Reconstructing Transient Images from Single-Photon Sensors: Supplemental Document

Matthew O'Toole, Felix Heide, David B. Lindell, Kai Zang, Steven Diamond, Gordon Wetzstein
Stanford University

1. Derivation of Joint Denoising and Deconvolution

1.1. Image Formation

In many imaging problems, such as low-light photography, microscopy, and also single-photon imaging, Poisson-distributed shot noise dominates the image formation. Note that no engineering effort can mitigate this, because shot noise is an inherent property of the particle nature of light. The image formation can be expressed as

$$\mathbf{h} \sim \mathcal{P}\left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right), \tag{1}$$

where $\mathcal{P}$ models a Poisson distribution. Here, we represent the temporal impulse response (*i.e.*, the transient image), the measured histogram, and the dark count as discrete column vectors $\boldsymbol{\tau}, \mathbf{h}, \mathbf{d} \in \mathbb{R}^{n_x n_y n_t \times 1}$. The matrix $\mathbf{A} \in \mathbb{R}^{n_x n_y n_t \times n_x n_y n_t}$ encodes the convolution of the transient image with the laser pulse and SPAD jitter $(g * f)[t]$ as well as other scale factors. Each transient image has a resolution of $n_x \times n_y$ pixels and each pixel has n_t time bins.

The probability of having taken a measurement at one particular SPAD and histogram bin i is thus given as

$$p\left(\mathbf{h}_i|\mathbf{A}\boldsymbol{\tau}\right) = \frac{\left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right)_i^{\mathbf{h}_i} e^{-\left(\mathbf{A}\boldsymbol{\tau}+\boldsymbol{d}\right)_i}}{\mathbf{h}_i!}. \tag{2}$$

Using the notational trick $u^v = e^{\log(u)v}$, the joint probability of all measurements is expressed as

$$p\left(\mathbf{h}|\mathbf{A}\boldsymbol{\tau}\right) = \prod_{i=1}^{M} p\left(\mathbf{h}_i|\mathbf{A}\boldsymbol{\tau}\right) = \prod_{i=1}^{M} e^{\log\left(\left(\mathbf{A}\boldsymbol{\tau}+\boldsymbol{d}\right)_i\right)\mathbf{h}_i} \times e^{-\left(\mathbf{A}\boldsymbol{\tau}+\boldsymbol{d}\right)_i} \times \frac{1}{\mathbf{h}_i!}, \tag{3}$$

where $M = n_x n_y n_t$. Note that it is essential to model the joint probability for this image formation, because elements in $\boldsymbol{\tau}$ may affect any or all measurements due to the mixing matrix $\mathbf{A}$. The log-likelihood of this expression is

$$\begin{aligned} \log\left(L\left(\boldsymbol{\tau}\right)\right) &= \log\left(p\left(\mathbf{h}|\mathbf{A}\boldsymbol{\tau}\right)\right) \\ &= \sum_{i=1}^{M} \log\left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right)_i \mathbf{h}_i - \sum_{i=1}^{M} \left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right)_i - \sum_{i=1}^{M} \log\left(\mathbf{h}_i!\right) \\ &= \log\left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right)^T \mathbf{h} - \left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right)^T \mathbf{1} - \sum_{i=1}^{M} \log\left(\mathbf{h}_i!\right) \end{aligned} \tag{4}$$

and its gradient is

$$\nabla \log\left(L\left(\boldsymbol{\tau}\right)\right) = \mathbf{A}^T \mathrm{diag}\left(\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}\right)^{-1} \mathbf{h} - \mathbf{A}^T\mathbf{1} = \mathbf{A}^T\left(\frac{\mathbf{h}}{\mathbf{A}\boldsymbol{\tau} + \boldsymbol{d}}\right) - \mathbf{A}^T\mathbf{1}. \tag{5}$$

1.2. ADMM-based Reconstruction

As outlined in the primary text, the objective function for recovering transient images from blurry images that are corrupted by Poissonian noise is a maximum a posteriori (MAP) problem:

$$\begin{aligned} \underset{\{\tau\}}{\text{minimize}} \quad & -\log\left(p\left(\mathbf{h}|\mathbf{A}\boldsymbol{\tau}\right)\right) + \Gamma\left(\boldsymbol{\tau}\right), \\ \text{subject to} \quad & 0 \leq \boldsymbol{\tau} \end{aligned} \tag{6}$$

Without loss of generality, we replace the constrained objective by an unconstrained cost function

$$\underset{\{\boldsymbol{\tau}\}}{\text{minimize}} \quad -\log\left(p\left(\mathbf{h}|\mathbf{A}\boldsymbol{\tau}\right)\right) + \mathcal{I}_{\mathbb{R}_+}\left(\boldsymbol{\tau}\right) + \Gamma\left(\boldsymbol{\tau}\right) \tag{7}$$

Here, $\mathcal{I}_{\mathbb{R}_+}\left(\cdot\right)$ is the indicator function that enforces the nonnegativity constraints

$$\mathcal{I}_{\mathbb{R}_+}\left(x\right) = \begin{cases} 0 & x \in \mathbb{R}_+ \\ +\infty & x \notin \mathbb{R}_+ \end{cases} \tag{8}$$

where $\mathbb{R}_+$ is the closed nonempty convex set representing nonnegative real-valued numbers.

