Precomputed Compressive Sensing for Light Transport Acquisition

Satoshi Yamamoto*, Yasumasa Itakura[1], Masashi Sawabe[1], Gimpei Okada[1],
Toshiya Nakaguchi[2], and Norimichi Tsumura[1]

Department of Japanese–Oriental (Kampo) Medicine, Graduate School of Medicine, Chiba University, 1-8-1 Inohana, Chuo-ku, Chiba 260-8670, Japan
[1]*Graduate School of Advanced Integration Science, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan*
[2]*Graduate School of Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan*

In this article, we propose an efficient and accurate compressive-sensing-based method for estimating the light transport characteristics of real-world scenes. Although compressive sensing allows the efficient estimation of a high-dimensional signal with a sparse or near-to-sparse representation from a small number of samples, the computational cost of the compressive sensing in estimating the light transport characteristics is relatively high. Moreover, these methods require a relatively smaller number of images than other techniques although they still need 500–1000 images to estimate an accurate light transport matrix. Precomputed compressive sensing improves the performance of the compressive sensing by providing an appropriate initial state. This improvement is achieved in two steps: 1) pseudo-single-pixel projection by multiline projection and 2) regularized orthogonal matching pursuit (ROMP) with initial signal. With these two steps, we can estimate the light transport characteristics more accurately, much faster, and with a lesser number of images.

Keywords: image-based relighting, compressive sensing

1. Introduction

In computer graphics, it is important to reproduce a real-world scene under arbitrary lighting conditions. The conventional model-based approach accurately reproduces an object scene only if the object scene has no complex light transport effects, such as caustics, inter-reflection, subsurface scattering, and occlusion, which occur depending on the scene. In contrast, an image-based approach — the acquisition of light transport characteristics from an object scene — is suitable for such complex scenes, but requires a vast number of images. For this reason, a number of image-based methods with fewer images have been proposed. These proposed methods include environment matting,[1–3] dual photography,[4] symmetric photography,[5] and the kernel Nyström method.[6] Moreover, methods based on compressive sensing have attracted much attention.[7–9] Compressive sensing is a recently proposed method that recovers a high-dimensional sparse signal from a small number of nonadaptive samples. We

*E-mail address: may-s@umin.net

selected regularized orthogonal matching pursuit (ROMP)[10] because it is a fast and easily implementable compressive sensing method. However, the computational cost of ROMP is relatively high, as each pixel of the reproduced image must be calculated; moreover, it takes 5–7 hours by multithread computation with the CPU Intel^{R} Core^{TM} i7-965 Processor Extreme Edition. Moreover, the optimal reproduction parameters, the number of iterations, and the number of bases added for each iteration differ among scenes (Fig. 1). Smaller parameters are insufficient for studying complex effects, and larger parameters are also not always the best; larger parameters cause noise on simple effects by adding unnecessary non-zero coefficients. In other words, the best reproduction needs to assign the optimal parameters for each pixel separately. However, no such assignment methods have been reported yet, as they are considered to be very complicated and slow. Still, even if possible, diverse iteration is not suitable for parallel processing such as graphics processing unit (GPU) computing, although ROMP is naturally suitable for GPU computing.

In this article, we propose a series of methods termed "precomputed compressive sensing". Precomputed compressive sensing consists of the following two steps: 1) "pseudo-single-pixel projection by multiline projection" for estimating a coarse light transport matrix by projecting multiline patterns, and 2) "ROMP with initial signal", one cycle of ROMP with low parameters to sophisticate the coarse light transport matrix. In summary, the salient points of this article are as follows:

- We propose "pseudo-single-pixel projection by multiline projection" for estimating the coarse light transport matrix, which is a coarse transport matrix but accurate enough for determining an initial signal.
- We propose "precomputed compressive sensing", i.e., coarse light transport estimation and one cycle of ROMP, which allows us to calculate the matrix at low cost with greater accuracy and fewer images.
- Our proposed method is suitable for GPU computing. The conventional method[8] on GPU was approximately 30 times faster than that on CPU (central processing unit), and our proposed method on GPU is even 3 times faster than the conventional one.

2. Related Works

Light transport characteristics — reflectance field — have been defined as the eight-dimensional (8D) function composed of a four-dimensional (4D) incident light field (illumination) and a 4D exitant light field (view). Each 4D is composed of two-dimensional (2D) coordinates and 2D directions. Since this 8D function is too large to measure, many research studies are focusing on a slice of this 8D function by various methods. For example, brute-force sampling is the simplest method of 4D reflectance field acquisition, which entails capturing images and changing the light direction, one by one. Debevec *et al.* measured the 4D reflectance field from a fixed viewpoint under directional lighting condition with a light stage.[11] Wenger *et al.* extended this method to reproduce moving objects.[12] These methods are not suitable for reproducing high-frequency light transport effects owing to low sampling of the light sources. Hawkins *et al.* resolved this problem using Helmholtz reciprocity.[13] These methods, based on brute-force sampling, are simple, but require many images, which takes much time and a very large amount of data. Therefore, certain methods have been focusing on reducing the number of images.

The simplest reduction method is overlapping. Sen *et al.* measured the 4D reflectance field from a fixed viewpoint under dense sampling lighting from one direction.[4] They set as many lights as possible, being careful not to allow them to affect each other. Then the optimal overlapping pattern was decided by illuminating a multiscaled block pattern adapted to each scene. Masselus *et al.* also proposed an overlapping-based method of measuring a six-dimensional (6D) reflectance field without adaptation.[14] They projected horizontal and vertical block array images with respective blocks independent of each other and without interaction between them. In general, an adaptive approach requires complex implementation because measurement and calculation are performed simultaneously. Our proposed method is a nonadaptive one in both steps.

Other methods assumed the reflectance field as a parametric model function. Zongker *et al.* assumed the reflectance field as a block function.[3] Chuang *et al.* extended this method by replacing the box function with the oriented Gaussian function.[1] Matusik *et al.* proposed a weighted sum of the rectangular function to relight the natural scene under arbitrary lighting condition.[15] As these methods require nonlinear optimization for each pixel of the image, the computational cost is very high. Although our proposed method also recovers signals pixel by pixel, it is much faster since the calculation cost of the pixel is lower and it is suitable for GPU computing.

Wang *et al.* developed the Nyström method to estimate a low-rank matrix from a small number of column vectors and row vectors.[6] This method can measure the reflectance field with a small number of images (a few hundred). However, the measurement device needs a diffuser between the light emitter and the object scene, making it impossible to reproduce the object scene under high-frequency lighting. Our method also requires no more than a few hundred images, and no special device is required other than a projector and a camera.

Peers *et al.*, Gu *et al.*, and Sen and Darabi applied compressive sensing to assume the reflectance field. Peers *et al.* developed a novel hierarchical algorithm based on the correlation among image pixels.[7] However, their algorithm is complex and the calculation cost is high. Gu *et al.* recovered inhomogeneous participating media with compressive structured light.[16] In their method, inhomogeneous participating media such as milk drip in water were recovered properly, but this method is limited to monochrome media. Sen and Darabi developed a modified ROMP method by limiting the number of coefficients.[8)] In this method, all pixels are computed with the same parameters and the same number of iterations; however, this method is not suitable for a scene with diverse non-zero coefficients, as shown in Fig. 1. Although our proposed method uses a similar strategy to Sen and Darabi's method, our method can process each pixel with various parameters according to the light transport effect. Our proposed method is simple and accurate despite the fact that it does not consider pixel correlation.

3. Proposed Method

The schematic flow is shown in Fig. 2.

