1} \tag{28}$$

which essentially performs lasso ($x$ update) repeatedly.

#### 4.3.3 Affine Constrained Convex Optimization Problem

$$\begin{aligned} \underset{\mathbf{x}_1, \cdots, \mathbf{x}_N}{\text{minimize}} \quad & \left[ f(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}_i) \right] \\ \text{subject to} \quad & (\mathbf{x}_1, \cdots, \mathbf{x}_N) \in \mathcal{C} \end{aligned} \tag{29}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$, $f_i : \mathbb{R}^n \to (-\infty, \infty)$ are closed proper convex functions for $i = 1, \cdots, N$. Here we assume $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_N} \mid A\mathbf{x} = \mathbf{b}\}$ is an affine set. (29) has an equivalent form,

$$\begin{aligned} \underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} \quad & f(\mathbf{x}) + \delta_{\mathcal{C}}(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{x} - \mathbf{z} = 0 \end{aligned} \tag{30}$$

which can be solved using ADMM iterates of the form,

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \arg\min_{\mathbf{x}_i} \left( f_i(\mathbf{x}_i) + \frac{\rho}{2} \left\| \mathbf{x}_i - (\mathbf{z}_i^k - \mathbf{u}_i^k) \right\|_2^2 \right) = \mathsf{prox}_{(1/\rho) f_i}(\mathbf{z}_i^k - \mathbf{u}_i^k) \\ \mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} \left( \delta_{\mathcal{C}}(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{z} - (\mathbf{x}^{k+1} + \mathbf{u}^k) \right\|_2^2 \right) = \mathsf{prox}_{(1/\rho)\delta_{\mathcal{C}}}(\mathbf{x}^{k+1} + \mathbf{u}^k) \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1} \end{aligned}$$

where $\mathsf{prox}_{\lambda f} : \mathbb{R}^n \to \mathbb{R}^n$ is the proximal operator of $\lambda f$, $\lambda > 0$,

$$\mathsf{prox}_{\lambda f}(\mathbf{v}) = \arg\min_{\mathbf{x}} \left( f(\mathbf{x}) + \frac{1}{2\lambda} \|\mathbf{x} - \mathbf{v}\|_2^2 \right)$$

#### 4.3.4 Some Common Proximal Operator

Evaluating the proximal operator involves solving a convex optimization problem. We will show how we can compute the proximal operators relevant to our methods. (See section 6 of [13])

**Projection Onto Convex Set** The proximal operator of an indicator function onto a convex set $\mathcal{C}$ is simply the projection onto $\mathcal{C}$.

$$\mathsf{prox}_{(1/\rho)\delta_{\mathcal{C}}}(\mathbf{x}^{k+1} + \mathbf{u}^k) = \prod_{\mathcal{C}}(\mathbf{x}^{k+1} + \mathbf{u}^k)$$

When $\mathcal{C}$ is affine, there is an analytic expression for the projection

$$\begin{aligned}\prod_{\mathcal{C}}(\mathbf{v}) &= \mathbf{v} - A^{\dagger}(A\mathbf{v} - \mathbf{b}) \\ &= \mathbf{v} - A^T(A^TA)^{-1}(A\mathbf{v} - \mathbf{b}) \qquad \text{(if } A \in \mathbb{R}^{m\times n} \text{ has } m < n \text{ and full rank)}\end{aligned}$$

**Quadratic Function** Let $f$ be $\ell$-2 norm of an affine function, assuming $A \in \mathbb{R}^{p\times n}$

$$f(x) = \frac{1}{2}\|A\mathbf{x} - \mathbf{y}\|_2^2 = \frac{1}{2}\mathbf{x}^TA^TA\mathbf{x} - \mathbf{y}^TA\mathbf{x} + \frac{1}{2}\mathbf{y}^T\mathbf{y}$$

