Exploiting Loops in the Camera Array for Automatic Focusing Depth Estimation

Regular Paper

Tao Yang¹, Yanning Zhang¹,*, Rui Yu² and Ting Chen¹

¹ ShaanXi Provincial Key Laboratory of Speech & Image Information Processing, School of Computer Science, Northwestern Polytechnical University, Xi’an, China
² School of Electronic Engineering and Computer Science, Queen Marry, University of London, London, UK

* Corresponding author E-mail: ynzhangnwpu@gmail.com

Received 10 May 2012; Accepted 5 March 2013

© 2013 Yang et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract Autofocus is a fundamental and key problem for modern imaging sensor design. Although this problem has been well studied in single camera literature, unfortunately, little research has been done on large-scale camera arrays. Most of the existing synthetic aperture imaging systems still need to manually select the optimal focus plane when an object moves. Unlike the conventional autofocus method, which sweeps the focus plane to find the maximal contrast, we present a novel optimization framework to handle the above challenges. In particular, we formulate the camera array autofocus problem as a constrained optimization problem by minimizing the temporal and spatial correspondences error subject to global loop constraint. Then this problem is relaxed as a quadratic program and solved using sequential quadratic programming. The experimental results show that the proposed method achieves a better performance compared with the results of traditional methods. To the best of our knowledge, our proposed method is the first optimization framework for solving camera array autofocus problem and it is of great importance to improve the performance of the existing synthetic aperture imaging system.

Keywords Camera Array Autofocus, Global Loop Optimization, Synthetic Aperture Imaging

1. Introduction

Autofocus (or AF) is one of the most important characteristics of an optical system. An intelligent AF system can adjust the camera lens to obtain focus on the subject so as to achieve sharpness at the focus point and is the foundation of many algorithms and systems in the fields of image processing, computer vision and pattern recognition.

By camera array autofocus, we mean that the relative position of the object to the camera array can be calculated automatically and thereby all the camera images can be accurately aligned on the object. It is obvious that this ability to continuously focus on the object dynamically is essential for the camera array to track a moving object robustly in occlusion cases.

Although single camera AF algorithms and systems have been well studied in the literature, to the best of our knowledge little work has been done on the camera array.

One difficult problem comes from the dynamic motion of the observed object; another problem comes from partial or even complete occlusion in multiple camera views.
In this paper, we propose a novel camera array autofocus (CAAF) model. Differing from previous works, our model for the first time integrates the spatial and temporal constraints of a large-scale camera array into a global optimization framework, which can be formulated as a quadratic program and solved via sequential quadratic programming. Extensive experiments with qualitative and quantitative evaluation demonstrate the accuracy, robustness and superiority of our approaches compared with state of the art AF algorithms.

The rest of the paper is structured as follows. We review work related to single camera AF and multiple camera AF in Section 2. In Section 3, we present both framework and details of our CAAF model. The challenging case of occlusion handling is discussed in Section 4. Extensive experimental results are given in Section 5, followed by conclusions and future work in Section 6.

2. Related Work

2.1 Single Camera Autofocus

Broadly speaking, previous single camera AF systems can be roughly classified into two categories: active AF systems and passive AF systems.

Instead of analysing the input image, active AF systems measure the distance to the observed object independently of the optical system. For instance, Polaroid cameras emit ultrasonic sound waves to estimate distance, while Nikon and Canon use infrared light to obtain the distance to an object. One drawback of an active AF system is that due to sound and infrared light reflection by the glass, the system usually cannot focus through windows and the accuracy of the focus is often considerably lower than that of passive systems. Another downside is the increased optical system complexity with additional multiple sensors.

Passive AF systems directly estimate the focus plane by analysing the image captured by the optical system without any assistance from other sensors. Typical passive AF methods include phase detection and contrast measurement. Phase detection is often used in single-lens reflex cameras and is achieved by dividing the incoming light into pairs of images and comparing them. Usually, the two images are analysed for similar light intensity patterns (peaks and valleys) and the separation error is calculated in order to find if the object is in front focus or back focus position.

