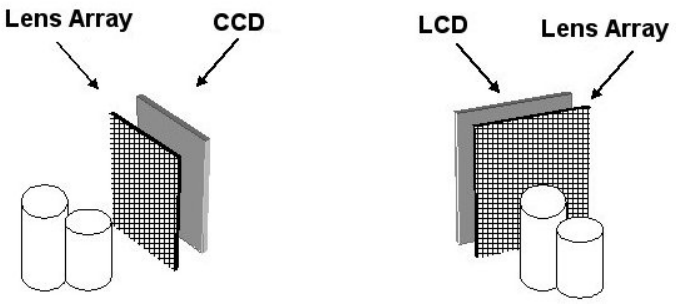
Enhanced Reconstruction of 3D Shape and Texture from Integral Photography Images

G. Passalis, N. Sgouros, S. Athineos and T. Theoharis

Department of Informatics and Telecommunications,

University of Athens, Ilisia 15784, Athens, Greece

A novel method for the reconstruction of 3D shape and texture from Integral Photography (IP) images is presented. Sharing the same principles with stereoscopic-based object reconstruction, it offers increased robustness to noise and occlusions due to the unique characteristics of IP images. A coarse to fine approach is used, employing a novel grid refinement step in order to increase the quality of the reconstructed objects. The proposed method's unique properties include configurable depth accuracy and direct and seamless triangulation. We evaluate our method using synthetic data from a computer simulated IP setup as well as real data from a simple yet effective digital IP setup. Experiments show reconstructed objects of high quality indicating that IP can be a competitive modality for 3D object reconstruction.



(a) (b)

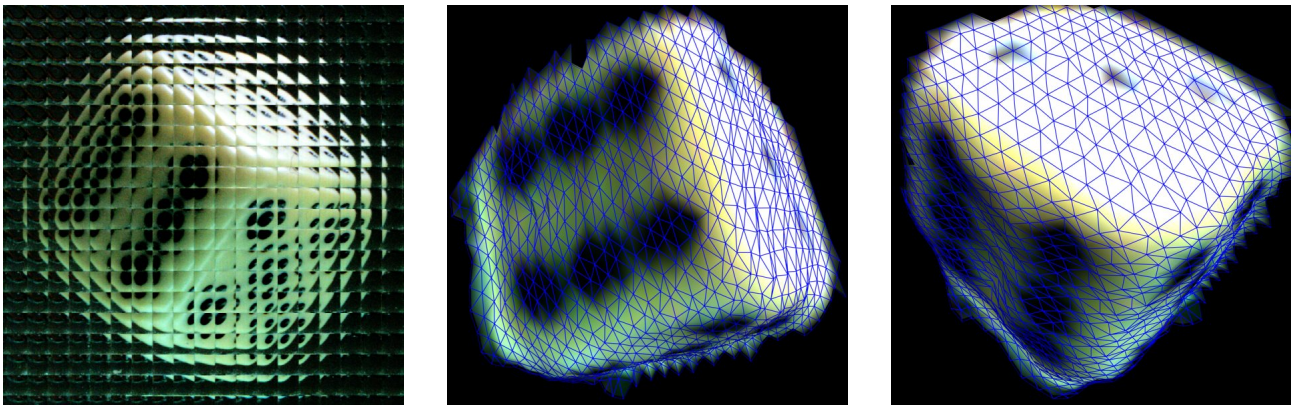
Fig. 1. Operating principles: (a) acquisition and (b) reproduction of IP images.

1. Introduction

Integral Photography is considered as one of the most promising techniques for delivering high quality three-dimensional (3D) content. Even though the working principles[1] of IP were established in 1908, only relatively recently has there been an increased interest in both display[2,3] and acquisition[4,5] setups.

The operating principle of IP is the reproduction of a 3D scene through the use of multiple 2D views. In conventional IP, a Charge Coupled Device (CCD) sensor is placed behind a lens array for acquisition (Fig. 1 (a)), while for display a LCD panel is placed behind the lens array (Fig. 1 (b)). Images produced by IP acquisition setups are pseudoscopic and can be real or virtual. Several techniques[4,5] have been presented to compensate for these effects using optical solutions. In our application, there is no need for real (as opposed to virtual) images so the IP acquisition setup can be simplified. The pseudoscopic effect can be rectified using one of the techniques described by Coral *et al.*[6]

IP images have unique characteristics that can be used for object reconstruction. Contrary to previous approaches our goal is to derive a 3D polygonal model, not just renderings of the object



(a) (b) (c)

Fig. 2. Reconstruction of a dice: (a) Input IP image (52×52 pixels/lens, f=3.3mm) and (b,c) Reconstructed 3D object rendered with triangulation superimposed.

from different views. The proposed method reconstructs quality textured 3D polygonal models (Fig. 2) from a single stage digital IP image capture. Naturally, such a model can then be rendered from arbitrary 3D viewpoints. We employ a coarse to fine approach in order to provide accurate and robust reconstruction of the shape and texture.

1.A. Related Work

In the past, two camera setups (stereovision) were mainly used for 3D object reconstruction. Recently, this has been extended to multiple views, either by the use of multiple cameras[7] or by the use of IP images. A comparative study between the two approaches is presented by Park *et al.*.[8] A number of modifications in the basic IP camera setup like time multiplexed IP cameras[9] has been used in order to increase the resolution of IP cameras.[10,11] Several attempts[9,10,11,12] have been reported in the literature aiming to tackle the problem of 3D object reconstruction using IP images.

The reconstruction algorithms proposed in previous works, are mainly applied to IP images of

small objects that do not span many images with their size. This limitation is enforced in order to avoid stitching problems during the reconstruction stage caused by abrupt discontinuities due to depth estimation errors.[11] Moreover elemental image modification techniques are proposed in an effort to increase depth accuracy.[12] In several works (such as Shin *et al.*[13]) the term *3D object reconstruction* is used to describe the generation of 2D images from multiple views and focus depths from a single IP image.