Then proximal operator of $(1/\rho)f$ has a closed form expression

$$\mathsf{prox}_{(1/\rho)f}(\mathbf{v}) = (\rho I + A^TA)^{-1}(\rho\mathbf{v} + A^T\mathbf{y})$$

which can be solved efficiently with conjugate gradient, as $(\rho I + A^TA) \succ 0$. If $\rho$ is fixed throughout, we can use Cholesky factorization to factor $(\rho I + A^TA)$ in $\mathcal{O}(n^3)$. Any subsequent computation of the inverse with back-solve would only cost $\mathcal{O}(n^2)$. When $p << n$, we can exploit this by using the matrix inversion lemma,

$$(\rho I + A^TA)^{-1} = \frac{1}{\rho}I - \frac{1}{\rho}A^T\left(\rho I + AA^T\right)^{-1}A$$

The dominant cost is computing $(AA^T)^{-1} \in \mathbb{R}^{p\times p}$. Each $x$ update now costs $\mathcal{O}(np^2)$. If use Cholesky factorization to factor $(\rho I + AA^T)$ once, subsequent iteration can be carried out in $\mathcal{O}(np)$ flops, which is essentially costs for matrix-vector multiply.

**$\ell$-1 Norm** Let $f(\mathbf{x}) = \|\mathbf{x}\|_1$, then the proximal operator of $\lambda f$ is

$$\mathsf{prox}_{\lambda f}(\mathbf{v}) = \mathcal{S}_{\lambda}(\mathbf{v})$$

where $\mathcal{S}$ is element-wise soft shrinkage operator

$$(\mathcal{S}_{\lambda}(\mathbf{v}))_i = (1 - \lambda/|\mathbf{v}_i|)_+\mathbf{v}_i \qquad \mathbf{x}_+ = \max(\mathbf{x}, 0)$$

**RED Explicit Regularizer** Let $f(\mathbf{x}) = (1/2)\mathbf{x}^T(\mathbf{x} - \mathcal{D}(\mathbf{x}))$ for some denoiser $\mathcal{D}$, the explicit regularizer in RED.[3]. We use one fixed point iteration evaluate the approximate proximal operator for $\lambda f$. Specifically, we want to evaluate

$$\mathsf{prox}_{(\lambda/\rho)f}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{\lambda}{2}\mathbf{x}^T(\mathbf{x} - \mathcal{D}(\mathbf{x})) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{v}\|_2^2$$

Setting the gradient to zero, we arrive at the fixed point iteration

$$\mathbf{x}^{(k)} \leftarrow \frac{1}{\rho + \lambda}(\lambda\mathcal{D}(\mathbf{x}^{(k-1)}) + \rho\mathbf{v})$$

If we only iterate once, then

$$\mathsf{prox}_{(\lambda/\rho)f}(\mathbf{v}) = \frac{1}{\rho + \lambda}(\lambda\mathcal{D}(\mathbf{x}^{(0)}) + \rho\mathbf{v})$$

for some initialization value $\mathbf{x}^{(0)}$

### 4.4 Linear Inverse Problem

Numerous inverse problem in image processing are modeled using the following relationship

$$y = Ax + \epsilon$$

wherer $y \in \mathbb{R}^m$ is noisy measurement, $x \in \mathbb{R}^n$ are unknown image, $A \in \mathbb{R}^{m \times n}$ captures the forward relationship, $\epsilon \in \mathbb{R}^m$ is noise. The goal is to estimate $x$ from noisy $y$ by making assumptions about the noise distributioon $\epsilon$ and exploiting structure in the latent variable $x$.

#### 4.4.1 MAP inference under Gaussian Noise

A typical approach to solve the linear inverse problem relies on computing max a posterior (MAP) estimator while assuming somer prior distirbution $p_{\mathsf{x}}$. In particular given observation $y$, we want to estimate $\hat{x}$ where

$$\hat{x} = \arg\max_x p_{\mathsf{x}|\mathsf{y}}(x|y) \tag{31}$$

Assuming isotropic Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, the conditional likelihood depends on the noise distribution, $\mathsf{y} \mid \mathsf{x} = x \sim \mathcal{N}(Ax, \sigma^2 I)$. The log likelihood is then

$$\log p_{\mathsf{y}|\mathsf{x}}(y|x) = -\frac{m}{2}\log(2\pi) - \frac{1}{2\sigma^2}\|Ax - y\|_2^2$$

