3D Reconstruction of Mirror-type Objects using Efficient Ray Coding

Siu-Kei Tin [*] Jinwei Ye [*] Mahdi Nezamabadi
Canon USA, Inc.
San Jose, CA 95134, USA

Can Chen [†]
University of Delaware
Newark, DE 19716, USA

Abstract

Mirror-type specular objects are difficult to reconstruct: they do not possess their own appearance and the reflections from environment are view-dependent. In this paper, we present a novel computational imaging solution for reconstructing the mirror-type specular objects. Specifically, we adopt a two-layer liquid crystal display (LCD) setup to encode the illumination directions. We devise an efficient ray coding scheme by only considering the useful rays. To recover the mirror-type surface, we derive a normal integration scheme under the perspective camera model. Since the resulting surface is determined up to a scale, we develop a single view approach to resolve the scale ambiguity. To acquire the object surface as completely as possible, we further develop a multiple-surface fusion algorithm to combine the surfaces recovered from different viewpoints. Both synthetic and real experiments demonstrate that our approach is reliable on recovering small to medium scale mirror-type objects.

Figure 1. Mirror-type objects “borrow” appearances from nearby environment.

1. Introduction

Recovering the 3D shape of an object is an important problem in computer vision. Successful reconstruction can benefit numerous applications in manufacturing, graphics modeling, and scene understanding, *etc*. However, most existing methods are focused on diffuse Lambertian surfaces. Recovering the shape of objects with complex reflectance (*e.g.*, specular, transparent or translucent) is still one of the few open problems in computer vision. In this paper, we propose a computational imaging method for recovering the 3D shape of mirror-type specular objects.

Mirror-type specular objects are difficult to reconstruct for several reasons: 1) The appearance of mirror-type object is determined by the environment, as shown in Fig. 1; 2) the reflection images are view-dependent, making it difficult to find correspondences; and 3) inter-reflections may occur when the shape is complex. Conceptually, most previous methods [9, 3, 39] use a continuous area illumination or a single display to cast coded patterns onto the mirror-type object and use a multi-view approach to resolve the surface shape. This class of methods suffer from the “depth-normal ambiguity” [15] because only one reference point is available on the illumination source. The depth-normal ambiguity is often resolved by using multiple viewpoints[4] or assuming additional surface constraints, such as planarity [16], smoothness [34], or integrability [37].

In this paper, we present a novel and simple computational imaging solution for reconstructing mirror-type specular objects. In particular, we adopt a two-layer liquid crystal display (LCD) setup to encode the directions of emitted light field. We optimize the illumination patterns by only encoding the useful rays. As a result, the number of captured images is reduced. By decoding the reflection images, the correspondences between illumination rays and camera rays can be directly obtained, as illustrated in Fig. 2. For accurate reconstruction, we integrate the normal field under the realistic perspective camera projection. The resulting surface is determined up to a scale. We demonstrate that the scale ambiguity is resolvable in a single viewpoint using backward ray tracing. To reconstruct the object surface as completely as possible, we capture multiple viewpoints by

[*] These authors contributed to this work equally.

[†] This work was performed when this author was an intern at Canon USA, Inc.

mounting the object on a rotary stage and further develop a multiple-surface fusion algorithm to combine the surfaces recovered from different camera views. We test our approach on both synthetic and real data and the experimental results show that our technique is reliable on recovering small to medium scale mirror-type objects.

In summary, our contributions include:

- Using a two-layer LCD setup as an active illumination source to resolve the depth-normal ambiguity.
- Designing an efficient ray coding scheme by only encoding the useful rays.
- Developing a surface reconstruction algorithm under perspective projection to generate complete profile of a mirror-type object.

2. Related Work

We first briefly review image-based techniques for recovering highly specular and mirror-type objects. Early approaches use image distortion to infer the shape of specular objects. Bonfort and Sturm [5] place a pattern on a reflection target and use multiple camera views to resolve the depth-normal ambiguity. Swaminathan *et al.* [35] study the caustic distortion in mirror reflections. In [37], Tarini *et al.* use the reflection pattern on a display and recover the surface by enforcing integrability of the normal field. Ding *et al.* [8] use general linear camera to model the distorted reflection of highly specular surfaces. Jacquet *et al.* [17] track the curved line images under a moving camera to recover the normal map of near-flat mirror surfaces.

Some approaches exploit the specular highlights caused by reflection for shape reconstruction. Ikeuchi [16] uses the reflectance map to determine the surface normal. Sanderson *et al.* [31] use an array of point light sources to generate a dense reflectance map for computing surface normal. Nayar *et al.* [26] model the specular highlight using extended Gaussian image (EGI). Oren and Nayar [27] propose the specular stereo to estimate mirror surface. Chen *et al.* [7] recover the mesostructure of a specular surface from the reflection of a distance point light source. Morris and Kutulakos [25] capture per-pixel reflectance and perform stereo matching on the reflectance map. Tunwattanapong *et al.* [38] determine the surface normal from reflectance field measurement. Roth *et al.* [30] use feature points in the reflection images of distant environment, or specular flow, to compute the geometry of specular objects. Adato *et al.* [2] use dense specular flow for shape estimation. Sankaranarayanan *et al.* [32] use specular flow to match correspondences on mirror-type objects and model surface as quadratic patches. The invariants in specular reflections are further generalized in [33]. Godard *et al.* [10] use silhouette to provide a rough reconstruction and then refine the surface using environment map. In this work, instead of using the distant environment light in specular flow, we consider near-field controlled illumination.

In the seminal work of [19], Kutulakos and Steger propose a generalized light-path triangulation framework to solve the 3D shape reconstruction problem with non-linear light path. They determine the triangulation rays by moving a display. In a similar vein, Liu *et al.* [23] translate a calibrated pattern to establish correspondences and derive a closed form solution for recovering the shape of specular objects. Chari and Sturm [6] exploit radiometric information as additional constraint. Grossberg and Nayar [12] use a translated displays for calibrating a catadioptric camera system. In our approach, we adopt a two-layer LCD system as illumination source and multiplex binary codes onto the two LCDs. Since our system has no physically movable parts and thus is much easier to calibrate and more portable. Liu *et al.* [22] propose to match for correspondences in the frequency domain for recovering the shape of transparent and specular objects. Wetzstein *et al.* [42, 43] use color-coded light field probe to measure transparent objects. Francken *et al.* [9] simulate a dense array of illumination sources using a display and use structured patterns to measure the mesostructure of a surface through specular reflection. Balzer *et al.* [4, 3] extend the multiple light source scheme by simulating a dense illumination array using LCD screen and encode the illumination using structured light patterns. Weinmann *et al.* [39] use a multi-view approach to reconstruct the full 3D shape of mirror objects. Gupta *et al.* [13, 14] develop optimized structured light patterns to recover the shape of surfaces with complex reflectance. O'Toole *et al.* [28] developed a structured light transport model to separate direct/indirect light transport and recover specular and transparent surfaces. Most recently, Matsuda *et al.* [24] propose a new structured light 3D scanning paradigm using a motion contrast sensor which is capable of capturing challenging surfaces with high specularity.

