

Online Robust Image Alignment via Iterative Convex Optimization

Yi Wu^{†b} Bin Shen[‡] Haibin Ling[†]

[†]Center for Data Analytics & Biomedical Informatics, Computer & Information Science Department,
Temple University, Philadelphia, PA 19122, USA

^bSchool of Information and Control Engineering, Nanjing University of Information Science & Technology, Nanjing, 210044, China

[‡]Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA.

{wuyi, hbling}@temple.edu, bshen@purdue.edu

Abstract

In this paper we study the problem of online aligning a newly arrived image to previously well-aligned images. Inspired by recent advances in batch image alignment using low rank decomposition [16], we treat the newly arrived image, after alignment, as being linearly and sparsely reconstructed by the well-aligned ones. The task is accomplished by a sequence of convex optimization that minimizes the ℓ_1 -norm. After that, online basis updating is pursued in two different ways: (1) a two-stage incremental alignment for joint registration of a large image dataset which is known a priori, and (2) a greedy online alignment of dynamically increasing image sequences, such as in the tracking scenario. In (1), we first sequentially collect basis images that are easily aligned by checking their reconstruction residuals, followed by the second stage where all images are re-aligned one-by-one using the collected basis set. In (2), during the tracking process, we dynamically enrich the image basis set by the new target if it significantly distinguishes itself from existing basis images. While inheriting the benefits of sparsity, our method enjoys the great time efficiency and therefore be capable of dealing with large image set and real time tasks such as visual tracking. The efficacy of the proposed online robust alignment algorithm is verified with extensive experiments on image set alignment and visual tracking, in reference with state-of-the-art methods.

1. Introduction

In the last two decades, the increasing popularity of smart phones and consumer cameras has led to a dramatic increase in the amount of visual data. Further, the image and video sharing web sites, such as Facebook, Flickr and YouTube, make these data easily available online. Analyzing such increasing data challenges existing computer vision algorithms with scalability and image corruptions as il-

lumination variations, occlusions, misalignment. In particular, lack of reliable *and* efficient alignment algorithms for a large amount of images makes it difficult for many image analysis tasks such as face recognition, image classification, and the increasing amount of data requires good scalability of algorithms. Therefore, how to align these large increasing amount of data with both time and memory efficiency has become an urgent problem to be solved.

A lot of work has been done toward the batch image alignment problem, where all the images of an object or objects of interest are aligned to a fixed canonical template. The congealing algorithm was proposed in [10] by seeking an alignment that minimizes the sum of entropies of pixel values at each pixel location in the batch of aligned images. In [4], a least squares congealing approach is presented that seeks an alignment that minimizes the sum of squared distances between image pairs. In [18], minimization of a log-determinant cost function is used for image joint alignment. In [5], an EM algorithm is introduced to optimize a low-rank objective function with respect to domain transformations drawn from a known group.

Our study is motivated by the recent work by Peng *et al.* [16], where a batch image alignment algorithm, named *robust alignment by sparse and low-rank decomposition* (RASL), is proposed to seek an optimal set of image domain transformations such that the matrix of transformed images can be decomposed as the sum of a sparse matrix of errors and a low-rank matrix of recovered aligned images. RASL has been verified over a wide range of realistic misalignments and corruptions. However, when aligning a new coming image to the aligned ones, RASL needs to re-adjust all the transformations of previous images to achieve the rank minimization, which can be very time- and memory-demanding when the image set is large or when it grows quickly, such as in the tracking scenario. This limits the scalability and scope of applications of RASL. Similar alignment frameworks have also been exploited in [8].

To overcome the limitation of batch alignment, we pro-

pose an *online robust image alignment* (ORIA) method that can be applied to large image sets or dynamically increasing ones. Similar as in [16], we assume that the aligned images are linearly correlated and each image, when aligned, can be linearly reconstructed by others. Consequently, for a newly arriving image I , our method seeks an optimal alignment for I such that after alignment I can be linearly reconstructed by previously well-aligned image basis denoted as A . We solve the task by a sequence of convex optimization that minimizes an ℓ_1 -norm. This step is similar as in RASL except we do not take into account of rank minimization. Instead, we keep the rank fixed and assume that the aligned image without corruption lies on the linear low rank subspace spanned by A . This assumption is reasonable for the images of objects drawn from the same category. In other words, we only need to align the newly added image to the well-aligned images one at a time, which is much more efficient in both computation and memory than batch alignment. So, this strategy enables our approach to handle large scale problems.