Contrast measurement is performed by analysing the contrast of the input image. As the intensity difference between adjacent pixels usually increases with good focus, we can adjust the focus depth until the maximal contrast is detected. Since it does not use a separate sensor, this method can be more flexible and potentially more accurate. However this approach still has several defects. First, it cannot handle textureless objects with only one uniform colour. Second, it is sensitive to illumination and may fail under dim light. Third, local searching processing is needed to find the maximal contrast, which is generally very time-consuming.

2.2 Camera Array Autofocus

Among existing multiple camera techniques, recently the camera array has proven to be a powerful tool for seeing objects through occlusion. The pioneer work of synthetic aperture imaging is proposed by Levoy et al. They setup a two dimensional Stanford light field camera array which consisted of 128 firewire cameras and for the first time multiple cameras were aligned to a focus plane to approximate a camera with a very large aperture. The MIT computer graphics group used 64 USB webcams to obtain dynamic depth of field effects. Wilburn et al. used 100 custom video cameras to approximate a video camera with a large synthetic aperture. Synthetic aperture focusing consists of warping and adding together the images in a 4D light field so that objects lying on a specified surface are aligned and thus in focus, while objects lying off this surface are misaligned and hence blurred. Veeraraghavan et al. presented a novel design to reconstruct the 4D light field from a 2D camera image without any additional refractive elements as required by previous light field cameras. Liang et al. introduced a system including a novel component called a programmable aperture and two associated post-processing algorithms for high-quality light field acquisition. The shape of the programmable aperture can be adjusted and used to capture light fields at full sensor resolution through multiple exposures without any additional optics and without moving the camera. Maître et al. used a planar camera array to perform surface reconstruction. Lei et al. developed a cluster-based system for camera array application, which consists of eight nodes and 16 cameras. Schuchert et al. estimated the 3D object structure, motion and rotation based on 4D affine optical flow using a multi-camera array. Ding et al. constructed a 3x3 camera array to recover fluid surfaces. Although camera arrays have been well studied in the literature, little work has been done on the problem of camera array autofocus. Most of the existing systems need "manual" adjustment of the focus plane. In SIGGRAPH2006, Joshi et al. took the lead on presenting a camera array "autofocus" algorithm. Their approach is similar to the contrast measurement of single camera AF, where a camera varies its focus and picks the setting that gives the strongest gradients in pre-defined regions. The measurement they used was the intensity variance, because low variance at a depth plane usually implies that a number of features are aligned and thus an object is present. They sweep a synthetic focal plane through the
observed scene, sum the variance at each plane with nine focus regions and pick the depth that gives the minimum variance in one of these regions.

The first limitation of Neel’s approach is that they assume the variance of the background is several orders of magnitude larger than that of the foreground so that this is not the case, this method will fail. The second limitation is that the foundation of this approach is based on low variance implying good focus, which is not valid since the variance of the foreground will be greatly increased under occlusion.

In contrast, we seek to continuously focus and see occluded objects even under object motion and occlusion, which is a general and difficult problem in the field of synthetic aperture imaging. Our main contribution is two-fold: first, we introduce a general camera array autofocus framework, which has the unique ability to continuously focus and see an object under challenging conditions. In this novel framework, very few parameters need to be adjusted and the problems of object motion and partial or even complete occlusion in several camera views can be naturally handled. Second, we demonstrate that the CAAF problem can be formulated as a constrained optimization problem and solved robustly by sequential quadratic programming.

3. Camera Array Autofocus Model

In this section, we first give the mathematical model of the CAAF problem. In the second step, three independent constraints are introduced, including a continuity constraint, a multi-view constraint and a global loop constraint. It is also shown that the calibration information of the camera array can be accounted for in multi-view constraints. Finally, the problem is formulated as a quadratic program and solved via sequential quadratic programming.

3.1 Formalization

As shown in Fig. 1, our algorithm takes a reference image and all the current camera images as the input. In this paper, the reference image is fixed and we simply choose the first frame of the fourth camera as the reference image. With the assumption that the object can be approximated as a planar object, our algorithm aims to estimate the object depth and the homography transformation with regard to the object between the reference image and all the cameras as well as between different cameras.