Next, we follow the general approach of the alternating direction method of multipliers (ADMM) [1] and split the unknowns while enforcing consensus in the constraints

$$\begin{aligned} \underset{\{\boldsymbol{\tau}\}}{\text{minimize}} \quad & \underbrace{-\log\left(p\left(\mathbf{h}|\mathbf{z}_1\right)\right)}_{g_1(\mathbf{z}_1)} + \underbrace{\mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_2\right)}_{g_2(\mathbf{z}_2)} + \underbrace{\Gamma\left(\mathbf{z}_3\right)}_{g_3(\mathbf{z}_3)} \\ \text{subject to} \quad & \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{I} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{K}} \boldsymbol{\tau} - \underbrace{\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{bmatrix}}_{\mathbf{z}} = 0 \end{aligned} \tag{9}$$

The Augmented Lagrangian of this objective is formulated as

$$L_\rho\left(\boldsymbol{\tau}, \mathbf{z}, \mathbf{y}\right) = \sum_{i=1}^{3} g_i\left(\mathbf{z}_i\right) + \mathbf{y}^T\left(\mathbf{K}\boldsymbol{\tau} - \mathbf{z}\right) + \frac{\rho}{2}\left\|\mathbf{K}\boldsymbol{\tau} - \mathbf{z}\right\|_2^2. \tag{10}$$

An iterative solver can now be constructed that sequentially minimizes the Augmented Lagrangian w.r.t. each of the variables $\boldsymbol{\tau}$ and $\mathbf{z}_i$. For convenience, the scaled form of ADMM is used that uses the scaled dual variable $\mathbf{u} = (1/\rho)\mathbf{y}$ instead of the Lagrange multiplier $\mathbf{y}$. This leads to the following iterative updates:

for $k = 1$ **to** $maxiter$

$$\boldsymbol{\tau} \leftarrow \mathbf{prox}_{\|\cdot\|_2}\left(\mathbf{v}\right) = \underset{\{\boldsymbol{\tau}\}}{\arg\min}\, L_\rho\left(\boldsymbol{\tau}, \mathbf{z}, \mathbf{y}\right) = \underset{\{\boldsymbol{\tau}\}}{\arg\min}\, \frac{1}{2}\left\|\mathbf{K}\boldsymbol{\tau} - \mathbf{v}\right\|_2^2, \quad \mathbf{v} = \mathbf{z} - \mathbf{u} \tag{11}$$

$$\mathbf{z}_1 \leftarrow \mathbf{prox}_{\mathcal{P},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_1\}}{\arg\min}\, L_\rho\left(\boldsymbol{\tau}, \mathbf{z}, \mathbf{y}\right) = \underset{\{\mathbf{z}_1\}}{\arg\min}\, g_1\left(\mathbf{z}_1\right) + \frac{\rho}{2}\left\|\mathbf{v} - \mathbf{z}_1\right\|_2^2, \quad \mathbf{v} = \mathbf{A}\boldsymbol{\tau} + \mathbf{u}_1 \tag{12}$$

$$\mathbf{z}_2 \leftarrow \mathbf{prox}_{\mathcal{I},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_2\}}{\arg\min}\, L_\rho\left(\boldsymbol{\tau}, \mathbf{z}, \mathbf{y}\right) = \underset{\{\mathbf{z}_2\}}{\arg\min}\, g_2\left(\mathbf{z}_2\right) + \frac{\rho}{2}\left\|\mathbf{v} - \mathbf{z}_2\right\|_2^2, \quad \mathbf{v} = \boldsymbol{\tau} + \mathbf{u}_2 \tag{13}$$

$$\mathbf{z}_3 \leftarrow \mathbf{prox}_{\Gamma,\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_3\}}{\arg\min}\, L_\rho\left(\boldsymbol{\tau}, \mathbf{z}, \mathbf{y}\right) = \underset{\{\mathbf{z}_3\}}{\arg\min}\, g_3\left(\mathbf{z}_3\right) + \frac{\rho}{2}\left\|\mathbf{v} - \mathbf{z}_3\right\|_2^2, \quad \mathbf{v} = \boldsymbol{\tau} + \mathbf{u}_3 \tag{14}$$

$$\underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \end{bmatrix}}_{\mathbf{u}} \leftarrow \mathbf{u} + \mathbf{K}\boldsymbol{\tau} - \mathbf{z}$$

end for

1.2.1 Proximal Operator for Quadratic Term (Eq. (11))

For the proximal operator of the quadratic subproblem, we formulation the closed-form solution via the normal equations:

$$\begin{aligned}\mathbf{prox}_{\|\cdot\|_2}(\mathbf{v}) &= \arg\min_{\{\boldsymbol{\tau}\}} \frac{1}{2}\| \underbrace{\begin{bmatrix}\mathbf{A}\\ \mathbf{I}\\ \mathbf{I}\end{bmatrix}}_{\mathbf{K}} \boldsymbol{\tau} - \underbrace{\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \mathbf{v}_3\end{bmatrix}}_{\mathbf{v}} \|_2^2 = \left(\mathbf{K}^T\mathbf{K}\right)^{-1}\mathbf{K}^T\mathbf{v} \\ &= \left(\mathbf{A}^T\mathbf{A} + 2\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3\right) \quad (15)\\ &= \mathcal{F}_t^{-1}\left(\frac{\mathcal{F}_t\{c\}^* \cdot \mathcal{F}_t\{v_1\} + \mathcal{F}_t\{v_2\} + \mathcal{F}_t\{v_3\}}{\mathcal{F}_t\{c\}^* \cdot \mathcal{F}_t\{c\} + 2}\right) \quad (16)\end{aligned}$$

where $\mathcal{F}_t$ is the 1D Fourier transform along the temporal dimension of a 3D datacube, the operators $\cdot$ and $\frac{\cdot}{\cdot}$ are element-wise multiplication and division, $c \in \mathbb{R}^{n_x \times n_y \times n_t}$ is a per-SPAD temporal convolution kernel combining laser pulse shape and SPAD jitter, which can be different for each SPAD. Note that this proximal operator is basically an inverse filter and does not require explicit matrix-vector products to be computed. The structure of this specific problem allows for a closed-form solution to be expressed using a few Fourier transforms as well as element-wise multiplications and divisions. Typically, $\mathcal{F}_t\{c\}^*$ and the denominator can be precomputed and do not have to be updated throughout the ADMM iterations.

The difference between Equations (15) and (16) is that the former uses vectorized discrete variables whereas the latter uses the non-vectorized version of the same quantities. So $\boldsymbol{\tau}, \mathbf{v}_{1/2/3} \in \mathbb{R}^{n_x n_y n_t \times 1}$ and $\tau, v_{1/2/3}, c \in \mathbb{R}^{n_x \times n_y \times n_t}$. The reason for this change in notation is that we can express the matrix vector multiplication $\mathbf{A}\boldsymbol{\tau}$, which represents a temporal convolution for each SPAD, via the convolution theorem, as a multiplication in the frequency domain, *i.e.*

$$\mathbf{A}\boldsymbol{\tau} = \text{vec}\left(c *_t \tau\right) = \text{vec}\left(\mathcal{F}_t^{-1}\left\{\mathcal{F}_t\{c\} \cdot \mathcal{F}_t\{\tau\}\right\}\right), \quad (17)$$

where $*_t$ is a 1D convolution along the time dimension of a 3D datacube and the operator $\text{vec}(\cdot)$ vectorizes a 3D datacube into a single column vector.

1.2.2 Proximal Operator for Poisson Term (Eq. (12))

Recall, this proximal operator is defined as

$$\mathbf{prox}_{\mathcal{P},\rho}(\mathbf{v}) = \arg\min_{\{\mathbf{z}_1\}} J(\mathbf{z}_1) = \arg\min_{\{\mathbf{z}_1\}} -\log\left(p\left(\mathbf{h}|\mathbf{z}_1\right)\right) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}_1\|_2^2 \quad (18)$$

Using Equation (5), we write the objective function $J(\mathbf{z}_1)$ for this subproblem as

$$J(\mathbf{z}_1) = -\log\left(\mathbf{z}_1 + \mathbf{d}\right)^T\mathbf{h} + \left(\mathbf{z}_1 + \mathbf{d}\right)^T\mathbf{1} + \frac{\rho}{2}\left(\mathbf{z}_1 - \mathbf{v}\right)^T\left(\mathbf{z}_1 - \mathbf{v}\right). \quad (19)$$

Next, we equate the gradient of the objective to zero

$$\nabla J(\mathbf{z}_1) = -\text{diag}\left(\mathbf{z}_1 + \mathbf{d}\right)^{-1}\mathbf{h} + \mathbf{1} + \rho\left(\mathbf{z}_1 - \mathbf{v}\right) = 0. \quad (20)$$

This proximal operator removes the need for including the system matrix $\mathbf{A}$, which makes the noisy observations *independent* of each other, so we do not need to account for the *joint probability* of all measurements. This can be seen by looking at the gradient of J w.r.t. individual elements in $\mathbf{z}_1$:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{z}_{1_j}} &= -\frac{\mathbf{h}_j}{\mathbf{z}_{1_j} + \mathbf{d}_j} + 1 + \rho\left(\mathbf{z}_{1_j} - \mathbf{v}_j\right) \quad (21)\\ &= \mathbf{z}_{1_j}^2 + \frac{1 + \rho\mathbf{d}_j - \rho\mathbf{v}_j}{\rho}\mathbf{z}_{1_j} - \frac{\mathbf{h}_j + \rho\mathbf{d}_j\mathbf{v}_j - \mathbf{d}_j}{\rho} = 0 \quad (22)\end{aligned}$$

This expression is a classical root-finding problem of a quadratic, which can be solved independently for each $\mathbf{z}_{1_j}$. The quadratic has two roots, but due to the nonnegativity constraints in our objective function we are only interested in the positive one. Thus, we can define the proximal operator as

$$\mathbf{prox}_{\mathcal{P},\rho}(\mathbf{v}) = -\frac{1 + \rho\mathbf{d} - \rho\mathbf{v}}{2\rho} + \sqrt{\left(\frac{1 + \rho\mathbf{d} - \rho\mathbf{v}}{2\rho}\right)^2 + \frac{\mathbf{h} + \rho\mathbf{d}\mathbf{v} - \mathbf{d}}{\rho}}. \quad (23)$$

This operator is independently applied to each pixel and can therefore be implemented analytically without any iterations. Note that we still need to chose a parameter ρ for the ADMM updates. Heuristically, small values (e.g. $1e^{-5}$) work best for large signals with little noise, but as the Poisson noise starts to dominate the image formation, ρ should be higher.