In this method, we assumed the use of one fixed projector and one fixed camera, restricting the direction of incident/exitant light, and then the reflectance field is determined as 4D function — 2D for the incident light and the other 2D for the exitant light.

The projector resolution is denoted by $p \times q$ for emitting a dense sampling lighting pattern, and the camera resolution $p' \times q'$ for capturing images of the object. Then the relationship between the camera image and the projector image is expressed as

$$\boldsymbol{c} = \mathbf{T}\boldsymbol{l}, \tag{1}$$

where $\boldsymbol{l}$ denotes a column vector of $p \times q$ length, representing a lighting pattern, $\boldsymbol{c}$ denotes a column vector of $p' \times q'$ length, representing a captured image corresponding to each lighting pattern, and $\mathbf{T}$ is a matrix representing the light transport between the camera image and the projector image. The matrix $\mathbf{T}$ is called the "light transport matrix", and our goal is to acquire $\mathbf{T}$ accurately with a low calculation time and as small a number of images as possible. An accurate $\mathbf{T}$ allows us to reproduce the object scene image under arbitrary lighting conditions more accurately. Herein, each element t_{ij} of the matrix $\mathbf{T}$ represents the light transport coefficient from the j-th projector pixel to the i-th camera pixel. The i-th row vector of the matrix $\mathbf{T}$ is called the "reflectance function" of the i-th camera pixel. The j-th column vector of the matrix $\mathbf{T}$ represents the image that is illuminated in the object scene only by the j-th pixel of the projector.

Below we present the first step — pseudo-single-pixel projection by multiline projection — in §3.1, and the concept of the second step — ROMP with initial signal — in §3.2. We demonstrate a detailed method of recovering the light transport matrix in §3.3.

3.1 Pseudo-single-pixel projection by multiline projection

In this section, we present a method of acquiring a coarse light transport matrix by projecting multiline patterns. The schematic flow is shown in Fig. 3. A rough matrix was recovered from a smaller number of images and directed to the next step, ROMP. First, we illuminated the horizontal and vertical line strip patterns with a width of w pixels. To reduce the image counts, we projected multiple lines simultaneously. The acquired images were then binarized and clustered by k-mean clustering, and each cluster was correlated with each line by its sequence. As all the projector pixels within a row and a column are assumed to be independent of each other, pseudo-single-pixel images are approximated depending on the scene's geometric and material properties. When a single line was projected, each pixel (x, y) of each pseudo-single-pixel image $A_{m,n}$ is calculated as

$$A_{m,n}(x,y) = \sqrt{H_m(x,y) \cdot V_n(x,y)}$$
$$(H_m^{bin}(x,y) = 1 \ \wedge \ V_n^{bin}(x,y) = 1)$$
$$\tag{2}$$
$$A_{m,n}(x,y) = 0$$
$$(H_m^{bin}(x,y) = 0 \ \vee \ V_n^{bin}(x,y) = 0),$$

where H_m denotes the image illuminated with an m-th horizontal line strip pattern, and H_m^{bin} is its binarized image. V_n denotes the image illuminated with an n-th vertical line strip pattern, and V_n^{bin} is its binary image.

Then we extended this to multiple lines. Assuming each clustered line to be a single-line image, we separated the cross-sectional image into respective pseudo-single-pixel illuminated images. When 2 lines are projected simultaneously, this process reduces the number of images to 25%, and to as low as 11% when 3 lines are projected. Generally, it requires $(p + q)/zw$ images, where p and q are the projector resolution $p \times q$, z is the number of lines simultaneously projected, and w is the width of a projected line. Although this pseudo-pixel estimation contains estimation errors due to wrong clustering and different complexities of scenes in a pseudo-pixel, such errors are corrected by the following ROMP process. That is, the "crude" estimation is more than sufficient. Then, the coarse light transport matrix $\mathbf{T}^{ini}$ was constructed as follows: 2D pseudo-single illumination images were transformed into one-dimensional (1D) column vectors, and set to the corresponding points of $\mathbf{T}^{ini}$, where the single-pixel illumination pattern is expected to be. In other words, we stacked duplicates of each 1D column vector $w \times w$ times into $\mathbf{T}^{ini}$. In the next step, $\boldsymbol{t}^{ini}$, each row vector of $\mathbf{T}^{ini}$, was employed as a default signal. A relighting scene with the coarse transport matrix is shown in Fig. 4

3.2 ROMP with initial signal

In this section, we explain the theory of compressive sensing. First, to recover the sparse signal $\boldsymbol{v}$ from $\boldsymbol{x}$, we employ

$$\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{v}, \tag{3}$$

where $\boldsymbol{v}$ denotes a column vector of n length, $\boldsymbol{x}$ denotes a column vector of m length, and $\boldsymbol{\Phi}$ is an $m \times n$ matrix, called the sensing matrix. We can easily recover the signal $\boldsymbol{v}$ accurately, if only $m \geq n$. On the other hand, in the case of $m \ll n$, compressive sensing is utilized. To perform compressive sensing, $\mathbf{\Phi}$ has to meet the restricted isometric condition (RIC):[10)]

$$(1-\varepsilon)\left\|\boldsymbol{v}\right\|_2 \leq \left\|\mathbf{\Phi}\boldsymbol{v}\right\|_2 \leq (1+\varepsilon)\left\|\boldsymbol{v}\right\|_2, \tag{4}$$

with the parameters (ε, δ), where $\varepsilon \in (0,1)$ for all δ-sparse vectors $\boldsymbol{v}$. RIC enables any set of δ column vectors of matrix $\mathbf{\Phi}$ to form an approximate orthogonal set. To meet RIC, certain methods have been proved useful, namely, the use of the Gaussian, binary, and Fourier ensembles.[17)] In this article, we employed the binary ensemble, in which the Bernoulli matrices are used for the sensing matrix, because in this kind of projector-camera system, black and white images such as Bernoulli patterns are suitable because such images have the highest contrast; the Bernoulli matrix is composed of randomly selected 1 and -1, with a mean of 0.

There are two strategies for signal recovery — $L1$ minimization and greedy pursuit. They have trade-offs: the former has good performance with high computational cost and the latter has relatively poor performance and low cost. We selected the latter, i.e., greedy pursuit, for the fast estimation of $\mathbf{T}$. In this greedy-pursuit-based compressive sensing, there are a number of methods; we selected ROMP, which is especially suitable for fast estimation.[10)] The algorithms of modified ROMP by Sen and Darabi[8)] and our second step are shown in Fig. 5, where it and b are the parameters that we determine when we start the calculation.

ROMP consists mainly of 3 steps, i.e., identification, regularization, and optimization. In the identification and regularization steps, ROMP searches contributing bases for the output signal. This is the same as finding a non-zero coefficient index of the recovering signal. This process requires large matrix-vector multiplication and the sorting of vector values. These processes are performed for each iteration and are therefore time-consuming. Additionally, in the simple lighting scene, the performance to find correct contributing bases deteriorates with increasing numbers of unnecessary non-zero coefficients in the recovering signal. Once the incorrect bases are added to the index list, ROMP cannot recover the exact signal because incorrect bases are not removed at the following iteration; ROMP is a non-zero addition algorithm and not a deletion algorithm. Our proposed method of precomputed compressive sensing — precomputed ROMP — is shown in Fig. 5(b). The conjugate gradient method was used in the optimization step.