The transformation from each camera to reference image is represented as a temporal homography $H_{i,s}^t$, while the spatial homography between camera $i$ and camera $j$ is encoded as $H_{i,j}^s$. Therefore, the CAAF problem can be formulated as an optimization problem of $H_{i,s}^t$, $H_{j,s}^t$ and $H_{i,j}^s$.

![Figure 1. Homography constraint graph of the camera array. An example of global loop is visualized by the bold black line.](image)

We present a unified framework to obtain the optimal focus plane through global optimization of temporal and spatial homography wrappings inside and across different camera views. This optimization framework mainly consists of three independent constraints: a temporal local continuity constraint, a spatial multi-view constraint and a global loop constraint.

The first constraint comes from temporal local feature correspondences between the reference image and the current image, which enforces a constraint on the temporal homography of each camera individually. The second constraint stems from spatial feature correspondences between different camera views. It is shown in Section 3.3 that a simple spatial homography representation, which takes into account the specificity of the camera array, can be introduced to tremendously simplify the problem. The third constraint is based on the fact that temporal and spatial homography establish a loop between each camera pair and the reference image.

As can be seen from Fig. 1, the continuity constraint is a unary constraint on $H_{i,s}^t$, and the multi-view constraint is a unary constraint on $H_{i,j}^s$, while the global loop enforces a ternary constraint between $H_{i,s}^t$, $H_{j,s}^t$ and $H_{i,j}^s$.

Instead of solving this optimization problem using a Markov random field or other general optimization methods, we formulate this problem as a quadratic program. Specifically, the optimization problem is
formulated as a constrained optimization problem by minimizing the temporal and spatial correspondence error subject to global loop constraints. Then this problem is relaxed as a quadratic program and solved using sequential quadratic programming.

The main advantage of this approach is that if a solution is found, it is guaranteed to be consistent with all the global loops, which makes the homography estimation result more stable as all the homography matrices are related together.

In this paper, the number of cameras used is eight. For initialization, we specify the object depth and manually label the object on the reference camera. The homography transformation between the reference image and all the individual cameras, as well as between different cameras can be directly calculated. For successive frames, we simply use the results of the previous frame to initialize these homography wrappings. All the correspondences inside or across different camera views are obtained by SIFT [16] feature matching.

3.2 Continuity Constraint

Our continuity constraint is defined between each camera and the reference image. By calculating corresponding feature points \( X_t \) between camera i and the reference image, we can estimate the corresponding homography matrix. Since a homography has eight degrees of freedom, we can get the homography result from at least four point-point correspondences. This estimation is usually done by finding a transformation \( H \) that minimizes the cost function. As all the cameras in the camera array are related together the geometric error is difficult to optimize. In this paper we take a DLT algorithm to minimize the algebraic error and formulate the constraint as a system of linear equations. For camera i we get the following equation:

$$\begin{align*}
H^i_t X^i_t &= x^0_t \\
\text{where } X^i_t \text{ and } x^0_t \text{ represent the corresponding points in camera } i \text{ and the reference image respectively. As equation (1) is determined up to a scale factor, if there are } n_i \text{ point correspondences in } X^i_t, \text{ we have only } 2n_i \text{ independent equations. Then the equation becomes:}
\end{align*}$$

$$A_t h^i_t = 0 \tag{2}$$

In this equation, \( A_t \) is a \( 2n_i \times 9 \) matrix and \( h^i_t \) is a nine dimension vector made up of the entries of the matrix \( H^i_t \).

3.3 Multi-view Constraint

Except the feature point correspondences between each camera and the reference image, we can also establish correspondence within each camera pair. Suppose \( X^i_s \) are corresponding points between camera i and camera j, we have the following equation:

$$H^i_s x^i_t \equiv x^j_t \tag{3}$$

where \( H^i_s \) is a common homography matrix with eight degrees of freedom. However, as mentioned before, camera array calibration information can be accounted for in this constraint. We will show in the following that \( H^i_s \) can be rewritten as a more much simple representation of \( H^i_d \), where the symbol \( H^i_d \) indicates that the homography is only related with object depth \( d \) and thus the freedom of \( H^i_d \) is only one.

