Robust and accurate transient light transport decomposition via convolutional sparse coding

Xuemei Hu,[1,2] Yue Deng,[1,2] Xing Lin,[1,2] Jinli Suo,[1,2] Qionghai Dai,[1,2,*] Christopher Barsi,[3] and Ramesh Raskar[3]

[1]*Department of Automation, Tsinghua University, Beijing 100084, China*

[2]*Beijing Key Laboratory of Multi-dimension and Multi-scale Computational Photography (MMCP), Tsinghua University, China*

[3]*Media Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, Massachusetts 02139, USA*

**Corresponding author: qhdai@tsinghua.edu.cn*

Received January 16, 2014; revised April 14, 2014; accepted April 16, 2014;
posted April 18, 2014 (Doc. ID 204602); published May 23, 2014

Ultrafast sources and detectors have been used to record the time-resolved scattering of light propagating through macroscopic scenes. In the context of computational imaging, decomposition of this transient light transport (TLT) is useful for applications, such as characterizing materials, imaging through diffuser layers, and relighting scenes dynamically. Here, we demonstrate a method of convolutional sparse coding to decompose TLT into direct reflections, inter-reflections, and subsurface scattering. The method relies on the sparsity composition of the time-resolved kernel. We show that it is robust and accurate to noise during the acquisition process.

OCIS codes: (070.2025) Discrete optical signal processing; (100.3190) Inverse problems; (150.1135) Algorithms.
http://dx.doi.org/10.1364/OL.39.003177

Light transport (LT) is the interaction between light and the scene through which it propagates. It represents the changes in the direction and intensity of light rays from the source to the detector. LT can be separated into two different components: direct reflections and global lighting. The latter component includes translucencies, volumetric scattering, inter-reflections, and subsurface scattering (sss). For conventional imaging, only the sum of all LT phenomena is measured, making it difficult to infer individual scene properties. Much research, therefore, focuses on decomposing the measured LT into its components. This methodology, for example, is similar to constructing volumetric information from surface measurements in diffuse optical tomography [1].

From a systems perspective, LT is characterized by a matrix that maps the input illumination to the signal detected. Current methods of LT decomposition are based on estimating these matrix elements. However, these methods require measurements from many illumination patterns or complex reconstruction algorithms [2–4].

Alternatively, transient LT (TLT) leverages the relative propagation delays of different ray paths. Well known in ultrafast optics, it has been studied only recently in the context of LT for computer vision and graphics applications. In this regime, TLT systems record the picosecond-scale per-pixel time profiles of intensity and use this added information for LT decomposition. This technique has been explored recently for depth estimation and 3D reconstructions [5–7], and has provided new visualizations for ultrafast light propagation in macroscopic scenes [8].

TLT decomposition has a wide range of applications and benefits [8–10]. An intuitive gradient-based decomposition (GradBM) method was used recently in this context. However, noisy data tend to corrupt its results, especially at edges, requiring more robust methods. Furthermore, complex geometries usually require GradBM to be run manually several times, so that full LT decomposition cannot be completed automatically. This is especially true for multiple inter-reflections overlapping in time with an sss component.

Here, we propose a convolutional sparse coding decomposition (ConvSCD) to decompose TLT robustly and accurately. There are three main contributions. First, we note that the TLT kernel composition for a given scene is sparse and is, therefore, amenable for ConvSCD. Second, we verify that with no *a priori* knowledge about the TLT structure, our method automatically detects and separates each TLT component. Third, we show that ConvSCD is robust in the presence of noise.