Our method pre-optimizes and adds just one loop of ROMP for the initial signal — the coarse light transport matrix. Here, $\boldsymbol{I}^{ini}$ and $\boldsymbol{v}^{ini}$ were induced from $\boldsymbol{t}^{ini}$ as default signals, where $\boldsymbol{I}^{ini}$ denotes the possession of non-zeros in $\boldsymbol{t}^{ini}$, and $\boldsymbol{v}^{ini}$ denotes the number of non-zeros in $\boldsymbol{t}^{ini}$. With this initialization, the respective pixels have diverse non-zero coefficients. In other words, this method allows us to acquire the optimal parameters for each pixel, which is stated as impossible in §2. Thus, pre-optimization and only one cycle of ROMP with small parameters are necessary for sophistication. In Fig. 1, we showed that extra non-zeros induced noise on the simple effect, but that the addition of small non-zeros in this process did not cause any visible change.

The number of bases to be added b was set at 5 empirically, which is so called “small”. Actually, additional ROMP is not necessary for all the pixels. The first pre-optimization is sufficient in most cases even in complex and simple scenes, but it is necessary for the edges of objects (Fig. 6). This is because of the very small non-zeros on the edges where normal is nearly orthogonal to the light direction. Additional ROMP effectively eliminated this noise. As those parts cannot have many non-zeros, additional 1 cycle and 5 non-zeros were more than sufficient.

3.3 Light transport matrix recovery

In this section, we introduce a detailed method of light transport recovery. This process is similar to that discussed by Sen and Darabi,[8] but is included here for completeness. Now, returning to eq. (1), which is extended for plural images as follows:

$$\begin{gathered}\mathbf{C} = \mathbf{TL} \\ (\mathbf{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k]\,, \quad \mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_k]),\end{gathered} \tag{5}$$

where k denotes the number of images, $\boldsymbol{c}_k$ denotes the column vector representing captured images under each illumination pattern, and $\boldsymbol{l}_k$ denotes the column vector representing an illumination pattern. Transposing the two sides of eq. (3), we get $\mathbf{C}^T = \mathbf{L}^T\mathbf{T}^T$. The i-th camera pixel values from all the acquired images $\boldsymbol{c}_i$ are written as

$$\boldsymbol{c}_i = \mathbf{L}^T \boldsymbol{t}_i, \tag{6}$$

where $\boldsymbol{t}_i$ is the row vector of the matrix $\mathbf{T}$. Equation (6) is similar to eq. (3), and thus we reconstructed $\boldsymbol{t}_i$ from the 3 vectors $\boldsymbol{c}_i$, $\mathbf{L}$, and $\boldsymbol{t}^{ini}$ with the compressive sensing theory, because it fits the condition for compressive sensing, where $\boldsymbol{t}_i$ is sufficiently sparse. When $\boldsymbol{t}_i$ is not sufficiently sparse, compression with appropriate bases such as a wavelet is necessary. We confirmed that $\boldsymbol{t}_i$ is recovered adequately without compression in this experiment.

3.4 Implementation on GPU

Our proposed method and ROMP are suitable for parallel processing because the recoveries of the respective $\boldsymbol{t}_i$ are independent of each other. We used NVIDIA CUDA 2.2 SDK and implemented three steps of ROMP, i.e., identification, regularization and optimization, as kernels. First, we loaded all matrices on the main memory and then a partial matrix representing 1024 reflectance functions was copied to the global memory of GPU. Data on the global memory were processed by multi-processors of GPU in parallel, and results on the global memory were copied to the main memory (Fig. 7).

4. Experiments and Results

Our setup is built inside a darkroom, as shown in Fig. 8. We used an XGA DLP projector Mitsubishi LVP-XD205R for illumination and a single-lens reflex digital camera Nikon D3x for image acquisition. To illuminate the -1 value of the Bernoulli pattern, we illuminated the positive and negative components separately and subtracted the negative image from the positive one. Three images were taken for each scene at different exposures; each image had a 480×640 resolution, a 3-color channel, and 8 bits for each channel. Then, the images were merged into a high-dynamic-range (HDR) image. A GPU NVIDIA Tesla$^{\mathrm{TM}}$ C1060 and a CPU Intel$^{\mathrm{R}}$ Core$^{\mathrm{TM}}$ i7-965 Processor Extreme Edition were employed for computation.

Our proposed method was compared with the conventional method by Sen and Darabi.[8)] Fig. 9 shows the recovery result with the fewest images. Each image is recovered using 128 images; 64 Bernoulli patterns and 64 line strip patterns are used for our proposed method, and 128 Bernoulli patterns are used for the conventional method. Our proposed method shows better recovery when the smallest number of images is used. The details are shown in Fig. 11, which present the recovery result with various types of scenes — subsurface scattering, complex object, transparent object, and inter-reflection. We set the projector resolution to 128×128. The projection line width was set to 2 pixels, and 2 lines were emitted simultaneously. We set these numbers empirically since the optimal parameters vary between scenes. In short, 64 images were taken in the first step. To refine the initial light transport matrix, we used 512, 256, 128, 64 Bernoulli patterns for ROMP. Additionally, Fig. 10 shows the relighting result under various lighting conditions. Our proposed method was applied to other scenes with indirect illumination, simulating such conditions as those in a movie theater (Fig. 12). In these scenes, a model-based approach is common; however, an image-based approach shows better simulation on a slightly irregular surface of a dish and a complex-shaped glass. Our proposed method can be extended to display-camera systems by displaying the same patterns on TV, simulating a room where a TV is left on.

Computational cost was measured with both CPU and GPU, as ROMP is supposed to fit with GPU computing. The conventional method was implemented on CPU first with C^{++}, and then was reimplemented on GPU with CUDA. As the conventional method ran very fast on GPU, the proposed method was implemented on GPU. Table I shows the calculation time for each method. The conventional method on GPU was approximately 30 times faster than that on CPU, and the proposed method on GPU was even 3 times faster than that.

5. Limitations

First, the clustering of multiline images is not applicable to some geometry, because the line strips in the acquired image are distributed over various locations in the image. In such a case, we have to emit a single line strip, and the number of required images increases. Second, although the proposed method works well for subsurface scattering and inter-reflection, it does not work well for transparent objects such as glass when relatively fewer images are used, owing to the many non-zero coefficients especially on edges of transparent objects. Third, dot noise occurs when vivid patterns are projected. This noise is observed in the same pixels on the image sensor, but not on the real image by the same image sensor. Our scene reproduction first calculates a matrix when each pixel of the projector illuminated the scene, and integrates them for arbitrary illumination. Therefore, a small difference in sensitivity among pixels might have been accumulated despite the calibration of the image sensor.

6. Conclusions and Future Work

In this article, we propose a precomputed compressive sensing method of measuring the light transport characteristics of a real-world scene. To obtain the initial light transport matrix, we propose a method of acquiring coarse light transport characteristics using a few hundred images at a low calculation cost. Moreover, we applied a procedure with GPU computing to shorten the calculation time. Our method is nonadaptive and superior to the conventional compressive-sensing-based method in image number and calculation time. For our future study, we are planning to develop a method of acquiring coarse light transport characteristics with even fewer images and to reconstruct the reflectance function. Improvement in the presently proposed method will be necessary to recover an appropriate basic level for the succeeding compressive sensing. Furthermore, to reproduce a higher dimension of light transport characteristics, we are currently trying another slice of 8D function using the proposed method.

References

1) Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski: Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), 2000, p. 121.

2) P. Peers and P. Dutré: Proc. 14th Eurographics Workshop on Rendering (EGRW '03), 2003, p. 157.

3) D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin: Proc. 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99), 1999, p. 205.

