A SPARSE AND LOW-RANK APPROACH TO EFFICIENT FACE ALIGNMENT FOR PHOTO-REAL TALKING HEAD SYNTHESIS

King Keung Wu\textsuperscript{1,2}, Lijuan Wang\textsuperscript{1}, Frank K. Soong\textsuperscript{1} and Yeung Yam\textsuperscript{2}

\textsuperscript{1}Microsoft Research Asia, Beijing, China
\textsuperscript{2}Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, China

\{kkwu, yyam\}@mae.cuhk.edu.hk, \{lijuanw, frankkps\}@microsoft.com

ABSTRACT

In this paper, we propose a framework for practical large-scale face alignment, based on the recent development of Robust Alignment by Sparse and Low-rank Decomposition for linearly correlated images (RASL). Unfortunately, the original implementation is not applicable in large image dataset. We extend this technique to deal with the situation with millions of images, with the aid of $l_1$-regularized least squares. Our proposal is applied onto the photo-real talking head, a challenging application which requires highly precise alignments of faces from video sequences. We verify the efficacy of our algorithm with experiments using real talking head data. Our method attains comparable quality to RASL in the experiments.

Index Terms— face alignment, photo-real, talking head, $l_1$-regularized least squares

1. INTRODUCTION

Photo-real talking heads have a wide variety of applications in human-computer interaction (HCI), from entertaining purposes in video games to educational software assisting language learning. A vividly lip-sync talking head provides a user-friendly interface, capable of engaging users in HCI. Such an animated talking head can be implemented by selecting an optimal sequence of lips images from a video training dataset, then stitching them back to a background head video. This topic has already been studied for a decade, and many successful models have been proposed and implemented\textsuperscript{1,2,3}.

In this paper, we focus on the alignment of faces in the video training set which is crucial in synthesizing natural lips and head movements. Consider the situation where the human subject being recorded keeps nodding his/her head while speaking, so the head pose varies among the raw image frames. Without additional treatment, the synthesized lip motion would probably be peculiar due to significant misalignment. Thus, face alignment is the first step in generating a talking head.

One way for the purpose is to use 3D model-based head pose tracking\textsuperscript{4,5}. The method estimates a pose transformation by matching a 3D mesh model to the 2D image. Although it enables fast large scale alignment, this method does not satisfy our requirement in precision.

Recently, Robust Alignment by Sparse and Low-rank Decomposition for linearly correlated images (RASL) is proposed\textsuperscript{6}, which allows a robust and highly accurate batch alignment of faces in images, despite occlusions, corruptions, and even illumination variations. This method formulates the batch alignment problem as the solution of convex programs, with the aid of latest advances in rank minimization. It is applicable in our video dataset for synthesis of talking head. Unfortunately, it has limitation on scalability. The memory constraint and computational cost restrain its application on very large dataset. For example in our talking head, several thousand images have to be aligned. This motivates us to extend the results of RASL so that it can be employed in large scale situations.

Instead of aligning the images in batch, we propose the one-by-one approach; we align the images individually using some well aligned images. In other words, if we are given $n$ RASL-aligned images, our approach would try to align the $(n + 1)$th image using the information provided. One of the advantages is that it relaxes the constraint of memory; at each time, we only have to store the $n$ RASL-aligned images and the $(n + 1)$th image, while for RASL, all the images have to be taken into the memory for the batch alignment.

This paper is organized as follows. Section 2 gives an overview of RASL. Section 3 introduces our one-by-one alignment approach. Section 4 gives the evaluations of our proposed method using several experiments. Finally, Section 5 concludes with a discussion of future work.

2. OVERVIEW OF RASL

Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images (RASL)\textsuperscript{6} is a scalable optimization technique for batch linearly correlated images alignment. One of its applications is to robustly align a dataset of human faces based on the knowledge that if the faces are well-aligned, they should show good low-rank structure up to some sparse corruptions. So the idea is to search for a set of transformation $\tau$ such that the rank of the transformed images becomes as small as possible and at the same time the sparse errors are compensated. The transformation we apply here is 2D affine transform, where we implicitly assume the face of a person is approximately on a plane in 3D space. The problem can be formulated as follows.