The experimental setup is shown in Fig. 1(a) [11]. A Ti:Sapphire pulsed laser (50 fs) is focused onto a diffuser at point O to create a pulsed spherical energy front,

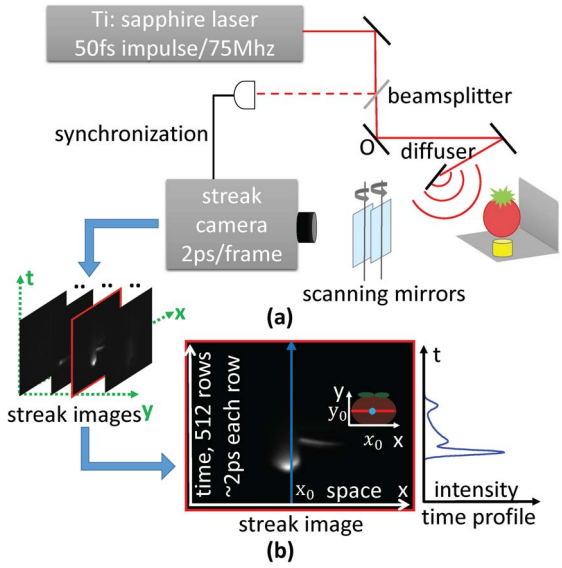

Fig. 1. (a) TLT measurement system. (b) Streak image and time profile.

which illuminates a static scene. The time-resolved scattering is recorded by a streak camera with 2 ps resolution. The one-dimensional camera aperture is scanned with a rotating mirror, and a full (x, y, t) data cube is reconstructed from this stack of streak images. Thus, for a given (x, y) observation point, we record its entire time profile. ConvSCD is based on using these per-pixel time profiles for decomposing LT into its components.

We model the measured time profile $p(t)$ as a convolution:

$$p(t) = c(t) * e(t) * i(t) + n(t), \tag{1}$$

where $*$ denotes convolution, t denotes time, $i(t)$ is the incident light source, $e(t)$ is the light–scene interaction, $c(t)$ is the camera system's point spread function (PSF), and $n(t)$ is additive noise.

Equation (1) is described graphically in Fig. 2. The camera records the time profile at observation point Q. From the illumination, we see that light rays arrive at Q via three main paths: direct reflections (yellow), inter-reflections (green), and sss (blue). The time profile is the sum of all possible paths, so that $e(t)$ comprises these individual effects. Because the pulse duration is less than the camera resolution, we treat direct and inter-reflections as delta-like contributions delayed by the total path length and scaled by an appropriate reflectance amplitude: $e^{(d)}(t) = a_0\delta(t - t_0)$, $e^{(r)} = \sum_{r'=1}^{R} a_{r'}\delta(t - t_{r'})$, where a_0 and $a_{r'}(r' = 1, \ldots, R)$ denote the amplitude coefficients of the reflections, and t_0 and $t_{r'}$ denote the time delays from the source. R is the number of inter-reflections. sss is modeled as a delayed decaying exponential: $e^{(s)}(t) = b\delta(t - t_s) * [\exp(-\alpha t)u(t)]$, where b, t_s, and α are the amplitude, time delay, and attenuation coefficient, respectively, and $u(t)$ is the unit step function. This sss modeling is typical [10], given the exponential decay of intensity with propagation through scattering volumes. These scattering effects are shown in Fig. 2(c). The total light–scene interaction, $e(t)$, is the sum of these effects:

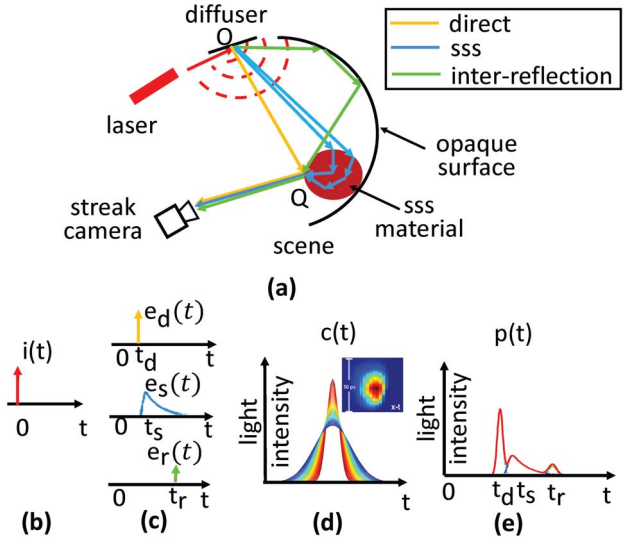

Fig. 2. Forward model of time profile. (a) Schematic of TLT system. (b) Input laser pulse. (c) Scene interaction effects. (d) Time-resolved PSF of the system. (e) Time profile of a pixel Q.