4) P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. Lensch: ACM Trans. Graph. **24** (2005) 745.

5) G. Garg, E.-V. Talvala, M. Levoy, and H. P. A. Lensch: Proc. Eurographics Symp. on Rendering, 2006, p. 251.

6) J. Wang, Y. Dong, X. Tong, Z. Lin, and B. Guo: ACM Trans. Graph. **28** (2009) No. 29, 1.

7) P. Peers, D. K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec: ACM Trans. Graph. **28** (2009) No. 3, 1.

8) P. Sen and S. Darabi: Comput. Graphics Forum **28** (2009) 609.

9) P. Sen and S. Darabi: IEEE Trans. Vis. Comput. Graphics **17** (2011) 487.

10) D. Needell and R. Vershynin: Found. Comput. Math. **9** (2009) 317.

11) P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar: Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), 2000, p. 145.

12) A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec: ACM Trans. Graph. **24** (2005) 756.

13) T. Hawkins, P. Einarsson, and P. Debevec: Proc. Eurographics Symp. on Rendering, 2005, p. 91.

14) V. Masselus, P. Peers, P. Dutré, and Y. D. Willems: ACM Trans. Graph. **22** (2003) 613.

15) W. Matusik, M. Loper, and H. Pfister: Proc. Eurographics Symp. on Rendering, 2004, p. 299.

16) J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi: Proc. 10th European Conference on Computer Vision (ECCV 2008), 2008, p. 845.

17) E. Candès and T. Tao: IEEE Trans. Inf. Theory **52** (2006) 5406.

Figure captions

Fig.1.

(Color online) Relighting with conventional ROMP. The top image represents the object scene. Each boxed scene was magnified; red: the scene with subsurface scattering, blue: with inter-reflection, green: single reflection with no non-zero coefficient. Ground truth and reproduced images are shown in the respective columns; left: ground truth, real-scene image, center: image reproduced by ROMP with large parameter ($b = 10, it = 10$), right: image reproduced by ROMP with small parameter ($b = 3, it = 3$). The parameters are shown in Fig. 5.

Fig.2.

(Color online) Flow of proposed method “precomputed complex sensing”.

Fig.3.

(Color online) Flow of pseudo-single-pixel projection by multiline projection. The coarse light transport matrix was computed by multiline projection, clustering, and pseudo-single-pixel elimination.

Fig.4.

(Color online) Relighting scene with coarse light transport matrix. Red: magnified image. Blue: corresponding position of ground truth, real-scene image

Fig.5.

Modified ROMP and proposed method. b and it are the optimization parameters. We set $b = 5$, empirically for the proposed method. The conjugate gradient method was used in the optimization step.

Fig.6.

(Color online) Effects of additional ROMP cycle. (a) Proposed method with additional ROMP, and (b) without additional ROMP. Noises are apparent on the edge after pre-optimization (b) but eliminated by additional ROMP (a).

Fig.7.

(Color online) Proposed method with GPU computing.

Fig.8.

(Color online) Photograph of measurement setup. Our setup is built in a darkroom, and the scene is set in a box lined with black cloth. A DLP projector illuminates the respective patterns, and a camera records the illuminated scene.

Fig.9.

(Color online) Reproduction result with fewest images. Each image is recovered using 128 images. (a) Our proposed method recovered with 64 Bernoulli patterns and 64 line strip patterns, and (b) conventional method recovered with 128 Bernoulli patterns.

Fig.10.

(Color online) Relighting result of the proposed method under arbitrary lighting condition.

Fig.11.

(Color online) Reproduction result with various scenes. (a) Subsurface scattering, (b) complex-shape object, (c) transparent object, and (d) inter-reflection. The conventional method is of Sen and Darabi.[8] In each scene, the top-left image shows the ground truth; from the left, first column images are reproduction results obtained by conventional method ($b = 10, it = 10$), second column images are those obtained by conventional method ($b = 3, it = 3$), third column images are those obtained by proposed method. Each row represents the number of images used for estimation: 512, 256, 128, and 64 Bernoulli patterns from the top. Additional 64 images were used for the proposed method, in the first step.

Fig.12.

(Color online) Reproduction result with indirect illumination. (a) Scene with specular reflection (china dish) and (b) scene with transmitted light (glass). Irregular surface of the dish and complex shape of the glass are magnified in the center.

Table

Table I. Calculation time (min). The calculation time under each condition is shown. The numbers above show the numbers of Bernoulli patterns used for calculation. The maximal non-zero coefficient was set to 50 for comparison.

		Number of Bernoulli patterns			
		1024	512	256	128
Proposed	(GPU)	13.66	7.79	5.84	5.36
Conventional	(GPU)	29.82	21.33	18.53	18.22
Conventional	(CPU)	1394	721.5	385.0	212.9

Conventional method

We set the parameters for $b = 10$ and $it = 5$, which sets the maximum number of non-zero coefficients to 50. This method is of Sen and Darabi.[8)]

Proposed method

We used 50 non-zero coefficients for all initial reflectance functions, which sets the maximal coefficient to 55. Additional 64 images were used the proposed method (*e.g.*, 1024 + 64 = 1088 images, for the first column)

