Multi-leaf Alignment from Fluorescence Plant Images*

Xi Yin[1], Xiaoming Liu†[1], Jin Chen[1,2], and David M. Kramer[2,3]

[1]Department of Computer Science and Engineering, Michigan State University
[2]MSU-Department of Energy Plant Research Lab
[3]Department of Biochemistry and Molecular Biology, Michigan State University
[1,2,3]{*yinxi1,liuxm, jinchen,kramerd8*}@*msu.edu*

Abstract

In this paper, we propose a multi-leaf alignment framework based on Chamfer matching to study the problem of leaf alignment from fluorescence images of plants, which will provide a leaf-level analysis of photosynthetic activities. Different from the naive procedure of aligning leaves iteratively using the Chamfer distance, the new algorithm aims to find the best alignment of multiple leaves simultaneously in an input image. We formulate an optimization problem of an objective function with three terms: the average of chamfer distances of aligned leaves, the number of leaves, and the difference between the synthesized mask by the leaf candidates and the original image mask. Gradient descent is used to minimize our objective function. A quantitative evaluation framework is also formulated to test the performance of our algorithm. Experimental results show that the proposed multi-leaf alignment optimization performs substantially better than the baseline of the Chamfer matching algorithm in terms of both accuracy and efficiency.

1. Introduction

Photosynthesis is a fundamental biological process interested to a number of scientific fields, such as plant biology, physiology and bio-energy [16]. Owing to the fast growing sensing technology and computing power, non-invasively phenotyping plant photosynthesis is receiving increasing attention because it paves the way for quantitative high-throughput plant phenotyping and deeper understanding of a wide range of plant physiological problems [25].

*The fluorescence measurements and software development were supported in part by the Chemical Sciences, Geosciences, and Biosciences, Office of Basic Energy Sciences, Office of Science, US Department of Energy (award number DE-FG02-91ER20021). The instrumentation was supported by the MSU Center for Advanced Algal and Plant Phenotyping.
† Corresponding author: Xiaoming Liu, liuxm@msu.edu.

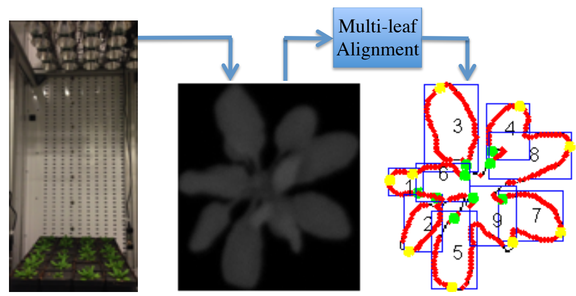

Figure 1. Computational plant growth modeling and leaf alignment. Given the fluorescence images of the plants captured during their growth period, we develop a novel optimization approach to multi-leaf alignment, i.e., estimating the number of leaves and their individual leaf structure such as the tips of a leaf. Leaf alignment is critical to fine-grained computational plant growth modeling, e.g., a leaf-level analysis of the photosynthetic effects.

Figure 1 shows a typical plant photosynthesis phenotyping framework in a growth chamber. The environmental conditions (light intensity, temperature, humidity, CO_2 concentration, etc.) are periodically changed by light sources on the top and by the growth chamber. A fluorescence camera is placed in between the light sources to capture a series of fluorescence images of the plants during their growth period [18], where the pixel intensity of such images represents the photosynthetic efficiency at a particular leaf spot. Because leaves at different developmental ages may response to the change of the environmental conditions in very different ways [7], it is important to provide a *leaf-level* analysis of the photosynthetic effects, i.e., to answer questions such as which leaf or which part of a leaf has higher photosynthetic efficiency under a specific condition. Obviously a prerequisite for such analysis is to segment each leaf from the image and accurately estimate the structure of a leaf, meaning the leaf tips and skeleton. This computer vision problem, named *leaf alignment*, is the main focus of this paper. Note that leaf alignment is a more advanced task than leaf segmentation [20], which only concerns segmenting leaves without estimating their structures.

Leaf alignment is a challenging problem due to a number of factors. First of all, unlike the images captured by popular RGB cameras, the fluorescence image is of low resolution and therefore the leaf size can be very small. Secondly, there are various degrees of overlap among plant leaves, which pose significant challenges in estimating their leaf boundaries. Thirdly, different leaves on the same plant may have large variations in their shapes, sizes and orientations. Such variations are even larger across different genetic variations of the same type of plant, which practically should be handled by one robust leaf alignment algorithm.

To the best of our knowledge, there is no previous study focusing on leaf alignment from fluorescence images of plants. To address this new problem, we develop a framework based on the well-known Chamfer matching (CM) algorithm [1], which computes the distance between two sets of edge points. To apply CM to our problem in a straightforward manner, we may first generate the edge maps of an array of leaf templates, each with different shapes, sizes, and orientations of the target leaves. Then the CM distances between the leaf templates and the edge map of a to-be-aligned plant image are computed, where the minimum distance leads to one aligned leaf. This process may be repeated to find the next minimum distance until all leaves are aligned. However, this naive procedure has limitations due to the limited representation power of leaf templates and leaf overlaps, as well as the fact that CM is fundamentally designed to align *one* object instance in an image.