The second component in ORIA is the update of basis A , which is pursued in two different schemes. The first scheme deals with joint alignment of a fixed image set that is too large to be easily handled by batch alignment. In this case, we propose a two-stage solution. In the first stage, we collect sequentially basis images that are easily aligned according to the alignment residuals. In other words, the basis A is dynamically increased by including reliably aligned images. Then, in the second stage, we re-align all images with large alignment residuals one by one using the basis generated in the first stage. The second scheme deals with dynamically increasing image sequences, such as in the visual tracking scenario. For such scenarios, ORIA dynamically update the image basis by integrating the newly arrived image if it significantly distinguishes itself from current basis images.

Our approach not only inherits the benefits of sparsity, but also enjoys time and memory efficiency. It is able to process one image in less than one second regardless the amount of images in the dataset on a standard PC with small memory occupation. As we will verify with extensive experiments on real image data, the algorithm runs efficiently on a large dataset with thousands of images. Furthermore, we adapt our online alignment algorithm to the visual tracking problem, which cannot be handled by batch aligning algorithm, and it again confirms the efficiency of our algorithm.

In the rest of the paper, we will first describe the proposed online alignment in §2. In particular, the online alignment is formulated in §2.1 and the online basis update in §2.2. After that, experimental results are reported in §3 to showcase the efficacy of our method on real images, followed by the conclusion in §4.

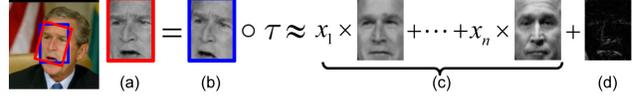


Figure 1. Online image alignment. (a) Aligned image; (b) Original image; (c) Combination of well-aligned images; (d) Sparse error.

2. Online Robust Image Alignment

In this section, we first describe the robust image alignment given known basis images, and then introduce the online basis update procedure. The summary of the algorithm is given in Algorithm 1.

2.1. Robust Image Alignment

Task Formulation. Given a newly arrived image, our task is to align it with previously aligned image basis. Following the formulation in [16], we treat the task as searching for the optimal image transformation such that the input image is warped to have a linear decomposition of a sparse error vector and the aligned basis images. The idea is illustrated in Fig. 1. In the following we use similar notations as in [16] to formulate the task.

Let us denote the newly arrived image as $I \in \mathbb{R}^{w \times h}$, where w and h are width and height of the image. The basis image set containing n images is defined as $\mathcal{I}_n = \{I_i \in \mathbb{R}^{w \times h} : 1 \leq i \leq n\}$. In practice, each image is stacked as a vector function $\text{vec} : \mathbb{R}^{w \times h} \rightarrow \mathbb{R}^d$, where $d = w \times h$ is the vector’s dimension. For conciseness, we use notation ‘ $\vec{\cdot}$ ’ to denote the vectorization operation, i.e., $\vec{I} \doteq \text{vec}(I)$. Finally, we organize the basis images into a matrix $A = [\vec{I}_1, \vec{I}_2, \dots, \vec{I}_n]$.

To align I with A , we need to seek an optimal transformation $\tau : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ in a certain Lie group \mathbb{G} . In the following, we apply the notation τ to the vectorized image \vec{I} such that $\vec{I} \circ \tau \doteq \text{vec}(I \circ \tau)$, where $I \circ \tau(x, y) \doteq I(\tau(x, y))$ denotes the image warping.

Given the above notation, the sparsity regularized alignment can be formulated as an ℓ_1 minimization problem as

$$\min_{\mathbf{x}, \tau} \|\mathbf{e}\|_1 \quad \text{s.t.} \quad \vec{I} \circ \tau = A\mathbf{x} + \mathbf{e}, \quad (1)$$

where \mathbf{x} denotes the reconstruction coefficients and \mathbf{e} denotes the reconstruction error.

The intuition of the above formulation is two-fold. On the one hand, it is known that the global appearance of an object under different illumination and viewpoint conditions lies approximately in a low dimensional subspace. Consequently, linearly decomposing the appearance of an object using an image basis set is commonly used in many vision tasks. On the other hand, such representation in practice is subject to corruptions such noise, occlusions and background clutters. The sparse reconstruction errors have

been used to capture such corruptions. Such a sparse error model has been successfully applied to many vision problems, such as face recognition [20] and visual tracking [14].