Figures

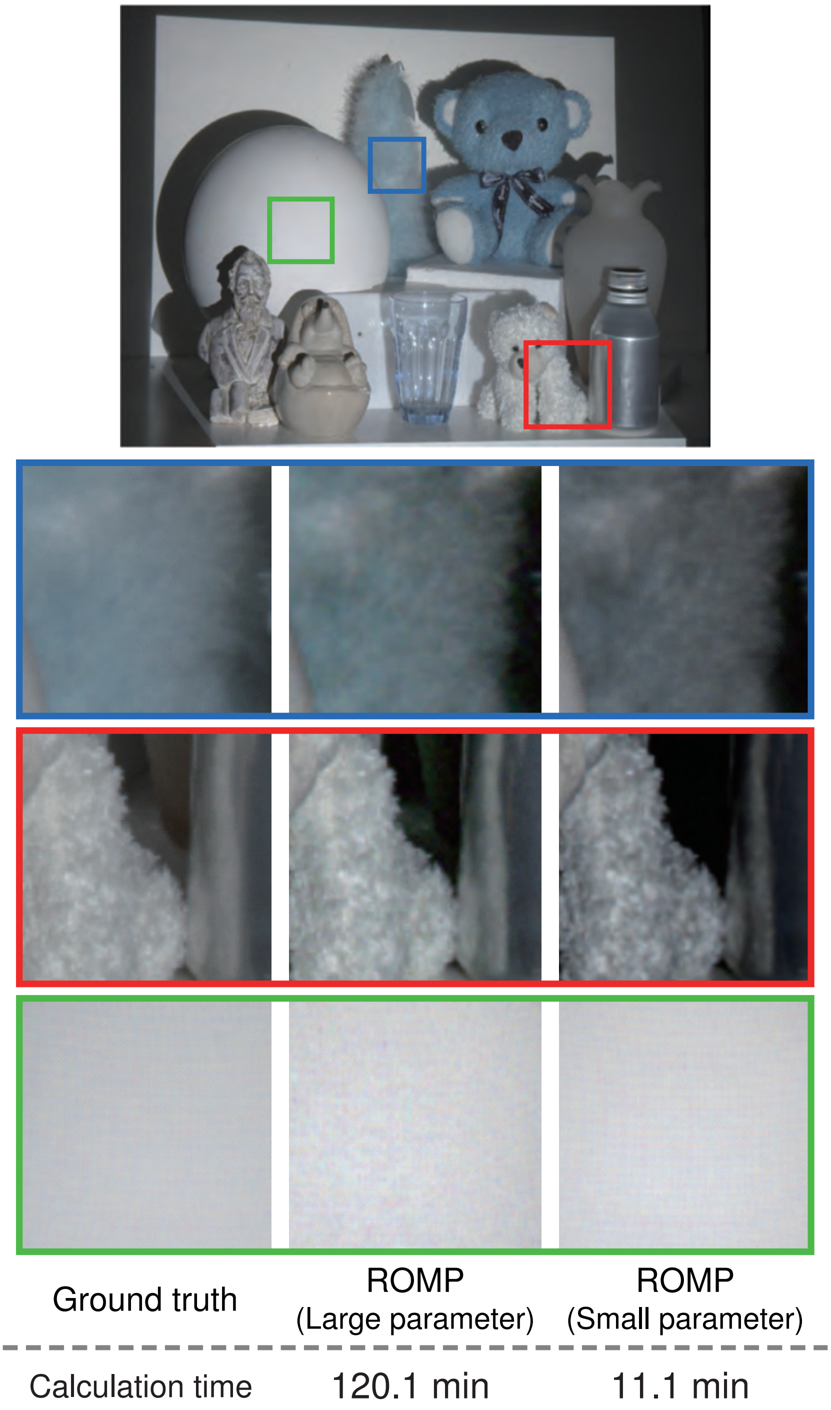


Fig. 1.

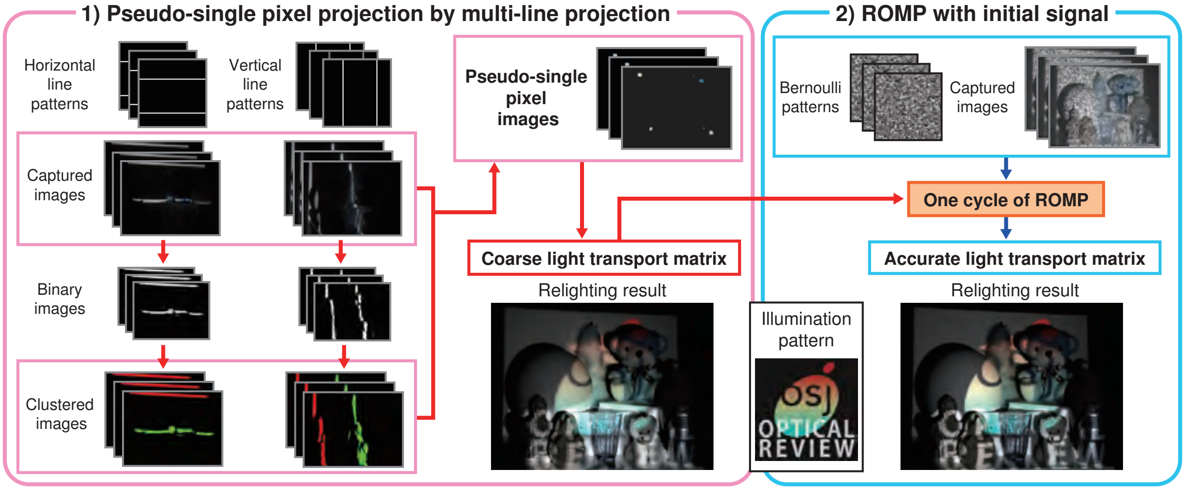

1) Pseudo-single pixel projection by multi-line projection
Horizontal line patterns
Vertical line patterns
Captured images
Binary images
Clustered images
Pseudo-single pixel images
Coarse light transport matrix
Relighting result
2) ROMP with initial signal
Bernoulli patterns
Captured images
One cycle of ROMP
Accurate light transport matrix
Relighting result
Illumination pattern
OSJ
OPTICAL REVIEW

Fig. 2.

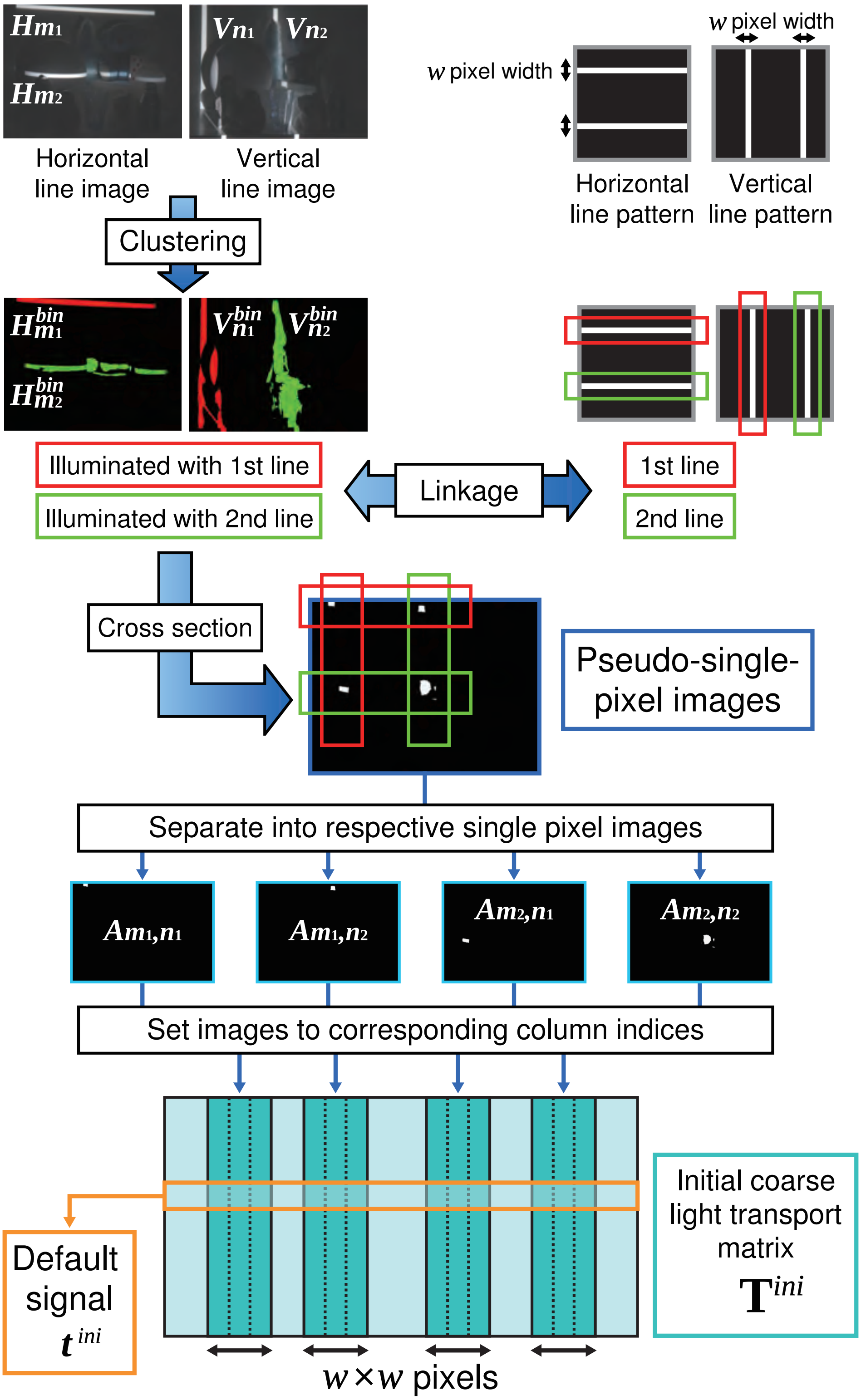

Hm_1
Hm_2
Vn_1
Vn_2
Horizontal line image
Vertical line image
w pixel width
w pixel width
Horizontal line pattern
Vertical line pattern
Clustering
$H^{bin}_{m_1}$
$H^{bin}_{m_2}$
$V^{bin}_{n_1}$
$V^{bin}_{n_2}$
Illuminated with 1st line
Illuminated with 2nd line
Linkage
1st line
2nd line
Cross section
Pseudo-single-pixel images
Separate into respective single pixel images
A_{m_1,n_1}
A_{m_1,n_2}
A_{m_2,n_1}
A_{m_2,n_2}
Set images to corresponding column indices
Default signal t^{ini}
Initial coarse light transport matrix $\mathbf{T}^{ini}$
w×w pixels

Fig. 3.