Therefore, this paper proposes a novel framework to jointly estimate the alignment of multiple leaves in an image. Our approach is motivated by the crowd segmentation work [10], where both the number and locations of pedestrians are unknown and estimated simultaneously. Specifically, we formulate an optimization problem of an objective function with three terms: $1)$ the average of CM distances of all aligned leaves, $2)$ the number of estimated leaves, and $3)$ the distance between the synthesized mask of the selected leaf candidates and the original mask of the test image. Minimizing such objective function leads to the joint estimation of the number, locations, shapes, sizes and orientations of multiple leaves. We perform the qualitative and quantitative analysis of the algorithm performance on a set of test images with manually labeled ground truth of leaf structures. The experimental results demonstrate the effectiveness and efficiency of our proposed approach.

In summary, this paper has three main contributions:

⋄ We identify a novel computer vision problem of leaf alignment from fluorescence plant images, which performs leaf segmentation and structure estimation simultaneously. We collect a dataset for this novel problem and make it publicly available to facilitate future research and comparison.

⋄ We propose a novel extension of the Chamfer matching algorithm. By optimizing a joint objective function, our method can handle the alignment of multiple overlapping object instances within one image.

⋄ We set up a quantitative evaluation framework for the leaf alignment problem. We show the improved alignment performance of our novel approach in comparison with the baseline Chamfer matching method.

2. Prior Work

Shape and appearance modeling of leaves or plants are well-studied problems in the computer graphics [3, 19], where the goal is to render photo-realistic images of plants. For instance, the recent image-based foliage modeling approach can create a detailed leaf model from a number of high-quality images. However, this approach may not be directly applicable to low-resolution fluorescence images.

In the computer vision community, the prior work on leaves range from leaf segmentation [8, 20] and alignment [4, 23], to retrieval and identification [5, 9, 17]. Image segmentation is a long-lasting research topic and some approaches may be used to solve the leaf segmentation problem. For example, an automatic marker-controlled watershed segmentation method [24] is introduced to segment leaf images with complicated background, in which the markers are implemented to avoid over-segmentation in traditional watershed segmentation. Teng et al. [20] develop a leaf segmentation and classification system from natural images with the manual assistance from humans. A similar system is also developed by using 3D points from a depth camera [8]. The existing work on leaf alignment are all targeting at images either with a single leaf on a clean background [4, 6] using a parametric model, or with the single dominant leaf in the natural setting [11, 23].

Chamfer matching [1] has been widely used to align a template to an image based on their edge maps. Researchers have developed a wide variety of extensions, such as hierarchical CM [2], the fusion of CM and shape context [22], efficient directional CM that considers the orientations of matching edge points [12], and the boosting CM to suppress false detections [15]. However, both the traditional CM and its extensions are typically applied to detect or align a single object instance. In contrast, our proposed algorithm explicitly extends it toward the alignment of multiple object instances within one image.

From the application perspective, plant phenotyping is an interesting topic to plant biologists. A method called Plant Area Estimation (PAE) [21] is proposed recently to estimate each leaf area by identifying the distance from the plant center to the leaf tip. The plant center is identified using some easy-to-segment leaves. Clearly, this method will face challenges without such easy-to-segment leaves.

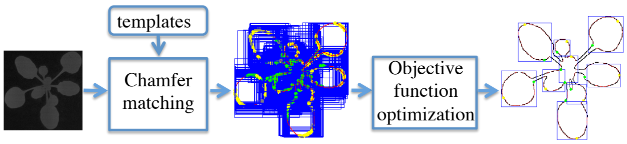

Figure 2. The overview of our multi-leaf alignment algorithm.

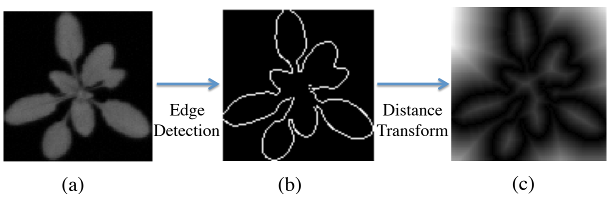

Figure 3. Calculating a distance transform image: (a) input test image, (b) edge map V, and (c) distance transform image *DT*.

3. Multi-leaf Alignment Algorithm

The proposed multi-leaf alignment algorithm is composed of two steps shown in Fig. 2. Firstly, the Chamfer matching distances between a test image and an array of leaf templates are computed by an exhaustive search of all possible locations of a template. Each template generates a leaf candidate using the location with the smallest CM distance. Given an over-completed set of leaf candidates, we formulate an optimization problem to estimate an optimal subset of candidates according to a joint objective function. In this section we will introduce each step in detail.

3.1. Candidate Nomination via Chamfer Matching

We start by introducing the basis of Chamfer matching, a method computing the best alignment between two edge maps. Let $U = \{\boldsymbol{u}_i\}$ and $V = \{\boldsymbol{v}_i\}$ be the sets of edge points in a template and a test image respectively. The CM distance is computed as the average distance of each point in the template with its nearest edge point in the test image:

$$d(U,V) = \frac{1}{|U|} \sum_{\boldsymbol{u}_i \in U} \min_{\boldsymbol{v}_j \in V} \|\boldsymbol{u}_i - \boldsymbol{v}_j\|_2, \tag{1}$$

where $|U|$ is the number of edge points in U. The matching score can be computed efficiently via a pre-computed distance transform image $DT(\boldsymbol{p}) = \min_{\boldsymbol{v}_j \in V} \|\boldsymbol{p} - \boldsymbol{v}_j\|_2$, which calculates the distance of each coordinate $\boldsymbol{p}$ to its nearest edge point in the test edge map V (Fig. 3). During the Chamfer matching process, an edge template U is superimposed on the distance transform image and the average *DT* value sampled by the template edge points $\boldsymbol{u}_i$ equals to the CM distance, i.e., $d(U,V) = \frac{1}{|U|} \sum_{\boldsymbol{u}_i \in U} DT(\boldsymbol{u}_i)$.