Our work is also related to compressive light field displays that comprise multiple LCD layers [40, 41]. By using the LCDs as light attenuator, compressive light field displays have been used for presenting glasses-free 3D content. Lanman *et al.* [20] introduce polarization field displays that use LCD layers as polarization rotator to generate dynamic light field. In this work, we use a two-layer LCD setup as an active illumination source to generate a dense light field for sampling the surface normal.

3. Acquisition System

Fig. 2 shows our mirror object acquisition system. Essentially, we use a two-layer LCD setup to encode the illumination light field. The two LCD layers resemble the two-plane parameterization (2pp) of light field [21, 11]: each illumination ray $\vec{r}$ is uniquely determined by its in-

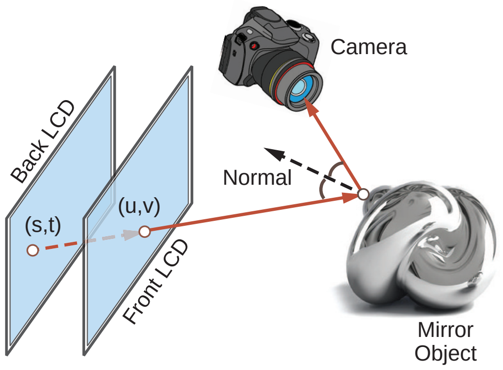

Figure 2. An illustration of our mirror-type object acquisition system using two-layer LCD.

tersections (*i.e.*, $[u, v]$ and $[s, t]$) with the two LCDs. A viewing camera is positioned at the side of the two LCD layers to capture the reflection on the mirror object surface. We use binary code patterns to encode the illumination rays for robust decoding. By mapping the captured reflection images to LCD pixel indices, we can directly establish a dense set of correspondences between the illumination rays and the camera rays that can be used to reconstruct the object surface.

LCD Polarization Modulation. To enable binary ray coding, we strategically configure the polarization rotation of the two LCDs such that the pixel operation between the two LCDs is linear in the binary field $\mathbb{F}_2$. In particular, we remove the original polarizers of the two LCDs and apply a pair of perpendicular linear polarization layers: a horizontal polarizer is applied to the diffuse backlight and a vertical polarizer to the front LCD, as shown in Fig. 3.

Recall that a LCD utilizes the polarization modulation properties of liquid crystal to form images: the display image appears white when the light is twisted 90° by the liquid crystal, otherwise black. In our two-layer LCD, consider a light ray $\vec{r} = [u, v, s, t]$ emitted from the unpolarized backlight. After passing through the first polarization layer, the ray becomes horizontally polarized. In order to pass the second vertical polarization layer and become visible, the ray needs to be twisted once (*i.e.*, polarization rotates 90°) by the two liquid crystal layers. When the ray is untwisted or twisted twice (*i.e.*, polarization rotates 180°), it would be blocked and not visible. This resembles the logical exclusive or (XOR) operator that outputs true only when both inputs are different. Thus the observed binary code B_r for the ray $\vec{r}$ can be written as $B_r(\vec{r}) = B_f(u, v) \oplus B_b(s, t)$, where $\oplus$ is the XOR operator, B_f and B_b are the binary code pattern on the front and back LCDs respectively.

Since XOR is linear in the binary field $\mathbb{F}_2$ (*i.e.*, addition modulo two), this enables code multiplexing on the two LCD layers using linear combinations.

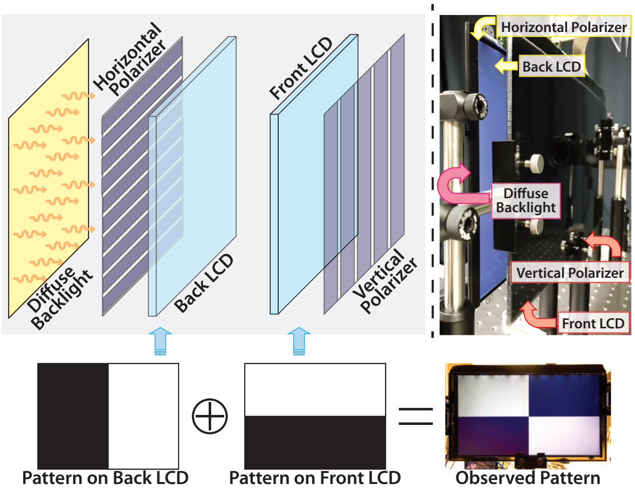

Figure 3. The two-layer LCD setup. We place the two LCDs between a pair of perpendicular linear polarizers such that the pixel operation between the two LCDs is XOR in the binary field $\mathbb{F}_2$.

4. Efficient Ray Coding

Our goal is to design a minimum binary code book for illumination rays such that every ray has a unique binary code sequence.

A naïve approach is to encode all the rays in the emitted light field. If each LCD has N pixels, then the total number of rays in the light field is N^2. However, as shown in Fig. 4, only a small subset of light field rays are reflected by the object and finally captured by the camera. We call the subset effective light field. Assume for each pixel on the front LCD panel, a cone comprising $\sim k$ rays intersect with the object, where $k << N$, then the number of rays in the effective light field is $k \times N << N^2$. Therefore, by only encoding rays in the effective light field, we can reduce the size of binary code book and save the acquisition time as a result.

In our experiment, we first determine a bounding sphere according to the hardware setup and pre-compute the effective light field by tracing rays from the two LCDs and intersecting with the bounding sphere. After obtaining the effective light field, we use an iterative code projection approach to generate the binary code pattern for the two LCDs.

