Object Tracking via Non-Euclidean Geometry: A Grassmann Approach

Sareh Shirazi, Mehrtash T. Harandi, Brian C. Lovell, Conrad Sanderson

NICTA, GPO Box 2434, Brisbane, QLD 4001, Australia
University of Queensland, School of ITEE, QLD 4072, Australia
Queensland University of Technology, Brisbane, QLD 4000, Australia

Abstract

A robust visual tracking system requires an object appearance model that is able to handle occlusion, pose, and illumination variations in the video stream. This can be difficult to accomplish when the model is trained using only a single image. In this paper, we first propose a tracking approach based on affine subspaces (constructed from several images) which are able to accommodate the abovementioned variations. We use affine subspaces not only to represent the object, but also the candidate areas that the object may occupy. We furthermore propose a novel approach to measure affine subspace-to-subspace distance via the use of non-Euclidean geometry of Grassmann manifolds. The tracking problem is then considered as an inference task in a Markov Chain Monte Carlo framework via particle filtering. Quantitative evaluation on challenging video sequences indicates that the proposed approach obtains considerably better performance than several recent state-of-the-art methods such as Tracking-Learning-Detection and MILtrack.

1. Introduction

Visual tracking is a fundamental task in many computer vision applications including event analysis, visual surveillance, human behaviour analysis, and video retrieval [18]. It is a challenging problem, mainly because the appearance of tracked objects changes over time. Designing an appearance model that is robust against intrinsic object variations (*e.g.* shape deformation and pose changes) and extrinsic variations (*e.g.* camera motion, occlusion, illumination changes) has attracted a large body of work [4, 24].

Rather than relying on object models based on a single training image, more robust models can be obtained through the use of several images, as evidenced by the recent surge of interest in object recognition techniques based on image-set matching. Among the many approaches to image-set matching, superior discrimination accuracy, as well as increased robustness to practical issues (such as pose and illumination variations), can be achieved by modelling image-sets as linear subspaces [10, 11, 12, 20, 21, 22].

In spite of the above observations, we believe modelling via linear spaces is not completely adequate for object tracking. We note that all linear subspaces of one specific order have a common origin. As such, linear subspaces are theoretically robust against translation, meaning a linear subspace extracted from a set of points does not change if the points are shifted equally. While the resulting robustness against small shifts is attractive for object recognition purposes, the task of tracking is to generally maintain precise locations of objects.

To account for the above problem, in this paper we first propose to model objects, as well as candidate areas that the objects may occupy, through the use of generalised linear subspaces, *i.e.* affine subspaces, where the origin of subspaces can be varied. As a result, the tracking problem can be seen as finding the most similar affine subspace in a given frame to the object's affine subspace. We furthermore propose a novel approach to measure distances between affine subspaces, via the use of non-Euclidean geometry of Grassmann manifolds, in combination with Mahalanobis distance between the origins of the subspaces. See Fig. 1 for a conceptual illustration of our proposed distance measure.

To the best of our knowledge, this is the first time that appearance is modelled by affine subspaces for object tracking. The proposed approach is somewhat related to adaptive subspace tracking [13, 19]. Ho *et al.* [13] represent an object as a point in a linear subspace, which is constantly updated. As the subspace was computed using only recent tracking results, the tracker may drift if large appearance changes occur. In addition, the location of the tracked object is inferred via measuring point-to-subspace distance, which is in contrast to the proposed method, where a more robust subspace-to-subspace distance is used.

Ross *et al.* [19] improved tracking robustness against large appearance changes by modelling objects in a low-dimensional subspace, updated incrementally using all preceding frames. Their method also involves a point-to-subspace distance measurement to localise the object.

The proposed method should not be confused with subspace learning on Grassmann manifolds proposed by Wang *et al.* [25]. More specifically, in [25] an online subspace learning scheme using Grassmann manifold geometry is devised to learn/update the subspace of object appearances. In contrast to the proposed method, they also consider the point-to-subspace distance to localise objects.