Compared with the formulation of RASL [16], which minimizes the error sparsity and the rank of aligned images simultaneously, ORIA only minimizes the error sparsity and preserves the rank of the aligned images by assuming all the well-aligned images lie in a low dimensional linear subspace. Each time ORIA aligns and projects one image to the subspace while RASL needs to align all the images by mining the underlying low rank structure of the dataset. When the dataset changes (adding or deleting images), RASL needs to re-align all the images in the dataset to discover the new low rank structure. Therefore, ORIA can be viewed as an online adaption of RASL. In practice, we can employ RASL on the subset of the dataset to generate the aligned basis images for ORIA and then ORIA can align the newly coming images on-the-fly.

Iterative Convex Optimization. The optimization problem (1) is non-convex in that the constraint $\vec{I} \circ \tau = \mathbf{A}\mathbf{x} + \mathbf{e}$ is nonlinear in $\tau \in \mathbb{G}$. Inspired by the work in [16, 19, 23], we solve the problem by an iterative convex optimization framework. The key idea is to linearize the constraint and iteratively improve the estimated solution. The linearization is the first order approximation as $\vec{I} \circ (\tau + \Delta\tau) \approx \vec{I} \circ \tau + J\Delta\tau$, where $\Delta\tau \in \mathbb{R}^p$ and $J \in \mathbb{R}^{d \times p}$ is the Jacobian of \vec{I} w.r.t. the transformation τ , which is determined by p parameters. This way, the optimization problem (1) reduces to

$$\min_{\mathbf{x}, \mathbf{e}, \Delta\tau} \|\mathbf{e}\|_1 \quad s.t. \quad \vec{I} \circ \tau + J\Delta\tau = \mathbf{A}\mathbf{x} + \mathbf{e}. \quad (2)$$

This linearized formulation is now a convex programming and is amenable to solve efficiently. By iteratively solving the problem in (2), we can approximate the solution to the original non-convex problem in (1). Such an iterative linearization scheme is a common technique in optimization to solve nonlinear problems, which converges quadratically to a local minimum of the original non-linear problem [16].

To solve the convex programming in (2), *augmented Lagrange multiplier* (ALM) [12] is utilized for efficiency. Specifically, we have the following augmented Lagrangian function for our problem (2):

$$\mathcal{L}_\mu(\mathbf{x}, \mathbf{e}, \Delta\tau, \lambda) = \|\mathbf{e}\|_1 + \lambda^\top \mathbf{h}(\mathbf{x}, \mathbf{e}, \Delta\tau) + \frac{\mu}{2} \|\mathbf{h}(\mathbf{x}, \mathbf{e}, \Delta\tau)\|_2^2, \quad (3)$$

where $\mathbf{h}(\mathbf{x}, \mathbf{e}, \Delta\tau) = \vec{I} \circ \tau + J\Delta\tau - \mathbf{A}\mathbf{x} - \mathbf{e}$ denotes the constraints; $\lambda \in \mathbb{R}^d$ is a Lagrange multiplier vector; and μ is a positive penalty scalar.

Intuitively, when μ is sufficiently large, the augmented Lagrangian function shares the same minimizer as the original constrained optimization problem [2] for appropriately chosen Lagrange multiplier vector λ . Consequently, we can

Algorithm 1 The ORIA Algorithm

- 1: **Input:** Image basis set $A \in \mathbb{R}^{d \times n}$, image $I \in \mathbb{R}^{w \times h}$, and initial transformation $\tau = \tau_0 \in \mathbb{G}$
 - 2: Initialize $\vec{I} \circ \tau_0 = \text{vec}(I \circ \tau_0)$
 - 3: **while** not converged ($n = 0, 1, \dots$) **do**
 - 4: Update the Jacobian matrix: $J_n = \left. \frac{\partial(\vec{I} \circ \zeta)}{\partial \zeta} \right|_{\zeta = \tau_n}$
 - 5: Initialize $\mathbf{x}_0 = 0$, $\mathbf{e}_0 = 0$, $\Delta\tau_0 = 0$, $\lambda_0 = \vec{I} \circ \tau_n$, $\mu_0 = 1.25$
 - 6: **while** not converged ($k = 0, 1, \dots$) **do**
 - 7: $(\mathbf{x}_{k+1}, \mathbf{e}_{k+1}, \Delta\tau_{k+1}) = \arg \min_{\mathbf{x}, \mathbf{e}, \Delta\tau} \mathcal{L}_{\mu_k}(\mathbf{x}, \mathbf{e}, \Delta\tau, \lambda_k)$
 - 8: $\lambda_{k+1} = \lambda_k + \mu_k \mathbf{h}(\mathbf{x}_{k+1}, \mathbf{e}_{k+1}, \Delta\tau_{k+1})$
 - 9: $\mu_{k+1} = \rho \cdot \mu_k$
 - 10: **end while**
 - 11: Update the transformation: $\tau_{n+1} = \tau_n + \Delta\tau_k$;
 - 12: Update the transformed image vector: $\vec{I} \circ \tau_{n+1} = \text{vec}(I \circ \tau_{n+1}) / \|\text{vec}(I \circ \tau_{n+1})\|_2$
 - 13: **end while**
 - 14: Update basis set A according to §2.2
 - 15: **Output:** Solution \mathbf{x}^* , \mathbf{e}^* , τ^* to problem (1), an updated basis set A .
-