Then the problem (31) reduces to

$$\hat{x} = \arg\max_x \left[\log p_{\mathsf{y}|\mathsf{x}}(y|x) + \log p_{\mathsf{x}}(x)\right] = \arg\min_x \left[\frac{1}{2\sigma^2}\|Ax - y\|_2^2 - \log p_{\mathsf{x}}(x)\right] \tag{32}$$

### 4.5 Image Priors

The choice of regularization has been an important research topic in image processing. Handcrafted priors have been successful in a number of different image recovery tasks. For example, we can choose to enforce task-specific priors: (1) the sparsity of $\mathbf{x}$ with $\ell$-1 norm in image deblurring [14] (2) total variation in image denoising [15] (3) cross-channel correlation in color image demosaicing [16] (4) dark channel prior in image dehazing [17], etc. More exhotically, randomly initialized neural network can inject inductive bias to the optimization and act as image priors. [18]

In addition to hand-crafted priors, there has been interest in algorithm induced priors. Alternating direction method of multipliers (ADMM) is a common convex optimization method for inverse problem where the objective function is separable with respect to the *data term* and the *regularizer*. Each primal update involves an evaluation of a proximal operator, which can be interpreted as performing denoising on some iterate. [2, 19, 20] proposed plug-and-play priors where the choice of regularization is implicitly specified by the denoiser used. [3] proposed an explicit laplacian-based expression for the regularizer and generalizes the method to a number of different iterative optimization algorithms.

The convergence of plug-and-play ADMM is studied by a number of papers. [21] showed fixed point convergence of plug-and-play ADMM. [3] showed convergence of the algorithm under some mild conditions of the denoiser, which are satisfied by some state of the art denoisers like *Block-matching and 3D filtering (BM3D)* [22] and *Trainable Nonlienar Reaction Diffusion (TNRD)* [23]. Most recently, [24] established convergence given that the denoising network satisfy certain Lipschitz condition.

Some proposed to learn proximal operator from data. [25] used a CNN denoiser [26]. Instead of substituting the proximal operator with a denoiser, [27] learns a projection mapping to the space of natural images by training a single neural network and showed impressive results on a number of different linear inverse problems.

## References


[1] Mian Wei et al. "Coded Two-Bucket Cameras for Computer Vision". en. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Vol. 11207. Cham: Springer International Publishing, 2018, pp. 55–73. ISBN: 978-3-030-01218-2 978-3-030-01219-9. DOI: 10.1007/978-3-030-01219-9_4.

[2] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. "Plug-and-Play Priors for Model Based Reconstruction". In: *2013 IEEE Global Conference on Signal and Information Processing*. Dec. 2013, pp. 945–948. DOI: 10.1109/GlobalSIP.2013.6737048.

[3] Yaniv Romano, Michael Elad, and Peyman Milanfar. "The Little Engine That Could: Regularization by Denoising (RED)". In: *arXiv:1611.02862 [cs]* (Nov. 2016). arXiv: 1611.02862 [cs].

[4] Joaquim Salvi, Jordi Pagès, and Joan Batlle. "Pattern Codification Strategies in Structured Light Systems". In: *Pattern Recognition*. Agent Based Computer Vision 37.4 (Apr. 2004), pp. 827–849. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2003.10.002.

[5] Joaquim Salvi et al. "A state of the art in structured light patterns for surface profilometry". In: *Pattern Recognition* 43.8 (Aug. 1, 2010), pp. 2666–2680. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2010.03.004. URL: http://www.sciencedirect.com/science/article/pii/S003132031000124X (visited on 05/24/2020).

[6] E. Horn and N. Kiryati. "Toward optimal structured light patterns". In: *Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134)*. Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134). ISSN: null. May 1997, pp. 28–35. DOI: 10.1109/IM.1997.603845.

[7] D. Moreno, K. Son, and G. Taubin. "Embedded Phase Shifting: Robust Phase Shifting with Embedded Signals". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 2301–2309. DOI: 10.1109/CVPR.2015.7298843.

[8] Tomislav Pribanić, Saša Mrvoš, and Joaquim Salvi. "Efficient multiple phase shift patterns for dense 3D acquisition in structured light scanning". In: *Image and Vision Computing* 28.8 (Aug. 1, 2010), pp. 1255–1266. ISSN: 0262-8856. DOI: 10.1016/j.imavis.2010.01.003. URL: http://www.sciencedirect.com/science/article/pii/S0262885610000107 (visited on 05/26/2020).