For more details on this proximal operator, please refer to the paper by Dupe et al. [5].

1.2.3 Proximal Operator for Indicator Function (Eq. (13))

The proximal operator for the indicator function representing the constraints is rather straightforward

$$\mathbf{prox}_{\mathcal{I},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_2\}}{\arg\min}\, \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_2\right) + \frac{\rho}{2}\left\|\mathbf{v}-\mathbf{z}_2\right\|_2^2 = \underset{\{\mathbf{z}_2 \in \mathbb{R}_+\}}{\arg\min}\, \left\|\mathbf{z}_2-\mathbf{v}\right\|_2^2 = \Pi_{\mathbb{R}_+}\left(\mathbf{v}\right), \tag{24}$$

where $\Pi_{\mathbb{R}_+}\left(\cdot\right)$ is the element-wise projection operator onto the convex set $\mathbb{R}_+$

$$\Pi_{\mathbb{R}_+}\left(\mathbf{v}_j\right) = \begin{cases} 0 & \mathbf{v}_j < 0 \\ \mathbf{v}_j & \mathbf{v}_j \geq 0 \end{cases} \tag{25}$$

For more details on this and other proximal operators, please refer to [1, 8].

1.2.4 Proximal Operator for Regularization Term Γ (Eq. (14))

The objective function of Equation (14) is

$$\mathbf{prox}_{\Gamma,\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_3\}}{\arg\min}\, \Gamma\left(\mathbf{z}_3\right) + \frac{\rho}{2}\left\|\mathbf{v}-\mathbf{z}_3\right\|_2^2 \tag{26}$$

Total Variation Prior Total variation (TV) is one of the most widely used priors in image reconstruction. An isotropic transverse TV prior was also recently proposed for denoising range data captured with SPADs [9]. Solving Equation (26) for $\Gamma\left(\mathbf{z}_3\right) = \left\|\mathbf{z}_3\right\|_{TV}$ can be implemented via a soft-thresholding operator. Please see Boyd et al. [1] for more details.

Self-similarity Prior Self-similarity priors are commonly used in state-of-the-art denoising approaches, such as non-local means [2] and block-matching and 3D filtering (BM3D) [4]. Similar to standard videos, our data is three-dimensional so we chose video BM3D (VBM3D) [3] as the self-similarity prior of choice.

Following previous work on image optimization (e.g., [7]), we note the similarity between Equation (26) and a general maximum a posteriori (MAP) formulation that occurs for any image denoising problem with Gaussian noise:

$$\underset{\{x\}}{\text{minimize}}\ \frac{1}{2\sigma^2}\left\|y-x\right\|_2^2 + \Gamma\left(x\right), \tag{27}$$

with y being the noisy measurements and x the unknown, noise-free image. This similarity allows us to use *any* algorithm for denoising Gaussian images as a proximal operator for the regularization term. VBM3D, for example, uses a patch-based self-similary prior for Γ, which we can directly exploit as

$$\mathbf{prox}_{\Gamma,\rho}\left(\mathbf{v}\right) = \text{vec}\left(\text{VBM3D}_{\sqrt{1/\rho}}\left(v\right)\right). \tag{28}$$

We simply use existing, optimized implementations of VBM3D for this proximal operator by setting the standard deviation parameter of the Gaussian denoiser to $\sigma = \sqrt{1/\rho}$ and letting it denoise the temporary variable $v \in \mathbb{R}^{n_x \times n_y \times n_t}$ which is the un-vectorized form of $\mathbf{v}$.

2. Extended Results & Comparisons

Simulations Figure 1 provides a comparison of our reconstruction procedures for additional frames and RGB data.

Experiments Figures 2 through 8 show the Poisson-TV (2D), Poisson-TV (3D) and Poisson-VBM3D reconstruction results for all scenes.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, Jan. 2011. 2, 4

[2] A. Buades and B. Coll. A non-local algorithm for image denoising. *Proc. CVPR*, pages 60–65, 2005. 4

[3] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. *Proc. European Signal Processing Conference*, pages 145–149, Sept. 2007. 4

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. on Image Processing*, 16(8):2080–2095, 2007. 4

[5] F.-X. Dupe, M. Fadili, and J.-L. Starck. Inverse problems with Poisson noise: Primal and primal-dual splitting. *IEEE International Conference on Image Processing*, pages 1901–1904, 2011. 4

[6] G. Gariepy, N. Krstaji, R. Henderson, C. Li, R. R. Thomson, G. S. Buller, B. Heshmat, R. Raskar, J. Leach, and D. Faccio. Single-photon sensitive light-in-fight imaging. *Nature Communications*, 6(6021), Jan. 2015. 5