Suppose homography projections on to the reference calibration plane and relative camera displacements with respect to the reference camera have been calculated with a plane+parallax calibration algorithm [16], we can get the following equation based on synthetic aperture focusing:

$$\Delta H_d H^i_d x^i_t \equiv \Delta H_d H^j_d x^j_t \tag{4}$$

where \( H^i_d \) and \( H^j_d \) represent the homography projection on to the calibration plane of camera i and camera j respectively, \( x^i_t \) and \( x^j_t \) are multi-view feature correspondences between camera i and j and \( \Delta H_d \) and \( \Delta H_d \) are the translation transformation eliminating the parallax between cameras. Given the depth \( d \) of focusing plane, \( \Delta H_d \) and \( \Delta H_d \) can be written as:

$$\Delta H_d = \begin{bmatrix}
1 & 0 & \Delta x_i (d - d_{ref}) / d \\
0 & 1 & \Delta y_i (d - d_{ref}) / d \\
0 & 0 & 1
\end{bmatrix} \tag{5}$$

In this equation, \( \Delta x_i \) and \( \Delta y_i \) are the relative camera displacements in the x and y direction, \( d_{ref} \) and \( d \) mean the calibration depth and current depth. From Equation (4) and Equation (5), we have:

$$H^i_d \Delta H_d H^j_d x^j_t \equiv x^j_t \tag{6}$$

Thus, \( H^i_s \) can be written as \( H^i_d \), which is defined as:

$$H^i_d = H^{-1}_j \Delta H_d H^j_i \tag{7}$$

So Equation (3) becomes:

$$H^i_d x^i_t \equiv x^j_t \tag{8}$$

It should be noted that Equation (4) is true only when all the correspondences are on the same frontoparallel plane. When this is not the case, we should compute corresponded \( H^i_d \) for each point individually. Therefore, Equation (8) is only true when the object is located on a frontoparallel plane relative to the reference calibration plane. However, we find that this equation is
effective in most cases, including with tilted plane objects and a nonplanar human face. More details will be discussed in Section 4.

Similar to the discussion in Section 3.2, if there are \( n_{ij} \) correspondences between camera pair \( i \) and \( j \), we have \( 2n_{ij} \) independent linear equations, all of which are about the object depth \( d \).

\[
c_{ij}d = b_{ij}
\]

(9)

where \( c_{ij} \) and \( b_{ij} \) are both \( 2n_{ij} \times 1 \) vectors.

### 3.4 Global Loop Constraint

Our global loop constraint is a ternary constraint between \( H_{t_i}^i \), \( H_{t_j}^i \), and \( H_{d_{ij}}^i \). As seen from Fig. 1, any two cameras and the reference image form a loop. For instance, the loop between cameras \( i \) and \( j \) gives rise to the constraint:

\[
H_{d_{ij}}^i (H_{t_j}^i)^{-1} H_{t_i}^i \cong I_{3\times3}
\]

(10)

To avoid a matrix inverse operation, we rewrite Equation (10) as:

\[
H_{t_j}^i (H_{d_{ij}}^i)^{-1} \cong H_{t_i}^i
\]

(11)

Equation (11) defines a set of eight polynomial constraints on \( H_{t_i}^i \), \( H_{d_{ij}}^i \), and \( H_{t_j}^i \). As there are 28 loops in the camera array, we end up with 224 constraints.

Differing from the above two constraints, which are either on each camera independently or on the depth of the object, the global loop constraint combines the camera homography and object’s depth together. The main advantage of introducing this constraint is that the solution is guaranteed to be consistent with the camera array and the CAAF framework can still work well even if the object is severely occluded in the majority of cameras.