Our approach does not generate 2D images that represent different viewpoint images or different depth planes but it is focused on the estimation of a fully 3D polygonal model. It is based on a rather simple hardware setup that produces high resolution IP images using a scanning CCD sensor in conjunction with a large area lens array, in order to overcome resolution limitations imposed by the fixed size of the CCD sensor arrays. The proposed 3D reconstruction algorithm efficiently reconstructs shape and texture of the initial 3D object. The reconstruction is based on a matching algorithm similar to the one presented by Park *et al.*[14] but it also handles cases where an object spans multiple lenses. It is avoiding stitching problems by introducing a novel grid refinement step, thus allowing for seamless triangulation. Additionally, our method does not have to rely on IP reshuffling techniques in order to increase the accuracy of the reconstructed object.

1.B. Overview

We propose a novel approach for the reconstruction of 3D shape and texture using IP images. We implemented what we believe is the simplest possible IP acquisition setup. A scanning CCD from a flatbed scanner is placed directly behind a lens array (that uses square lenses), eliminating the need for an objective lens commonly found in most IP acquisition setups. Algorithmically, we extend the classic stereo correspondence problem[15] to two dimensions and multiple correspondences.

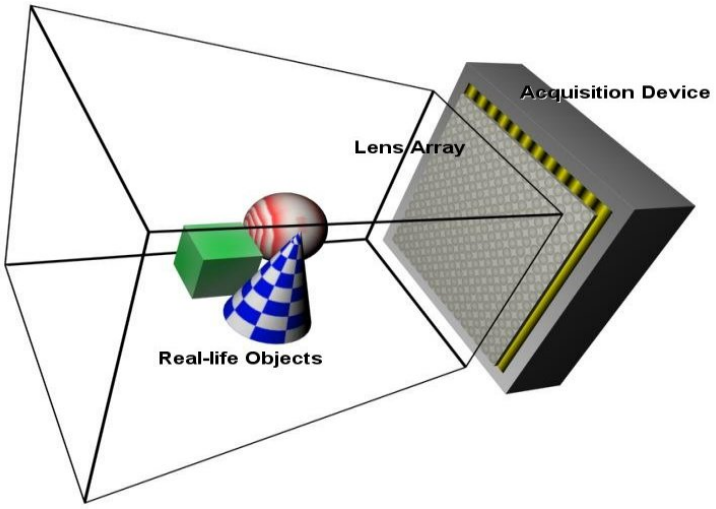


Fig. 3. Hardware setup of the proposed approach.

The output of the process is the 3D polygonal representation of the object's shape and texture. We thus provide a robust and simple way of capturing IP images and yet produce accurate 3D polygonal data using a pure computer vision approach. Finally, we show that this approach has unique properties compared to previous work and alleviates many of the problems that are found even in commercial 3D optical sensors that are based on stereo vision. Unique properties include configurable depth accuracy, direct and seamless triangulation and the ability of handling cases where the field of view of the elemental lenses is less than the acquired object's size at a certain distance.

The rest of this paper is organized as follows: Sec. 2 describes the setup used to obtain the integral photography images, Sec. 3 describes the algorithm that reconstructs the 3D shape from these images, Sec. 4 demonstrates the capabilities of our approach using synthetic and challenging real-life objects, while Sec. 5 summarizes our approach and outlines future directions.

2. Acquisition Setup

Our experimental setup for capturing real world objects uses a scanning CCD and a lens array as depicted in Fig. 3. The scanning CCD sensor allows the capture of images over a 101×127mm area. We utilize the mechanism of a commonly available commercial flatbed scanner as the scanning CCD. This is an inexpensive solution that can offer a scanning area up to 210×297mm. To the best of our knowledge, this is the first application of such setup in Integral Photography using commercial lens arrays.

The lens array provides two dimensional parallax. It uses a square lens geometry over a rectangular grid, as this ensures processing flexibility and has no optical gaps. No objective lens was used, making the setup easier to calibrate and more resilient to optical errors. The characteristics of our lens array are summarized in the following table:

Focal Length	$3.3mm$
Number of Elemental Lenses	$100(H) \times 120(V)$
Elemental Lens Pitch	$1.04mm$

In previous IP acquisition setups the use of a relatively small-sized CCD imposed a limitation on the lenses' pitch. This results in a tight tradeoff between the number of lenses and the resolution of the elemental images. Our setup suffers from no such limitation, since the large scanning area allows for a large number of lenses without sacrificing elemental image resolution. However, due to the scanning CCD, our system cannot handle dynamic objects.

A preprocessing stage is required in order to calibrate the captured IP image. In detail, a small tilt of the lens array during its placement over the CCD sensor results in a skewed IP image, causing registration errors. To this end a skew detection process was developed utilizing the Hough

Transform. Additionally, the positioning errors of the stepper motor moving the CCD over the scanning area were recovered, using a heuristic technique that takes advantage of the IP image's regular structure.[16] Note that this calibration needs to be performed only once, when the lens array is mounted on the flatbed scanner.

3. Reconstruction Algorithm

Our algorithm estimates the 3D shape and texture of an object from a single IP image. To this end a two step process is applied: first, 3D points (vertices) on the surface of the object are computed and second, these points are connected in a polygonal (e.g. triangular) mesh. Our reconstruction algorithm can be summarized as follows:

- *Vertex Grid Computation*: Vertices are computed using the central pixel of each lens, forming a rough regularly sampled vertex grid.
- *Grid Refinement and Triangulation*: The grid is subdivided, new vertices are computed and the refined grid is triangulated.
- *Post-Processing*: The final grid is filtered in order to improve reconstruction quality (e.g. noise reduction).

3.A. The Correspondence Problem

The principle idea behind our approach is an extension of classical stereo vision.[15] In order to estimate the depth of a point in three dimensional Euclidean space (R^3) we need to have its projection in at least two images from cameras with different orientation, and determine the *corresponding* pixels, i.e. pixels in the images of the different cameras onto which the same object point is

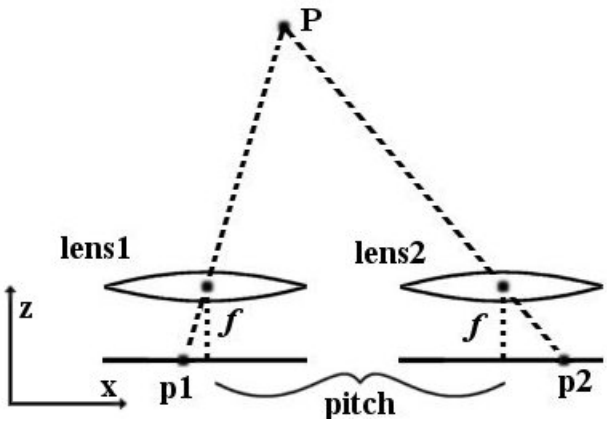


Fig. 4. Point P is projected through two lenses on their image planes at p_1 and p_2. The distance of the optical center from the image plane is equal to f.

mapped. This is known as the *correspondence problem*. The difference from classical stereo vision is that we now have multiple cameras instead of just two as each elemental image essentially corresponds to the image that would be captured by a different camera. In addition these cameras offer horizontal as well as vertical parallax.