Given $I_1, \cdots, I_n \in \mathbb{R}^{w \times h}$ as the original misaligned grayscale images of a person’s face. Define $\text{vec} : \mathbb{R}^{w \times h} \rightarrow \mathbb{R}^{wh}$ as the operator that selects an $m$-pixel region of interest (e.g. the face part with main features such as eyes, nose and mouths) from an image and stacks to a vector. Denote $\tau = \{\tau_1, \ldots, \tau_n\}$ as the set of transformation, and $D \circ \tau$ as shorthand for $[\text{vec}(I_1 \circ \tau_1) \ldots \text{vec}(I_n \circ \tau_n)] \in \mathbb{R}^{m \times n}$, where $I \circ \tau$ represents image $I$ after transformed by $\tau$. The problem is formulated as the minimization in Lagrangian form:

$$
\min_{A,E,\tau} \text{rank}(A) + \gamma \|E\|_0 \quad \text{s.t.} \quad D \circ \tau = A + E
$$

(1)
Section 2, producing a low rank dictionary

A deal with thousands or even millions images. Therefore, we extend we select τ would give us the optimal solutions respectively. The problem (1) becomes:

$$\min_{A,E,\tau} \|A\|_* + \lambda \|E\|_1 \text{ s.t. } D \circ \tau = A + E$$ (2)

For the nonlinear constraint $D \circ \tau = A + E$, we can approximate it by linearizing about the current estimate of $\tau$ for small change of $\tau$. Then (2) can be written as:

$$\min_{A,E,\tau} \|A\|_* + \lambda \|E\|_1 \text{ s.t. } D \circ \tau = A + E + J \Delta \epsilon_i$$ (3)

where $J_i = \frac{\partial}{\partial \epsilon_i} \text{vec}(I \circ \epsilon_i)|_{\epsilon_i=0}$ is the Jacobian of the $i$-th image with respect to the transformation parameters $\tau_i$, $\tau = [\tau_1, \ldots, \tau_n]$ and $\epsilon_i$ denotes the standard basis for $\text{I}^n$.

Although RASL can give a very accurate alignment for faces as illustrated in [6], it is not applicable when $n$ is very large, say, $n \approx 10^5$. In many applications, such as the talking head, we need to deal with thousands or even millions images. Therefore, we extend the method of RASL to align $N$ face images, where $N >> n$.

3. ONE-BY-ONE ALIGNMENT APPROACH

We propose an extension to RASL, from $n \to N >> n$, by reformulating the problem with one-by-one alignment approach. First, we select $n$ frames to align with RASL, just like that described in Section 2 producing a low rank dictionary $A^*$. Next, the $(n+1)$th image is aligned with $A^*$ which contains the information for the previously aligned $n$ images. Finally, we repeat this step to all the images in the dataset, regardless of the size of the dataset.

3.1. Align $n$ images with RASL

In this step, the procedure is basically the same as described in Section 2. Applying RASL on the $n$ images chosen from the dataset would give us the optimal solutions $\tau^*, A^*, E^*$. We form $A \in \text{R}^{n \times \text{rank}(A^*)}$ whose columns consist of $\text{rank}(A^*)$ (out of $n$) independent columns of $A^*$. It acts as a dictionary for the aligned images, which will be used in the next step. RASL extracts and stores all the important features of the aligned faces in $A^*$. Therefore, we can use this set of occlusion-free images to be the basis. We would like our dictionary to cover as many features and variations as possible. Empirical results show that uniformaly random selection rather than consecutive sampling can ensure the convergence of $\tau_{n+1}$ with fewer selected images. It is reasonable since consecutive frames usually have similar features (e.g. the variations in illumination), which do not provide sufficient variations to form the basis.

In choosing the $n$ input images, there is a tradeoff between quality of dictionary and computational cost. Selecting more images (larger $n$) would probably lead to a better dictionary, however at the same time increases the speed of computation, mainly due to increase in size of dictionary.

3.2. From $n$ to $n+1$

Here we are going to align an additional image with those $n$ images already aligned by RASL. Let $I_{n+1}$ be a new image. We formulate the problem to the following $l_1$-regularized Least Squares ($l_1$-LS) problem:

$$\min_{x,\tau_{n+1}} \frac{1}{2} \|I_{n+1} \circ \tau_{n+1} - \hat{A}x\|_2^2 + \mu \|x\|_1$$ (4)

Here $\hat{A}$ is the dictionary we defined in Section 3.1. The goal of this optimization is to search for optimal $\tau_{n+1}$ such that $\hat{A}x$ forms the best approximation of $I_{n+1} \circ \tau_{n+1}$ with the least number of columns of $\hat{A}$. $x$ is a vector with dimension $\text{rank}(A^*)$ which represents the coefficients of the linear combination by columns of $\hat{A}$. $\mu$ is the weight that trades off the least square error and the sparsity of $x$.