### 3.5 Sequential Quadratic Programming

As shown in Fig. 1, the estimation of \( H_{t_i}^i \), \( H_{t_j}^i \), and \( H_{d_{ij}}^i \) can be easily formulated as a graph optimization problem. Since temporal local mappings and spatial multi-view mappings are related together in the global loop constraint, it is difficult to solve the optimization problem by directly combining all the three constraints. Here we introduce a quadratic program formulation and the problem is then solved by sequential quadratic programming. Let \( \Gamma = [h^T, d]^T \) and \( h = [h_{1,1}^T, h_{2,1}^T, \ldots, h_{n,1}^T]^T \) denote the variable standing for the homography of all cameras and the object depth respectively. Continuity constraints and multi-view constraints inside and across all the cameras can be expressed as an inhomogeneous linear equation system in \( \Gamma \):

\[
A\Gamma = b
\]

(12)

\[
A = \text{diag}(A^*, c^*)
\]

\[
A^* = \text{diag}(A_1, A_2, \ldots, A_8)
\]

\[
c^* = [c_{1,2}^T, c_{1,2}^T, \ldots, c_{7,8}^T]^T
\]

\[
b = [0^T, b^T]^T
\]

\[
b^* = [b_{1,2}^T, b_{1,3}^T, \ldots, b_{7,8}^T]^T
\]

In other words:

\[
\begin{bmatrix}
A^* & 0 \\
0 & c^*
\end{bmatrix}
\begin{bmatrix}
h \\
d
\end{bmatrix} =
\begin{bmatrix}
0 \\
b^*
\end{bmatrix}
\]

(13)

By gathering all the global loop constraints, we get:

\[
f_i(\Gamma) = 0, j = 1, 2, \ldots, 224
\]

(14)

Thus, it is natural to estimate \( \Gamma \) by solving:

\[
\min \| A\Gamma - b \| \quad \text{subject to} \quad f_i(\Gamma) = 0, j = 1, 2, \ldots, 224
\]

(15)

However, as the constraints are not linear, the problem is not convex and therefore is difficult to solve. In this paper, we linearize the constraint in its neighbourhood and relax the problem as a sequential quadratic program.

Given an initial value \( \Gamma_0 \), we propose to estimate \( \Gamma_k \) iteratively by solving the following quadratic program:

\[
\min \| A\Gamma - b \| \quad \text{subject to} \quad f_i(\Gamma_k) + \nabla f_i(\Gamma_k)(\Gamma - \Gamma_k) \leq \varepsilon
\]

(16)

where \( \varepsilon \) is a small positive parameter. This quadratic program can be effectively solved with many off the shelf optimization softwares, such as GPLK and SeDuMi. In this paper, we use convex optimization software package CVX to solve this problem. Experiments show that the solution converges rapidly and a solution with satisfactory accuracy is usually found within five iterations.

### 3.6 Algorithm Output

Based on the output of our algorithm, we can easily perform a synthetic aperture focusing on the object. There are two different focusing methods, one is based on the estimated object depth and the other is based on homography mappings. More details will be discussed in Section 5.
4. Handling Occlusion

As mentioned in Section 3, one of the main advantages in our camera array focusing system is that it has the ability to see-through occlusion and track objects robustly even under severe occlusion. In this paper, the meaning of severe occlusion is two-fold. The first refers to cases where the object is partly occluded in all cameras, like the case when the object is occluded by a blooming tree. The second deals with cases where the object is totally occluded in some cameras. For instance, when the occluder is a moving person, the target may be totally occluded in some cameras while visible in others. Besides, this visibility information changes dynamically as the occluder keeps moving. Neither of these two cases has been successfully solved in single camera focusing or tracking.

What about the performance of our algorithm in these two challenging cases? It is obvious that occlusion only reduces the number of corresponded feature points we can use and does not influence our optimization framework at all. Specifically, the continuity constraint as well as multi-view constraint will be weakened, but the global loop constraint will always remain unchanged. As long as the object is visible to a certain degree in all eight cameras of the camera array, our framework will work quite well. Therefore, occlusion can be naturally overcome within our framework in the two mentioned cases. As a result, focusing and tracking can be performed robustly except in very extreme cases.

![Image](image.png)

**Figure 2.** The layout of our camera array system

5. Experimental Results

5.1 System and Dataset

Since the existing public multi-view dataset does not satisfy the camera array autofocus requirements of our approach, to evaluate the performance of our algorithm, we have set up a linear array camera in our laboratory with eight Pointgrey Flea3 cameras (as shown in Fig. 2). The aperture size of this camera array is about one metre. The camera array is connected to a single data server via four PCI-E Firewire adaptors. Multiple view images are captured at a speed of 60fps with an image resolution of 640x480. The time synchronization accuracy of our system is less than 125 microseconds.