We model our lenses in the array as perfect cameras that use perspective projection. A homogeneous vertex $P = [x\ y\ z\ 1]^T$ in R^3 is projected onto $p = [u\ v]^T$ in R^2 as follows:

$$\begin{bmatrix} U \\ V \\ H \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

and $u = \frac{U}{H}$ and $v = \frac{V}{H}$ where f is the focal length of the lenses.

As in stereo vision, in order to solve the correspondence problem for a point P in R^3 we need two projections (p_1 and p_2) of this point in elemental images from different lenses. If the coordinates of

P are defined in the coordinate system of the first lens, then they can be expressed in the coordinate system of the second lens as $P' = \mathbf{M} \cdot P$ where $\mathbf{M}$ is a 4×4 transformation matrix computed during the calibration of the acquisition setup. For each pixel in the first elemental image, we seek its corresponding pixel in the second elemental image along a line (which consists of candidate corresponding pixels), using a minimum difference criterion (see next section).

In practice, due to the regularity of the lens array, the lenses do not need calibration. By construction, they have parallel orientations and the distance between two successive optical centers is always equal to the lenses' $pitch$ in each direction. In the example of Fig.4, the transformation matrix for the two adjacent lenses shown is $\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & pitch \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. From the above equations, for given $p_1 = [u_1 v_1]^T$ and $p_2 = [u_2 v_2]^T$, we derive the following: $v_1 = v_2 = -f\frac{y}{z}$, $u_1 = -f\frac{x}{z}$ and $u_2 = -f\frac{x+pitch}{z}$ from which P is estimated (3 equations with three unknowns). The determination of corresponding pairs $[u_1 v_1]^T$ and $[u_2 v_2]^T$ is the subject of the next section.

3.B. Vertex Grid Computation: Determining Correspondences

Given an IP image produced by a lens array with known focal length f, we first compute the 3D vertices that correspond to the central pixels of each elemental image. These vertices form a regularly sampled grid that is refined in the following step of the algorithm. Note that all elemental images have the same pixel resolution R which is determined by the acquisition device.

We define the distance $D(p_1, p_2)$ between two pixels ($p_1 = [u_1 v_1]^T$ and $p_2 = [u_2 v_2]^T$) from

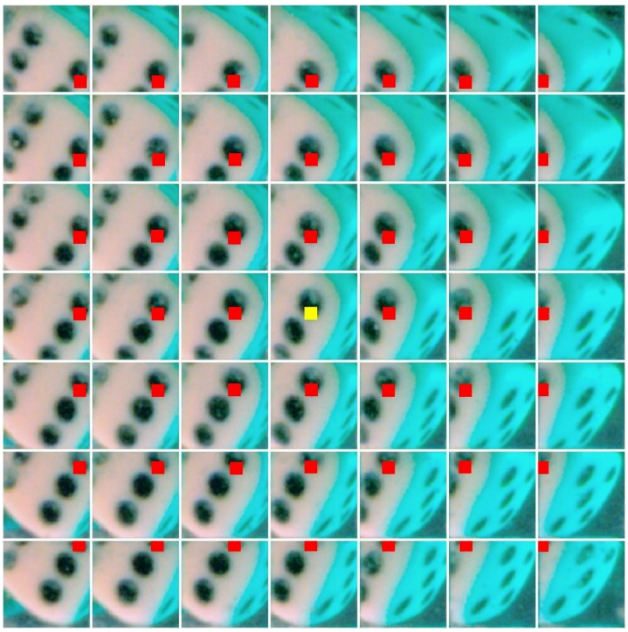


Fig. 5. An IP image of a dice using 124×124 pixels per elemental image, $f = 3.3mm$ and $N = 3$: A pixel (marked yellow) and its correspondences (marked red). The colored boxes denote the window area ($W = 8$).

different elemental images using a simple but effective L_1 metric:

$$D(p_1, p_2) = \sum_{j=-W}^{W} \sum_{i=-W}^{W} |E_1(u_1 + i, v_1 + j) - E_2(u_2 + i, v_2 + j)|$$

where E_1 and E_2 are the two elemental images, and W defines the size of the comparison window area. The value of W depends on the size of texture patterns in the elemental images. If the window intersects the elemental image's boundary, we only use the pixels inside the elemental image and normalize the total distance by dividing by the number of pixels used.

We subsequently extend the above distance metric to more than two elemental images. In practice, we use $2N + 1$ neighboring elemental images per direction, thus forming a symmetrical neighborhood area of radius N around each elemental image (Fig.5). The best correspondence has

the minimum sum of the distances over all neighbors, as follows:

For the central pixel $p_{k,l}$ of each elemental image $E_{k,l}$

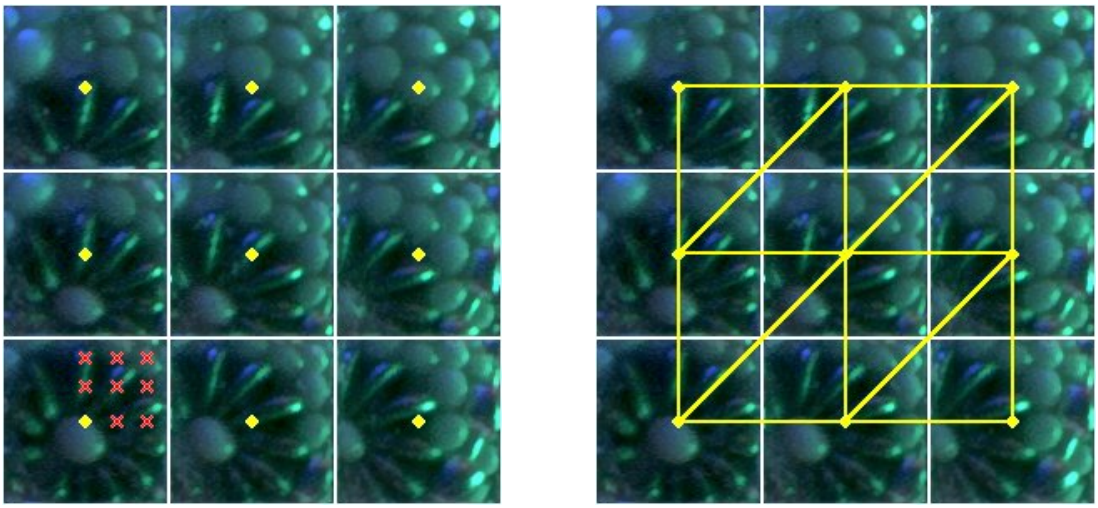
- Find the 3D vertex P that minimizes the expression:

$$D_{total} = \sum_{j=-N}^{N} \sum_{i=-N}^{N} D(p_{k,l}, p_{k+i,l+j})$$

  where $p_{k+i,l+j}$ is the projection of P in elemental image $E_{k+i,l+j}$.

In the above algorithm, the vertex P lies on the line through the central pixel of elemental image $E_{k,l}$ that is perpendicular to the image plane. Therefore the search of the optimal vertex requires only to sample this line at various points (that correspond to different depth values) and then projecting them to neighboring elemental images to evaluate their distance.

The use of multiple images to determine correspondences is a unique feature of IP based reconstruction approaches. The inherent advantage is that it offers more robustness compared to a stereo vision approach, as problems like partial occlusion or noisy data can be overcome. If we determine that the distance of the best correspondence of a given neighbor is above a certain threshold we can discard that neighbor, and compute the total difference from the remaining neighbors. Depending on the expected quality of the IP image, we can differentiate our strategy, for example using median instead of mean in the computation of D_{total}. Note that since our acquisition setup does not produce noisy data, we did not use the median in the computation of D_{total} in any of the examples in this paper.



(a) (b)

Fig. 6. Elemental images from a real-life object with 98×98 pixels per lens and $f = 3.3mm$: (a) A 3×3 neighborhood with central pixels marked yellow. The correspondences of the central pixels in the bottom-left elemental image are marked red. (b) Triangulation of the same neighborhood using only central pixels, projected onto R^2 and superimposed over the IP image.

3.C. Vertex Grid Refinement and Triangulation

In order to refine the vertex grid, we introduce a subdivision parameter S, which defines how many vertices we will compute between the computed central vertices. The reason for separating this step from the previous one is to allow seamless triangulation. As seen in Fig. 6 (a) we can project the central pixels from neighboring elemental images (computed in the previous step) onto a specific elemental image. Additional vertices that we will use for refinement will be derived only from pixels between the central pixels of the lens and the correspondences of the central pixels of the neighboring lenses. The algorithm that subdivides the grid of a lens can be summarized as follows:

Let the current lens be denoted by L_{00}, and its up and diagonal neighbors by L_{10},L_{01} and L_{11} respectively. Let the 3D vertices computed using the central pixel of the above lenses (in the previous step) be denoted by V_{00}, V_{10}, V_{01} and V_{11}.

1. Project V_{00}, V_{10}, V_{01} and V_{11} in L_{00} as p_{00}, p_{10}, p_{01} and p_{11} respectively.

2. For j from 1 to $S-1$ do

   For i from 1 to $S-1$ do

   Estimate $p' = \frac{(S-i)\cdot(S-j)}{S} \cdot p_{00} + \frac{i\cdot(S-j)}{S} \cdot p_{10} + \frac{(S-i)\cdot j}{S} \cdot p_{01} - \frac{j\cdot i}{S} \cdot p_{11}$,

   Compute the reconstructed vertex V' using the algorithm of Sec. 3.B

   and add it to the grid.

In Fig. 6 (b) a triangulation example is shown without refinement. Since the refined vertices are ordered, the refined grid remains regularly sampled thus allowing straightforward triangulation. An issue with this approach is that errors introduced in the computation of the initial vertex grid (Sec. 3.B), can cause undesired effects in the refinement step (Sec. 3.C).

Our approach has the unique property of producing a seamless triangulation without the need for stitching among different neighbors. Compared to other approaches,[12] our method does not have to reshuffle the initial IP image.

3.D. Post-Processing Filters

In order to improve the quality of the final reconstruction, we apply a series of post-processing filters on the grid of vertices. The filters are applied in the following order:

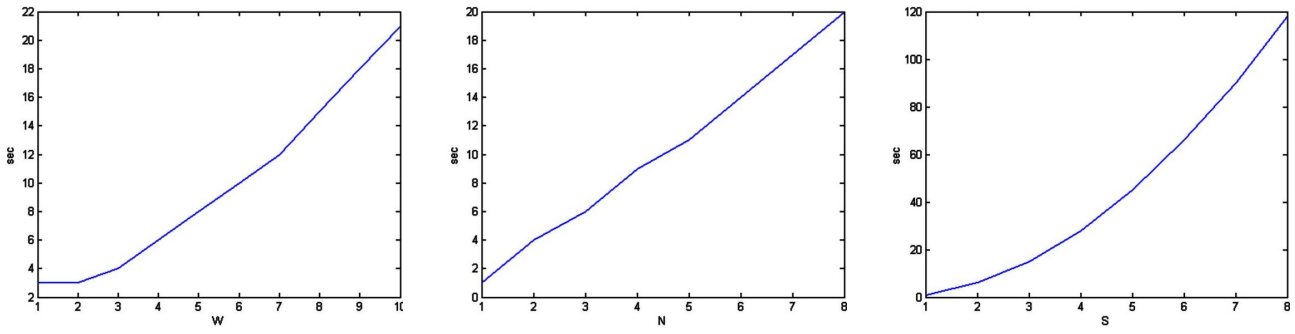
- *Median Cut:* A median cut filter with a 3×3 kernel is used to remove spikes. Spikes are introduced by incorrect reconstruction. If this filter is omitted, the error is diffused into neighboring vertices by the subsequent two filters resulting in bulges on the polygonal surface.