To take advantage of the efficient computation of CM distances, we use it to produce a potential list of leaf candidates for a fluorescence test image. That is, we first apply the conventional edge detection operator such as "Sobel" on the test image and generate an edge map V. Since there are a large amount of variations in leaf shape, size and orientation, it is infeasible to match leaves with only one template U. Therefore, as shown in Fig. 4, we employ an array of N leaf templates, $\{U_{hsr}\}$, where $h = 1, \dots, H$, $s = 1, \dots, S$, $r = 1, \dots, R$, $N = HSR$, and H, S, R are the numbers of chosen leaf shapes, sizes, orientations respectively. These templates can be obtained by first computing the edge maps from H leaves with representative shapes, and then varying their sizes and orientations according to S and R. Note that the yellow and green points in Fig. 4 are the two labeled leaf tips. They will be used to estimate the tips of the aligned leaf based on the point correspondence from Chamfer matching, which is the leaf structure information desired in leaf alignment.

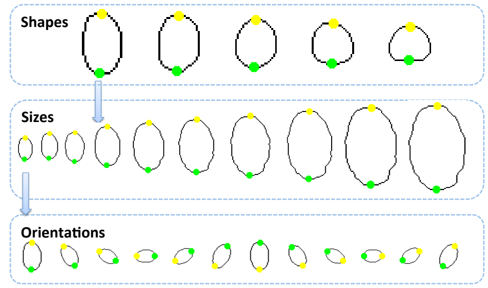

Figure 4. Leaf templates of various shapes, sizes and orientations.

For each template U_{hsr}, we shift it with all possible locations $\boldsymbol{u}^0$ on V and compute the corresponding CM distances, where the location associated with the minimum CM distance is recorded as $\boldsymbol{u}^0_{hsr}$, i.e, $\boldsymbol{u}^0_{hsr} = \arg\min_{\boldsymbol{u}^0} d(U_{hsr} + \boldsymbol{u}^0, V)$. After applying all templates, we can generate the same number of leaf candidates, each corresponding to one template. For clarity we denote the set of leaf candidates and their corresponding minimum CM distances as $\{\boldsymbol{l}_n, d_n\} \doteq \{U_{hsr} + \boldsymbol{u}^0_{hsr}, d(U_{hsr} + \boldsymbol{u}^0_{hsr}, V)\}$, where $n = 1, \dots, N$.

Note that for each template we only keep its minimum CM distance, because we assume that no two leaves can be best aligned with a single template. Even if this assumption cannot be met, owing to a large number of templates, a slightly different template will likely be included in the candidate pool so that we would not miss a leaf. We now have an over-completed pool of N leaf candidates, which includes all potential leaf configurations on the test image. The next critical question is how to select the best *subset* or *combination* of candidates according to certain objectives, which will be described in the remainder of this section.

3.2. Objective Function

To select the best combination of leaf candidates, one approach is to rank the CM distances d_n of N candidates and sequentially select *one* candidate with the minimum d_n. However, Chamfer matching only concerns the alignment of a single leaf in a local region, which is not sufficient for our problem where multiple leaves are present and overlap with each other in a larger spatial domain, since the selection of one leaf candidate may affect the selection of neighboring overlapping leaves. Therefore, for our problem, we need to define an objective function that goes beyond CM distances and consider the selection of *all* candidates jointly.

We first describe the rationality behind our multi-term objective function and then present its mathematical formulation. The *first* objective of our optimization is to select leaf candidates with the minimum average CM distances, which prompts candidates matching well with the edges of the test image. The *second* objective is to select the minimal number of leaf candidates. This is understood since our optimization aims to reduce the number of candidates N to be minimal, yet still be able to explain the multi-leaf plant image. Finally, we may envision that all selected candidates are placed together to compose a synthesized mask that should well approximate the test image mask, if the candidates are well selected. Thus, the *third* objective is to minimize the difference between these two masks. It encourages the selected leaf candidates to *jointly* cover the entire leaf region in the test image, and hence reduce the miss detection of leaves. In summary, our objective function seeks the minimal number of leaf candidates with small CM distances to best cover the test image mask.

To formulate the objective function, we define a N-dim indicator vector $\boldsymbol{x}$ to be the unknown parameter being estimated by our optimization, where $x_n = 1$ means that the leaf candidate $\boldsymbol{l}_n$ is selected and $x_n = 0$ otherwise. Hence $\boldsymbol{x}$ uniquely specifies a combination of candidates from a pool of N candidates. By denoting the CM distances of all candidates as a N-dim vector $\boldsymbol{d} = [d_1, \dots, d_N]^\intercal$, the first term in our objective function, the average CM distances, can be formulated as $\frac{\boldsymbol{d}^\intercal \boldsymbol{x}}{\|\boldsymbol{x}\|_1}$, where $\|\boldsymbol{x}\|_1$ indicates the number of selected leaf candidates. Similarly, the second term, the number of selected candidates, is $\|\boldsymbol{x}\|_1$.