[7] F. Heide, S. Diamond, M. Niessner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein. ProxImaL: Efficient image optimization using proximal algorithms. *ACM Trans. on Graph.*, 35(4):84:1–84:15, 2016. 4

[8] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Machine Learning*, 1(3):123–231, 2014. 4

[9] D. Shin, F. Xu, D. Venkatraman, R. Lussana, F. Villa, F. Zappa, V. K. Goyal, F. N. C. Wong, and J. H. Shapiro. Photon-efficient imaging with a single-photon camera. *Nature Communications*, 7(12046):1–7, 2016. 4

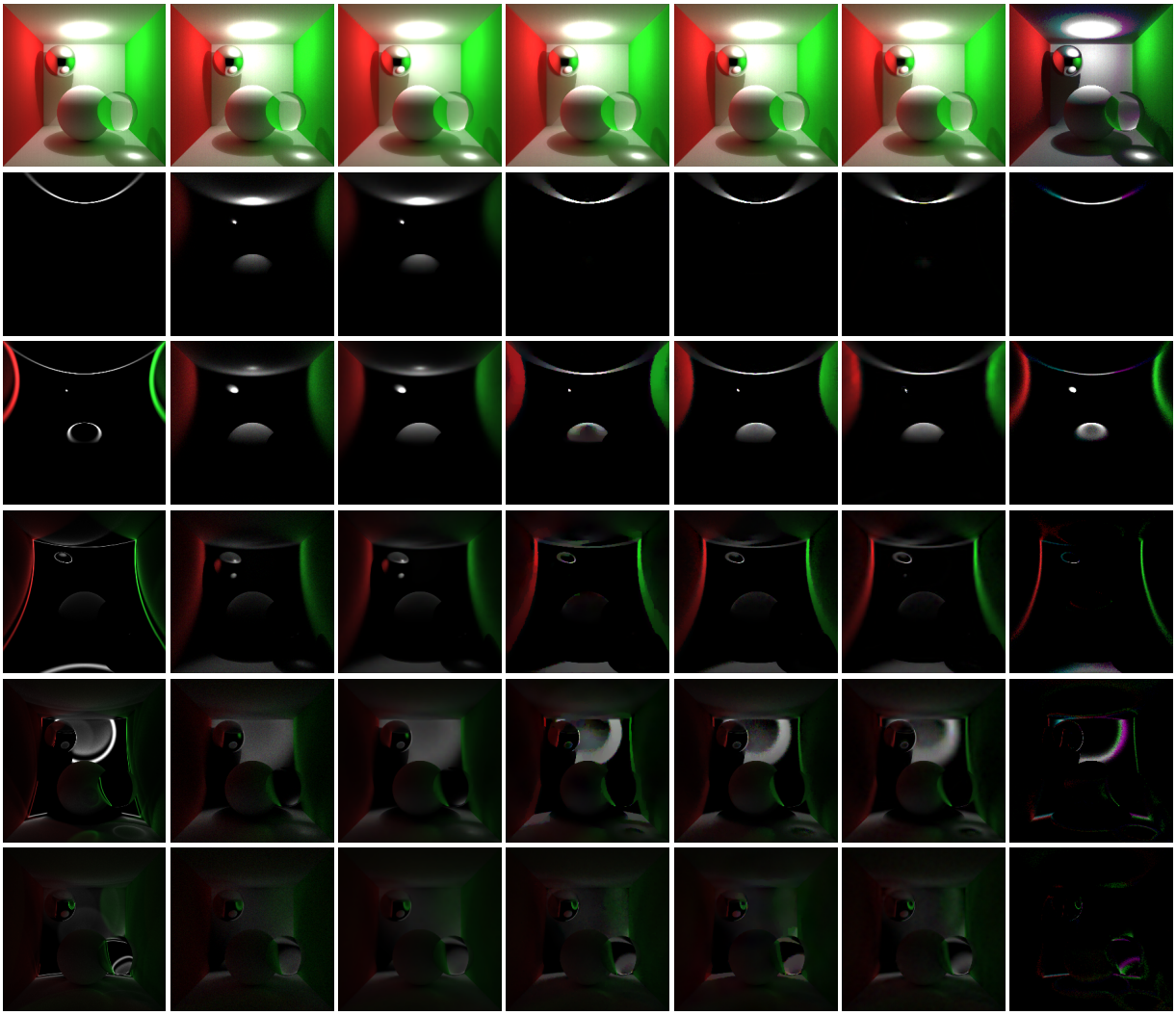

Figure 1. Denoising & deblurring results from a simulated transient image.