- *Smoothing:* A smoothing filter is applied to remove white noise introduced by the acquisition process. A 3×3 kernel is utilized, allowing smoothing only with the immediate neighbors.

- *Sub-sampling:* The reconstruction algorithm can compute as many vertices as desired, up to the number of pixels per lens divided by two. Note that using more vertices than the number of pixels does not offer any real increase to the resolution. Such a reconstruction resolution is well above the requirements of most real-life applications. Instead of reconstructing the object directly in the desired lower resolution, the object can be reconstructed at a higher resolution. Subsequently, the grid of vertices can be sub-sampled to obtain the lower resolution. This results in reconstructions of higher quality and robustness but has an increased computational cost.

3.E. Computational Requirements

Due to the two dimensional nature of the problem, the theoretical computational requirements of the reconstruction process are $O(W^2N^2S^2)$. However, due to several optimizations in the implementation, we achieved almost linear behavior for all three parameters within the useful parameter range. An optimization example is the use of only vertical horizontal neighbors in the matching



(a) (b) (c)

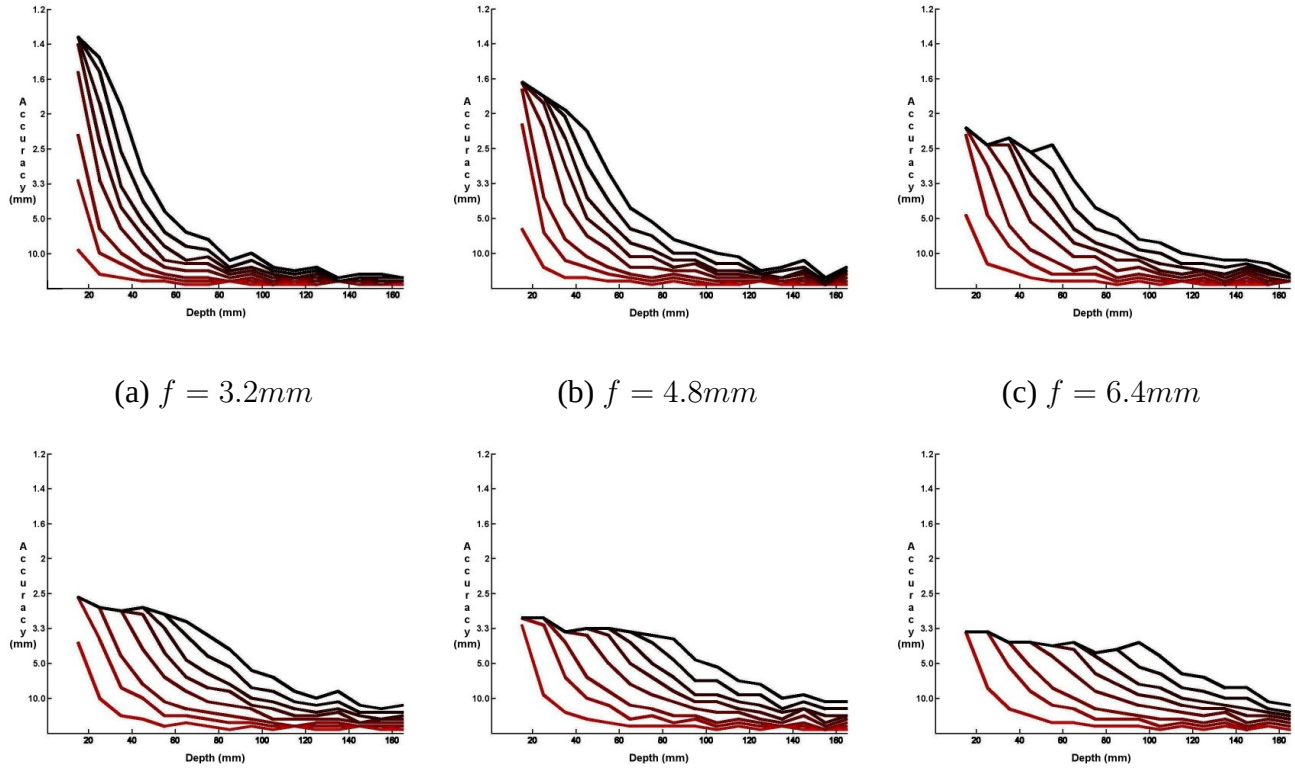
Fig. 7. Computational requirements of reconstruction: (a) Varying W with $S = 1$, $N = 3$; (b) Varying N with $S = 1$, $W = 4$ and (c) Varying S with $N = 1$, $W = 1$.

process. We performed an experiment to practically measure the computational requirements of the reconstruction process. We used 64×64 elemental lenses, with 64×64 pixels each. The focal length was 3.0 and the pitch was 1.0. We varied the N, W and S parameters inside their useful range. The results are depicted in Fig. 7. All measurements were carried out on a Pentium 4 3.0Ghz with 1GB of RAM.

3.F. Depth Accuracy Analysis

In a classical stereo vision reconstruction approach, the depth accuracy depends only on the number of pixels in the stereo images. By increasing the image resolution, the number of different depth values that can be computed increases, thus increasing the sensor's accuracy. The accuracy generally decreases proportionally to the distance from the cameras. This property may be useful if we want more accuracy near the cameras, but is a drawback if we require even distribution of the accuracy across the entire depth range of the sensor.

Our lens array based reconstruction offers a configurable depth accuracy distribution. In our



(a) $f = 3.2mm$ (b) $f = 4.8mm$ (c) $f = 6.4mm$

(d) $f = 8.0mm$ (e) $f = 9.6mm$ (f) $f = 11.2mm$

Fig. 8. Accuracy distribution for variable focal length f (from 3.2 to 11.2), using $R = 32 \times 32$. Each plot shows graphs utilizing from $N = 1$ to 8 neighbors (color coded from bright red to black respectively).

novel approach, the depth accuracy is a function of the parameters N, R and f. The effect of parameter R is straightforward: the more pixels we have in each elemental image, the higher the reconstruction accuracy. Parameter N affects the accuracy in a similar way: the more neighbors we utilize, the more pixels we take advantage of and thus the greater the depth accuracy. The interesting feature is that each successive neighbor does not distribute its additional depth accuracy across the entire depth range, but it only affects the accuracy after a certain depth. This depth is

equal to $2 \cdot k \cdot f$ for the k^{th} neighbor. Therefore, the effect of the focal length is twofold: it determines how fast the depth accuracy of each lens dissipates, and it controls the interleave between the useful depth ranges of successive lenses.