The formulation of the third term depends on the image masks. As shown in Fig. 5, given a test image, we apply foreground segmentation to generate a test image mask, whose pixel is 1 at the multi-leaf region and 0 elsewhere, and convert it to be a K-dim row vector $\boldsymbol{m}$ by raster scan, where K is the number of pixels in the test image mask. Similarly, for each candidate $\boldsymbol{l}_n$ we generate a mask $\boldsymbol{M}_n$, whose size is the same as the test image mask and whose pixel is 1 within the leaf region and 0 elsewhere. We convert the mask $\boldsymbol{M}_n$ to a K-dim row vector $\boldsymbol{a}_n$, the collection of which from all leaf candidates is denoted as a $N \times K$ matrix $\boldsymbol{A}$. Note that $\boldsymbol{x}^\intercal \boldsymbol{A}$ is indicative of the synthesized mask, except that the values of overlapping pixels are larger than 1. In order to make it in the range of 0 to 1 so as to be

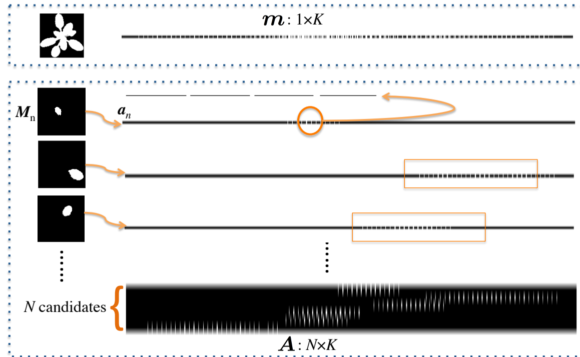

Figure 5. The process of generating $\boldsymbol{A}$ and $\boldsymbol{m}$.

comparable with $\boldsymbol{m}$, we employ the $\arctan()$ function,

$$f(\boldsymbol{x}) = \frac{1}{\pi} \arctan(C(\boldsymbol{x}^\intercal \boldsymbol{A} - \frac{1}{2})) + \frac{1}{2}, \tag{2}$$

where C is a constant controlling how closer the $\arctan()$ function approximates the step function. Similar $\arctan()$ function has been used in prior image alignment work [13, 14]. Note that the actual step function cannot be used here since it is not differentiable and thus is difficult to optimize. The constant $\frac{1}{2}$ within the parentheses is a flip point where the value of $\boldsymbol{x}^\intercal \boldsymbol{A}$ will be pushed toward either 0 or 1.

Finally, our objective function has three terms:

$$J(\boldsymbol{x}) = J_1 + J_2 + J_3 = \frac{\boldsymbol{d}^\intercal \boldsymbol{x}}{\|\boldsymbol{x}\|_1} + \lambda_1 \|\boldsymbol{x}\|_1 + \lambda_2 \frac{\|f(\boldsymbol{x}) - \boldsymbol{m}\|_2}{|\boldsymbol{m}|}, \tag{3}$$

where λ_1 and λ_2 are the weights for the two terms and $|\boldsymbol{m}|$ is the number of pixels in the test image mask. From different perspectives, the three terms jointly provide guidance on what constitutes an optimal combination of leaf candidates.

3.3. Gradient Descent based Optimization

We now discuss how to minimize the objective function in Eqn. 3. This is basically a combinatorial optimization problem searching for the best combination of leaf candidates, where the exhaustive search is not feasible due to the high computational cost. Also, because of the nonlinear function $\arctan()$, integer programming can not be applied either. Therefore, we propose a suboptimal gradient descent-based optimization to solve this problem, which is possible owing to the smooth objective function. Specifically, the derivative of the objective function w.r.t. $\boldsymbol{x}$ is:

$$\frac{dJ}{d\boldsymbol{x}} = -\frac{\boldsymbol{d}^\intercal \boldsymbol{x}}{\|\boldsymbol{x}\|_1^2} \text{sign}(\boldsymbol{x}^\intercal) + \frac{\boldsymbol{d}^\intercal}{\|\boldsymbol{x}\|_1} + \lambda_1 \text{sign}(\boldsymbol{x}^\intercal)$$
$$-\frac{2\lambda_2 C}{\pi|\boldsymbol{m}|} \left[(f(\boldsymbol{x}) - \boldsymbol{m}) \oslash (1 + (C(\boldsymbol{x}^\intercal \mathbf{A} - \frac{1}{2}))^2) \right] \boldsymbol{A}^\intercal, \tag{4}$$

where sign() is a function returning the sign of each element in vector $\boldsymbol{x}$, and $\oslash$ is the element-wise division of vectors.

Table 1. HR and LR datasets.

Dataset	Image size	#Images	#Labeled
HR	[210×180-240×220]	556	56
LR	[40×40-100×100]	88,650	56

Table 2. Templates for HR and LR datasets.