In this section, we present our results from three experiments. First, we test our autofocus algorithm on a planar image and a nonplanar human face with frequent object depth variation, large tilt and rotation. Comparison results with state of the art imaging methods (standard Synthetic Aperture Imaging [17]) are given respectively. Second, to demonstrate the advantages of our approach in handling occlusion, we ask several volunteers to walk in front of the observed moving object. Experimental results show that our approach is better than AutoFocusing [18] and can robustly solve partial and even complete occlusion in several camera views. Finally, to evaluate the tracking performance of our algorithm, we compare our method with several state of the art tracking methods, including: ContextTracker [18] and PNTacker [19]. Tracking results show that our algorithm outperforms other methods and can estimate the object focus plane accurately and track objects robustly even under occlusion.

Our main data set consists of 11 video sequences. Considering the possible surface depth variation of the object, the chosen objects for data acquisition include a planar image and a nonplanar human face. Additionally, complex scenarios with frontoparallel translation, forward and backward walking, large tilt and rotation, as well as severe occlusion are also included.

5.2 Autofocus Results with Depth and Pose Variation

We begin the experiment with a study of autofocus of a planar object. Fig. 3 and Fig. 4 show the autofocus comparison results of our algorithm with standard synthetic aperture imaging method [17].

The test sequences consist of two handheld planar images. Fig. 3 shows the cover of the second edition of Multiple View Geometry written by Richard Hartley and Andrew Zisserman, which contains printed text and colourful circles. Fig. 4 shows the Bell Tower of Xi’an City, which is typically Chinese classical architecture. A volunteer rotates and tilts the planar image while walking inside the overlapping field of view of the camera array. These sequences test the ability of our autofocus algorithm to cope with planar object depth and pose variation.
In all cases our method is more accurate and the difference is especially large when the object moves far away from the initial focus plane. Please note that due to the spatial and temporal constraints and the global loop optimization, our method estimates the focus plane accurately under planar image movement, large rotation and tilting. In contrast significant blurs appear in the method of standard SAI [17] when the depth changes dynamically.

To further evaluate the effectiveness of our autofocus algorithm, we manually labelled the ground truth of the object depth in each frame. To obtain the ground truth-value of object depth, we performed standard SAI on sampled depths and select the one with best imaging result. In our test, the depth error of our algorithm is under 0.15m throughout the whole sequence, which is accurate enough considering the tilt and non-frontoparallel motion involved.

Next we apply our method to nonplanar face cases (as shown in Fig. 5). The data involves two human face sequences captured at shoulder level. The difficulty of this case comes from two parts. The first part is the depth variation of the nonplanar face surface, the second part is the unknown relative motion between the people and the camera array. Although strictly speaking, the planar homography transform assumption is no longer valid in this case, Fig. 5 shows that our algorithm still works well in spite of the depth variation of the nonplanar face surface. The reason behind this is that in our autofocus framework and homography transformation provide a good approximation when the depth variation is small as a result of global optimization the solution is guaranteed to be consistent with the camera array.

5.3 Autofocus Results with Occlusion

We then tested our algorithm under severe occlusion. As shown in Fig. 6, Fig. 7 and Fig. 8, there are three different test sequences. These test sequences contain frequent occlusion of moving handheld planar images. They are very challenging because not only do the objects contain large scale and rotation changes, but also they are partially or even completely occluded in several camera views.