This property is demonstrated in the graphs of Fig. 8. In this figure, the accuracy distribution was practically measured, using 32×32 pixels per lens and varying focal lengths (from 3.2 to 11.2). Note that our practical measurements conform with the theoretical depth accuracy limits of parallel camera setups described by Son *et al.*.[17] Within each graph, the accuracy distribution for varying number of neighbors ($N = 1$ to 8) is depicted. In lower focal lengths (thinner lens arrays), each lens distributes most of its depth accuracy at a near range and at the same time there is substantial interleave between the depth distributions of successive lenses. This results in peak accuracy near the lenses and a fast accuracy degradation away from the lenses. In higher focal lengths there is less interleaving, slower accuracy degradation, but also a lower peak value. Therefore, the focal length is the key to configure the accuracy distribution. Note that the total accuracy (total number of distinguishable depth values) is the same in all cases, since it is in fact cumulative and depends only on N and R.

In a practical application, for a given target accuracy, we can estimate the required N, R and f parameters in order to maintain it stable for a given range. For example, in Fig. 8, with a focal length of 11.2 and using 8 neighbors, we can maintain an almost stable depth accuracy of $5mm$ for depths ranging from $2cm$ to $14cm$. For resolutions of 64×64 and 128×128 this accuracy is improved to $2.5mm$ and $1.25mm$ respectively.

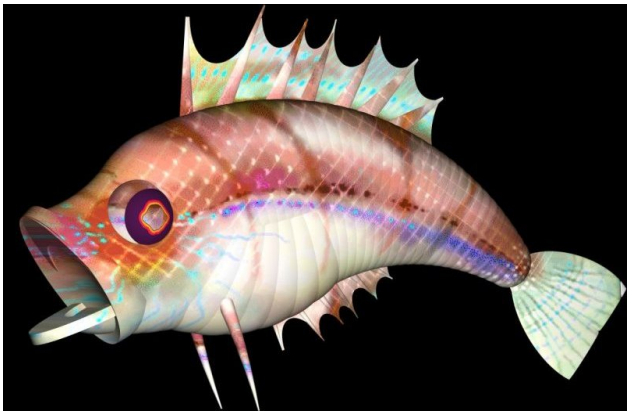


Fig. 9. Synthetic fish used in the virtual lens array experiments.

4. Results

We present results using both synthetic and real-life data. We utilize the synthetic data in order to demonstrate the effect of the focal length of the lenses on the quality of the reconstructed object. We utilize the real data in order to demonstrate the robustness and accuracy of our prototype setup and our algorithmic approach. For visualization purposes we show both the IP images used for reconstruction and the resulting reconstructed object.

4.A. Synthetic Data

Using a commercial ray tracer (PovRay) we implemented a virtual lens array setup[18] with similar properties to the real one utilized in the prototype setup. The virtual lens array's parameters are configurable; in our experiments we opted for varying the focal length while retaining all other parameters unchanged. We used 64×64 pixels per lens and 64×64 lenses. The pitch was $1mm$. This setup has a total resolution of 4096×4096 pixels and a scanning area of $64 \times 64mm$. The synthetic object we used for the experiment was a fish of approximate size $40mm$ along its principal

axis (Fig. 9). We used three different focal lengths ($3mm$, $6mm$ and $9mm$) thus obtaining three IP images, depicted in Fig. 10 (a,b,c). In order to keep the object inside the useful depth range of the lenses we positioned it at $12mm$, $24mm$ and $36mm$ away from the lens array respectively, a distance equal to $4f$. We performed 3D reconstruction using $W = 6$, $N = 4$ and $S = 4$ for all three cases and the results are visualized in Fig. 10 (d,e,f). Note that the texture information used in the renderings of Fig. 10 is acquired by the same setup.

The quality of the reconstructed polygonal object in all three cases is almost identical. No discrepancies were noted demonstrating our method's robustness to different focal lengths. However, a detailed visual inspection of the results indicated that the reconstruction with the higher focal length was of slightly better quality in certain areas. Synthetic experiments, such as the one described in this section, combined with the analysis of depth accuracy of Sec. 3.F allow the selection of the optimum lens array for a given hardware acquisition device.

4.B. Real Data

We acquired real IP data using the prototype setup described in Sec. 2. We reconstructed a small hemispherical case with rough texture. We selected this object because of its simple shape that allowed us to evaluate the reconstruction visually. The scanned IP image is depicted in Fig. 11 (a). Note that we used a scanning resolution close to 2400dpi to achieve a resolution of 98×98 pixels per lens (elemental image). In practice however, the real scanning resolution of the device is significant lower than the one specified by the manufacturer.

We depict the resulting reconstruction in Fig. 11 (b, c). The reconstructed polygonal surface is rendered with the acquired texture and with the triangulation superimposed. It can be seen that the reconstruction is of high quality with no significant artifacts. In Fig. 11 (d, e, f) we depict the

reconstruction using a more refined triangular grid (S is 2, 3 and 4 respectively); the triangulation is not superimposed because the grid is extremely dense and the object's texture would not be visible. Note that the background in the IP image is not reconstructed correctly; this could be dealt with using a background removal algorithm.

5. Conclusion

The use of an IP system for the reconstruction of 3D shape and texture purposes is advocated. An algorithmic method and a hardware setup that allows robust and accurate estimation of a polygonal 3D surface of real life 3D objects is presented. Using a scanning CCD and a lens array, an IP is acquired that is subsequently processed by a novel algorithm in order to compute the 3D reconstruction of shape and texture.

It is shown that our system has unique properties over other optical approaches. The seamless triangulation regardless of the object's position in the lens array field of view maximizes the effective range. The configurable depth accuracy allows the adjustment of the setup based on the application needs in order to optimally utilize the available depth resolution. The redundancy provided by the multiple views increases robustness.

Future work will be directed towards scanning various real life textureless objects. These objects posed a significant challenge to optical scanners in the past, since the lack of texture made the solution of the correspondence problem difficult. Most optical systems rely on active devices (e.g., structured light projectors) to tackle this issue. We will investigate how our system's optical redundancy may alleviate this problem.