Dataset	H	S	R	N
HR	5	10	24	1200
LR	4	7	24	672

During the initialization, all elements in $\boldsymbol{x}$ are set to be 1, i.e., all candidates are valid leaves in the test image. In each iteration of gradient descent, $\boldsymbol{x}$ is updated by $\boldsymbol{x} = \boldsymbol{x} - \alpha \frac{dJ}{d\boldsymbol{x}}$, where α is a step size. Note that all elements of $\boldsymbol{x}$ should be either 0 or 1, while the gradient descent updating will apparently violate this assumption. Therefore, after $\boldsymbol{x}$ is updated at each iteration, the element in $\boldsymbol{x}$ with the largest change will be chosen, which means that this element has a relatively larger influence in minimizing the objective function. Then we verify whether this element should be fixed to either 0 or 1 in order to obtain a smaller $J(\boldsymbol{x})$. Once this element has been fixed, its value remains unchanged in future iterations. The iteration continues until all elements in $\boldsymbol{x}$ are fixed to either 0 or 1. Finally, the elements in $\boldsymbol{x}$ equal to 1 provide the combination of candidates for a test image.

4. Experiments

4.1. Dataset and Templates

We test the proposed multi-leaf alignment algorithm on Arabidopsis fluorescence images taken every 15 minutes periodically during the plant growth. Since all images have the same resolution. The more plants being captured, the lower the resolution of each plant is. We apply our method to two dataset: high-resolution (HR) fluorescence images, each including only 4 plants, and low-resolution (LR) fluorescence images, each including around 40 to 50 plants. We perform image segmentation so that each plant within a fluorescence image is saved as an individual image. The basic information of both datasets are shown in Tab. 1. To facilitate future research and performance comparison, the two labeled databases are publicly available[1].

Leaf templates can have a large influence on the leaf alignment performance. In our experiments, we observe the fluorescence images and select several representative leaves from images outside the testing set, as shown in the top row of Fig. 4. We label the leaf tips only for these representative leaves. These labels are mapped and recorded while synthesizing templates at multiple sizes and rotations. Table 2 shows the template information for both datasets.

4.2. Performance Evaluation

In order to quantitatively evaluate the leaf alignment performance, we need to provide manual labels on the ground-truth locations and structures of all leaves in test plant images. Given the small leaf changes between consecutive images, we decide to label 56 plant images for each dataset that are taken at a larger interval. For each plant image, we manually label the two tips of each of its L leaves, which are denoted as $\boldsymbol{p}_1^l$ and $\boldsymbol{p}_2^l$ $(l = 1, \ldots, L)$.

Given a test image $\boldsymbol{I}$, our optimization approach converges to A estimated leaves or selected leaf candidates via $\boldsymbol{x}$, i.e., $A = \|\boldsymbol{x}\|_1$. The tips of each estimated leaf, denoted as $\hat{\boldsymbol{p}}_1^a$ and $\hat{\boldsymbol{p}}_2^a$ $(a = 1, \ldots, A)$, are computed according to the correspondence of template tips during the Chamfer matching process. Since the numbers of estimated leaves A and labeled leaves L may not be the same, we establish the leaf correspondence as follows. For every labeled leaf, we compute the average distance e_{la} of its labeled tips to those of every estimated leaf, normalized by the leaf length:

$$e_{la} = \frac{||\boldsymbol{p}_1^l - \hat{\boldsymbol{p}}_1^a||_2 + ||\boldsymbol{p}_2^l - \hat{\boldsymbol{p}}_2^a||_2}{2||\boldsymbol{p}_1^l - \boldsymbol{p}_2^l||_2}. \tag{5}$$

This results in a $L \times A$ matrix E with e_{la} as its element. Within E we then find the minimum $\min(L, A)$ elements that meet the requirements of not sharing the same column or row. We implement this by first finding the minimum element in E and then delete the corresponding column and row of that element. And then we find the next minimum element in the new subset of E. This process is repeated until we find $\min(L, A)$ elements. We denote these elements as a vector $\boldsymbol{e}_i$ of length $\min(L, A)$, where the index of $\boldsymbol{e}_i$ in E determines corresponding leaves between the two leaf sets and the value indicates the landmark estimation error. Note that the remaining $|L - A|$ leaves will contribute to either false alarm counts or miss detection counts, depending on whether they are from the labeled set or the estimated set.

Although we have determined the correspondence and associated landmark error $\boldsymbol{e}_i$, we should still make a decision on whether the distance is small enough to be considered as a valid alignment or not. To do that, we define a threshold τ to compare with each element of $\boldsymbol{e}_i$. If one element is larger than τ, we claim the leaf in the labeled set as one miss detection count while the one in the estimated set as one false alarm count. If one element is smaller than τ, it will be added into a pool of "well aligned leaves".