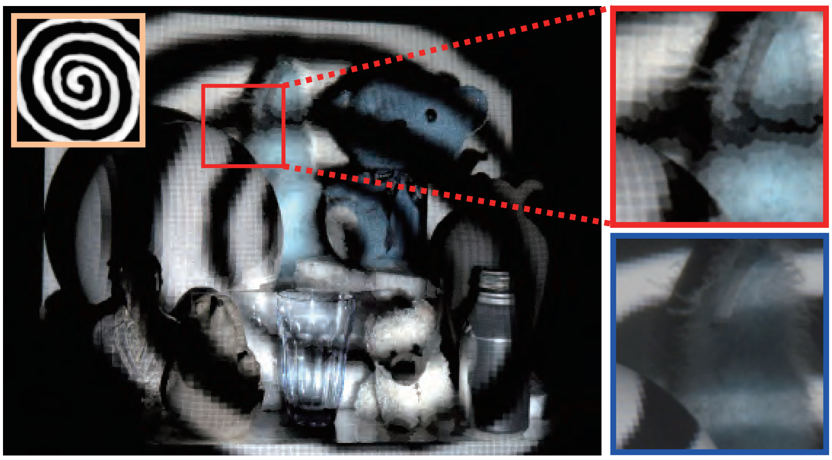
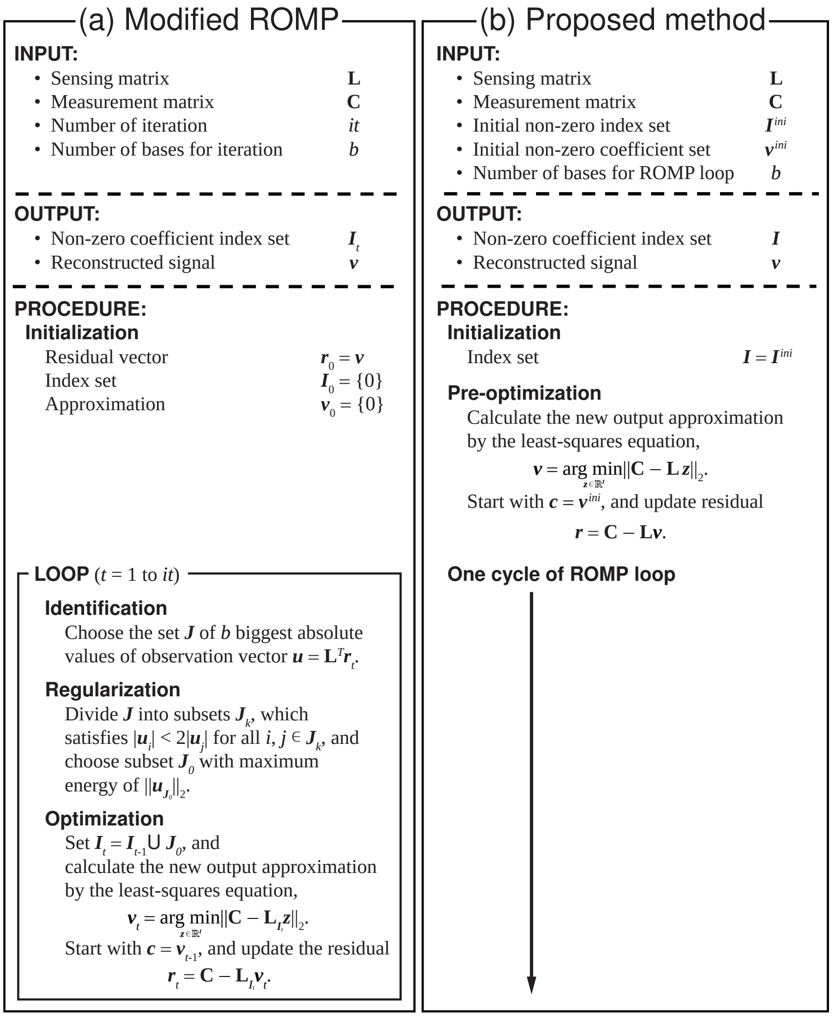


Fig. 4.



(a) Modified ROMP

INPUT:

- Sensing matrix $\mathbf{L}$
- Measurement matrix $\mathbf{C}$
- Number of iteration it
- Number of bases for iteration b

OUTPUT:

- Non-zero coefficient index set $\boldsymbol{I}_t$
- Reconstructed signal $\boldsymbol{v}$

PROCEDURE:

Initialization

Residual vector $\boldsymbol{r}_0 = \boldsymbol{v}$

Index set $\boldsymbol{I}_0 = \{0\}$

Approximation $\boldsymbol{v}_0 = \{0\}$

LOOP (t = 1 to it)

Identification

Choose the set $\boldsymbol{J}$ of b biggest absolute values of observation vector $\boldsymbol{u} = \mathbf{L}^T\boldsymbol{r}_t$.

Regularization

Divide $\boldsymbol{J}$ into subsets $\boldsymbol{J}_k$, which satisfies $|\boldsymbol{u}_i| < 2|\boldsymbol{u}_j|$ for all $i, j \in \boldsymbol{J}_k$, and choose subset $\boldsymbol{J}_0$ with maximum energy of $\|\boldsymbol{u}_{J_0}\|_2$.

Optimization

Set $\boldsymbol{I}_t = \boldsymbol{I}_{t\text{-}1} \cup \boldsymbol{J}_0$, and calculate the new output approximation by the least-squares equation,

$$\boldsymbol{v}_t = \arg\min_{\boldsymbol{z}\in\mathbb{R}^I} \|\mathbf{C} - \mathbf{L}_{I_t}\boldsymbol{z}\|_2.$$

Start with $\boldsymbol{c} = \boldsymbol{v}_{t\text{-}1}$, and update the residual

$$\boldsymbol{r}_t = \mathbf{C} - \mathbf{L}_{I_t}\boldsymbol{v}_t.$$

(b) Proposed method

INPUT:

- Sensing matrix $\mathbf{L}$
- Measurement matrix $\mathbf{C}$
- Initial non-zero index set $\boldsymbol{I}^{ini}$
- Initial non-zero coefficient set $\boldsymbol{v}^{ini}$
- Number of bases for ROMP loop b

OUTPUT:

- Non-zero coefficient index set $\boldsymbol{I}$
- Reconstructed signal $\boldsymbol{v}$

PROCEDURE:

Initialization

Index set $\boldsymbol{I} = \boldsymbol{I}^{ini}$

Pre-optimization

Calculate the new output approximation by the least-squares equation,

$$\boldsymbol{v} = \arg\min_{\boldsymbol{z}\in\mathbb{R}^I} \|\mathbf{C} - \mathbf{L}\,\boldsymbol{z}\|_2.$$

Start with $\boldsymbol{c} = \boldsymbol{v}^{ini}$, and update residual

$$\boldsymbol{r} = \mathbf{C} - \mathbf{L}\boldsymbol{v}.$$

One cycle of ROMP loop

Fig. 5.

