13th International Symposium on Particle Image Velocimetry – ISPIV 2019
Munich, Germany, July 22-24, 2019

Particle Matching and Triangulation using Light-Field Ray Bundling

Chris Clifford[1], Zu Puayen Tan[1], Elise Hall[1], Brian Thurow[1*]

[1] Advanced Flow Diagnostics Lab, Auburn University, Auburn, USA

[*] thurow@auburn.edu

Abstract

This paper aims to provide a 3D high-resolution spatial matching and triangulation algorithm called Light-Field Ray Bundling. The algorithm represents particle observations as rays in 3D space, then attempts to "bundle" those rays by identifying common intersections. A bundle of rays represents a matched list of observations from different perspectives for one particle. That list of observations is combined with a light-field calibration to retrieve the (x,y,z) coordinates of said particle in object space. The work presented here demonstrates quick execution times and high accuracy for volumes with low to moderate particle densities. At high particle densities, an iterative approach performs nearly as well as volumetric reconstructions. This preliminary work shows promising results of the Light-Field Ray Bundling approach with substantial potential for further improvements.

1 Introduction

Particle tracking velocimetry (PTV) provides a number of advantages over correlation-based particle image velocimetry (PIV). Most notably, PTV enables the acquisition of a large number of long particle trajectories in both space and time (Maas et al., 1993). This facilitates detailed study of complex flow physics, but is currently limited to modest particle densities. PTV can be broadly divided into three sequential components: ① image segmentation and identification, ② spatial matching and triangulation, and ③ temporal matching and tracking. At high particle densities, ambiguity in the first two components limit the spatial resolution of PTV (Maas et al., 1993). This paper aims to provide a 3D high-resolution spatial matching and triangulation algorithm that clusters light rays directly in ray-space by utilizing the calibrated light-field of a plenoptic camera.

Direct light-field calibration (DLFC; Hall et al., 2017) uses polynomials to describe a direct mapping from object to image space for a plenoptic camera. This mapping takes the form $s = \mathcal{P}_s(x,y,z,u,v)$ and $t = \mathcal{P}_t(x,y,z,u,v)$ where (s,t) are coordinates on the image plane, (u,v) are coordinates on the aperture plane, and (x,y,z) are object space coordinates in the experimental volume. For tomographic reconstruction algorithms, this mapping replaces the weighting matrix allowing for efficient reconstruction of an arbitrary voxel without sampling every pixel on the image sensor. For gridless particle triangulation, the inverse is required: given matched pairs of (s,t) and (u,v) values (*i.e.* given a series of observations from multiple known viewing angles), determine the (x,y,z) position in object space. If a list of (s,t,u,v) can be provided for a given particle, then (x,y,z) may be solved using a simple least-squares method. Since plenoptic cameras often have 100 or more unique views, this triangulation method is robust against missing or corrupt views (Hall et al., 2019a,b). The challenge becomes determining the uniqueness of particles in a field and thus assigning them the correct list of samples (*i.e.* spatial matching).

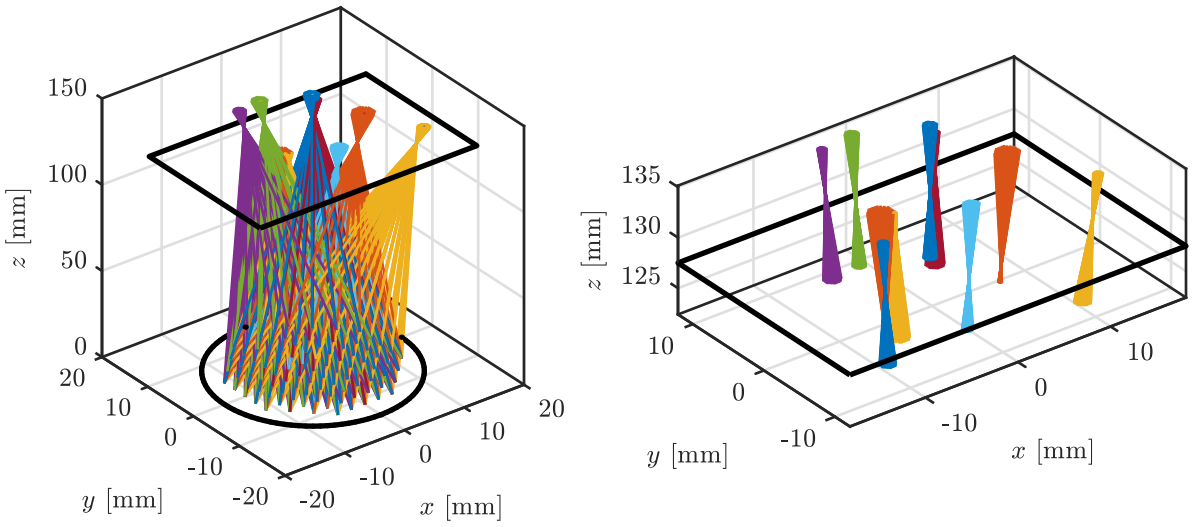


Figure 1: Ray-space visualization of a ten-particle field. Left: The full ray-space from aperture plane (circle) through image plane (rectangle). Right: The ray-space near the image plane (rectangle) where particles exist.