After performing the above operation for all test images, we can quantitatively evaluate the alignment performance using three metrics, based on a chosen τ. The first one is the **Landmark error** $\bar{e}$, which is the average of all elements in the "well aligned leaves" pool. The second one is the **Miss detection** D, which is the total number of miss detection counts divided by the number of test images. The third one is the **False alarm** F, which is the total number of false

[1] http://www.cse.msu.edu/~liuxm/plant

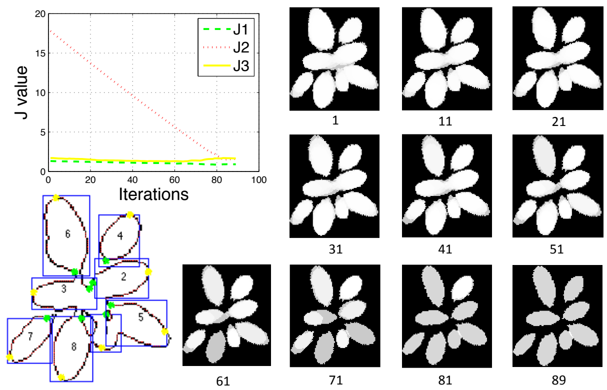

Figure 6. An examplar optimization process. The up-left part shows the changes of three terms during the iterations. The bottom-left part is the final alignment result. The right part is the synthesized mask $f(\boldsymbol{x})$ for different iterations, with the iteration index below the image.

alarm counts divided by the number of test images. Note that both miss detection and false alarm can attribute to two sources: the $|L - A|$ leaves without correspondence, and the corresponded pair whose landmark error is larger than τ. By varying τ, we can have a series of measurements for $\bar{e}$, D and F. To visualize these in one figure, we plot a 2D curve with $\bar{e}$ and $\frac{1}{2}(F + D)$ as two axes.

4.3. Experimental Results

Multi-leaf Alignment Optimization We apply all templates to a test image to find the best location of each template with the minimum CM distance. This results in a large number of candidates, which have heavy overlap as shown in Fig. 2. In order to narrow down the search space for optimization, we compute the overlap of each candidate mask with the test image mask. A candidate is deleted if the overlap is less than 90% of the candidate mask. Otherwise we preserve it in the candidate pool to generate the matrix $\boldsymbol{A}$. As shown in Eqn. 3, the objective function has three parameters: λ_1, λ_2 and C. We experimentally determine the best parameter setting to be: $\lambda_1 = 0.2$, $\lambda_2 = 20$, $C = 3$. And the step size of gradient descent is set to be $\alpha = 0.001$.

Figure 6 is an example of the optimization process. All elements in $\boldsymbol{x}$ are initialized as 1, i.e., all candidates are selected. We iteratively compute the gradient, update $\boldsymbol{x}$, and fix one element. The number of iteration is the same as that of available candidates. We see that J_2 decreases very quickly, which means most candidates are deleted at first due to the heavy overlap. J_1 and J_3 together make sure that the candidates overlapping with others and/or with larger CM distances will be deleted first. As the iteration goes on, the synthesized mask of the remaining candidates will be less dense and finally approximate the test image mask.

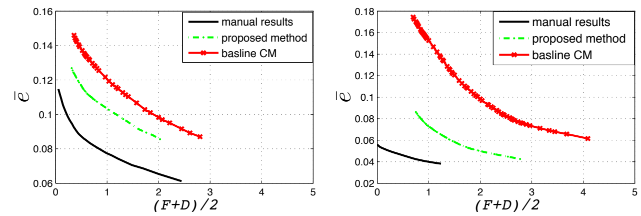

Figure 7. Performance comparison on two datasets: LR (left) and HR (right).

Baseline Chamfer Matching The basic idea of Chamfer matching is to align one object in an image. We compare our algorithm with the iterative version of the Chamfer matching method. Specifically we apply all templates to the edge map of a test image to find a large pool of candidates, which is exactly the same as the first step in our algorithm. Different from our joint leaf selection, the baseline CM aligns one leaf at a time. The template with the minimum CM distance is selected and declares an aligned leaf. We update the edge map of the test image by deleting the matched edge points of the aligned leaf. All templates are then applied to the modified edge map in order to find the next aligned leaf. The iteration continues until 90% of the edge pixels have been deleted. We use the same templates in Tab. 2 for this baseline Chamfer matching.

Manual Results Since the leaf templates might not perfectly represent the leaves in an unseen test image, it is good to know the upper bound of a leaf alignment algorithm. That is, if we know the labeled locations of all leaf tips, what is the optimal set of leaf candidates and what is the associated performance for this set? To answer this question, for each labeled leaf, we find the leaf candidate from $\{\boldsymbol{l}_n\}$ that has the minimum landmark error e computed from the labeled tips and the estimated tips. This is performed for all leaves in the 56 labeled images, and a performance curve can be obtained by applying a threshold τ to $\{e\}$. We call this curve as the "manual results". As shown in Fig. 7, this result is not perfect due to the limited representation power of a finite set of leaf templates. Nevertheless, this curve still serves as an upper bound for evaluating our leaf alignment algorithm.

Results of Accuracy We evaluate our proposed algorithm and the baseline CM method on both HR and LR datasets. We set the threshold τ to vary within $[0.1 : 0.01 : 1]$ in the evaluation process and generate the performance curves for both algorithms and the manual results. The performance curves are shown in Fig. 7, and some results of leaf alignment are shown in Fig. 8.