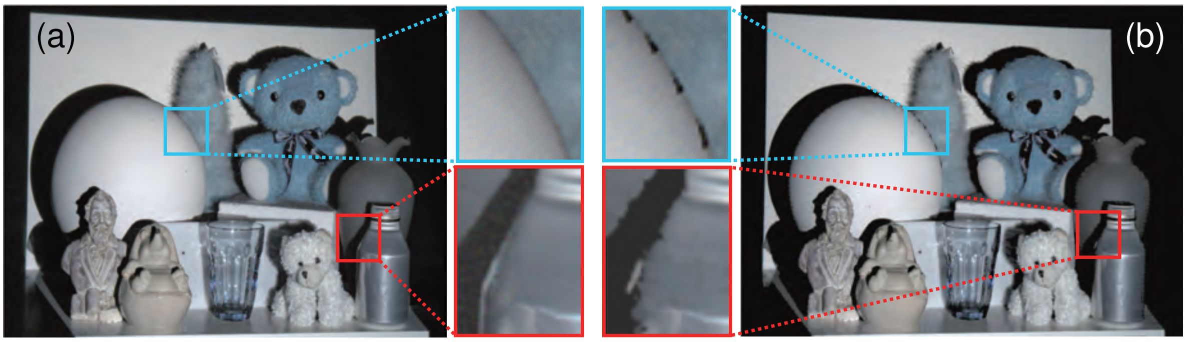

(a)
(b)

Fig. 6.

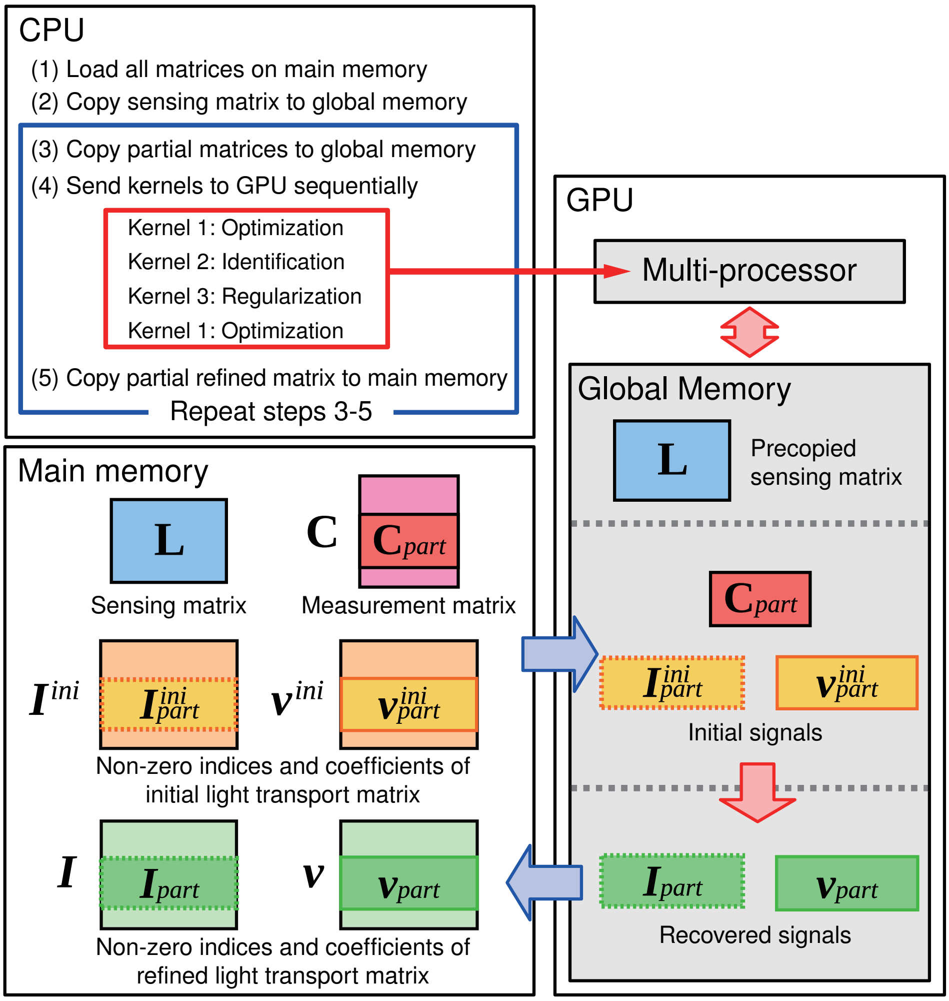

CPU
(1) Load all matrices on main memory
(2) Copy sensing matrix to global memory
(3) Copy partial matrices to global memory
(4) Send kernels to GPU sequentially
Kernel 1: Optimization
Kernel 2: Identification
Kernel 3: Regularization
Kernel 1: Optimization
(5) Copy partial refined matrix to main memory
Repeat steps 3-5
GPU
Multi-processor
Global Memory
L
Precopied sensing matrix
$\mathbf{C}_{part}$
$\mathbf{I}^{ini}_{part}$
$\mathbf{v}^{ini}_{part}$
Initial signals
$\mathbf{I}_{part}$
$\mathbf{v}_{part}$
Recovered signals
Main memory
L
Sensing matrix
C
$\mathbf{C}_{part}$
Measurement matrix
$\mathbf{I}^{ini}$
$\mathbf{I}^{ini}_{part}$
$\mathbf{v}^{ini}$
$\mathbf{v}^{ini}_{part}$
Non-zero indices and coefficients of initial light transport matrix
$\mathbf{I}$
$\mathbf{I}_{part}$
$\mathbf{v}$
$\mathbf{v}_{part}$
Non-zero indices and coefficients of refined light transport matrix

Fig. 7.

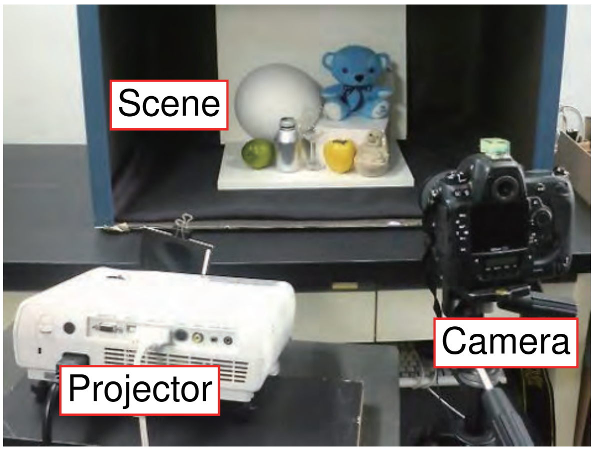


Fig. 8.