In order to determine the code for effective rays given the LCD code patterns, we define a *code transfer matrix* A that indicates the pixel correpondences of effective rays on the two LCDs. Specifically, A is a $l \times 2N$ matrix, where l is the number of rays in the effective light field and N is the number of display pixels in the 1D case. If the ith ray is defined by pixel u_i and s_i on the two displays respectively, then in the ith row of A: $A(i, u_i) = 1$ and $A(i, N + s_i) = 1$ (elsewhere are zeros). Given the composite binary code

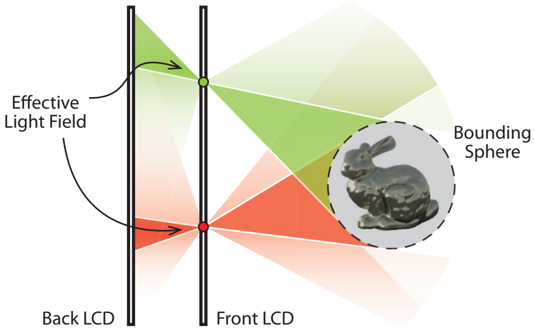

Figure 4. A small subset of the emitting light field intersect with the bounding sphere that encloses the object. We only consider the effective light field for efficient ray coding.

sequence matrix X for the two LCDs, the resulting binary code book R for the effective rays is given as

$$R = AX \tag{1}$$

where X is an $2N \times M$ binary matrix which indicates the M sets of binary code patterns displayed on the two LCDs and R is an $l \times M$ binary matrix for the ray codes. Please note that the linearity of the XOR operation enables this matrix multiplication representation.

We therefore set out to find X such that R has unique row vectors, *i.e.*, each ray will receive a unique (row) code vector. We start from a known solution X_0 with dimension $2N \times M_0$ such that R_0 has unique rows. One example of the known solution is the Gray code, $X_0 = \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix}$. However, the Gray code is redundant for a reduced set of rays. In order to reduce the number of code sets, we apply a code projection matrix P with dimention $M_0 \times M_p$ (where $M_p < M_0$) on Eqn. 1: $R_0P = A(X_0P)$. If the rows of $R = R_0P$ are unique, then $X = X_0P$ is a new solution.

Note that right multiplying corresponds to mixing columns of X. Therefore, this code projection process can be interpreted as multiplexing binary patterns on the two LCDs corresponding to different bit planes. Different from conventional code multiplexing (*e.g.*, Hadamard multiplexing) that uses linear combination over real numbers $\mathbb{R}$, here we use binary addition (*i.e.* XOR), or linear combination over the binary field $\mathbb{F}_2$.

Brute force search of the projection matrix P is computationally expensive. In our solution, we break down the projection into elementary projections along vectors. The projection vectors are chosen to make sure that after each projection, each ray will continue to receive a unique code. We repeat this process until the code projection space is null. We demonstrate the existence of the code projection matrix and an optimal selection algorithm in the Appendix.

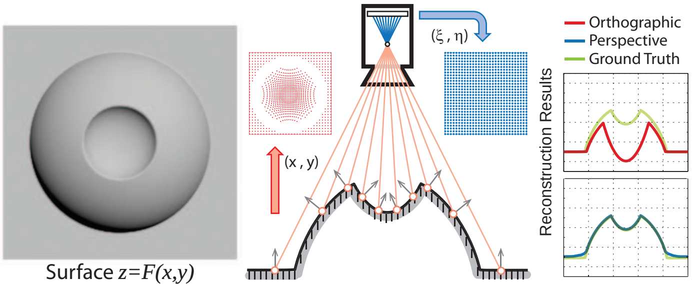

Figure 5. Perspective camera projection result in non-uniform samples in the world coordinate (x, y). Simply integrating the surface normal field under orthographic camera assumption results in large error. In contrast, our proposed integration scheme produces reliable reconstruction.

5. Surface Reconstruction

After decoding the captured images, we obtain the display pixel index maps for each reflection region. We compute the directions of incident illumination rays $\vec{r}_{in}$ by connecting the indices of corresponding pixels on the two LCDs. The reflected camera rays $\vec{r}_{re}$ are given by the camera calibration. Finally, we compute the surface normal $\vec{n}$ by taking the half-way vector of $\vec{r}_{in}$ and $\vec{r}_{re}$:

$$\vec{n} = \frac{\vec{r}_{in} + \vec{r}_{re}}{||\vec{r}_{in} + \vec{r}_{re}||} \tag{2}$$

Theoretically, we can also obtain the 3D position of the surface point by intersecting the incident rays and reflected rays. However, due to the large display pixel size and short distance in between the two LCDs, the intersection points are very sensitive to noise (similar to triangulation in multi-view stereo) and the resulting surface is prone to large errors. We have therefore opted to recover the surface by normal integration for robust and accurate reconstruction.

For simplicity, most existing methods assume the orthographic camera model and directly integrate the normal field using Poisson method. However, this assumption can result in large reconstruction error due to the non-uniform sampling caused by the perspective effect (see Fig. 5 for a toy case example). We therefore adopt the more realistic perspective camera model for accurate reconstruction. We derive a normal integration algorithm under perspective projection. Similar to perspective photometric stereo [36, 29], the initial integrated surface is determined up to a scale. We resolve the scale ambiguity by backward ray tracing. Furthermore, to obtain an object surface as completely as possible, we develop a joint optimization algorithm to merge surfaces from multiple viewpoints.

5.1. Normal Integration under Perspective Camera

Assume the origin is at the camera center of projection (CoP) and the focal length is f. Let (ξ, η) be the image

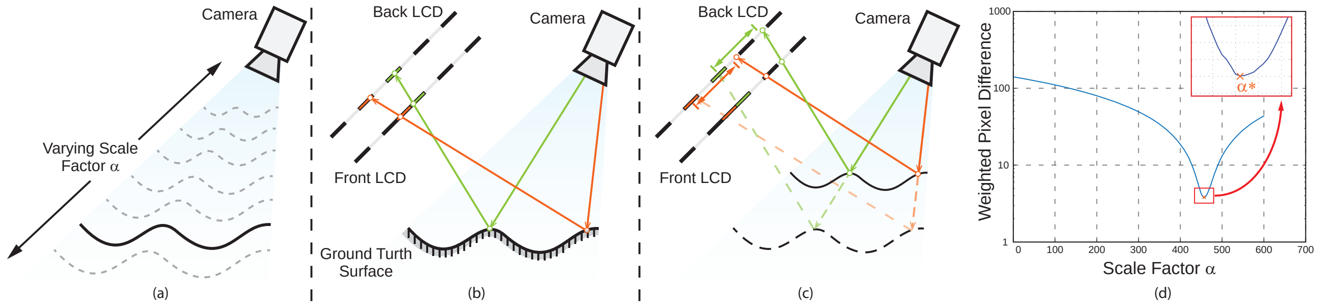

Figure 6. Single view scale factor estimation using backward ray tracing. (a) We search through a range of scale factors for an integrated surface; (b) We assume the measured display indices corresponds the surface with correct scale; (c) For each scale factor, rays are traced from the viewing camera, reflected by the mirror surface, and intersect with the two LCDs. We compare the LCD intersections with the measured indices; (d) An example curve of the objective function. The optimal scale factor corresponds to the one with minimum weighted display pixel error.