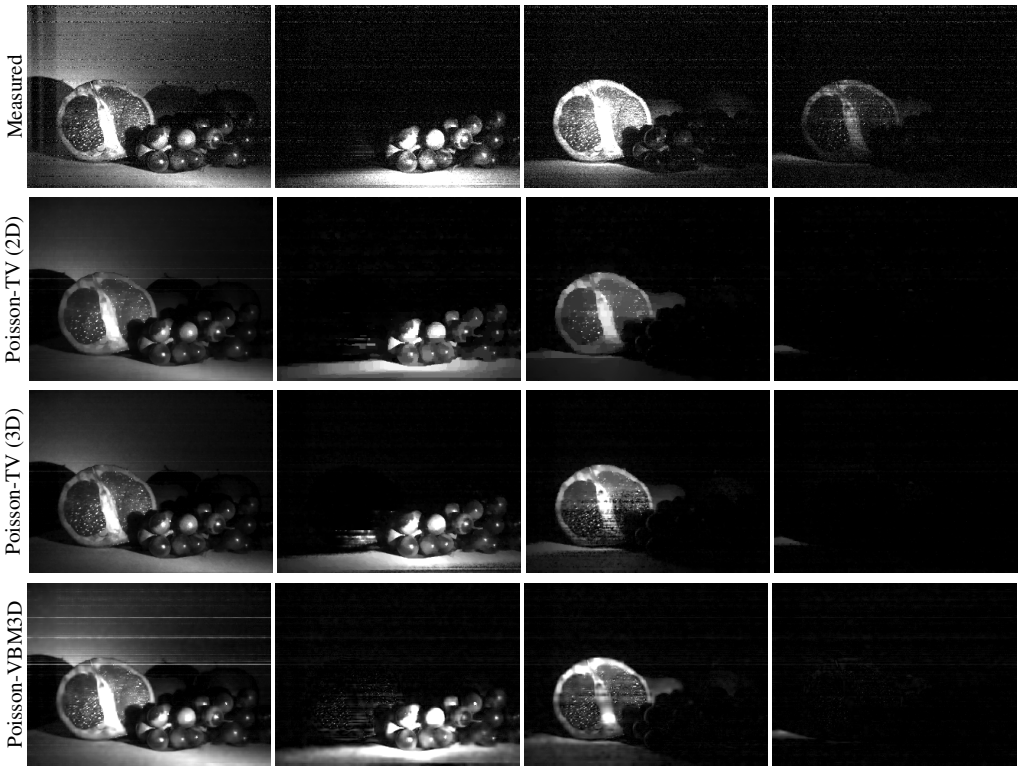

Figure 2. *Fruit* scene. **Column 1**: Images of the scene under laser illumination, *i.e.*, the result of integrating the transient image over the temporal dimension. **Column 2-4**: Frames from a transient image appearing in chronological order.

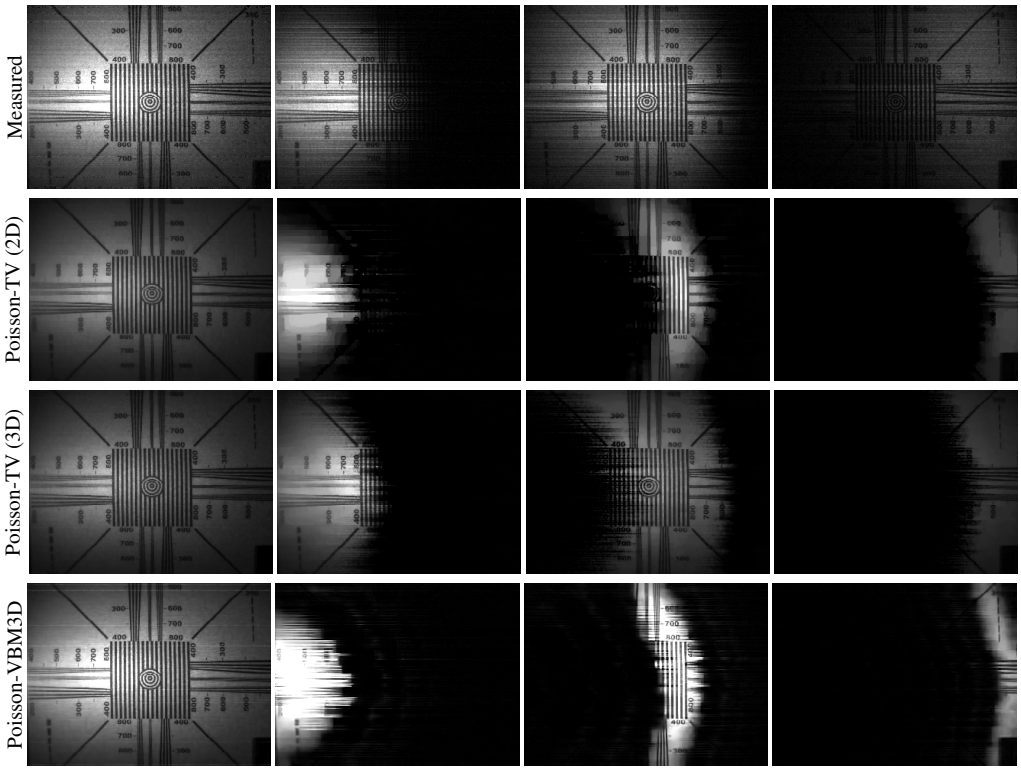

Figure 3. *Resolution chart* scene.