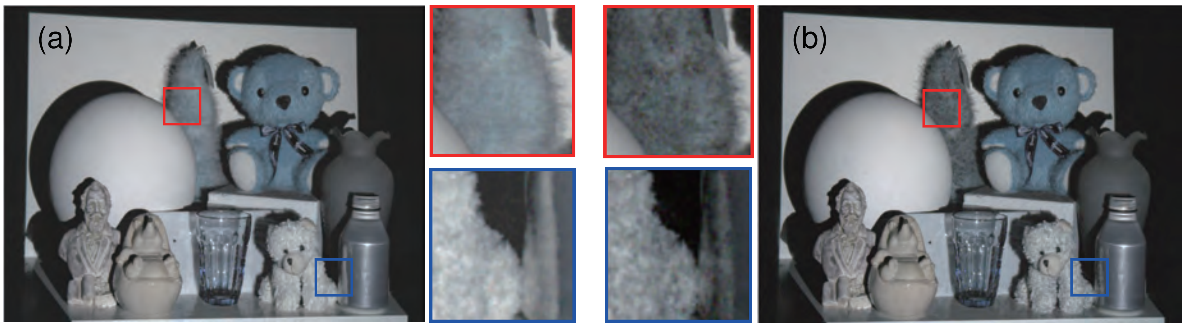


Fig. 9.

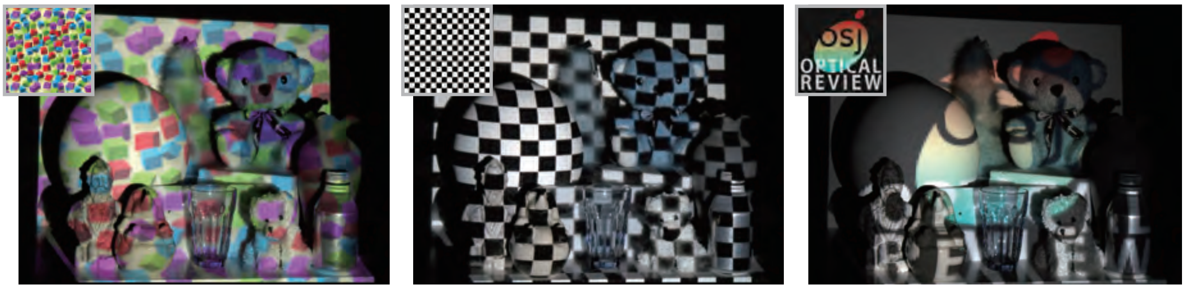


Fig. 10.

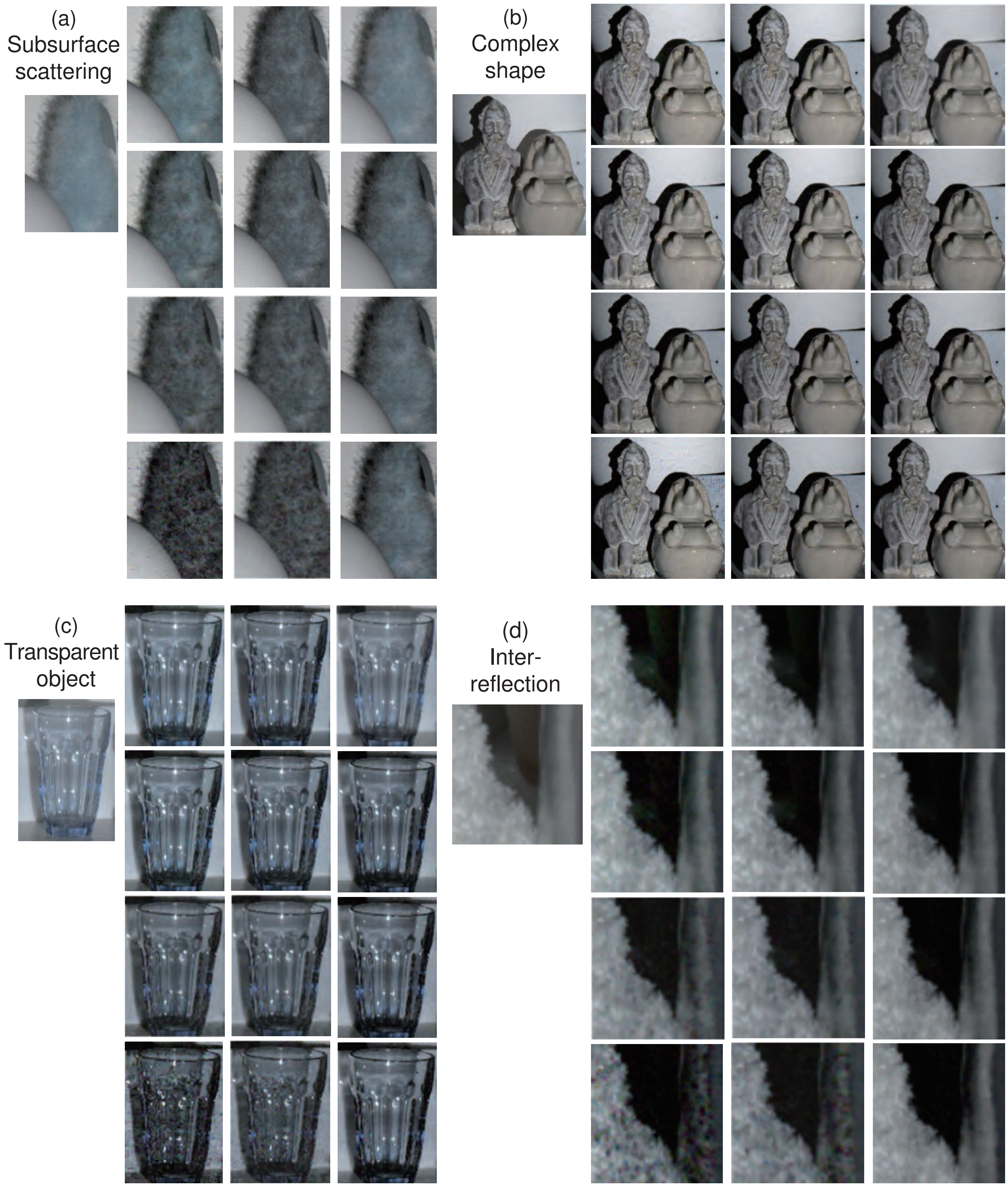

(a)
Subsurface scattering
(b)
Complex shape
(c)
Transparent object
(d)
Inter-reflection

Fig. 11.

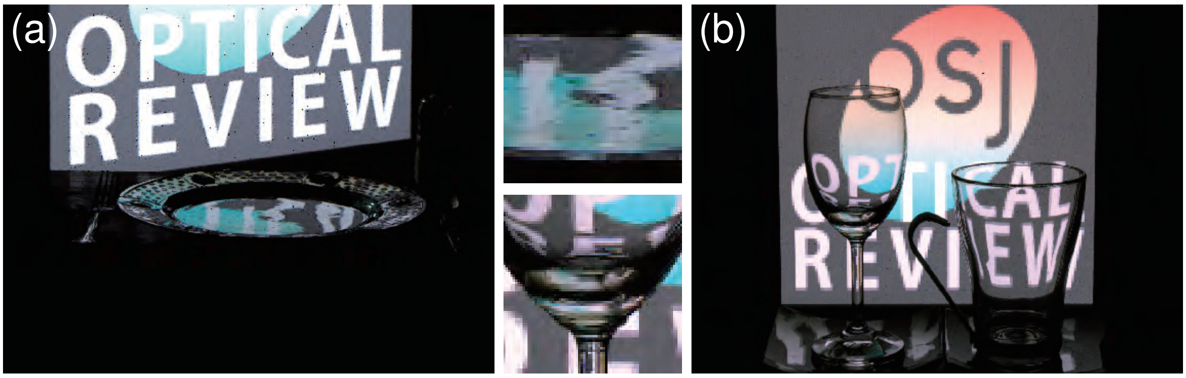


Fig. 12.