1 Methods

1.1 Joint Demultiplex and Upsample

One approach which we can consider 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 phase and albedo using ZNCC decoding [1]

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. Jointly upsample and demultiplex enforces a prior knowledge of image formation and reduces accumulation of error at each image processing step [2]. This approach allows encoding arbitrary subsampling schemes in the forward operator and can be adapted to any frame number F. 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 $p_{\mathsf{x}}(x) \propto \exp\left\{-(\lambda/\sigma^2)R(\mathbf{x})\right\}$ where $R : \mathbb{R}^{SP} \to \mathbb{R}$ 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

$$\hat{\mathbf{x}}(\mathbf{y}) = \arg\min_{\mathbf{x}} \frac{1}{2}\left\|\mathbf{A}\mathbf{x} - \mathbf{y}\right\|_2^2 + \lambda R(\mathbf{x}) \tag{1}$$

where λ is weight for the regularizer. We solve (1) using PnP/RED [3, 4], which use denoisers in place of a proximal operator under a ADMM style updates. We can specialize the update equations to our problem. First we notice that $\mathbf{A}\mathbf{A}^T$ is diagonal when $\mathbf{W}$ is the optimal bucket multiplexing matrix constructed from Hadamard matrix specified in [5]. The x-update can be simplified [6]

$$\mathbf{x}^{k+1} = \tilde{\mathbf{x}} + \mathbf{A}^T \left[\tfrac{(\mathbf{y}-\mathbf{A}\tilde{x})_1}{\zeta_1+\rho}, \cdots, \tfrac{(\mathbf{y}-\mathbf{A}\tilde{\mathbf{x}})_{SP}}{\zeta_{SP}+\rho}\right]^T \tag{2}$$

where ρ is the parameter for augmented lagrangian, $\tilde{\mathbf{x}} = \mathbf{z}^k - \mathbf{u}^k$, and $\boldsymbol{\zeta} = \mathbf{diag}\left(\mathbf{A}\mathbf{A}^T\right)$, which can be precomputed. (2) is fast because it consists of 2 sparse matix-vector multiply and a few element-wise vector operations. The update equations is then

$$\mathbf{x}^{k+1} = \tilde{\mathbf{x}} + \mathbf{A}^T \left[\tfrac{(\mathbf{y}-\mathbf{A}\tilde{x})_1}{\zeta_1+\rho}, \cdots, \tfrac{(\mathbf{y}-\mathbf{A}\tilde{\mathbf{x}})_{SP}}{\zeta_{SP}+\rho}\right]^T \qquad \tilde{\mathbf{x}} = \mathbf{z}^k - \mathbf{u}^k \tag{3}$$

$$\mathbf{z}^{k+1} = \frac{1}{\rho+\lambda}\left(\lambda\mathcal{D}(\mathbf{z}^k) + \rho(\mathbf{x}^{k+1} + \mathbf{u}^k)\right) \tag{4}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1} \tag{5}$$

1.2 Experimental Results

Ground Truth Ground truth phase/disparity is obtained by computing ZNCC decoding on sinusoids of periods 1,2,5,17,31, each shifted 30 times. Each measurement under a unique illumination pattern is stacked form 250 noisy images.

Empirical Performance Upper Bound In Figure 1, we experimentally determine what is empirically possible for the decoding algorithm, without either noise nor recosntruction error, when given measurement under different number of phase shifts, assuming sinusoidal pattern with period 1.

latex

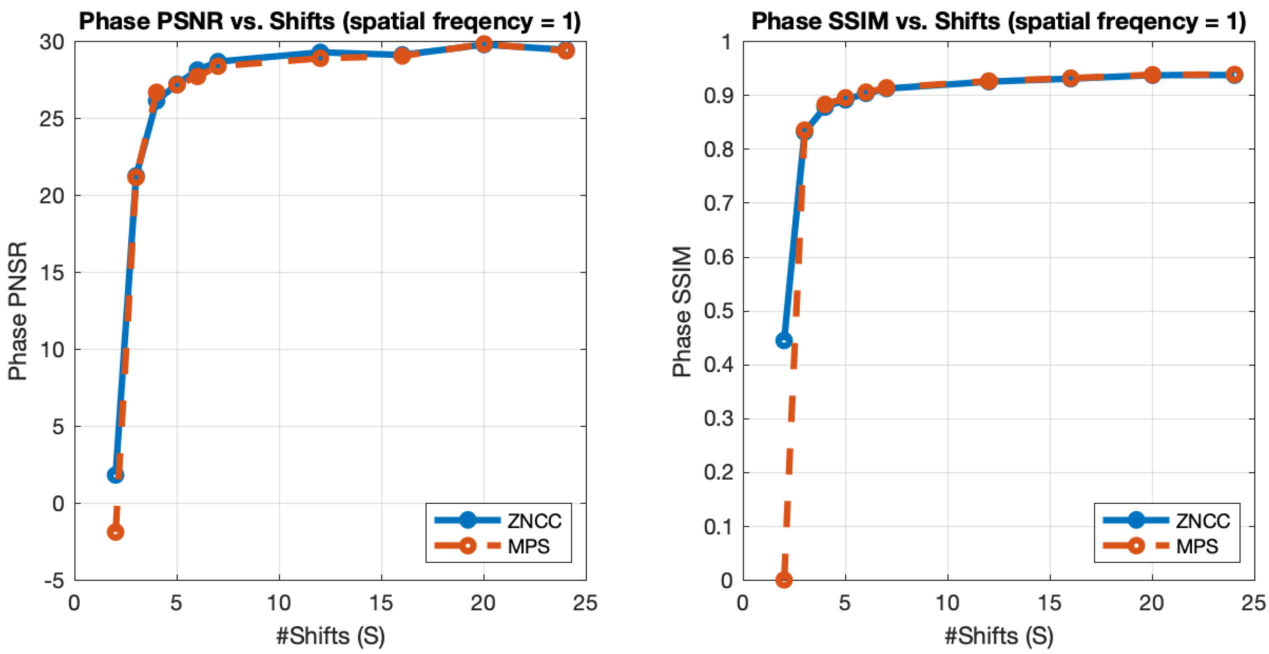

Figure 1: PSNR/SSIM computed by comparing phase obtained by ZNCC/MPS decoding on stacked noiseless images for sinusoidal pattern with period of 1 with varying number of phase shifts to the ground-truth phase. Both decoder performed similarly. We see large improvement when $S = 3, 4$ and minimal improvement when $S > 7$.

	S=2	S=3	S=4	S=5	S=6	S=7	S=12
ADMM-MF	1.92/0.436	20.44/0.747	24.49/0.828	25.47/0.832	26.35/0.848	26.68/0.848	27.19/0.861
ADMM-TNRD	1.92/0.436	19.13/0.728	19.66/0.816	20.71/0.812	19.35/0.814	19.59/0.830	21.16/0.856

Table 1: MyTableCaption

1.3 Solving Inverse Problem using RED

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 [4] 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,

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

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

$$\begin{array}{ll} \text{minimize} & f_1(\mathbf{x}_1) + \lambda_2 f_2(\mathbf{x}_2) + \lambda_3 f_3(\mathbf{x}_3) \\ \text{subject to} & (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \in \mathcal{C} \end{array} \tag{6}$$

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

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

1.4 Optimization

As shown below, the scaled form ADMM for solving (6) is given by

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