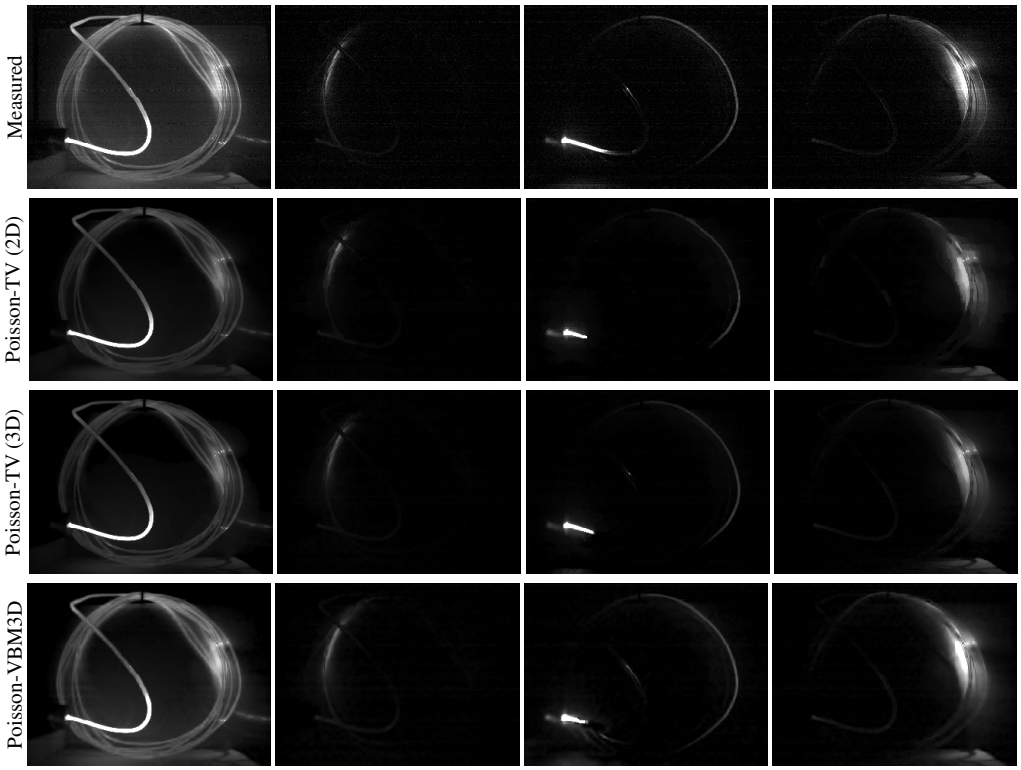

Figure 4. *Optical fiber* scene.

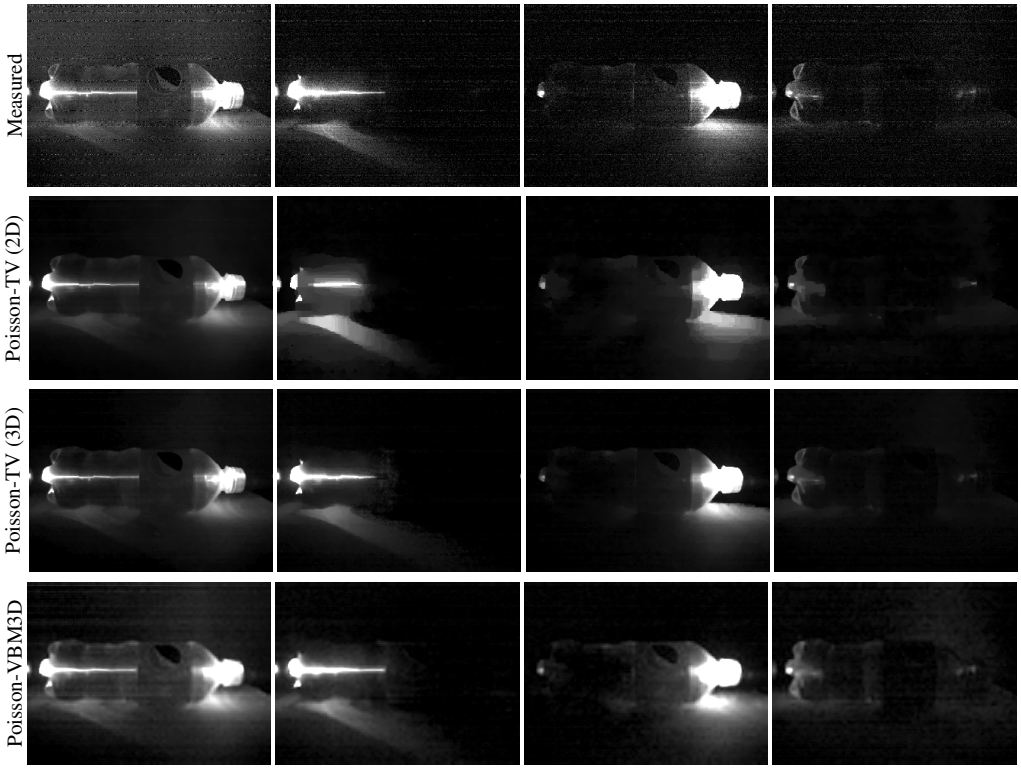

Figure 5. *Soda bottle* scene.