As we can see from Fig. 7 and Fig. 8, the proposed algo-

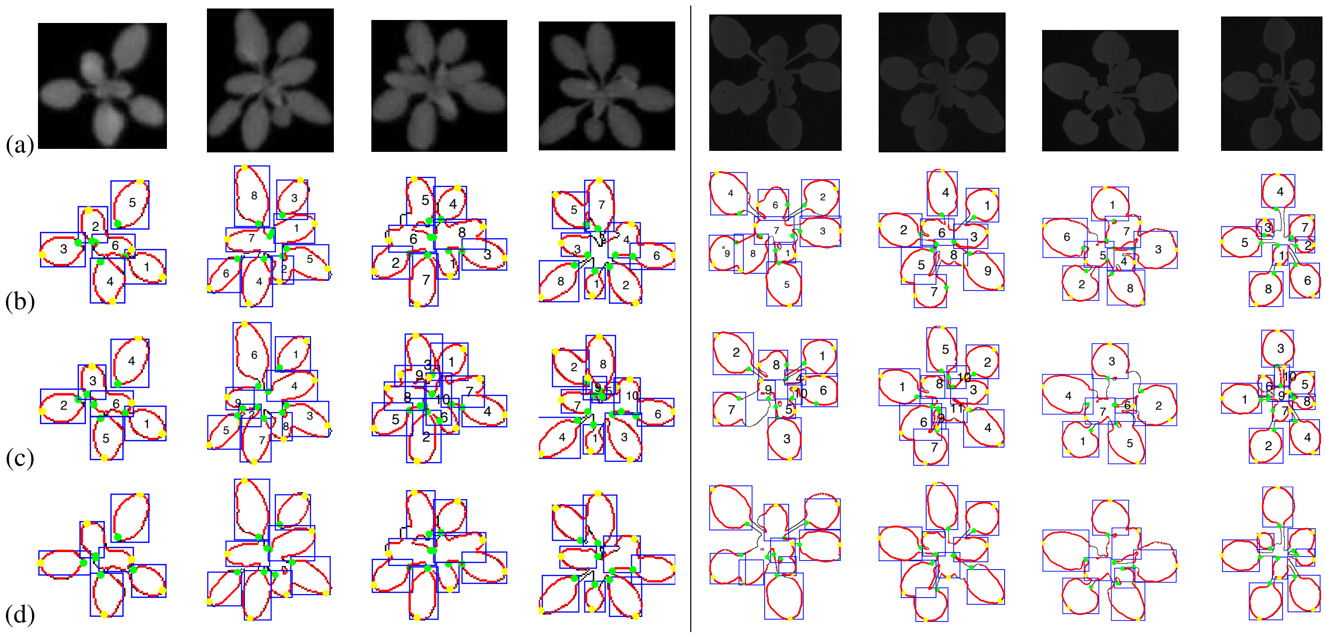

Figure 8. Examples of leaf alignment results. Vertically from top to bottom each row shows: (a) the input test images; (b) results of our proposed method; (c) results of the baseline CM method; (d) manual results. Horizontally the left and right 4 columns are from the LR and HR dataset respectively. The number on a leaf indicates the order of the leaf being aligned during our optimization iteration. The estimated leaf tips are shown in yellow and green and they contribute to the quantitative evaluation as Eqn. 5. The blue rectangle is the bounding box of the selected leaf candidate. Note the superior performance of our method compared to the baseline method. Best viewed in color.

rithm performs substantially better than the baseline CM on both HR and LR datasets. In the LR dataset, for an easy-to-align plant (1^{st} column of Fig. 8), both methods can work well to align all leaves in the test image. However, for more complicated plants, the baseline CM is more likely to generate false alarms due to the more crowded edge points in the test images. In the HR dataset, the leaf areas are larger and with more overlaps among neighboring leaves. The baseline CM is more likely to generate false alarms as well as miss detections mostly due to the incomplete leaf edge in the overlapping region. Therefore when the correct candidate is matching with the overlapped leaf, the CM distance is relatively large and this candidate will less likely to be chosen in the iterative alignment process, which causes the miss detection. Furthermore, the small-sized leaf candidates might still match with the incomplete leaf edge with smaller CM distances, which result in false alarms.

In contrast, in addition to the CM distance constraint (J_1 in Eqn. 3), the J_2 and J_3 terms in our objective function are defined to tackle the false alarm and miss detection respectively. Also, the joint optimization has the potential to better detect overlapped leaves even with slightly larger CM distances. Figure 7 shows the performance gap between our proposed method and the baseline CM algorithm. This gap is relatively larger on the HR dataset, which includes more overlapped leaves. This shows that our method can better handle the overlap problems in the fluorescence images compared to the baseline CM algorithm. One interesting question is that to what extend our method can correctly identify leaves in the overlapping region. We answer this question using a simple synthetic example. As shown in Fig. 9, our method performs well when the percentage of overlap is less than 23%. Otherwise it only identifies one leaf, which appears to be reasonable when the percentage is higher (e.g., 59%).

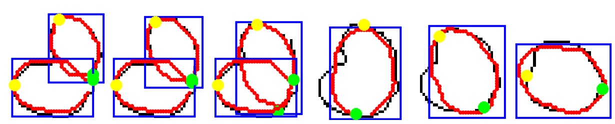

Figure 9. Leaf alignment results on synthetic leaves with various amount of overlap. From left to right, the percentage of overlap w.r.t. the smaller leaf is 10%, 15%, 22%, 23%, 36% and 59%.

Results of Efficiency Table 3 illustrates the average execution time of both algorithms in two steps, as shown in Fig. 2. Our method is superior to the baseline Chamfer matching in terms of efficiency, especially on the LR dataset, where the number of templates N is relatively smaller. The time is calculated based on a Matlab™ implementation on a conventional laptop.