coordinates on the sensor plane and (x, y, z) be the world coordinates. For each image pixel (ξ, η), we have the measured normal $\vec{n} = (n_1, n_2, n_3)$. Our goal is to estimate an optimal surface that best fits the given normal field sampled at the image coordinates. Traditional approach assumes orthographic camera and obtains gradients from normals using equations $z_x = -n_1/n_3, z_y = -n_2/n_3$ and solve the optimal surface using Poisson Equation solver. In our case, however, due to the perspective projection, the surface samples in world coordinates (x, y, z) are not linear relative to the image coordinates (ξ, η). Therefore, the Poisson method cannot be applied directly. To integrate the surface, we first transform the surface gradients (z_x, z_y) into image coordinates by applying the perspective projection (*i.e.*, $x = \xi \cdot z/f, y = \eta \cdot z/f$):

$$\begin{aligned} z_\xi &= \frac{z_x}{f}(z + z_\xi \cdot \xi) + \frac{z_y}{f} z_\xi \cdot \eta \\ z_\eta &= \frac{z_x}{f} z_\eta \cdot \xi + \frac{z_y}{f}(z + z_\eta \cdot \eta) \end{aligned} \tag{3}$$

However (z_ξ, z_η) is not directly integrable since they are functions of the unknown surface z itself. To eliminate z in Eqn. 3, we substitute z with a new variable $t = \ln z$. By applying the chain rule, we have

$$\begin{aligned} t_\xi &= \frac{z_x}{f - z_x \xi - z_y \eta} \\ t_\eta &= \frac{z_y}{f - z_x \xi - z_y \eta} \end{aligned} \tag{4}$$

Then (t_ξ, t_η) can be integrated using standard Poisson method. The integration results in $t = t_0 + c$ for some arbitrary constant c. Substituting t with z, we have $z = \alpha \cdot e^{t_0}$, where $\alpha = e^c$ is a multiplicative arbitrary constant that indicates the scale of the surface.

In the case of the orthographic projection, the additive integration constant c manifests as an unknown translation along the camera axis, which is usually not necessary because in most applications, the goal is not absolute depth but the 3D model of the object. In the case of the perspective camera, this constant appears, through exponentiation, as an unknown multiplicative constant α which can be considered as the scale factor of the object and is critical to the reconstruction.

5.2. Resolving Scale Ambiguity in Single View

In order to determine the scale factor, a naïve approach is to use triangulation points to fit the integrated surface. However, the triangulation points are not reliable and produce large errors in the scale factors. One underlying reason is that the display pixel size is relatively big (*e.g.*, 0.179mm in our case) and result in large uncertainty in the display rays. Since the camera pixel size is much smaller (*e.g.*, 0.0043mm), we consider the camera rays reliable. Thus we back-trace rays from the camera and use the decoded display pixel indices as prior to determine the scale factor with a maximum likelihood method.

Specifically, we first integrate the surface using the algorithm described in Section 5.1 and re-compute the normal field on the integrated surface for backward ray tracing. As shown in Fig. 6, we search through a range of scale factors. For each scale factor α, we trace rays from camera pixels in the reflection regions and compute the reflected rays to intersect with the two LCDs. Given the measured LCD indices as prior, we compute the display pixel errors to determine α. Due to the noise in light transport, the decoded LCD indices are subject to errors that are related to the object geometry and LCD distance (*e.g.*, the back LCD panel tend to have larger error than the front one), we then use the inverse of the standard deviations of the display index maps in a small neighborhood as weights for balancing the index errors. Therefore, our objective function is formulated as:

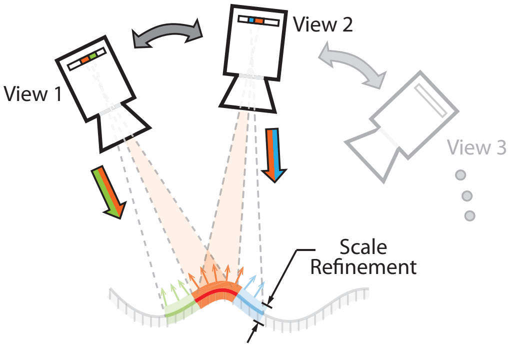

Figure 7. We combine integrated surfaces from multiple viewpoints by refining the scale factors.

$$\alpha^* = \arg\min_{\alpha} \sum_{i \in R} \Big(\big(\frac{\hat{u}_i - u_i(\alpha)}{\sigma^x_{i,f}} \big)^2 + \big(\frac{\hat{v}_i - v_i(\alpha)}{\sigma^y_{i,f}} \big)^2 + \big(\frac{\hat{s}_i - s_i(\alpha)}{\sigma^x_{i,b}} \big)^2 + \big(\frac{\hat{t}_i - t_i(\alpha)}{\sigma^y_{i,b}} \big)^2 \Big) \tag{5}$$

where i is the camera pixel index in the reflection region R; $(\hat{u}_i, \hat{v}_i)$ and $(\hat{s}_i, \hat{t}_i)$ are the measured display pixel indices for the front and back LCDs respectively; $(u_i(\alpha), v_i(\alpha))$ and $(s_i(\alpha), t_i(\alpha))$ are the intersection points on the front and back LCDs for the scale factor α; and $\sigma^x_{i,f}, \sigma^y_{i,f}, \sigma^x_{i,b}, \sigma^y_{i,b}$ are the standard deviations at pixel i for horizontal/vertical index maps of the front and back LCD respectively.

The scale factor is solved by minimizing the objective function. We use the golden section search to solve the problem. This objective function corresponds to the assumption that the display pixel error distribution is Gaussian and the global minimum of the error occurs at maximum likelihood of α.

5.3. Multiple-View Fusion

To reconstruct the object surface as completely as possible, we rotate and capture the object to acquire multiple surface patches and then fuse the different views by solving a joint optimization problem. As shown in Fig. 8, the direct fusion using the single view scale factors has artifacts of misalignment due to slight errors in the scale factors. We therefore refine the initial scale factors when multiple views of the surface are available, as shown in Fig. 7

Specifically, in each view, by applying the initial scale factor $\alpha^{(0)}_{\Omega}$ estimated by single view backward ray tracing, we have a disjoint union of surfaces Ω that receive LCD reflections $\{\alpha^{(0)}_{\Omega} W_{\Omega}\}$ in the camera coordinate system. In order to compare between different views, the scaled surfaces must be brought together into one common coordinate system, or the world coordinate. By applying the rotation and translation matrix, we have the disjoint union of surfaces in the world coordinate: $\{R^{-1}(\alpha^{(0)}_{\Omega} W_{\Omega} T)\}$.