iteratively minimize (3) with an increasing set of μ , with the following steps:

$$\begin{aligned} (\mathbf{x}_{k+1}, \mathbf{e}_{k+1}, \Delta\tau_{k+1}) &= \arg \min_{\mathbf{x}, \mathbf{e}, \Delta\tau} \mathcal{L}_{\mu_k}(\mathbf{x}, \mathbf{e}, \Delta\tau, \lambda_k), \\ \lambda_{k+1} &= \lambda_k + \mu_k \mathbf{h}(\mathbf{x}_{k+1}, \mathbf{e}_{k+1}, \Delta\tau_{k+1}), \quad (4) \\ \mu_{k+1} &= \rho \cdot \mu_k, \end{aligned}$$

where $\rho > 1$ controls the steps of μ and $\{\mu_k > 0, k = 0, 1, \dots\}$ forms a monotonically increasing sequence.

Note that the proposed solution is similar to the one used in RASL. The main difference lies in that RASL includes an additional penalty over the rank of the basis A . The solver for (4) is also similar as that in RASL, the details can be referred to [16].

2.2. Online Basis Update

Once a new image, say I , is aligned with current basis, say A , we will update A according to the task requirement. We pursue basis update in two different schemes. The first scheme, which contains two stages, deals with joint alignment of a large amount of images from the same category. The second scheme focuses on dynamically increasing image sequences, such as in the visual tracking scenario.

Joint Image Set Alignment. It is impractical, in terms of both running time and memory request, to batch aligning a large image data set. Instead, ORIA adopts an online solution with a two-stage basis update scheme. The first stage focuses on the easy cases, which are identified by $\|\mathbf{e}\|_1$, and

after the basis set is improved in this stage, the next stage will focus on the hard cases.

In the first stage, we start with a synthetic basis set A generated from the first image by disturbing it with small transformations. By this way, the images in A are visually similar but A has full column rank. Although the representative ability of this initial subspace is not powerful, we can use it to collect informative basis online. In the online process, when an image is well-aligned to current basis set, i.e., with small reconstruction residual $\|e\|_1 \leq t_0$ for some threshold t_0 (10 in our experiments), we update basis A with this image. In particular, the synthetic basis images will be replaced first. After all the synthetic bases are replaced, the size of basis set increases when adding the new basis. For computation efficiency, we add constraint to the size of basis set (e.g., 100). If the size constraint is reached, when adding a new basis, it will replace the oldest one, except the first one, in the basis set.

Intuitively, when $\|e\|_1$ is large, the image may be occluded seriously or still be mis-aligned. In this case, the image will not serve as a basis and it will be re-aligned in the second stage. We run the first stage to enrich the basis set and then start the second stage.

Online Image Sequence Alignment. For the alignment of image sequences, in particular in the scenario of visual tracking, the image set is not known a priori. In this case, we can not apply the two-stage basis update, while instead updating the basis on-the-fly. Specifically, in the first frame, a synthetic basis set A is generated in the same way as in the joint image set alignment. Then, when a new image I arrives, it is first aligned with A using the method described in Section 2. Then, if the reconstruction residual $\|e\|_1$ is greater than a predefined threshold t_2 , we will update the basis A with the aligned result, i.e., $I \circ \tau$.

3. Experiments

In this section, we first demonstrate the speed and scalability of the proposed approach and then evaluate its efficiency on several image alignment tasks. After that, we adapt ORIA to the online visual tracking problem, where the results also verify the robustness of our approach.