References

1. G. Lippmann, “La photographie integrale,” *C.R. Acad. Sci.*, vol. 146, pp. 446–455, 1908.
2. J. Son and B. Javidi, “Three-dimensional imaging methods based on multview images,” *IEEE/OSA Display Technology*, 2005.
3. S. Nakajima, K. Nakamura, K. Masamune, I. Sakuma, and T. Dohi, “Three-dimensional medical imaging display with computer-generated integral photography,” *Comp. Med. Imag. Graphics*, vol. 25, no. 3, pp. 235–241, 2001.
4. J. Jang and B. Javidi, “Two-step integral imaging for orthoscopic three-dimensional imaging with improved viewing resolution,” *Optical Engineering*, vol. 41, no. 10, pp. 2568–2571, 2002.
5. J. Jang and B. Javidi, “Formation of orthoscopic three dimensional real images in direct pickup one-step integral imaging,” *Optical Engineering*, vol. 42, no. 7, pp. 1869–1870, 2003.
6. M. Corral, B. Javidi, R. Cuenca, and G. Saavedra, “Formation of real orthoscopic integral images by smart pixel mapping,” *Optics Express*, vol. 13, pp. 9175–9180, 2005.
7. H. Kim, “A 3D Modelling System Using Multiple Stereo Cameras”, *Ph.D thesis*, Yonsei University, Seoul, Korea, August, 2005.
8. J. Park, H. Choi, B. Lee “Comparative study of integral imaging and multi-camera pickup for acquisition of three-dimensional information”, *Conference on Optical Information Systems*, SPIE Annual Meeting, vol. 5202, San Diego, CA, USA, pp. 158–167, Aug. 2003.
9. S. Kishk and B. Javidi, “Improved resolution 3D object sensing and recognition using time multiplexed computational integral imaging,” *Optics Express*, vol. 11, pp. 3528–3541, 2003.
10. Y. Frauel and B. Javidi, “Digital three-dimensional image correlation by use of computer-reconstructed integral imaging,” *Applied Optitcs*, vol. 41, pp. 5488–5496, 2002.

11. J. Park, Y. Kim, J. Kim, S. Min, and B. Lee, “Three-dimensional display scheme based on integral imaging with three-dimensional information processing,” *Optics Express*, vol. 12, pp. 6020–6032, 2004.

12. J. Park, S. Jung, H. Choi, Y. Kim, and B. Lee, “Depth extraction by use of a rectangular lens array and one-dimensional elemental image modification,” *Applied Optics*, vol. 43, pp. 4882–4895, 2004.

13. D. Shin, E. Kim, B. Lee, “Computational reconstruction of three-dimensional objects in integral imaging using lenslet array”, *Japanese Journal of Applied Physics*, vol. 44, no. 11, pp. 8016–8018, 2005.

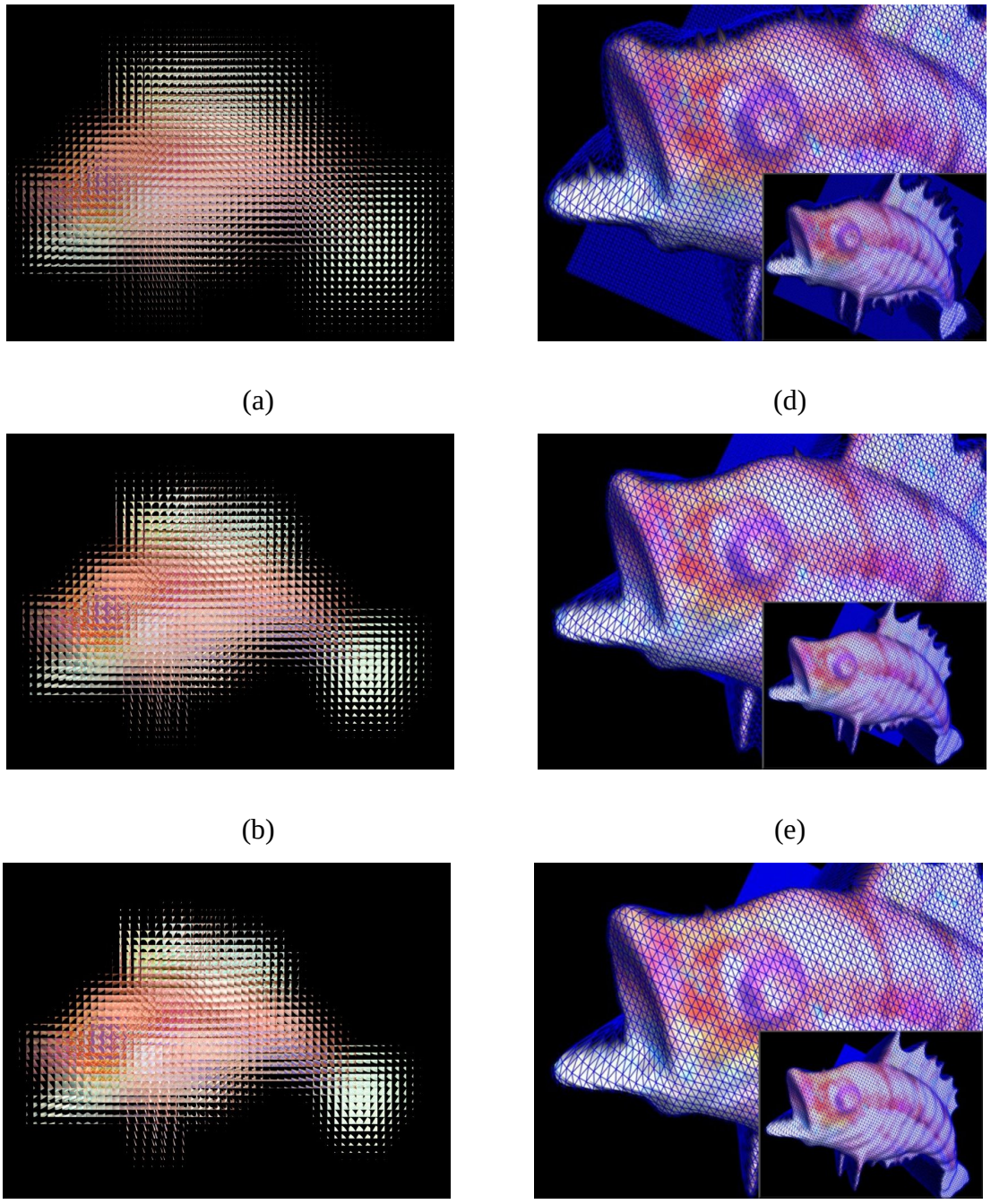
14. J. Park, S. Min, S Jung, B. Lee, “A new stereovision scheme using a camera and a lens array”, *Conference on Algorithms and Systems for Optical Information Processing V*, SPIE Annual Meeting, vol. 4471, San Diego, CA, USA, pp. 73–80, Jul.–Aug. 2001.