Fig. 6 shows the case when the object is totally occluded by a person in several views while visible in other cameras. Fig. 7 shows the case when all cameras are partially occluded by foliage, while Fig. 8 is a more challenging sequence containing person occlusion as well as object pose changes. The first row of each subfigure
displays the original input images of different camera views. The second row presents the comparison results of the AF method [15] and our method. Both AF and our method successfully focus on the moving object without occlusion in frame #37 (Fig. 6), #91 (Fig. 7) and #31 (Fig. 8). However, the AF method fails when there is severe occlusion, in frame #93 (Fig. 6), #211 (Fig. 7) and #125 (Fig. 8). This is not surprising as occlusion or object pose change violate the fixed window minimum variance principle of AF. However, our method accurately focuses on the object with rotation, depth variation and occlusion. The better performance of our method comes from the fact that occlusion only reduces the number of matched feature points and has no influence on our global loop constraint. As long as there exists a valid loop in the camera array, our algorithm can still estimate the depth and homographies accurately. It is shown that good depth estimation results can be achieved in our global optimization framework. Please note that half of the camera views are completely occluded in frame #101 (Fig. 6) and #125 (Fig. 8) and our algorithm still focuses on the object focus plane accurately.

![Figure 6](image.jpg)

Figure 6. Comparison of camera array autofocus results through occlusion (People occlusion sequence)

![Figure 7](image.jpg)

Figure 7. Comparison of camera array autofocus results through occlusion (Plant occlusion sequence)

![Figure 8](image.jpg)

Figure 8. Comparison of camera array autofocus results through occlusion (Rotation and occlusion sequence)

5.4 Comparison Results of Visual Tracking

Perhaps one of the most likely applications of our method is visual tracking. Due to the advantages of see-through occlusion, the camera array has proven to be a powerful tool in tracking, especially in occlusion cases. In our autofocus framework, tracking is performed naturally. Based on the homography wrappings between each camera and the reference image, we can easily calculate the tracking bounding box of the object on each camera as well as on the synthetic aperture image. In order to evaluate the tracking performance of our algorithm, we use several state of the art tracking methods for comparison, including the TLD tracker and Context Tracker (CT). All codes are from the original authors.

Tracking results from two difficult occlusion sequences are shown in Fig. 9 and Fig. 10 respectively. Fig. 9 is a static object under occlusion, while Fig.10 is a more challenging case with occlusion as well as object rotation. The first row of each subfigure gives comparison tracking results from the reference camera and the second row shows the corresponding autofocus imaging results of our method. It is shown in Fig. 9 that when there is no occlusion, all three methods successfully track the object. However, when occluded by a front person on the reference image, the tracking accuracy of the single camera trackers decreases rapidly, while our method remains stable and robust throughout the whole sequence due to the proposed camera array global optimization framework. In Fig. 10, when the object is rotated rapidly, PNTacker (red rectangle) and Context Tracker (green rectangle) fail simultaneously. However, in contrast our algorithm accurately tracks the object (blue rectangle) stably, even in the most challenging cases (with occlusion and object rotation, as shown in frame #121).

5.5 Discussion of Processing Speed

The most time consuming part of our algorithm is establishing point correspondences and the sequential
quadratic programming (SDP). In this paper, we use SIFT features to establish correspondences inside or across different camera views and the iteration times of the SDP is fixed as three. In this case, the running time of our algorithm is two seconds per frame with unoptimized Matlab code.

![Figure 9. Comparison results of tracking through occlusion](image)

Figure 9. Comparison results of tracking through occlusion

6. Occlusions and Future Work

Although the autofocus problem has been well studied for single cameras, little work has been done on large-scale camera arrays. In this paper we present a novel camera array autofocus framework which can estimate the optimal object focus plane dynamically and tracks objects robustly even under severe occlusion. The effectiveness of our approach benefits from global optimization based on the three constraints about the camera array introduced in this paper. To evaluate the proposed method, we have set up an active camera array.

Extensive experiments carried out on different objects and scenarios demonstrate the robustness, accuracy and efficiency of the proposed approach. As no visibility information is utilized to help segment the object in our framework currently, in future work, we plan to improve the imaging result and solve the occlusion problem by introducing a visibility analysis method into our framework.

7. Acknowledgments

We are very appreciate to the editor and anonymous reviewers for the time spent and valuable suggestions. This work is supported by the National Natural Science Foundation of China (No. 61272288, No.60903126, No.61231016), NPU Foundation for Fundamental Research (No.JC201120), Soaring Star of Northwestern Polytechnical University (No.12GH0311), the China Postdoctoral Science Foundation Special Class (No.201003685), and the China Postdoctoral Science Foundation (No.20090451397).