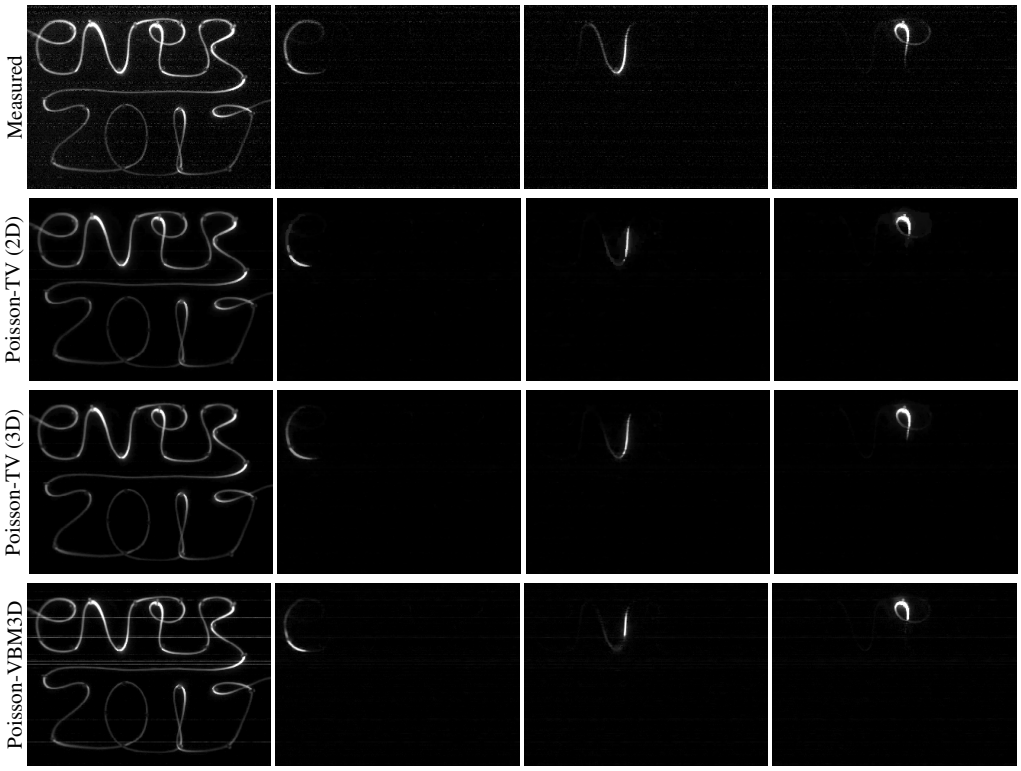

Figure 6. *CVPR logo* scene.

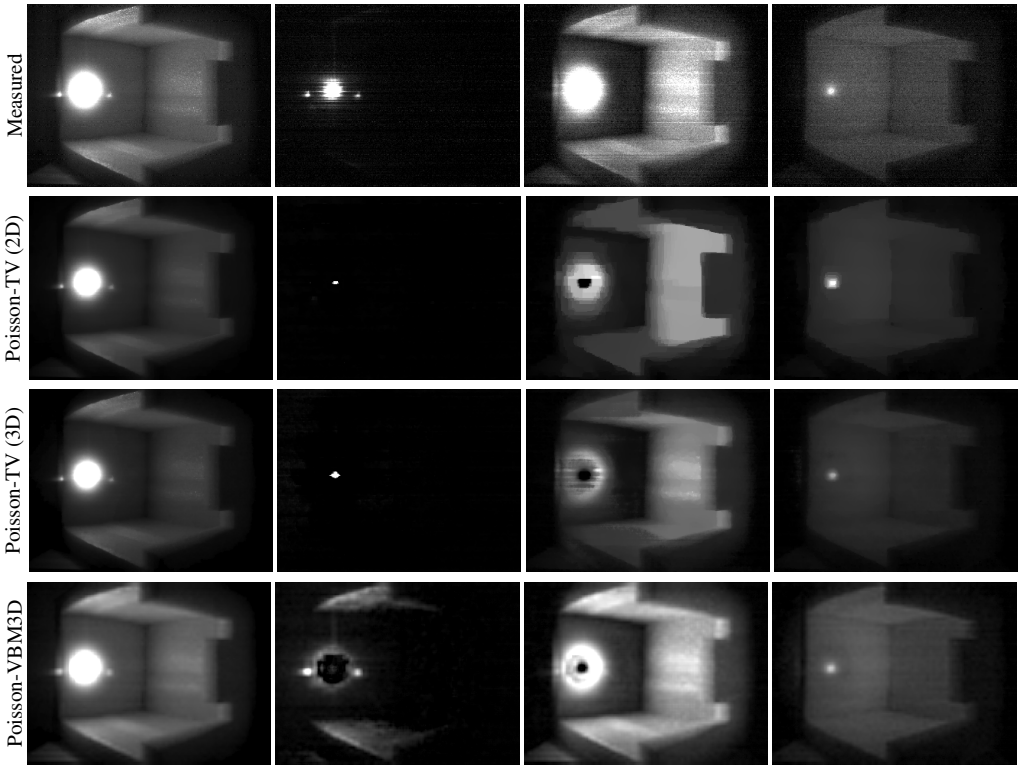

Figure 7. *Foam box* scene.

Figure 8. *Statue* scene.