$e(t) = e^{(d)}(t) + e^{(s)}(t) + e^{(r)}(t)$. Under the experimental conditions here, $c(t)$ is Gaussian: $c(t) = \exp[-t^2/(2\sigma_0^2)]$ [10]. (This can be validated by observing the direct reflection of a pulse from a Lambertian object.)

With these terms and $i(t) = \delta(t)$, we can rewrite Eq. (1) as

$$p(t) = \left\{\sum_{r'=0}^{R} a_{r'}\delta(t - t_{r'}) + b\delta(t - t_s) * [e^{-\alpha t}u(t)]*\right\} * e^{-t^2/(2\sigma_0^2)} + n(t). \tag{2}$$

Our goal is to recover all unknown time delays, amplitudes, and the attenuation coefficient from the measured time profile $p(t)$. To solve the problem numerically, we discretize the pixel response: $p(t) \to \mathbf{p} = [\mathrm{p}(T), \mathrm{p}(2T), \ldots, \mathrm{p}(LT)]$, where T is the time resolution of the camera (2 ps), and L is the number of time bins in the time profile. We consider a $G \times 1$ vector $\mathbf{k}_n^{(1)}$, $(\mathbf{k}_n^{(1)})_g = \exp[-(t_g - \mu)^2/(2\sigma_n^2)]$ $(t_g = gT, g = 1, \ldots, G)$, and an $(L - G + 1) \times 1$ vector $\mathbf{z}_n^{(1)}$. G is user-defined such that $G < L$ and μ is a constant to center $\mathbf{k}_n^{(1)}$ in the time window. If $\sigma_n = \sigma_0$, then $\mathbf{k}_n^{(1)} * \mathbf{z}_n^{(1)}$ will reproduce the contributions from all $(R + 1)$ direct and inter-reflection components, where the temporal location of each component corresponds to the locations of the nonzero elements in $\mathbf{z}_n^{(1)}$, and the values of these nonzero elements provide the amplitudes. Generally, $R \ll L - G$, so that $\mathbf{z}_n^{(1)}$ is sparse.

Similarly, we define a vector $\mathbf{k}_m^{(2)}$, $(\mathbf{k}_m^{(2)})_g = \exp[-(t_g - \mu)^2/(2\sigma_{n1}^2)] * [\exp(-\alpha_{n2}t_g)u(t_g)]$, where m is the vectorized index of the pair (n_1, n_2), i.e., $m = (n_1 - 1)N + n_2$. We also define a vector $\mathbf{z}_m^{(2)}$, such that $\mathbf{k}_m^{(2)} * \mathbf{z}_m^{(2)}$ for $\alpha_{n2} = \alpha$ reproduces the sss component in $\mathbf{p}$; $\mathbf{z}_m^{(2)}$ will contain a single nonzero element, whose position corresponds to the time delay for the sss component, and whose value is the reflection amplitude.

We allow varying values of σ_0 (due to nonlinearity in the streak timing) and α by summing over indices m and n to arrive at a discretized time profile $\mathbf{p}$:

$$\mathbf{p} = \sum_{n=1}^{N}(\mathbf{k}_n^{(1)} * \mathbf{z}_n^{(1)}) + \sum_{m=1}^{M}(\mathbf{k}_m^{(2)} * \mathbf{z}_m^{(2)}) + \mathbf{n}. \tag{3}$$