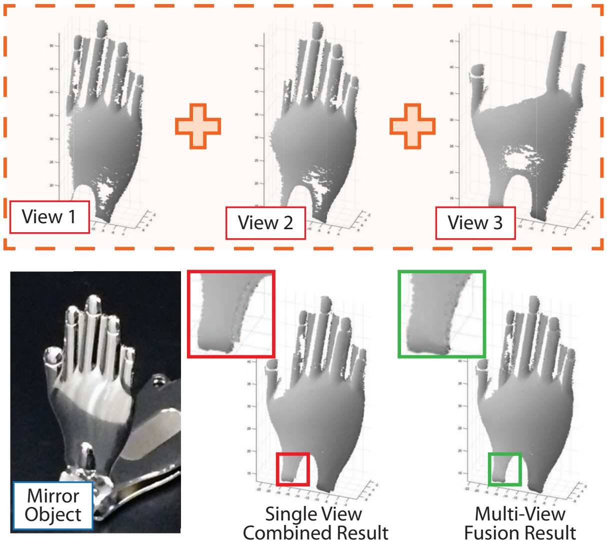

Figure 8. Merging surfaces from different views. Directly using the initial single view scale factors results in artifacts of misalignment.

We then combine multiple camera views by minimizing the differences between overlapped scaled surfaces. We measure the difference in both position and angle. Hence the objective function for matching a surface ω from view i and surface ω' from view j can be written as:

$$\begin{aligned} \varepsilon(\alpha_\omega, \alpha_{\omega'}) = {} & d_{C^0}(R_i^{-1}(\alpha_c W_\omega - T_i), R_j^{-1}(\alpha_{\omega'} W_{\omega'} - T_j)) \\ & + d_{C^1}(R_i^{-1}(\alpha_\omega W_\omega - T_i), R_j^{-1}(\alpha_{\omega'} W_{\omega'} - T_j)) \end{aligned} \tag{6}$$

where d_{C^0} measures topological closeness between the two surfaces (in mm), and d_{C^1} measures closeness in the tangent space of the two surfaces (in degrees).

Fig. 8 illustrates an example of merging different views using the proposed algorithm. As shown in the figure, the combined point cloud using the initial scale factor estimated by the single view has artifacts (*e.g.*, offsets between different views). By using the multiview refinement, the surfaces from multiple views align well.

6. Experiments

6.1. Synthetic data Simulation

We first apply our method on synthetic data to verify its accuracy. In the simulation, we set up the two-layer LCD with resolution 1920×1080. The display size is $38.4cm \times 21.6cm$ and the simulated display pixel size is $0.02cm$. The two-layer LCD emulates the setup in our real experiment.

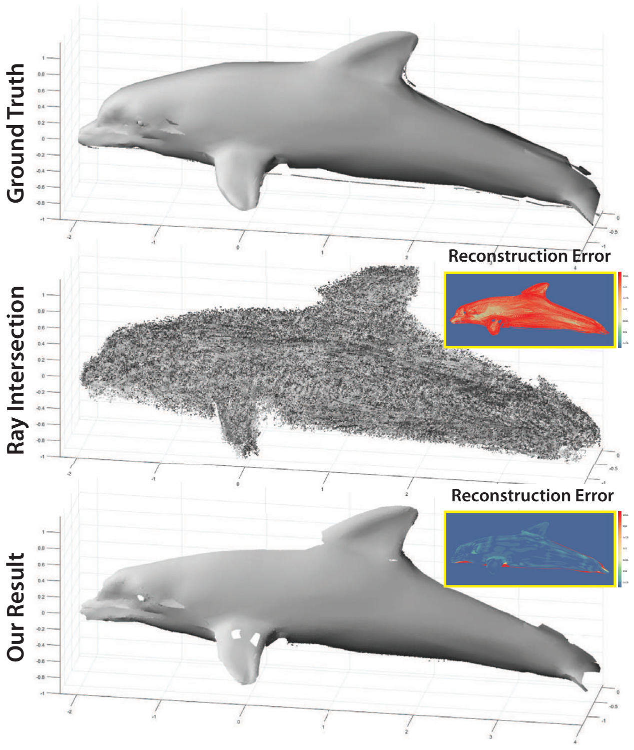

Figure 9. Simulation result on a "dolphin" model.

We place the mirror object model $5.2cm$ and render the reflection image using Pov-ray (http://www.povray.org/). We use a bounding sphere with diameter $2.5cm$ to enclose the object $\sim\ 2cm$. Less than 1% of the full light field intersect with the bounding sphere. We only encode the effective light field and generate binary patterns for the two LCDs. To acquire complete surface coverage, we rotate the camera vertically at $\pm 45°$ and horizontally at $\pm 30°$. We first integrate the surface in each camera view using our integration scheme under perspective projection and then estimate the initial scale factor using backward ray tracing. Finally, we apply the joint optimization algorithm to combine surfaces from multiple viewpoints. The reconstructed point cloud and the reconstruction error map are shown in Fig. 9. We also show the reconstruction result using direct ray triangulation. Due the large display pixel size, the recovered point cloud using triangulation is very noisy and inaccurate, where the average error is $0.2cm$. In contrast, with average error $0.005cm$, the normal integration result is much more accurate.

6.2. Real Scene Experiment

We also perform real experiments to validate our approach. We build our two-layer LCD system using the liquid cyrstal panels disassembled from two commodity $15.6''$ displays (ASUS MB168B+). Each panel has resolution 1920×1080 and the pixel size is $0.179mm$. The distance between the two LCDs is $30mm$. The larger the distance, the more accurate the incident rays are. However, the light throughput decreases. Our experimental setup is shown in Fig. 10. We place the mirror-type object on a rotary stage to acquire complete coverage of the surface. We use a commodity DSLR camera (Canon T3i with $50mm$ $f/1.8$ lens) and place the camera at the right side of the display to capture the display reflection on the mirror object. To reduce out-of-focus blur, we use small aperture ($f/22$) and long exposure for extended depth of field. To reduce noise, we also capture the inverse patterns. Due to the property of XOR operation, we only invert the patterns on one LCD layer for inverse patterns.