[9] V.I. Gushov and Yu.N. Solodkin. "Automatic processing of fringe patterns in integer interferometers". In: *Optics and Lasers in Engineering* 14.4 (1991), pp. 311–324. ISSN: 01438166. DOI: 10.1016/0143-8166(91)90055-X. URL: https://linkinghub.elsevier.com/retrieve/pii/014381669190055X (visited on 02/09/2020).

[10] M. Gupta and S. K. Nayar. "Micro Phase Shifting". en. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI: IEEE, June 2012, pp. 813–820. ISBN: 978-1-4673-1228-8 978-1-4673-1226-4 978-1-4673-1227-1. DOI: 10.1109/CVPR.2012.6247753.

[11] P. Mirdehghan, W. Chen, and K. N. Kutulakos. "Optimal Structured Light a La Carte". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 6248–6257. DOI: 10.1109/CVPR.2018.00654.

[12] Mohit Gupta and Nikhil Nakhate. "A Geometric Perspective on Structured Light Coding". In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 87–102. URL: http://openaccess.thecvf.com/content_ECCV_2018/html/Mohit_Gupta_A_Geometric_Perspective_ECCV_2018_paper.html (visited on 11/26/2019).

[13] Neal Parikh and Stephen Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (Jan. 2014), pp. 127–239. ISSN: 2167-3888. DOI: 10.1561/2400000003.

[14] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm with Application to Wavelet-Based Image Deblurring". In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. Apr. 2009, pp. 693–696. DOI: 10.1109/ICASSP.2009.4959678.

[15] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. "Nonlocal Image and Movie Denoising". en. In: *International Journal of Computer Vision* 76.2 (Feb. 2008), pp. 123–139. ISSN: 1573-1405. DOI: 10.1007/s11263-007-0052-1.

[16] H. S. Malvar, Li-wei He, and R. Cutler. "High-Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images". In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 3. May 2004, pp. iii–485. DOI: 10.1109/ICASSP.2004.1326587.

[17] Raanan Fattal. "Single Image Dehazing". In: *ACM SIGGRAPH 2008 Papers*. SIGGRAPH '08. New York, NY, USA: ACM, 2008, 72:1–72:9. ISBN: 978-1-4503-0112-1. DOI: 10.1145/1399504.1360671.

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *arXiv:1711.10925 [cs, stat]* (Nov. 2017). arXiv: 1711.10925 [cs, stat].

[19] Felix Heide et al. "FlexISP: A Flexible Camera Image Processing Framework". In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 231:1–231:13. ISSN: 0730-0301. DOI: 10.1145/2661229.2661260.

[20] Stanley H. Chan. "Algorithm-Induced Prior for Image Restoration". In: *ArXiv* abs/1602.00715 (2016). arXiv: 1602.00715.

[21] Stanley H. Chan, Xiran Wang, and Omar A. Elgendy. "Plug-and-Play ADMM for Image Restoration: Fixed Point Convergence and Applications". In: *arXiv:1605.01710 [cs]* (May 2016). arXiv: 1605.01710 [cs].

[22] K. Dabov et al. "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". In: *IEEE Transactions on Image Processing* 16.8 (Aug. 2007), pp. 2080–2095. ISSN: 1057-7149. DOI: 10.1109/TIP.2007.901238.

[23] Yunjin Chen and Thomas Pock. "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2017), pp. 1256–1272. ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2016.2596743. arXiv: 1508.02848.

[24] Ernest Ryu et al. "Plug-and-Play Methods Provably Converge with Properly Trained Denoisers". en. In: *International Conference on Machine Learning*. May 2019, pp. 5546–5557.

[25] Tim Meinhardt et al. "Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems". In: *arXiv:1704.03488 [cs]* (Apr. 2017). arXiv: 1704.03488 [cs].

[26] K. Zhang et al. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 26.7 (July 2017), pp. 3142–3155. ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2662206.

[27] J. H. Rick Chang et al. "One Network to Solve Them All — Solving Linear Inverse Problems Using Deep Projection Models". en. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 5889–5898. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.627.