The authors propose a geometric approach using two-plane parametrization of the ray-space. As described by Levoy (2006), each ray in the light field can be uniquely described using a point on two planes. For convenience, we choose the *s*-*t* and *u*-*v* planes, which correspond with the image and aperture planes, respectively. The observed ray-space of a ten-particle field is shown in Fig. 1. From this figure, the location (in image space) of each particle is apparent: each particle corresponds to a tight bundle of rays, forming a double cone with a particle at the apex. Each bundle has been given a unique color to aid in visualization, but the color of each ray is not known *a priori*. In fact, the problem may be recast in the following form: how do we assign a color (*i.e.* a unique identifier) to each ray such that the resulting bundles are uniform? The proposed solution is to **calculate the minimum distance between rays** from each view, then **cluster those rays into bundles** based on the distance and location of closest approach. A least-squares fit may then be applied to each bundle using the DLFC polynomials to triangulate the object space coordinates (x,y,z).

The proposed ray bundling algorithm, dubbed Light-Field Ray Bundling (LFRB), functions as follows. For each (u,v) perspective view, a particle finding algorithm finds the weighted centroids (s,t) of each visible particle. Each of these (s,t,u,v) positions defines a unique ray through 3D space. One of these rays from the first perspective view is chosen as a reference. The minimum distances between the reference ray and all rays from the other perspectives are calculated. If the resultant distance is suitably close, then it is considered for inclusion in the bundle. To be included, the point of closest approach must occur within the experimental volume (defined by the user) and must also be suitably close to previously bundled rays. This process is repeated until all rays in the first perspective are claimed. If additional rays are unclaimed in the remaining perspectives, then the process is repeated using the next perspective view as the reference, until all rays from all views have been claimed or discarded. A least-squares fit is then applied to each bundle using DLFC and the (x,y,z) coordinates are recovered.

2 Light-Field Ray Bundling

The details of the LFRB algorithm are included here with additional discussion regarding alternative methods. The specifics of this algorithm are tailored toward a single plenoptic camera, but are adaptable to any light-field sampling system.

Generate perspective views from plenoptic image. Plenoptic data in its raw form consists of over a hundred thousand, tightly packed, sub-aperture images corresponding with each microlens. This packing is often (but not necessarily) hexagonal, which is inconvenient to work with directly. Thus, the light-field data is resampled into a more amenable form by generating a sequence of traditional images, each corresponding with a unique perspective. The number of views N_v available without oversampling is approximately equal to the number of pixels behind each microlens. After resampling, each view has a (u,v) coordinate and each pixel has an (s,t) coordinate. For a single plenoptic camera, this may be done entirely in image space where (s,t) lies on the image plane and (u,v) lies on the aperture plane, enabling the use of two-plane parametrization. For multiple cameras (plenoptic or otherwise), it would be necessary to utilize a global coordinate system shared by all views.

Segment and identify particles in each view. Before being triangulated in 3D space, the particles must first be identified on the 2D images. For each perspective view, centroids are found for as many particles as possible. At low particle densities, this may be accomplished with simple blob finding techniques, such as MATLAB's `regionprops` or OpenCV's `SimpleBlobDetector`. At higher densities, such implementations will agglomerate multiple particles into a single blob. The image segmentation used in this work is based on `dynamic_image_segmentation_v3` from Prana (Aether Lab, 2015; Eckstein and Vlachos, 2009a,b) which excels at finding overlapping particles. First, a static threshold is applied to the image to remove background noise. For high noise levels, background subtraction and/or dynamic thresholding might be necessary. Local maxima are used to find peaks, where each peak is considered a particle. The extent of each particle is found by performing dilation about each peak until the previous threshold or another particle is encountered. Finally, the weighted centroid is calculated using the particle extents and the image intensity. From this process, a list of all (discovered) particles from all views is produced and ready for matching.

The success of particle matching and triangulation relies upon the sub-pixel accuracy of the image segmentation. Intensity-weighted center-of-area methods, such as that used here, suffer from background and sensor noise. More advanced methods, such as mask correlation (Ohmi and Li, 2000; Takehara and Etoh, 1998) and Gaussian fitting (Nobach and Honkanen, 2005; Rogers et al., 2007), offer several advantages. These methods tend to detect only particle-like objects, based on the mask or model definition, and automatically handle overlapping particles with high accuracy. Advanced image segmentation would likely benefit the following steps, and thus the overall algorithm.