However, the above optimization is non-linear which is hard to solve. Similar to that in RASL, we manage to linearize the optimization with iterative linearization. We write $I_{n+1} \circ \tau_{n+1} = I_{n+1} \circ (\tau_{n+1}^0 + \Delta \tau_{n+1}) = I_{n+1} \circ \tau_{n+1}^0 + J \Delta \tau_{n+1}$, where $J$ is the Jacobian matrix with respect to the affine transform $\tau_{n+1}$. Thus, the minimization becomes:

$$\min_{x,\Delta \tau_{n+1}} \frac{1}{2} \|I_{n+1} \circ \tau_{n+1}^0 + J \Delta \tau_{n+1} - \hat{A}x\|_2^2 + \mu \|x\|_1$$ (5)

which can easily be rewritten into the usual form of $l_1$-LS:

$$\min_{y} \frac{1}{2} \|y^T \tau_{n+1} - B y\|_2^2 + \mu \|Cy\|_1$$ (6)

where $\tau_{n+1}^0 = I_{n+1} \circ \tau_{n+1}^0$, $B = [\hat{A} -J]$, $y = [\Delta \tau_{n+1}]^T$, and $C = [I \ 0]$.

Since the linearization only holds locally, in order to find the minimal solution of (6), we have to perform an update about our current estimation of $\tau_{n+1}^0$ for many times until it converges. This step can be divided into two parts: outer loop and inner loop. The outer loop is the process of iterative linearization (shown in Algorithm 1. Inside the outer loop, there is an inner loop for the $l_1$-LS with split Bregman method which is a fast and efficient algorithm [2]. As we will see in Section 3 if the initial misalignment is not too large, this iteration recovers the correct transformations $\tau_{n+1}$ in an efficient manner.

Empirically, we find that $l_1$-LS is more stable than the conventional least squares (LS). LS has similar performance in the case when only Gaussian noise exists in the image $I_{n+1}$. However, even a small amount of non-Gaussian noise would generate many extra local minima in the objective function, leading to an incorrect optimal solution. The additional $l_1$-regularized term can act as a smoother, which eliminates the unwanted local minima by penalizing the number of atoms used in the dictionary $A$ to form the approximation.

3.3. From $n$ to $N$

We apply the same step as in Section 3.2 to all the remaining images. The $\hat{A}$ is kept unchanged. Therefore, the memory usage is independent of the number of images $N$ in the dataset. Empirically, we obtain comparable results to RASL in a reasonable time for thousands of images (as shown in [5]).

4. EXPERIMENTS

In this section, we demonstrate the capability and efficacy of our approach on large image datasets with two experiments. First, we compare our approach with the 3D pose tracking method [4] and RASL.
Algorithm 1 (Outer loop)

INPUT: Image \( I_{n+1} \in \mathbb{R}^{w \times h} \), RASL solution \( A^* \), initial transformation \( \tau_0 \) in affine group, weight \( \mu \)

WHILE not converged DO

Step 1: compute Jacobian matrices w.r.t. transformation:

\[
J \leftarrow \frac{\partial}{\partial \tau} \left( \frac{\text{vec}(I_{n+1} \circ \tau)}{\|I_{n+1} \circ \tau\|_2} \right)_{\tau = \tau_{n+1}}
\]

Step 2: warp and normalize the images:

\[
I_{n+1} \circ \tau_{n+1} \leftarrow \frac{\text{vec}(I_{n+1} \circ \tau_{n+1})}{\|\text{vec}(I_{n+1} \circ \tau_{n+1})\|_2}
\]

Step 3 (inner loop): solve the linearized \( l_1 \)-LS:

\[
(x^*, \Delta \tau^*_{n+1}) \leftarrow \text{arg min}_{x, \Delta \tau_{n+1}} \frac{1}{2} \left\| I_{n+1} \circ \tau_{n+1} + J\Delta \tau_{n+1} - A^* \right\|_2^2 + \mu \|x\|_1
\]

Step 4: update transformation:

\[
\tau_{n+1} \leftarrow \tau_{n+1} + \Delta \tau^*_{n+1}
\]

END WHILE

OUTPUT: solution \( \tau_{n+1} \) of optimization (4)

We test our method with an interview video obtained from the internet\(^1\). Totally there are 921 frames. 100 random images were selected

(i) Eye corners

Here we compare the eye corner positions of the alignments by the two approaches. To have a fair comparison, we only count the faces with eyes open, since the eye corners displace considerably when the eyes are closed. However, this does not mean that our algorithm is inapplicable to eye-closed case. We pick out 6633 images with eye-open and detect their eye corner positions. Table\(^2\) gives the statistics of errors in eye corners, calculated as the distances from the estimated eye corners to their center. Our approach produces alignments with one pixel accuracy, with standard deviations of half a pixel, which improves on the 3D pose tracking.