15. O. Faugeras, *Three-Dimensial Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

16. N. Sgouros, S. Athineos, M. Sangriotis, P. Papageorgas, N. Theofanous, “Accurate lattice extraction in integral images”, *Optics Express*, vol. 14, pp. 10403–10409, 2006.

17. J. Son, V. Bobrinev, K. Kim, “Depth Resolution and Displayable Depth of a Scene in 3 Dimensional Images”, *Journal of Optical Society of America A*, vol. 22, no. 9, pp. 1739–1745, 2005.

18. S. Athineos, N. Sgouros, P. Papageorgas, D. Maroulis, M. Sangriotis, N. Theofanous, “Photorealistic Integral Photography using a Ray Traced Model of Capturing Optics”, *Journal of Electronic Imaging*, vol. 15, no. 4, pp. 043007–043014, Oct.–Dec. 2006.

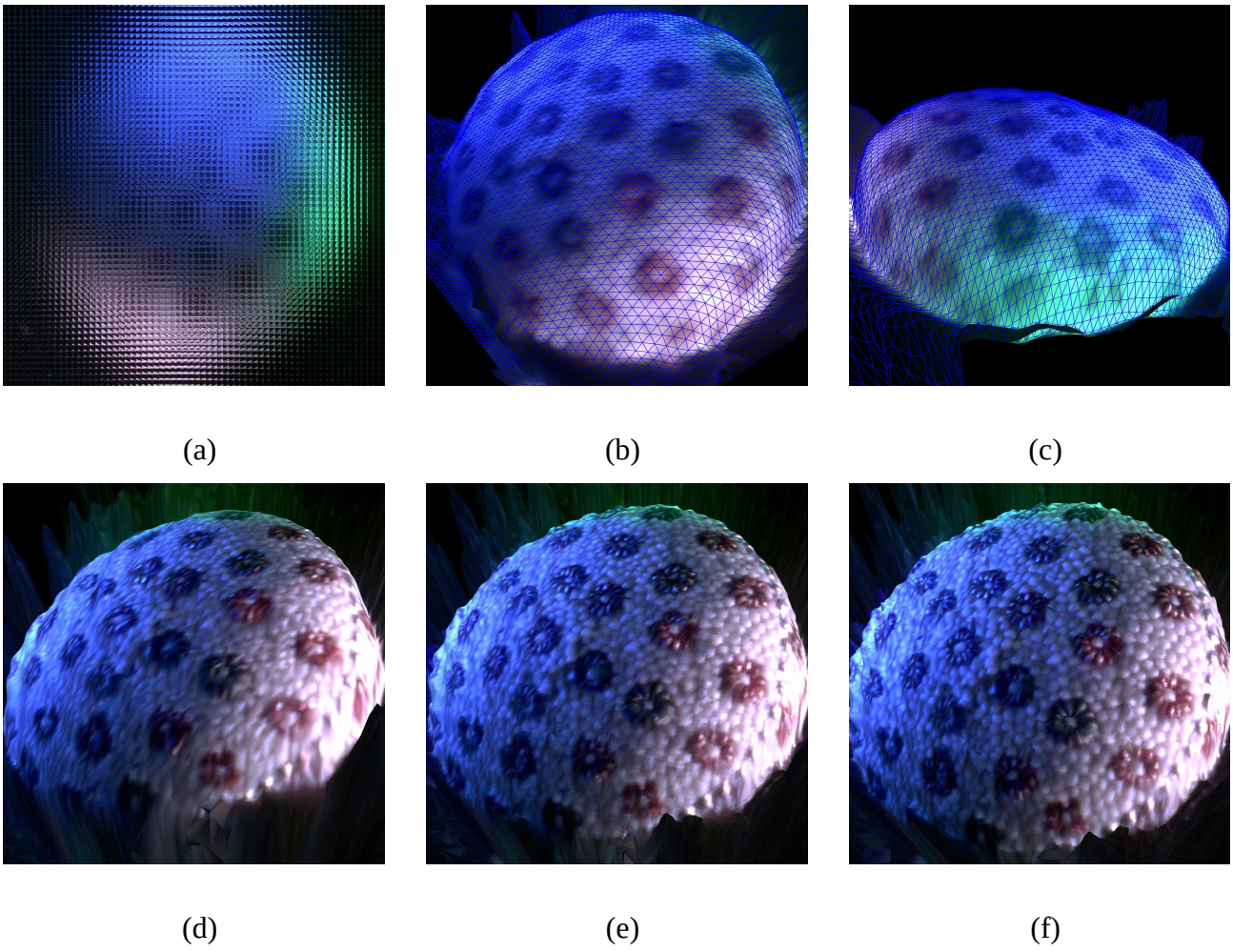


(a) (d)

(b) (e)

(c) (f)

Fig. 10. Reconstruction of a synthetic fish (Fig. 9) for variable focal lengths ($3mm$, $6mm$ and $9mm$ from top to bottom) using 64×64 pixels per lens, 64×64 lenses (elemental images) and $1mm$ pitch: (a,b,c) Input IP images and (d,e,f) corresponding renderings of reconstructed 3D object with the surface triangulation superimposed.



(a) (b) (c)

(d) (e) (f)

Fig. 11. Shape and texture reconstruction of a real-life object: (a) Input IP image using 98×98 pixels per lens and $f = 3.3mm$; (b,c) Renderings of reconstructed 3D object with the surface triangulation superimposed (using $S = 1$); (d,e,f) Renderings using more subdivision steps ($S = 2$, 3 and 4 respectively).