Represent particle centroids as rays. The LFRB matching procedure is based on ray bundles, which necessitates that all centroids be represented as light rays propagating through 3D space. As implemented here, this is done on-the-fly during the bundling process, but may require a discrete step for multi-camera setups. A ray propagating through space can be represented as

$$\mathbf{R} = \vec{p} + \lambda \vec{r} \tag{1}$$

where $\vec{p}$ is a point in space, $\vec{r}$ is the direction of propagation, and λ is a parametric variable. For the remainder of this paper, boldface $\mathbf{R}$ will represent the entire ray, while a vector hat $\vec{r}$ represents only the propagation vector. Using two-plane parametrization, a centroid may be represented as a ray by

$$\mathbf{R} = [\ u,\ v,\ 0\ ]' + \lambda[\ s-u,\ t-v,\ l\ ]' \tag{2}$$

where (s,t) are the image-plane coordinates, (u,v) are the aperture-plane coordinates, and l is the distance between planes.

Build bundles of rays. A bundle is initialized using one ray from the first view. This ray serves as the reference or anchor for building the remainder of bundle. In the next view, each ray is a

candidate for inclusion in the bundle. Note that rays from the same view are never compared, since they obviously belong to different particles. For a candidate ray to be included in the bundle, it must pass several checks. First, how far is the candidate (s,t) from the reference (s,t)? If the distance exceeds that predicted by the shift in (u,v), then the candidate is from another particle and may be safely skipped without further computation. Next, the minimum distance between rays is calculated. Given two rays **A** and **B** as illustrated in Fig. 2, there exists a line of closest approach. The length of that line is given by

$$d = \left| \frac{(\vec{p}_B - \vec{p}_A) \cdot \vec{n}}{\vec{n} \cdot \vec{n}} \right| \tag{3}$$

where $\vec{n} = \vec{a} \times \vec{b}$ is the normal vector between rays. The closest approach between the reference and all candidates from this view is calculated and sorted in ascending order. Thus, the first candidate is the one which passes closest to the reference. If the distance d is within some threshold (the "maximum approach distance"), then the rays are considered to cross and a particle might exist at their crossing. If the current size of the bundle is one (*i.e.* only the reference), then the closest candidate is included in the bundle. However, if the bundle has more than one ray, an additional check is necessary.

As illustrated in Fig. 3, an occluded particle may have multiple valid crossings. In this figure, the solid green line is the reference ray; the dashed green line is an already included ray; and the magenta and teal lines are candidate rays for inclusion. Note that both candidates originate from the same (u,v), since they belong to the same view. It is possible that the teal ray passes closer to the reference than does the magenta ray; however, it is apparent that the teal ray is not associated with the same particle. To address this problem, the approximate (uncalibrated) location of the particle is tracked. The end points of the closest approach line are given by

$$\vec{p}_{A|B} = [\ u_A,\ v_A,\ 0\ ]' + \frac{(\vec{b} - \vec{a}) \cdot \vec{n}_B}{\vec{a} \cdot \vec{n}_B} \vec{a} \tag{4}$$

$$\vec{p}_{B|A} = [\ u_B,\ v_B,\ 0\ ]' + \frac{(\vec{a} - \vec{b}) \cdot \vec{n}_A}{\vec{b} \cdot \vec{n}_A} \vec{b} \tag{5}$$

where $\vec{n}_A = \vec{a} \times \vec{n}$ and $\vec{n}_B = \vec{b} \times \vec{n}$ are planes normal to $\vec{n}$, coincident with their respective rays. The midpoint is then simply $\vec{p}_{\text{mid}} = (\vec{p}_{A|B} + \vec{p}_{B|A})/2$. The particle location is estimated by the cumulative average of all midpoints as they are bundled. Any additional rays must pass within a threshold (the bundle "core radius"; Fig. 3) of that location for inclusion. Other valid candidates remain in the pool for inclusion in other bundles.

The current implementation uses a fixed, user defined threshold for inclusion. However, a dynamic threshold could be implemented using the standard deviation or confidence interval of the particle location. Furthermore, the radius implemented here is spherical, but an ellipsoid based on the plenoptic camera properties would probably be better suited. A non-uniform optical transfer function could be utilized to construct a particle model dependent on experimental setup and particle location, similar to the work of Schanz et al. (2017).

Only one ray per perspective is permitted and a ray may only belong to one bundle. Thus, after a ray has been included in a bundle, it is removed from the pool of available rays, and the process continues with the next view until all N_v views have been examined. Then a new bundle is formed and the process repeated. Therefore, each bundle is built one ray at a time using this assignment process. This has the potential for path dependence, where the particle matching is based on the order that views are considered. Tight thresholds and the abundance of available views are expected to be sufficient to mitigate any path dependence. However, simultaneous execution is possible using k-means clustering, Gaussian mixture models, or similar; but significant memory and processing requirements are expected.