3.1. Speed and scalability of ORIA

The ORIA formulation consists of solving a sequence of convex optimization problems. Recent advances in ℓ_1 -norm minimization have enabled us to develop scalable algorithms for ORIA. We provide the running time for an example case to give an idea of the efficiency of our algorithm.

The speed comparison experiment between ORIA and RASL is carried out on a video sequence *gore* containing 140 images. The alignment results of both algorithms are shown in Fig.3. The affine transformations $\mathbb{G} = \text{Aff}(2)$ is

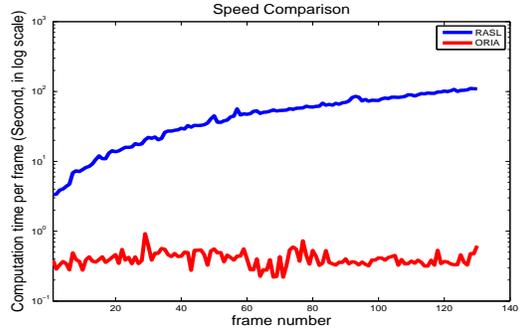


Figure 2. Running time comparison between ORIA and RASL

used. Each face is aligned to be an 80×60 canonical face. In order to compare the alignment results with RASL visually, we use the first ten images from the alignment results generated by RASL as the basis set of our ORIA. Then, the remaining images of *gore* are aligned to these well-aligned images one by one. To test the computation time of RASL w.r.t the growing amount of images in the dataset, RASL is running in a online batch mode, namely the data matrix as the input to RASL is added one column once. The computation time of the two approaches is illustrated in Fig. 2. We can see that the computation time of our ORIA keeps almost constant, while the computation complexity of RASL is growing linearly as the size of image set increases. Averagely, on a PC with Intel Q8200 2.33 GHz CPU with a MATLAB implementation, ORIA needs only 0.4 seconds to warp the newly arrived image to the subspace spanned by the well-aligned basis. RASL processes the first arrived images in 3.4 seconds, and when the last image arrives, it needs around 110 seconds to align all the images. Observed from experiments, the inner loop of our ORIA is at least 15 times faster than RASL. RASL cannot adopt the historical alignment information and wastes a large amount of computation resource to adjust the alignment of previous images.

Furthermore, RASL needs to keep all the previous images to carry out the batch alignment. This limits its application to large scale dataset or online vision problems. In contrast, our method needs only keep the basis A and the current arrived image. Both the computation and memory efficiency allows our method to be applied to the practical online vision problems. One example among these applications is visual tracking, which will be shown in §3.3.

3.2. Joint Image Set Alignment

We test our algorithm on a subset of the Labeled Faces in the Wild (LFW) [7] and handwritten digits. This subset of LFW contains 19 subjects and each has 35 images. The initial basis set is constructed from one face from Gloria Macapagal and then the basis set is updated by the following well-aligned faces. The average faces before and after alignment are shown in Fig.4(a) and Fig.4(b) respec-

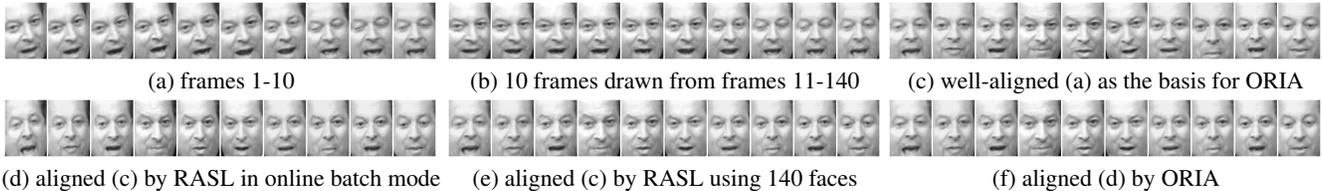


Figure 3. Image alignment on 140-frame video sequence *gore*. The original face images are cropped out by applying a face detector to each frame. (a) frames 1-10; (b) 10 frames uniformly drawn from frames 11-140; (c) the first 10 aligned faces from all the 140 faces aligned by RASL. It takes as the basis for ORIA. (d) aligned faces in (c) by RASL using online batch mode (the amount of images as the input of RASL is increasing). Note that the first image is not aligned to the other nine; (e) aligned frames in (c) by RASL taking all the 140 faces as input; (f) aligned faces in (c) by ORIA.

tively. We can see that the average faces after alignment is much clearer than those before alignment. The specific alignment results for Ariel Sharon and Donald Rumsfeld are illustrated in Fig.5. Note that RASL in [16] is used to align each subject while our ORIA here is used to align faces across subjects as well. Even though alignment across subjects is much more difficult than within each subject, ORIA achieves almost the same alignment results as RASL. This also verifies that the proposed basis update strategy is effective. We also tested ORIA to the alignment of digit images taken from the MNIST database. For this experiment, we use 100 images of the handwritten “3”, of size 29×29 pixels and the results are shown in. Fig.4 (c-d).