With the measured time profile $\mathbf{p}$ and the known form of $\mathbf{k}_n^{(1)}$ and $\mathbf{k}_m^{(2)}$, we seek to recover $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_m^{(2)}$. By design, the vast majority of elements in $\mathbf{z}_n^{(1)}$ and $\mathbf{z}_m^{(2)}$ are zero, allowing a sparsity-based optimization [12]:

$$\begin{aligned} &\min \sum_n \|\mathbf{z}_n^{(1)}\|_{ls} + \sum_m \|\mathbf{z}_m^{(2)}\|_{ls} \\ &s.t. \quad \mathbf{p} = \sum_n (\mathbf{k}_n^{(1)} * \mathbf{z}_n^{(1)}) + \sum_m (\mathbf{k}_m^{(2)} * \mathbf{z}_m^{(2)}) + \mathbf{n}, \\ &\|\mathbf{n}\|_2 < \varepsilon \end{aligned} \tag{4}$$

where $\| * \|_{ls}$ is the *log-sum* of a vector $(\|\mathbf{v}\|_{ls} = \sum_i \log |v_i|)$, which is an enhanced sparsity metric [13,14], $\| * \|_2$ indicates the l_2 norm, and ε is a small

non-negative constant. We can rewrite the optimization in Eq. (4) [15] as

$$\min \lambda \|\mathbf{p} - \sum_n (\mathbf{k}_n^{(1)} * \mathbf{z}_n^{(1)}) - \sum_m (\mathbf{k}_m^{(2)} * \mathbf{z}_m^{(2)})\|_2 + \sum_n \|\mathbf{z}_n^{(1)}\|_{ls} + \sum_m \|\mathbf{z}_m^{(2)}\|_{ls}, \quad (5)$$

where λ is a parameter that balances the fitting of the time profile (the first term) and the sparsity constraint (the second term). In other words, we seek to minimize an objective function with the constraint that the number of components is as small as possible. Due to the sparsity of LT convolution space in a time-resolved system, the inversion is robust. We can recover each LT component from the nonzero entries in $\mathbf{z}_n^1$ and $\mathbf{z}_m^2$. Here, ConvSCD is solved by the convolutional extension of the coordinate descent algorithm [16].

In our experiment, we choose the range of σ_n to be from 1 to 10 ps, and that of α_m to be from 0.01 to 0.05 ps^{-1}, with 500 equi-spaced samples each. We choose these parameter ranges with the statistical distribution of estimated parameters, over time profiles of the streak camera. Note that we adopt a globally uniform thresholding (at 5% of the maximum nonzero value) of the reconstructed result to exclude negligible nonzero values.

To demonstrate the performance of our algorithm, we first apply our method on noiseless synthetic data, as shown in Figs. 3(a)–3(c). We simulate multiple time profiles with different compositional complexity. Without any additional information, ConvSCD automatically decomposes the time profiles into their components, without any manual iterations.

Generally [17], there are three fundamental types of noises, i.e., photon noise (Poisson), detector noise (pink), and fluctuation noise (white). Noise can cause errors because it can be confused with a separate component, which causes failure of the decomposition. We simulate noise in our data, compare this possible failure condition in ConvSCD and GradBM. As shown in Fig. 3(d), which plots the reconstruction error with the noise-to-signal ratio (NSR), we see that ConvSCD is more robust in the presence of noise compared with GradBM. ConvSCD ensures 90% fidelity for an NSR of 0.055, whereas GradBM requires a 0.005 NSR to achieve the same results.