Table 3. Computational efficiency of two algorithms (sec./image).

Datasets	HR	LR
Both algorithms (step 1)	17.02	1.14
Multi-leaf Alignment (step 2)	97.90	2.58
Basline CM (step 2)	196.92	25.12

5. Conclusions

This paper identifies a new computer vision problem of leaf alignment from fluorescence plant images. This work is motivated by the need to provide leaf-level photosynthetic analysis in plant bio-energy related research. We propose a novel framework for multi-leaf alignment, which is an extension of the well-known Chamfer matching algorithm. This algorithm can estimate the best alignment of all leaves in a fluorescence image by minimizing an objective function with three terms. Experimental results demonstrate the effectiveness and efficiency of our proposed approach. Our multi-leaf alignment algorithm performs substantially better than the baseline Chamfer matching on both datasets with different resolutions. In addition, it can successfully handle the overlap problem that is very common in plant images. It should be noted that the proposed multi-object joint alignment algorithm does not utilize any domain knowledge of leaves or plants, and hence it is potentially applicable to other similar problems in computer vision.

References

[1] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report, DTIC Document, 1977. 2

[2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE T-PAMI*, 10(6):849–865, 1988. 2

[3] D. Bradley, D. Nowrouzezahrai, and P. Beardsley. Image-based reconstruction and synthesis of dense foliage. *ACM Trans. Graph.*, 32(4):74, 2013. 2

[4] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, and D. Coquin. Understanding leaves in natural images–a model-based approach for tree species identification. *CVIU*, 117(10):1482–1501, 2013. 2

[5] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, D. Coquin, et al. A model-based approach for compound leaves understanding and identification. In *ICIP*, pages 1471–1475, 2013. 2

[6] G. Cerutti, L. Tougne, A. Vacavant, and D. Coquin. A parametric active polygon for leaf segmentation and shape estimation. In *Advances in Visual Computing*, pages 202–213. Springer, 2011. 2

[7] C. P. Chen, X. G. Zhu, and S. P. Long. The effect of leaf-level spatial variability in photosynthetic capacity on biochemical parameter estimates using the farquhar model: A theoretical analysis. *Plant Physiology*, 148(2):1139–1147. 1

[8] Y. Chéné, D. Rousseau, P. Lucidarme, J. Bertheloot, V. Caffier, P. Morel, É. Belin, and F. Chapeau-Blondeau. On the use of depth camera for 3D phenotyping of entire plants. *Computers and Electronics in Agriculture*, 82:122–127, 2012. 2

[9] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*, pages 502–516. Springer, 2012. 2

[10] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume 1, pages 878–885. IEEE, 2005. 2

[11] X. Li, H.-H. Lee, and K.-S. Hong. Leaf contour extraction based on an intelligent scissor algorithm with complex background. In *2nd International Conference on Future Computers in Education*, pages 215–220, 2012. 2

[12] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, pages 1696–1703. IEEE, 2010. 2

[13] X. Liu. Discriminative face alignment. *IEEE T-PAMI*, 31(11):1941–1954, 2009. 4

[14] X. Liu, T. Yu, T. Sebastian, and P. Tu. Boosted deformable model for human body alignment. In *CVPR*, 2008. 4

[15] T. Ma, X. Yang, and L. J. Latecki. Boosting chamfer matching by learning chamfer distance normalization. In *ECCV*, pages 450–463. Springer, 2010. 2

[16] T. A. MacMahon and J. T. Bonner. *On size and life*. Scientific American Books, 1983. 1

[17] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet. Advanced shape context for plant species identification using leaf image retrieval. In *ICMR*, pages 49:1–49:8. ACM, 2012. 2

[18] L. Nedbal and J. Whitmarsh. Chlorophyll fluorescence imaging of leaves and fruits. In *Chlorophyll a Fluorescence*, pages 389–407. Springer, 2004. 1

[19] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM SIGGRAPH*, pages 599–604. ACM, 2006. 2

[20] C.-H. Teng, Y.-T. Kuo, and Y.-S. Chen. Leaf segmentation, its 3D position estimation and leaf classification from a few images with very close viewpoints. In *Image Analysis and Recognition*, pages 937–946. Springer, 2009. 2

[21] O. L. Tessmer, Y. Jiao, J. A. Cruz, D. M. Kramer, and J. Chen. Functional approach to high-throughput plant growth analysis. *BMC Systems Biology*, 2013. 2

[22] A. Thayananthan, B. Stenger, P. H. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, volume 1, pages 127–133. IEEE, 2003. 2

[23] A. Vacavant, T. Roussillon, B. Kerautret, and J.-O. Lachaud. A combined multi-scale/irregular algorithm for the vectorization of noisy digital contours. *CVIU*, 117(4):438–450, 2013. 2

[24] X.-F. Wang, D.-S. Huang, J.-X. Du, H. Xu, and L. Heutte. Classification of plant leaf images with complicated background. *Applied mathematics and computation*, 205(2):916–926, 2008. 2

[25] X. Zhang, R. J. Hause, and J. O. Borevitz. Natural genetic variation for growth and development revealed by high-throughput phenotyping in arabidopsis thaliana. *G3: Genes— Genomes— Genetics*, 2(1):29–34, 2012. 1