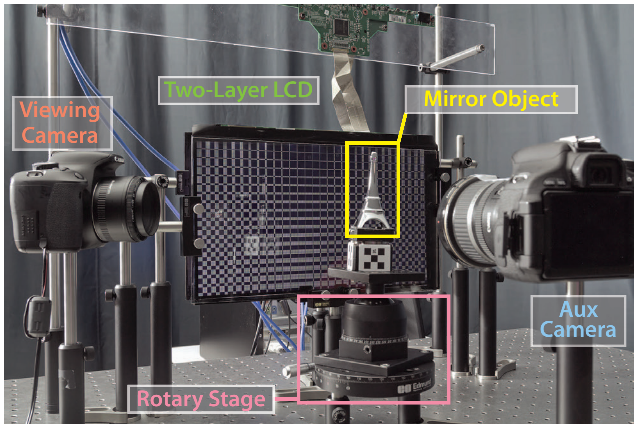

Figure 10. Our experimental setup with a two-layer LCD.

Calibration. We calibrate the viewing camera using the Matlab calibration toolbox [1]. Since the display is not directly visible in the viewing camera, calibration of the two-layer LCD is more challenging. To calibrate the LCDs, we use a second auxiliary camera pointing at the display. We first calibrate the LCDs relative to the auxiliary camera. Then we correlate calibrations of the viewing camera and the auxiliary camera using a checkerboard that is in common view to both cameras.

We test on two real mirror-type objects (*e.g.*, "Eiffel Tower" and "Big Ben"). The size of the mirror objects is $\sim\ 30mm \times 30mm \times 100mm$ and we place the object at $\sim\ 100mm$ in front of the display. We use a bounding sphere with diameter $150mm$ and less than 2% of the light field are effective. We generate the binary pattern for each LCD by code multiplexing in the binary field. Our efficient coding scheme results in 34 code patterns to be captured for each view, while the Gray code needs take 44 shots $(2 \times (\lceil \log_2 1920 \rceil + \lceil \log_2 1080 \rceil))$. To capture a complete profile, we rotate the object at $20°$ steps to obtain sufficient coverage of the object. To merge surfaces from multiple views, we use a 3×3 checkerboard rigidly attached to the object to estimate the world transformation (rotation and translation) for each viewpoint. Since the small checkerboard produces large extrinsic calibration error, after the multi-view scale factor refinement, we use the iterative closest point (ICP) algorithm to update the world transformation matrices and use the new world transformation to further refine the scale factor. We repeat this process until convergence. Our reconstruction results are shown in Fig. 11. Compared with the “Eiffel Tower”, the “Big Ben” is a more challenging case because it has more disjoint surfaces in each view. The multi-view joint optimization takes longer computational time. The top of Big Ben is truncated in our reconstruction due to out of the camera’s field of view. There are also some artifacts on the clock due to self-occlusion.

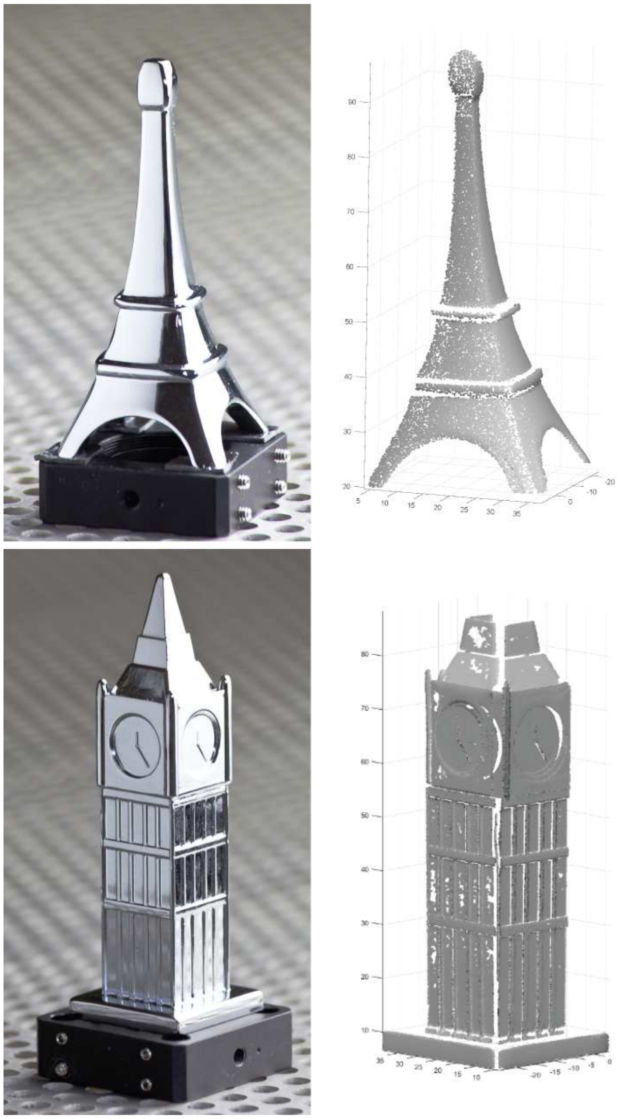

Figure 11. Results on real objects. Left: photographs of the mirror-type objects; Right: our reconstructed point clouds.

7. Conclusions and Discussions

In this paper, we have presented a two-layer LCD setup as active illumination source for recovering the shape of mirror-type objects. We have further derived a normal integration scheme under perspective camera projection. To resolve the scale ambiguity, we have developed single view approach for estimating the initial scale factor and a multi-view joint optimization approach to refine the scale factors. Finally, we have demonstrated the effectiveness of our approach using both synthetic and real-world experiments. Our solution provides reliable and accurate 3D reconstruction of small to medium scale mirror-type objects.

One limitation of our approach is that the reflection image only covers limited areas on the object surface and we need to rotate the object with fine steps to acquire full coverage. One can use a curved display to surround the object to obtain larger coverage of reflection. On the algorithm side, if the initial scale factor from the single view has large error, the search in the multi-view joint optimization may converge slowly. We can improve the efficiency of the search by considering additional surface constraints (*e.g.*, curvature and smoothness). Since the mirror reflection changes light polarization, one possible future direction is to use polarized display and camera [18] to increase the reconstruction accuracy and reduce acquisition time.

References

[1] Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[2] Y. Adato, Y. Vasilyev, O. Ben-Shahar, and T. Zickler. Toward a theory of shape from specular flow. In *International Conference on Computer Vision (ICCV)*, 2007.

[3] J. Balzer, D. A. Feliz, S. Soatto, S. Höfer, M. Hadwiger, and J. Beyerer. Cavlectometry: Towards holistic reconstruction of large mirror objects. In *International Conference on 3D Vision (3DV)*, 2014.

[4] J. Balzer, S. Hofer, and J. Beyerer. Multiview specular stereo reconstruction of large mirror surfaces. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[5] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In *International Conference on Computer Vision (ICCV)*, 2003.