By comparing the results of ORIA to those in [16] generated by RASL, we can see that our ORIA achieves very similar results as RASL. However, to generate the same results ORIA uses much less computation resources (both CPU and memory).

3.3. Visual Tracking

Visual tracking is a critical task in many computer vision applications. The challenges in designing a robust visual tracking algorithm are caused by the presence of noise, occlusions, varying viewpoints, background clutter, and illumination changes. Many tracking approach are based on the linear subspace representation. In [3], the eigenspace is adopted to represent the target. Further, the incremental eigenspace update is proposed in [17]. Recently, in [14] sparse representation is proposed for visual tracking, where the sparsity is achieved by solving an ℓ_1 -regularized least squares problem. Then, the group sparsity is used for improving tracking robustness [13]. Later, some approaches are proposed to accelerate the L1 tracker [15, 11]. In [22], the sparse representation is extended to the motion-blurred target tracking. Our ORIA tracker is also based on the linear subspace representation. Here, we will see that our ORIA is qualified for this challenging online task and outperforms many state-of-the-art trackers.

In our visual tracking experiments, each target is initialized manually by a bounding box in the first frame. The initial transformation for current target is the estimated trans-

formation of target in the last frame. One of critical problem in visual tracking is model update. Here, we adopted the strategy proposed for *online image sequence alignment*.

Our ORIA tracker is compared with five latest state-of-the-art trackers named Incremental Visual Tracking (IVT) [17], Multiple Instance Learning (MIL) [1], Visual Tracking Decomposition (VTD) [9], Incremental Covariance Tensor Learning (ICTL) [21], and Online AdaBoost (OAB) [6]. The tracking results of the compared methods are obtained by running the source code or binaries provided by their authors using the same initial positions in the first frame.

3.4. Tracking results

We first test ORIA on the sequence *rubik*¹, which lasts around 2000 frames. As illustrated in Fig.6(a) one planar surface of the rubik is undergoing affine transformation over time and our ORIA can robustly estimate the state of the target regardless the pose variations and occlusions. In the sequence *car*, the vehicle undergoes drastic illumination changes and some samples of the final tracking results are demonstrated in Fig.6(b). Our ORIA tracker can track target well even though the illumination changes. In the sequence *singer*, although the target is undergoing gradually large scale changes (#8, #350), severe illumination variations (#85, #133) and viewpoint changes (#8, #350), our ORIA tracker can track the object accurately as illustrated in Fig.6(c). Fig.6(d) shows the tracking results of sequence *woman*, where the target is severely occluded by another object (#197) and the scale changes gradually (#500). Our ORIA tracker can follow the target throughout the sequence in spite of these challenging conditions. In sequence *pole* (Fig.6(e)), the IVT fails to track the target from the beginning and the VTD also loses the target after #274. Our tracker and the rest trackers successfully track the target. Finally, we test our ORIA tracker on the sequence *sylv* (Fig.6(f)). Although the pose of the target changes severely together with lighting variations, our ORIA tracker follows the target throughout the sequence.

¹The implementation of VTD from the original authors can only output the results for the first 1000 frames of *rubik*.



(a) average faces before alignment (b) average faces after alignment (c) original digit images (d) aligned digit images by ORIA
 Figure 4. Alignment results of face images and handwritten digits. In (a) and (b), the results marked with red boxes are the average faces of the whole face set and others are the average faces of different subjects.



(a) original images (b) aligned by ORIA (c) original images (d) aligned by ORIA

Figure 5. Alignment of personal specific image sets.

	MIL	OAB	ICTL	VTD	IVT	ORIA
car	0.749	0.786	0.326	0.313	0.049	0.033
singer	0.299	0.466	0.503	0.056	0.155	0.082
woman	0.361	0.179	0.323	0.339	0.148	0.021
pole	0.007	0.010	0.008	0.049	0.572	0.003
sylv	0.069	0.058	0.096	0.203	0.197	0.031
Ave.	0.297	0.300	0.251	0.192	0.224	0.034

Table 1. The average tracking errors. The error is measured using the Euclidian distance of two center points, which has been normalized by the size of the target from the ground truth. The last row is the average error for each tracker over all the test sequences.