Optimize bundles. The resulting bundles can consist of one to N_v rays. Only two rays are required for triangulation, but accuracy will be limited. For sparse particle fields, as few as 10 rays have been

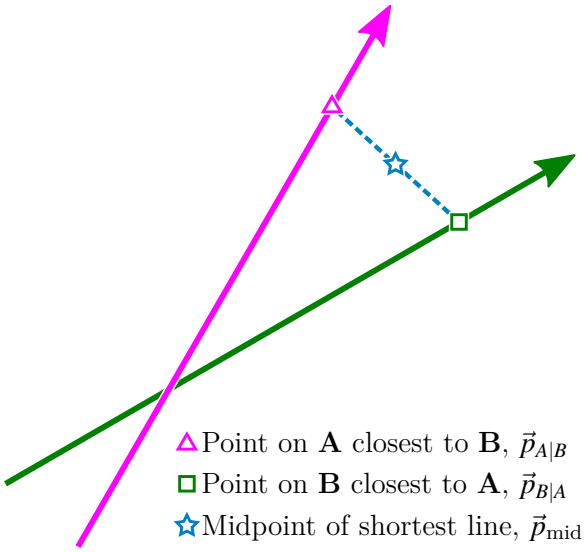


Figure 2: The shortest line between two rays in 3D space.

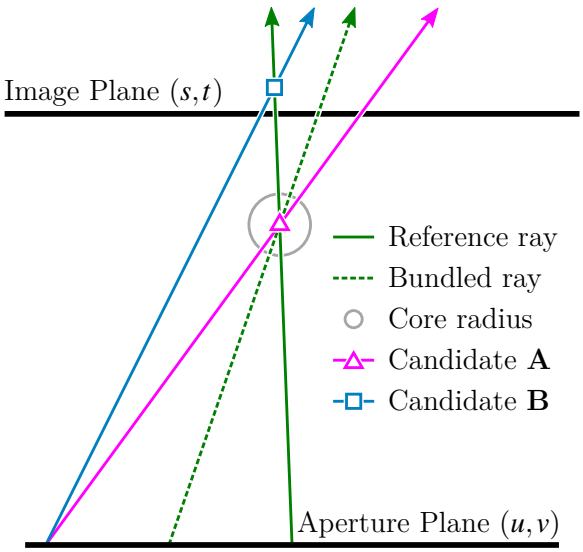


Figure 3: Example of an occluded particle with multiple valid crossings.

used to triangulate objects with fair accuracy. In this work, a bundle must have rays from at least 51% of available views. At high particle densities, a lower threshold can result in identifying the same particle multiple times. This could be addressed with an additional step that merges bundles, but for now bundles with fewer than this number of views are simply discarded. At this stage it is also possible to re-evaluate all rays in a bundle to discard any outliers, similar to the work done by Hall et al. (2019a); however, that has not been implemented at this time.

Triangulate particle positions. Now that particle identification and matching has been performed via ray bundling, all that remains is to triangulate the particle position in object space. DLFC uses the polynomials $s = \mathcal{P}_s(x,y,z,u,v)$ and $t = \mathcal{P}_t(x,y,z,u,v)$ to describe a direct mapping from object to image space. Each bundle is a list of (s,t,u,v) coordinates. The (x,y,z) location of the associated particle may be solved using a simple least-squares method. Non-linear least squares was implemented here using the Leveberg-Marquardt algorithm provided by ALGLIB (Bochkanov, 2015).

The execution time of the proposed algorithm is expected to scale linearly with the number of views N_v and $N_p \cdot \log(N_p)$ with the number of particles N_p.

3 Performance and Accuracy

To evaluate the LFRB algorithm, synthetic images are used such that the accuracy of the triangulation can be compared with a known solution. The details of the synthetic plenoptic image generator are provided by Fahringer et al. (2015). The simulated camera was an Imperx Bobcat B6640 29 MP (6600×4400 pixels, 5.5 µm pixel pitch) and a 471×362 hexagonally-packed microlens array with 77 µm pitch and 308 µm focal length. The simulated camera was equipped with an 85 mm main lens and operated at a magnification of -0.5. Particles 10 µm in diameter were randomly positioned within a 40 mm cube. Two synthetic datasets were generated: the first had only a single particle in each image, the second with particle density ranging from 0.001 to 0.4 particles per microlens (ppµ). Previous work (Fahringer and Thurow, 2018) has shown ppµ to be a good surrogate for the particles per pixel (ppp) metric used in tomographic studies, with typical values around 0.05 ppp.