[6] V. Chari and P. Sturm. A theory of refractive photo-light-path triangulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[7] T. Chen, M. Goesele, and H.-P. Seidel. Mesostructure from specularity. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[8] Y. Ding, J. Yu, and P. Sturm. Recovering specular surfaces using curved line images. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] Y. Francken, T. Cuypers, T. Mertens, J. Gielis, and P. Bekaert. High quality mesostructure acquisition using specularities. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[10] C. Godard, P. Hedman, W. Li, and G. J. Brostow. Multi-view reconstruction of highly specular surfaces in uncontrolled environments. In *International Conference on 3D Vision (3DV)*, 2015.

[11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH*, 1996.

[12] M. Grossberg and S. Nayar. The Raxel Imaging Model and Ray-Based Calibration. *International Journal on Computer Vision*, 61(2):119–137, Feb 2005.

[13] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. Narasimhan. Structured light 3d scanning in the presence of global illumination. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[14] M. Gupta and S. Nayar. Micro Phase Shifting. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Computer Graphics Forum*, 29(8), 2010.

[16] K. Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 1981.

[17] B. Jacquet, C. Hane, K. Koser, and M. Pollefeys. Real-world normal map capture for nearly flat reflective surfaces. In *International Conference on Computer Vision (ICCV)*, 2013.

[18] A. Kadambi, V. Taamazyan, B. Shi, and R. Raskar. Polarized 3d: High-quality depth sensing with polarization cues. In *International Conference on Computer Vision (ICCV)*, 2015.

[19] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. *International Journal of Computer Vision*, 76(1), 2008.

[20] D. Lanman, G. Wetzstein, M. Hirsch, W. Heidrich, and R. Raskar. Polarization fields: Dynamic light field display using multi-layer LCDs. *ACM Trans. Graph.*, 30(6), 2011.

[21] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, 1996.

[22] D. Liu, X. Chen, and Y.-H. Yang. Frequency-based 3d reconstruction of transparent and specular objects. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[23] M. Liu, K.-Y. K. Wong, Z. Dai, and Z. Chen. Specular surface recovery from reflections of a planar pattern undergoing an unknown pure translation. In *Asian Conference on Computer Vision (ACCV)*. 2011.

[24] N. Matsuda, O. Cossairt, and M. Gupta. Mc3d: Motion contrast 3d scanning. In *International Conference on Computational Photography (ICCP)*, 2015.

[25] N. Morris and K. Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *International Conference on Computer Vision (ICCV)*, 2007.

[26] S. K. Nayar, A. C. Sanderson, L. E. Weiss, and D. Simon. Specular surface inspection using structured highlight and gaussian images. *IEEE Transactions on Robotics and Automation*, 6(2), 1990.

[27] M. Oren and S. K. Nayar. A Theory of Specular Surface Geometry. *International Journal of Computer Vision*, 24(2):105–124, Sep 1997.

[28] M. O'Toole, J. Mather, and K. N. Kutulakos. 3d shape and indirect appearance by structured light transport. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[29] T. Papadhimitri and P. Favaro. A new perspective on uncalibrated photometric stereo. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[30] S. Roth and M. J. Black. Specular flow and the recovery of surface structure. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[31] A. Sanderson, L. Weiss, and S. Nayar. Structured highlight inspection of specular surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(1), 1988.

[32] A. C. Sankaranarayanan, A. Veeraraghavan, O. Tuzel, and A. Agrawal. Specular surface reconstruction from sparse reflection correspondences. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[33] A. Sankarnarayanan, A. Veeraraghavan, O. Tuzel, and A. Agrawal. Image invariants for smooth reflective surfaces. In *European Conference on Computer Vision (ECCV)*, 2010.

[34] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. In *European Conference on Computer Vision (ECCV)*, 2002.

[35] R. Swaminathan, M. Grossberg, and S. Nayar. A Perspective on Distortions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume II, pages 594–601, Jun 2003.

[36] A. Tankus and N. Kiryati. Photometric stereo under perspective projection. In *International Conference on Computer Vision (ICCV)*, 2005.

[37] M. Tarini, H. P. A. Lensch, M. Goesele, and H.-P. Seidel. 3d acquisition of mirroring objects using striped patterns. *Graphical Models*, 67(4), 2005.

[38] B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on Graphics*, 32(4), 2013.

[39] M. Weinmann, A. Osep, R. Ruiters, and R. Klein. Multi-view normal field integration for 3d reconstruction of mirroring objects. In *International Conference on Computer Vision (ICCV)*, 2013.

[40] G. Wetzstein, D. Lanman, W. Heidrich, and R. Raskar. Layered 3d: Tomographic image synthesis for attenuation-based light field and high dynamic range displays. In *SIGGRAPH*, 2011.

[41] G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar. Tensor displays: compressive light field synthesis using multilayer displays with directional backlighting. In *SIGGRAPH*, 2012.

[42] G. Wetzstein, R. Raskar, and W. Heidrich. Hand-Held Schlieren Photography with Light Field Probes. In *International Conference on Computational Photography (ICCP)*, 2011.

[43] G. Wetzstein, D. Roodnick, R. Raskar, and W. Heidrich. Refractive Shape from Light Field Distortion. In *International Conference on Computer Vision (ICCV)*, 2011.

Appendix:

Existence of Code Projection Matrix P

If $\mathbf{v}$ is a row vector in $\mathbb{F}_2^M$, the projection matrix $P_\mathbf{v}$ along $\mathbf{v}$ is an $M \times (M-1)$ binary matrix satisfying $\mathbf{v}P_\mathbf{v} = \mathbf{0}$. That is, the null space $\ker(P_\mathbf{v}) = \{\mathbf{0}, \mathbf{v}\}$ and only the vector $\mathbf{v}$ is annihilated. In order for the rows of R_0 to remain unique, all we need to do is to make sure that none of the pairwise differences of rows of R_0 gets annihilated by $P_\mathbf{v}$. Let $D(R)$ be the set of pairwise differences of rows of a code matrix R:

$$D(R) = \{R(i,:) \oplus R(j,:) | 1 \le i < j \le l\} \qquad (7)$$

where $R(i,:)$ denotes the ith row of matrix R. By the assumption that R has unique rows, we have $\mathbf{0} \notin D(R)$. Note that over $\mathbb{F}_2$, difference is the same as sum $\oplus$. Fig. 12 illustrates the geometry of the projection.

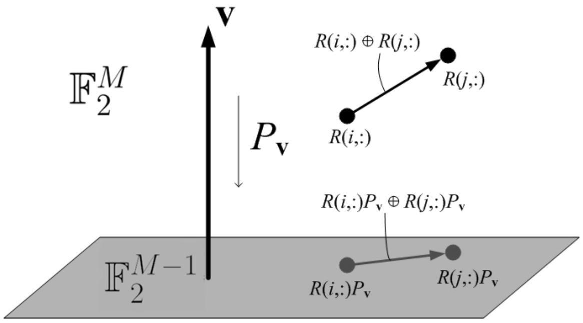

Figure 12. Projection in code space.