Qualitative comparison. To quantitatively compare robustness under challenging conditions, we manually annotated the target’s bounding box in each frame for all the test sequences except *rubik*. As shown in Table 1, the position differences of the ORIA tracker are smaller than those of the other trackers.

4. Conclusions

We have presented an online alignment method that can incrementally align images to the well-aligned basis images despite gross corruptions. Our approach seeks an optimal image domain transformation such that the transformed im-

age can be decomposed as the sum of a sparse error and a linear composition of well-aligned basis set by solving a sequence of convex optimization problems. Our method inherits the benefits of sparsity and enjoys both time and memory efficiency. We have shown the efficacy of our method with extensive experiments on image alignment under a wide range of real-world conditions. Further, we adapt our approach to an online vision application, visual tracking, where the experimental results demonstrate our method achieves higher accuracy than many existing methods.

Acknowledgment. We thank the reviewers for valuable comments and suggestions. The work is supported partly by NSF Grants IIS-0916624 and IIS-1049032. Wu is supported partly by NSFC (Grant No. 61005027) and PADA.

References

- [1] B. Babenko, M. Yang, and S. Belongie. “Visual Tracking with On-line Multiple Instance Learning”, in *CVPR*, 2009. 5
- [2] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2004. 3
- [3] M. J. Black and A. D. Jepson. “Eigenttracking: Robust matching and tracking of articulated objects using view-based representation”, *IJCV*, 26:63-84, 1998. 5
- [4] M. Cox, S. Lucey, S. Sridharan, and J. Cohn. “Least squares co-gealing for unsupervised alignment of images”, in *CVPR*, 2008. 1
- [5] B.J.Frey and N. Jojic. “Transformation-invariant clustering using the EM algorithm”, *PAMI*, 25(1):1-17, 2003. 1