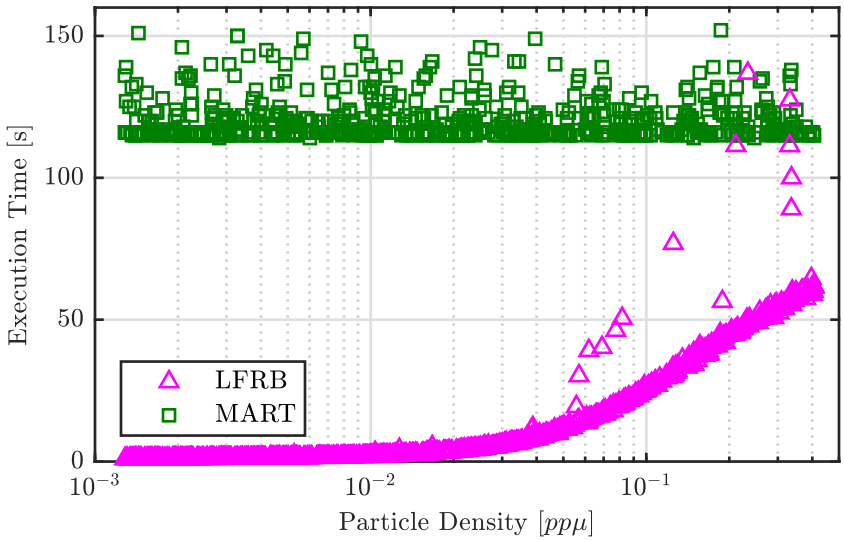
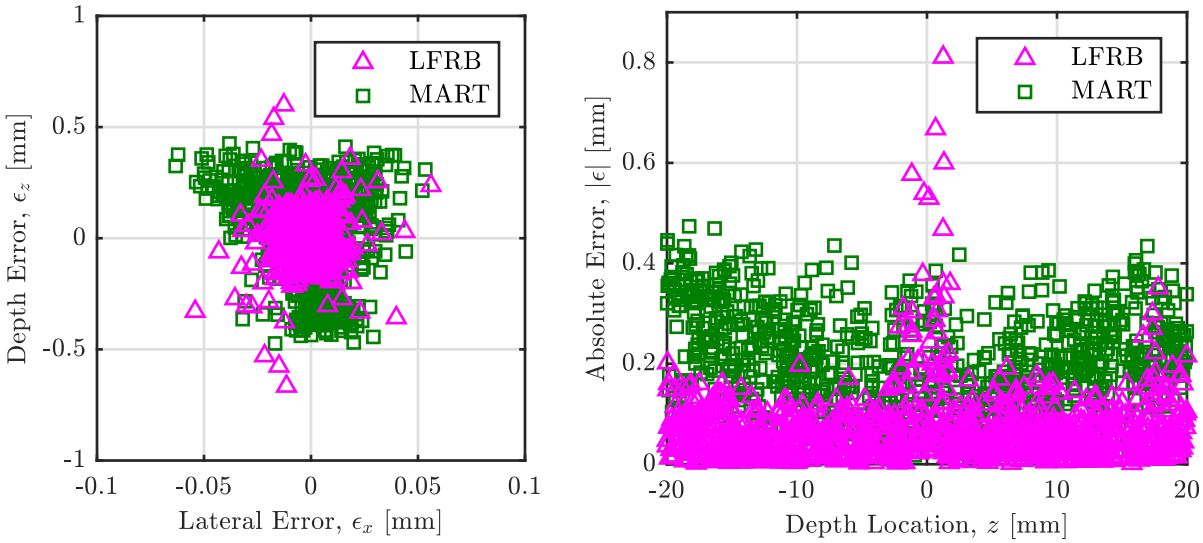


Figure 4: Execution time of LFRB and MART algorithms.

In addition to LFRB, volumetric reconstruction was performed using Multiplicative Algebraic Reconstruction Technique (MART). Five iterations were performed with no relaxation. The volumetric grid was 42×42×46 mm in size, with z being slightly larger to allow for particle elongation to be captured entirely within the volume. The cubic voxels were 0.13 mm on edge, resulting in approximately 1.2 voxels per microlens. The full details of the MART algorithm are provided by Fahringer and Thurow (2015).

The various user-defined thresholds used during LFRB are summarized here. During image segmentation, a minimum image intensity of 100 counts was required; all pixels less than that were considered noise. Blurry and out-of-focus particles had peak intensities of at least 150 counts, with sharply focused particles near 1000 counts and highly occluded particles in excess of 2000 counts. The experimental volume in image space was considered $0.9 \cdot l$ to $1.1 \cdot l$, which is -25.5 to 25.5 mm in object space, well in excess of the particle placement from -20 to 20 mm. This was enforced in two ways: first, the focal-plane distance was required to be less than 1/9 that of the aperture-plane distance for any subsequent computation to occur; second, the point of closest approach must occur between $0.9 \cdot l$ and $1.1 \cdot l$. The maximum approach distance for a valid crossing was $1.25 \cdot 10^{-4}$ mm. The bundle core radius was 0.75 mm. Finally, for a bundle to be considered valid for triangulation, it must contain observations from at least 51% of the 134 available perspective views.