Also, we define the complement set:

$$\tilde{D}(R) = \mathbb{F}_2^M \setminus (\{\mathbf{0}\} \cup D(R)) \qquad (8)$$

Then it is clear that any choice of $\mathbf{v} \in \tilde{D}(R)$ will give a projection matrix $P_\mathbf{v}$ that preserves unique rows of R. If $\tilde{D}(R) = \emptyset$, then no such projection is possible. On the other hand, if $\tilde{D}(R) \neq \emptyset$, it is typically the case that $\tilde{D}(R)$ will contain many vectors.

Optimal Selection of Code Projection Matrix P

Intuitively, we should choose a projection vector that will maximize the chance of another projection. In other words, we would like to choose a $\mathbf{v}$ such that $\tilde{D}(RP_\mathbf{v}) \neq \emptyset$, or better still, such that $\tilde{D}(RP_\mathbf{v})$ is a large set. This is formalized by introducing the notion of Code Sparsity of X:

$$\psi(X; A) = \frac{\tilde{D}(AX)}{2^M - 1} \times 100\% \qquad (9)$$

where M is the number of columns of X. A *locally optimal projection* is a projection matrix $P_{\mathbf{v}^*}$ given by a projection vector $\mathbf{v}^*$ satisfying

$$\mathbf{v}^* = \text{arg max}_{\mathbf{v} \in \tilde{D}(AX)} \psi(XP_\mathbf{v}; A) \qquad (10)$$

When $\tilde{D}(AX)$ is a large set, searching through vectors in it can be very time consuming even for an offline calculation. We implement an approximation to the locally optimal projection by applying a heuristic filter $\mathcal{F}$ on $\tilde{D}(AX)$ to reduce the size of the search set:

$$\hat{\mathbf{v}} = \text{arg max}_{\mathbf{v} \in \mathcal{F}(\tilde{D}(AX))} \psi(XP_\mathbf{v}; A) \qquad (11)$$

Let $\|\mathbf{v}\|_H$ denote the Hamming weight of a binary vector, *i.e.*, the number of 1's in the vector. We define the Minimum Weight Filter, $\mathcal{F}_{\text{MinWt}}$, by

$$\mathcal{F}_{\text{MinWt}}(R) = \left\{ \mathbf{v} \in R \mid \|\mathbf{v}\|_H = \min_{\mathbf{w} \in R} \|\mathbf{w}\|_H \right\} \qquad (12)$$

Empirical evidence suggests that performance of the Minimum Weight Filter is very close to the true local optimum. It has the additional benefit that the resulting projection minimally mixes the bit planes and therefore preserves some desirable error-deterrent properties of the Gray code.

Performance of the Efficient Ray Coding

We apply the efficient ray coding algorithm to an example configuration of the two-layer LCD system, shown in Figure 13.

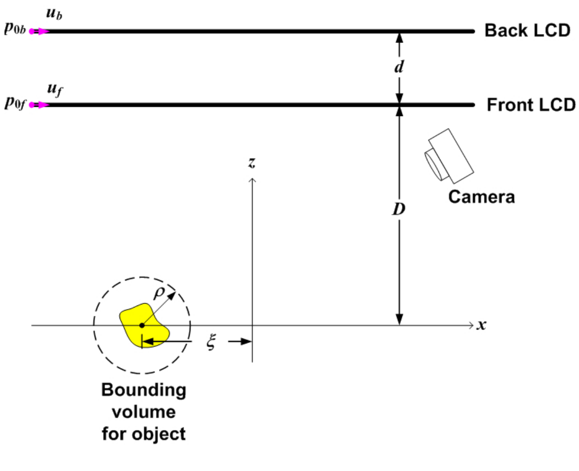

Figure 13. An example configuration of the two LCD system with a bounding volume.

Both LCDs have the same pixel resolution $N = 1080$ and pixel pitch $\|\mathbf{u}_b\| = \|\mathbf{u}_f\| = 0.179$ mm. The separation between the LCD layers is $d = 25$ mm while the bounding sphere is positioned relative to the front LCD with a vertical offset of $D = 95$ mm and a horizontal offset of

$\xi = 45.82\,\mathrm{mm}$. We vary the radius of the bounding sphere ρ to study the effect of sparsity of the light field on the coding.

Table 1 shows the efficiency of the optimized code patterns relative to the size of the bounding volume and the theoretical bound $\lceil \log_2 l \rceil$, where l is the number of useful rays that reach the bounding volume. As a reference, Gray code solution requires $2\lceil \log_2 1080 \rceil = 22$ shots.

ρ(mm)	l	$\lceil \log_2 l \rceil$	# projections	# shots
38.1	112080	17	4	18
31.75	**91572**	**17**	**5**	**17**
25.4	**72118**	**17**	**5**	**17**
19.05	53444	16	5	17
12.7	**35320**	**16**	**6**	**16**
6.35	**17594**	**15**	**7**	**15**

Table 1. Performance of code patterns relative to the size of the bounding volume. Boldfaced rows indicate that the theoretical bound has been achieved.

By fixing $\rho = 31.75mm$, we show an example code matrix X for the two-layer LCD in Fig 14.

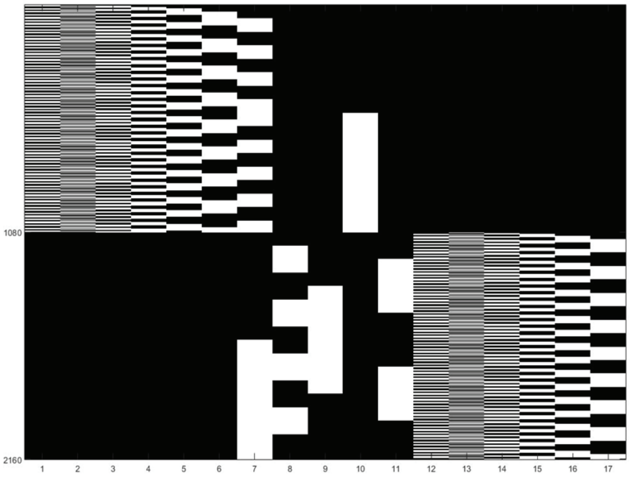

Figure 14. Example two-layer LCD code patterns (X).

It can be seen that while the Gray code patterns have been modified, the high spatial frequency bit patterns remain intact, which is due to the Minimum Weight Filter selection algorithm for projection vectors.