(ii) Accumulated variances of mouths

Here we compare the two methods using PCA of the mouths. For better alignments, the first few principal components of mouths should be in larger portions as they capture more meaningful features rather than those caused by misalignment. Fig.\(^3\) shows the first 20 principal components have a larger accumulated variance proportion and a smaller total variance after our alignment (about 20% less), which verifies our enhancement over 3D pose tracking.

4.1. Quantitative evaluations with talking head dataset

We verify the accuracy of our algorithm using a dataset with \( N = 9116 \) images which is practically being used in synthesizing a talking head. The images are collected from 35 video sequences of the same person. The original dimension of each images is 720 \( \times \) 576. The size of the face in canonical frame is 200 \( \times \) 200. To ensure fast convergence of our algorithm, we apply it on the faces after 3D pose tracking. Thus it acts as an enhancement of 3D pose tracking.

In this experiment, we choose the \( n = 100 \) samples uniformly in random over the whole dataset to perform RASL in the first step. The transformation \( \tau_{n+1} \) converges within 140 iterations for all 9116 images (on average 13.5 iterations per image), while if we choose the first 100 consecutive frames as the samples, \( \tau_{n+1} \) cannot converge within 300 iterations for some images.

In the following, we compare our quality of alignment with the result of 3D pose tracking as well as RASL.

4.1.1. Our approach vs. 3D pose tracking

The comparisons include using the eye corner positions and the accumulated variances of the mouths as the evaluation quantities.

\(^{1}\)This is a standard specification of personal computer nowadays.

\(^{2}\)The video is obtained from http://www.beet.tv/2008/09/microsofts-crai.html

### Table 1. Eye corners comparison of (a) 3D pose tracking and (b) our approach using 6633 frames with eye open. Here the distances are measured from the estimated eye corners to their center.

<table>
<thead>
<tr>
<th></th>
<th>Left eye</th>
<th>Right eye</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error</td>
<td>(a) 1.80</td>
<td>1.53</td>
<td>1.67</td>
</tr>
<tr>
<td></td>
<td>(b) 0.97</td>
<td>1.08</td>
<td>1.03</td>
</tr>
<tr>
<td>Standard error</td>
<td>(a) 1.09</td>
<td>0.86</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>(b) 0.58</td>
<td>0.56</td>
<td>0.57</td>
</tr>
<tr>
<td>Maximum error</td>
<td>(a) 8.76</td>
<td>7.93</td>
<td>8.35</td>
</tr>
<tr>
<td></td>
<td>(b) 4.31</td>
<td>5.50</td>
<td>4.91</td>
</tr>
</tbody>
</table>

### Table 2. Eye corners comparison of (a) RASL and (b) our approach with 250 frames. Here the distances are measured from the estimated eye corners to their center.

<table>
<thead>
<tr>
<th></th>
<th>Left eye</th>
<th>Right eye</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean error</td>
<td>(a) 0.94</td>
<td>0.90</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>(b) 0.94</td>
<td>0.86</td>
<td>0.90</td>
</tr>
<tr>
<td>Standard error</td>
<td>(a) 0.56</td>
<td>0.46</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>(b) 0.53</td>
<td>0.43</td>
<td>0.48</td>
</tr>
<tr>
<td>Maximum error</td>
<td>(a) 2.30</td>
<td>2.40</td>
<td>2.35</td>
</tr>
<tr>
<td></td>
<td>(b) 2.40</td>
<td>2.29</td>
<td>2.35</td>
</tr>
</tbody>
</table>

Fig. 1. Accumulated variances of mouths for first 20 principal components. The total variances of 3D pose tracking and our approach are $1.89 \times 10^{10}$ and $1.51 \times 10^{10}$.

5. CONCLUSIONS AND FUTURE WORK

While 3D pose tracking cannot get accurate alignment and RASL cannot deal with large scale, we have proposed a framework for practical and efficient face alignment that compensates for both of the above disadvantages, based on sparsity and low-rank structures in the linearly correlated face images. One possible future direction is to exploit the smoothness or small changes of adjacent frames of video sequence. For example, to use the video property to wisely choose a small set of images for RASL, or to achieve faster computation. Another future direction is to generate natural head movements using transformation parameters calculated by our method.

6. ACKNOWLEDGEMENT

We would like to thank Yi Ma and John Wright for their suggestions and help on the code of RASL.

7. REFERENCES