All computations were performed using highly-parallelized C/C++ code compiled with Intel MKL and run on a compute node with dual 14-core 3.3 GHz Xeon E5-2680 v4 processors. The execution time of each algorithm is shown in Fig. 4. MART shows a consistent performance independent of the number of particles, with an average execution time of 120 seconds. Note that the MART algorithm implemented here does not update voxels which have already converged to zero, thus a slight trend with particle density was expected. However, the execution is dominated by the initialization and first iteration time. At particle densities below 0.02 ppμ, the execution time of LFRB is a near constant 1.3 seconds. In this region, the execution is dominated by the generation of perspective views from the raw plenoptic data. This task scales with the number of perspectives and the number of pixels per perspective (*i.e.* the total number of pixels in the system), both of which were held constant throughout this work. At higher particle densities, $N_p \cdot \log(N_p)$ scaling is observed as predicted. Even at very high particle densities, the LFRB algorithm is twice as fast as MART. The crossover point



(a) Depth versus lateral error

(b) Absolute error as a function of depth

Figure 5: Error in particle centroid positions.

between LFRB and MART is estimated to occur at 2.7 ppμ, well beyond the point where individual particles could be discerned.

The dataset containing only a single particle per image was used to establish the triangulation error under ideal conditions. The positional error $\boldsymbol{\varepsilon}$ of particle centroids is shown in Fig. 5a for the lateral x versus depth z error of each particle. Note that the z-axis is an order of magnitude greater than the x-axis; this reaffirms existing literature (Fahringer et al., 2015) that demonstrates depth error 5–10× greater than lateral error in single-camera plenoptic systems. The distribution of LFRB errors is smaller than that of MART. In addition, two observations may be made about the shapes of each distribution. First, the MART errors occur in an hourglass pattern, compared to the elliptical pattern of the LFRB errors. Second, the apex or waist of the hourglass occurs at $\boldsymbol{\varepsilon}_z = -0.3$ mm (towards the camera), which causes a slight systematic error away from the camera. The absolute error $|\boldsymbol{\varepsilon}|$ of the particle centroid is shown in Fig. 5b as a function of depth in the volume. Errors in the MART algorithm are bounded by a V-shaped envelope with an additional spike near the focal plane ($z = 0$). This behavior is well characterized by Fahringer (2018): near the focal plane, information from a particle is captured only by a few microlenses, thus limiting the reconstruction accuracy; away from the focal plane, the circle of confusion scales with increasing distance from the focal plane, leading to increased particle blur. Errors in LFRB exhibit a larger spike near the focal plane, but a smaller overall value in z. The mean errors are 0.20 mm for MART and 0.07 mm for LFRB.

The fraction of particles found and their mean error are shown in Fig. 6. The MART reference is shown in gray while LFRB is shown in blue; the other lines are discussed below. The LFRB algorithm performs well for particle densities up to 0.02 ppμ. Beyond this point, the fraction of particles found starts falling behind MART. The accuracy of LFRB remains better than MART until 0.07 ppμ; however, the value of that accuracy is diminished by the missing particles, which would negatively impact PTV. At these high densities, the number of occluded particles leaves few rays that pass through only one particle, thus a significant portion of the rays in each bundle are corrupted.

An iterative approach inspired by Iterative Particle Reconstruction (IPR; Wieneke, 2013) and Dense Particle Identification and Reconstruction (DPIR; Troolin et al., 2018) can be utilized. Such methods have found success for particle densities up to 0.05 ppp. The necessary alteration to the