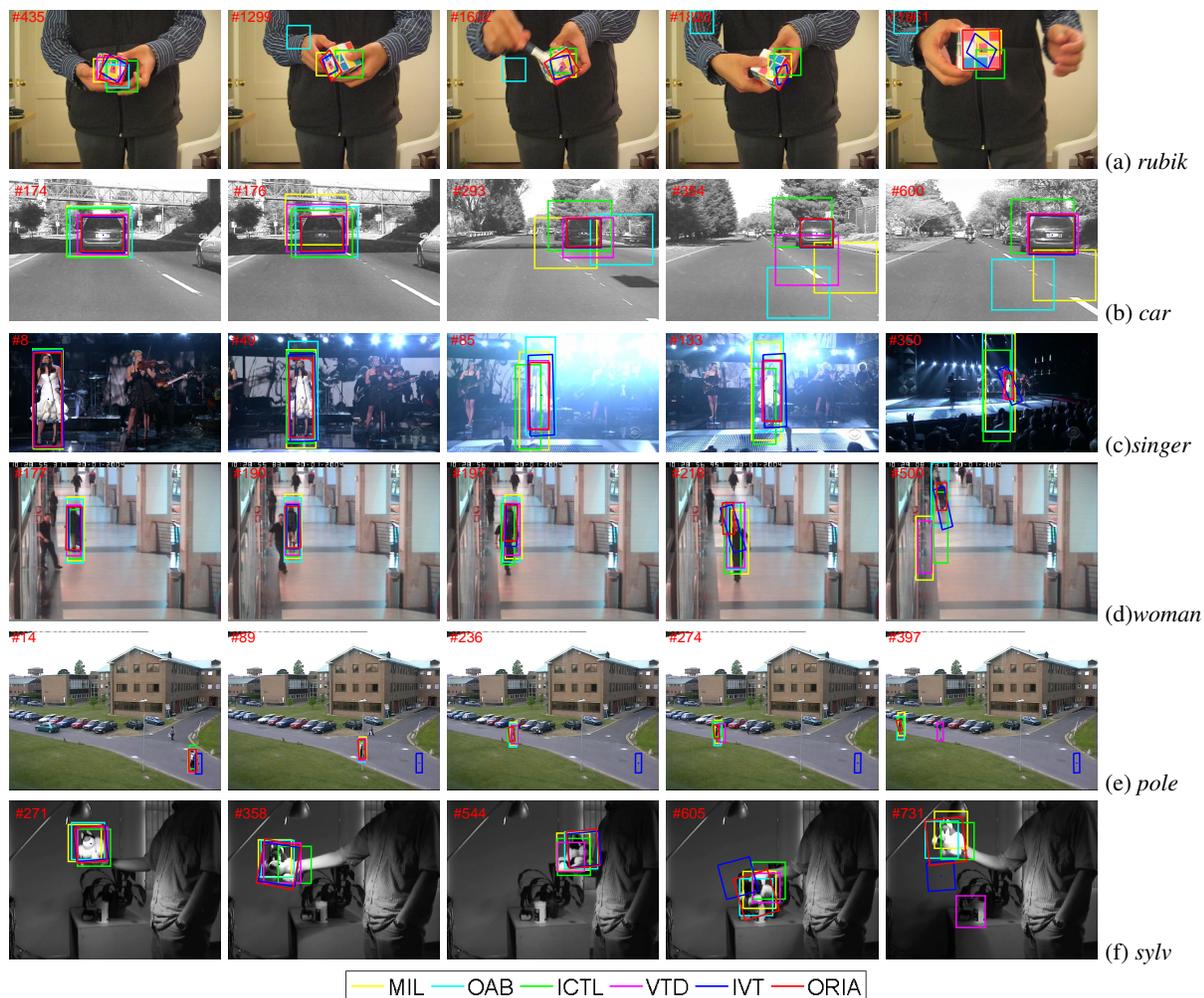


Figure 6. Tracking results of different algorithms. Sequence names are *rubik(a)*, *car(b)*, *singer(c)*, *woman(d)*, *pole(e)* and *sylv(f)*.

- [6] H. Grabner, M. Grabner, and H. Bischof. “Real-time tracking via online boosting”, in *BMVC*, 2006. 5
- [7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. “Labeled faces in the wild: A database for studying face recognition in unconstrained environments”, in *The ECCV Workshop on Faces in Real-Life Images*, 2008. 4
- [8] J. Huang, X. Huang, and D. Metaxas, “Simultaneous Image Transformation and Sparse Representation Recovery,” in *CVPR*, 2008. 1
- [9] J. Kwon and K. M. Lee. “Visual Tracking Decomposition”, in *CVPR*, 2010. 5
- [10] E. Learned-Miller. “Data driven image models through continuous joint alignment”, *PAMI*, 28(2):236-250, 2006. 1
- [11] H. Li, C. Shen, and Q. Shi. “Real-time Visual Tracking Using Compressive Sensing”, in *CVPR*, 2011. 5
- [12] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. “The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices”, *UIUC Technical Report UILU-ENG-09-2214*, 2010. 3
- [13] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. “Robust and fast collaborative tracking with two stage sparse optimization”, in *ECCV*, 2010. 5
- [14] X. Mei and H. Ling. “Robust Visual Tracking and Vehicle Classification via Sparse Representation”, *PAMI*, 33(11):2259-2272, 2011. 3, 5
- [15] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. “Minimum Error Bounded Efficient ℓ_1 Tracker with Occlusion Detection”, in *CVPR*, 2011. 5
- [16] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. “RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images”, *PAMI*, 2012, in press. 1, 2, 3, 5
- [17] D. A. Ross, J. Lim, R. Lin and M. Yang. “Incremental learning for robust visual tracking”, *IJCV*, 77:125-141, 2008. 5
- [18] A. Vedaldi, G. Guidi, and S. Soatto. “Joint alignment up to (lossy) transformations”, in *CVPR*, 2008. 1
- [19] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma. “Towards a Practical Face Recognition System: Robust Registration and Illumination via Sparse Representation”, in *CVPR*, 2009. 3
- [20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. “Robust Face Recognition via Sparse Representation”, *PAMI*, 31(1):210-227, 2009. 3
- [21] Y. Wu, J. Cheng, J. Wang, H. Lu, J. Wang, H. Ling, E. Blasch, and L. Bai. “Real-time Probabilistic Covariance Tracking with Efficient Model Update”, *T-IP*, 2012, in press. 5
- [22] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. “Blurred Target Tracking by Blur-driven Tracker”, in *ICCV*, 2011. 5
- [23] Z. Zhang, A. Ganesh, X. Liang and Y. Ma. “TILT: Transform Invariant Low-rank Textures”, *IJCV*, 2012, in press. 3