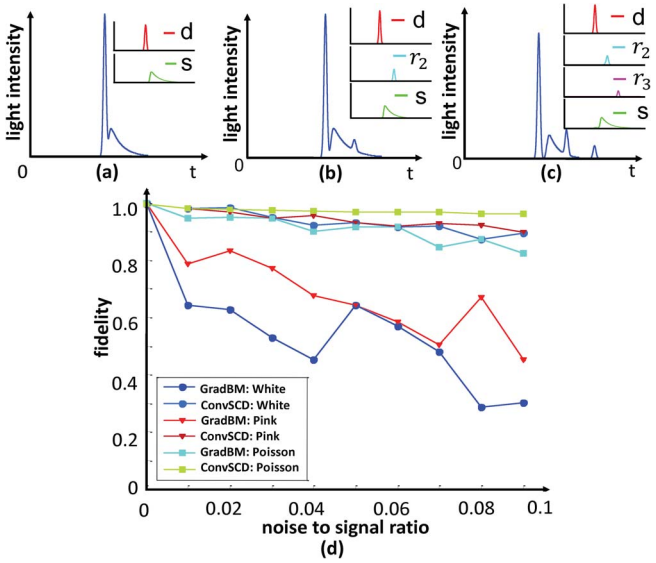

Fig. 3. Decomposition of time profile into its components. (a)–(c) Time profiles for three simulated pixels. Inset: decomposed components. d, s, r_2, and r_3, respectively, denote the decomposed direct component, sss, the second bounce, and the third bounce. (a) Time profile containing only d and s. (b) Time profile containing d, s, and r_2. (c) Time profile containing d, s, r_2, and r_3. (d) Decomposition fidelity compared with GradBM under three types of noise.

We implement ConvSCD on the experimental TLT data set used previously [10] to show that our results show improvement over GradBM [10] when separating direct reflections from sss. Figure 4 shows the TLT decomposition result of our method and the comparison with GradBM for a point containing a direct reflection and sss. In Fig. 4(b), the time profile extracted from point 1 is noisy before the direct component. Our method correctly separates the direct component regardless of noise, whereas GradBM confuses noise with the direct component. From experimental observation, this noise is especially common along depth edges in a scene. Our method, therefore, provides improvement over GradBM for depth discrimination. As can be seen in Figs. 4(c) and 4(d) (Media 1 and Media 2, respectively), we extracted the decomposed frames of the direct and sss components of the whole scene at a fixed time. The comparison shows that the robustness of our method ensures continuity in the whole scene, whereas the decomposed frame in GradBM is more noisy and discontinuous.

We next show the improvement of ConvSCD in separating inter-reflections from sss, as shown in Fig. 5. Figure 5 shows the decomposition result on the TLT of another scene. Due to the holder between the tomato and the floor, light bounces between the bottom of the tomato and the floor. Thus, the inter-reflection components cannot be neglected. As shown in Fig. 5(b), in the time profile of point 2, the two small peaks after the first one are inter-reflections. ConvSCD automatically detects and separates them, whereas GradBM failed to

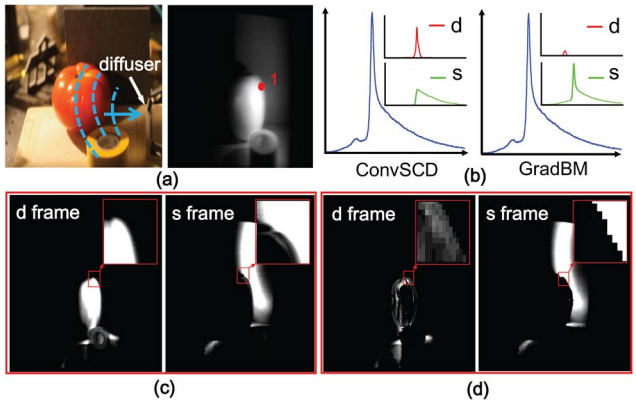

Fig. 4. Decomposition result on real TLT data. (a) Photo of Scene 1 and corresponding TLT data integrated over time. (b) Time profiles of point 1 and corresponding ConvSCD and GradBM decompositions. (c) Decomposed frames of d and s of ConvSCD (Media 1). (d) Decomposed frames of d and s of GradBM (Media 2). (Denotation is the same as that in Fig. 3.)