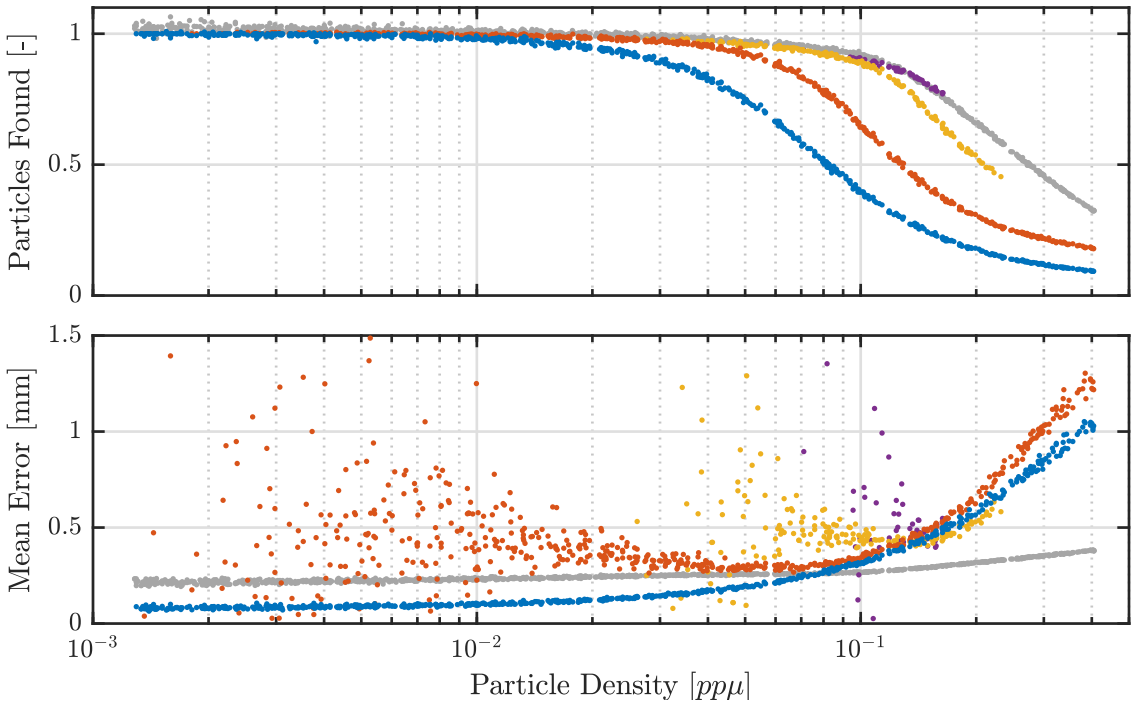


Figure 6: Performance of Light-Field Ray Bundling after 1 (—), 2 (—), 5 (—), and 10 (—) iterations; MART reference (—). Particles found are cumulative, while errors are for the labeled iteration only.

LFRB algorithm is simple: tighten the bundle inclusion requirements (if necessary) such that only particles with high-confidence are found; project found particles onto each view; calculate residual images by subtracting the projections from all views; restart the process and repeat until no (or very few) new particles are discovered. Ideally, the residual images contain only those particles which haven't been identified. The full details of this approach are beyond the scope of this paper, and are presented briefly only as a possibility for future enhancement.

The result of 2, 5, and 10 LFRB iterations are shown in Fig. 6 by the red, yellow, and purple lines, respectively. This modification of LFRB has been carried out in MATLAB which performs considerably slower than the compiled C/C++ code; as a result it was necessary to limit the total execution time, thus the higher iterations do not cover the full range of ppμ. The fraction of particles found are cumulative, however, the mean error is shown based only on particles found in that iteration. As additional iterations of LFRB are run, the number of particles found increases and matches that of MART even at very high particle densities. The mean errors are significantly more scattered due to the low number of samples, in many cases finding only one additional particle. When the number of additionally found particles increases, the lines converge toward that of the first iteration.

4 Additional Considerations

The work presented here does not make use of, nor attempt to address, the temporal component of velocimetry. To obtain velocimetry data, LFRB would need followed by a particle tracking algorithm. Many codes exist to form continuous particle tracks, including open source options such as OpenPTV (OpenPTV Consortium, 2014) and PTVlab (Brevis et al., 2011; Patalano and Wernher, 2009). Such codes initialize particle tracks using a two-frame approach, then continue with forward-predictive

tracking after a few frames. Shake The Box (STB; Schanz et al., 2016, 2013) is a robust PTV algorithm that iteratively predicts tracks-from-locations and locations-from-tracks, thus providing high spatial and temporal accuracies even in densely seeded flows. A similar technique could be used to predict ray bundles in sequential frames.

The authors also acknowledge the independent development of related plenoptic triangulation work by Hadfield (2018). That work utilized a focused plenoptic camera (often called "plenoptic 2.0") with a hexagonal microlens array of three different focal lengths. Particle matching was performed using Epipolar Triangular Connectivity (ETC) on the raw plenoptic image. Then triangulation was achieved by minimizing the distance between a point and all rays of that particle. This differs from the current work where particle matching is performed directly in ray-space and triangulation relies on a calibrated light-field.

5 Conclusions and Future Work

A new particle matching and triangulation algorithm called Light-Field Ray Bundling (LFRB) was presented. LFRB represents particle observations as rays in 3D space, then attempts to "bundle" those rays by identifying common intersections. A bundle of rays represents a list of (s,t,u,v) coordinates all associated with the same particle. That list of coordinates is combined with Direct Light-Field Calibration (DLFC) polynomials and a non-linear least squares solver to retrieve the (x,y,z) particle coordinates in object space. LFRB demonstrates execution times twice as fast and 3× as accurate as MART for particle densities up to 0.02 ppµ. A multi-iteration approach extends viability to particle densities of 0.07 ppµ, beyond which volumetric reconstruction offers superior accuracy.

The implementation shown here represents only a cursory investigation and leaves significant room for improvement. The user defined thresholds are flexible, but are statically applied to all rays during the bundling process. Dynamic thresholds would benefit the algorithm in two ways. First, thresholds based on camera parameters (such as main lens focal length and nominal magnification), would alleviate responsibility from the user. Second, thresholds that vary with the properties of each bundle (such as position within the volume) would produce bundles of higher quality. Bundles could be improved even further with an iterative rejection process similar to Hall et al. (2019a) to remove outliers. Finally, this work focused primarily on particle matching, but is preceded by image segmentation and succeeded by triangulation via least-squares fitting. Improvement of either of these processes would naturally improve the overall results.