8. References

[1] Levoy M, Ng R, Adams A, Footer M, Horowitz M (2006) Light field microscopy. ACM Transactions on Graphics, 25(3):924-934.
[2] Yang J C, Everett M, Buehler C, McMillan L (2002) A real-time distributed light field camera. Proceedings of the 13th Eurographics workshop on Rendering.
[3] Wilburn B, Joshi N, Vaish V, Talvala E V, Antunez E, Barth A, Adams A, Horowitz M, Levoy M (2005) High performance imaging using large camera arrays. ACM Transactions on Graphics, 24(3):765-776.
[4] Vaish V, Szeliski R, Zitnick C L, Kang S B, Levoy M (2006) Reconstructing occluded surfaces using synthetic apertures: stereo, focus and robust methods. Proceedings of the Conference on Computer Vision and Pattern Recognition: 2331-2338.
[5] Vaish V, Garg T, Talvala E, Antunez E, Wilburn B, Horowitz M, Levoy M (2006) Synthetic aperture focusing using a shear-warp factorization of the viewing transform. Proceedings of Workshop on Advanced 3D Imaging for Safety and Security, in conjunction with the Conference on Computer Vision and Pattern Recognition.
[6] Joshi N, Avidan S, Matusik W, Kriegman DJ (2007) Synthetic aperture tracking: tracking through occlusions. Proceedings of International Conference on Computer Vision:1-8.

[7] Yang T, Zhang Y N, Tong X M, Zhang X Q, Yu R (2011) Continuously tracking and see-through occlusion based on a new hybrid synthetic aperture imaging model. Proceedings of the Conference on Computer Vision and Pattern Recognition: 3409-3416.

[8] Pei Z, Zhang Y N, Yang T, Zhang X W, Yang Y H (2012) A novel multi-object detection method in complex scene using synthetic aperture imaging. Pattern Recognition, 45(4):1637-1658.

[9] Veeraraghavan A, Raskar R, Agrawal A, Mohan A, Tumblin J (2007) Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Transactions on Graphics, 26(3):69(1-69(12).

[10] Liang C K, Lin T H, Wong B Y, Liu C, Chen H H (2008) Programmable aperture photography: multiplexed light field acquisition. ACM Transactions on Graphics, 27(3): 55(1-55(10).

[11] Maitre M, Shinagawa Y, Do Mi N (2008) Symmetric multi-view stereo reconstruction from planar camera arrays. Proceedings of the Conference on Computer Vision and Pattern Recognition:1-8.

[12] Lei C, Chen X, Yang Y H (2009) A new multi-view spacetime consistent depth recovery framework for free viewpoint video rendering. Proceedings of International Conference on Computer Vision:1570-1577.

[13] Schuchert T, Schar H (2006) Estimation of 3D object structure, motion and rotation based on 4D affine optical flow using a multi-camera array. Proceedings of European Conference on Computer Vision:596-609.

[14] Ding Y Y, Li F, Ji Y, Yu J Y (2011) Dynamic fluid surface acquisition using a camera array. Proceedings of International Conference on Computer Vision: 2478-2485.

[15] Joshi N, Matusik W, Avidan S (2006) Natural video matting using camera arrays. ACM Transactions on Graphics, 25(3);779-786.

[16] Lowe D G (2006) Object recognition from local scale-invariant features. Proceedings of International Conference on Computer Vision:1150-1157.

[17] Vaish V, Wilburn B, Joshi N, Levoy M (2004) Using plane + parallax for calibrating dense camera arrays. Proceedings of the Conference on Computer Vision and Pattern Recognition:12-19.

[18] Dinh T B, Vo N, Medioni G (2011) Context tracker: exploring supporters and distracters in unconstrained environments. Proceedings of the Conference on Computer Vision and Pattern Recognition:1177-1184.

[19] Kalal Z, Matas J, Mikolajczyk K (2010) P-N learning: bootstrapping binary classifiers by structural constraints. Proceedings of the Conference on Computer Vision and Pattern Recognition:49-56.