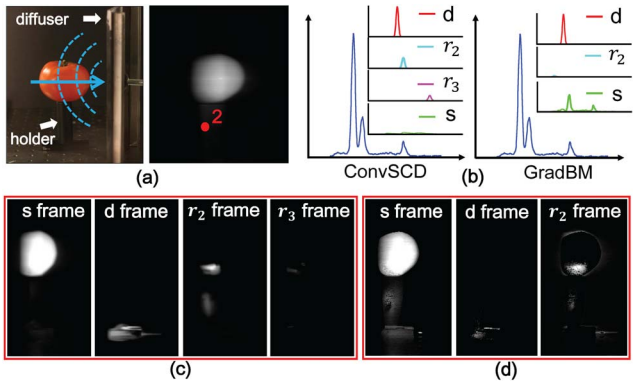

Fig. 5. Decomposition result on real TLT data. (a) Photo of Scene 2 and corresponding TLT data integrated over time. (b) Time profiles of point 2 and corresponding ConvSCD and GradBM decompositions. (c) Decomposed frames of s, d, r_2, and r_3 of ConvSCD (Media 3). (d) Decomposed frames of s, d, and r_2 of GradBM (Media 4). (Denotation is the same as that in Fig. 3.)

separate the second and third bounces from the sss component. Figures 5(c) and 5(d) (Media 3 and Media 4, respectively) show the decomposed result of the whole scene at a fixed time; our method successfully detects and separates the third bounce of the inter-reflection. At the same time, spatial continuity is improved than in the GradBM case, due to the noise robustness of our method.

Future work includes improving the computation time, which we defined as the time needed to decompose a data cube of a given pixel dimension. Because our convolutional sparse coding decomposition method is based on optimization, it is slower than the gradient-based method. GradBM requires approximately 1.5 h, whereas our method requires approximately 4.8 h. All experiments were run under four-thread parallel computation in Matlab with an eight-core Intel Xeon 2.27 GHz processor and 12 GB RAM.

In summary, we exploit the natural convolutional sparsity of TLT to decompose LT via a method called ConvSCD. This algorithm was applied to both simulated and captured TLT data, and the result shows that our method could accurately separate the TLT components, even in the presence of noise and complex geometries. This new comprehensive LT decomposition method is scalable to new modeling of LT component kernels and may be adopted to enable better reconstruction of hidden geometries and material properties.

This work was supported by the project of the National Natural Science Foundation of China (Nos. 61327902, 61035002, and 61120106003), NSF award 1115680, ISU award 6927356, and Charles Stark Draper award SC001-744. We thank D. Wu for valuable discussions.

References

1. S. Colak, D. Papaioannou, G. 't Hooft, M. Van der Mark, H. Schomberg, J. Paasschens, J. Melissen, and N. Van Asten, Appl. Opt. **36**, 180 (1997).
2. S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, ACM Trans. Graph. **25**, 935 (2006).
3. D. Reddy, R. Ramamoorthi, and B. Curless, in *Proceedings of the ECCV* (Springer, 2012), pp. 596–610.
4. S. M. Seitz, Y. Matsushita, and K. N. Kutulakos, in *Proceedings of the ICCV* (IEEE, 2005), pp. 1440–1447.
5. N. Abramson, Opt. Lett. **3**, 121 (1978).
6. D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, C. A. Puliafito, and J. G. Fujimoto, Science **254**, 1178 (1991).
7. J. Busck and H. Heiselberg, Appl. Opt. **43**, 4705 (2004).
8. A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar, Nat. Commun. **3**, 745 (2012).
9. N. Naik, S. Zhao, A. Velten, R. Raskar, and K. Bala, ACM Trans. Graph. **30**, 171 (2011).
10. D. Wu, M. O'Toole, A. Velten, A. Agrawal, and R. Raskar, in *Proceedings of CVPR* (IEEE, 2012), pp. 366–373.
11. M. B. A. Velten and R. Ramesh, *Imaging Systems and Applications* (Optical Society of America, 2011).
12. B. A. Olshausen and D. J. Fieldt, Vis. Res. **37**, 3311 (1997).
13. Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, IEEE Trans. Neural Netw. Learn. Syst. **24**, 383 (2013).
14. E. J. Candes, M. B. Wakin, and S. P. Boyd, J. Fourier Anal. Appl. **14**, 877 (2008).
15. S. S. Chen, D. L. Donoho, and M. A. Saunders, SIAM Rev. **43**, 129 (2001).
16. Y. Li and S. Osher, Inverse Probl. Imaging **3**, 487 (2009).
17. S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction* (Wiley, 2008).