Acknowledgements

This work was completed in part with resources provided by the Auburn University Hopper Cluster.

References

Aether Lab (2015) Prana. Available at www.github.com/aether-lab/prana

Bochkanov S (2015) ALGLIB numerical analysis library. Available at www.alglib.net

Brevis W, Niño Y, and Jirka GH (2011) Integrating cross-correlation and relaxation algorithms for particle tracking velocimetry. *Experiments in Fluids* 50:135–147

Eckstein A and Vlachos PP (2009a) Assessment of advanced windowing techniques for digital particle image velocimetry (DPIV). *Measurement Science and Technology* 20:075402

Eckstein A and Vlachos PP (2009b) Digital particle image velocimetry (DPIV) robust phase correlation. *Measurement Science and Technology* 20:055401

Fahringer T and Thurow B (2015) Comparing volumetric reconstruction algorithms for plenoptic-PIV. in *53rd AIAA Aerospace Sciences Meeting*. 2015-1221. Kissimmee, FL

Fahringer TW (2018) *On the Development of a Volumetric Velocimetry Technique using Multiple Plenoptic Cameras*. Ph.D. thesis. Auburn Unversity

Fahringer TW, Lynch KP, and Thurow BS (2015) Volumetric particle image velocimetry with a single plenoptic camera. *Measurement Science and Technology* 26:115201

Fahringer TW and Thurow BS (2018) Plenoptic particle image velocimetry with multiple plenoptic cameras. *Measurement Science and Technology* 29:075202

Hadfield J (2018) *Plenoptic Imaging in Particle Tracking Velocimetry Experiments*. Master's thesis. University of Alberta. Alberta, CA

Hall EM, Fahringer TW, Thurow BS, and Guildenbecher DR (2017) Volumetric calibration of a plenoptic camera. in *55th AIAA Aerospace Sciences Meeting*. 2017-1642. Grapevine, TX

Hall EM, Guildenbecher DR, and Thurow BS (2019a) Development and uncertainty characterization of 3D particle location from perspective shifted plenoptic images. *Optics Express* 27:7997–8010

Hall EM, Tan ZP, Guildenbecher DR, and Thurow BS (2019b) Refinement and application of 3d particle location from perspective shifted plenoptic images. in *AIAA SciTech 2019 Forum*. 2019-0268. San Diego, CA

Levoy M (2006) Light fields and computational imaging. *Computer* 39:46–55

Maas HG, Gruen A, and Papantoniou D (1993) Particle tracking velocimetry in three-dimensional flows. *Experiments in Fluids* 15:133–146

Nobach H and Honkanen M (2005) Two-dimensional Gaussian regression for sub-pixel displacement estimation in particle image velocimetry or particle position estimation in particle tracking velocimetry. *Experiments in Fluids* 38:511–515

Ohmi K and Li HY (2000) Particle-tracking velocimetry with new algorithms. *Measurement Science and Technology* 11:603–616

OpenPTV Consortium (2014) OpenPTV – open source particle tracking velocimetry. Available at www.openptv.net

Patalano A and Wernher B (2009) PTVlab – time-resolved digital particle tracking velocimetry (PTV) tool for Matlab. Available at www.mathworks.com/matlabcentral/fileexchange/41235

Rogers SS, Waigh TA, Zhao X, and Lu JR (2007) Precise particle tracking against a complicated background: polynomial fitting with Gaussian weight. *Physical Biology* 4:220–227

Schanz D, Gesemann S, and Schröder A (2016) Shake-The-Box: Lagrangian particle tracking at high image densities. *Experiments in Fluids* 57:70

Schanz D, Gesemann S, Schröder A, Wieneke B, and Novara M (2017) Non-uniform optical transfer functions in particle imaging: calibration and application to tomographic reconstruction. *Measurement Science and Technology* 24:024009

Schanz D, Schröder A, Gesemann S, Michaelis D, and Wieneke B (2013) 'Shake The Box': a highly efficient and accurate tomographic particle velocimetry (TOMO-PTV) method using prediction of particle positions. in *10th International Symposium on Particle Image Velocimetry*. Delft, Netherlands

Takehara K and Etoh T (1998) A study on particle identification in PTV particle mask correlation method. *Journal of Visualization* 1:313–323

Troolin D, Boomsma A, and Lai WT (2018) Dense particle identification and reconstruction (DPIR) technique applied to simulated volumetric PTV images. in *71st Annual Meeting of the APS Division of Fluid Dynamics*. volume 63. page D21.00009. Atlanta, GA

Wieneke B (2013) Iterative reconstruction of volumetric particle distribution. *Measurement Science and Technology* 24:024008