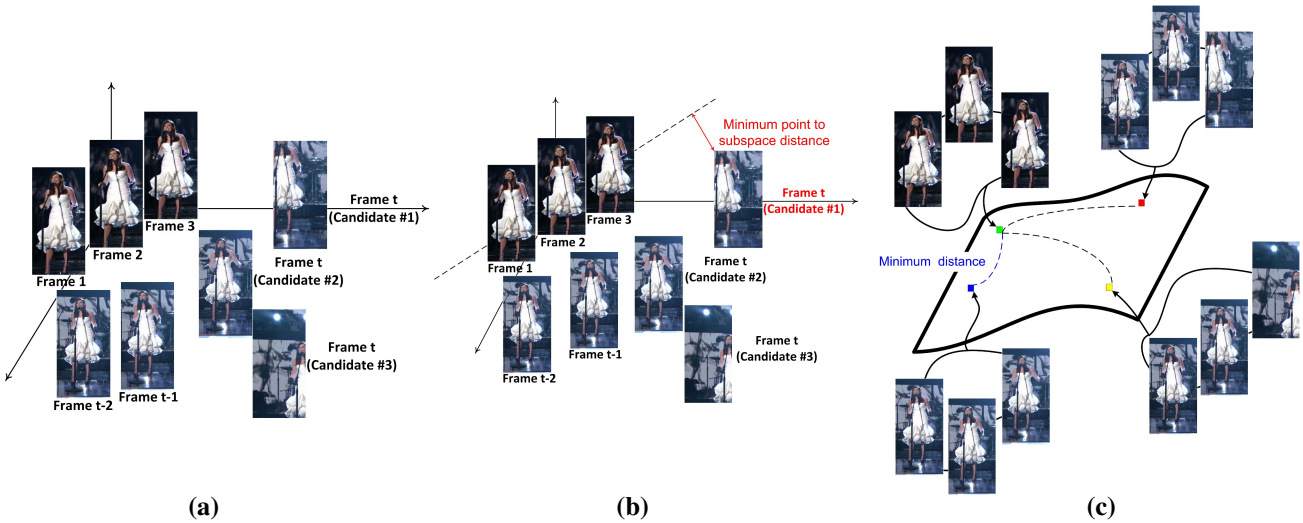

Figure 1. Difference between point-to-subspace and subspace-to-subspace distance measurement approaches. **(a)** Three groups of images, with each image represented as a point in space; the first group (top-left) contains three consecutive object images (frames 1, 2 and 3) used for generating the object model; the second group (bottom-left) contains tracked object images from frames $t-2$ and $t-1$; the third group (right) contains three candidate object regions from frame t. **(b)** Subspace generated based on object images from frames 1, 2 and 3, represented as a dashed line; the minimum point-to-subspace distance can result in selecting the wrong candidate region (*i.e.* wrong location). **(c)** Generated subspaces, represented as points on a Grassmann manifold; the top-left subspace represents the object model; each of the remaining subspaces was generated by using tracked object images from frames $t-2$ and $t-1$, with the addition of a unique candidate region from frame t; using subspace-to-subspace distance is more likely to result in selecting the correct candidate region.

2. Proposed Affine Subspace Tracker (AST)

The proposed Affine Subspace Tracker (AST) is comprised of four components, overviewed below. A block diagram of the proposed tracker is shown in Fig. 2.

1. **Motion Estimation**. This component takes into account the history of object motion in previous frames and creates a set of candidates as to where the object might be found in the new frame. To this end, it parameterises the motion of the object between consecutive frames as a distribution via particle filter framework [2]. Particle filters are sequential Monte Carlo methods and use a set of points to represent the distribution. As a result, instead of scanning the whole of the new frame to find the object, only highly probable locations will be examined.

2. **Candidate Subspaces**. This module encodes the appearance of a candidate (associated to a particle filter) by an affine subspace $A_i^{(t)}$. This is achieved by taking into account the history of tracked images and learning the origin $\mu_i^{(t)}$ and basis $U_i^{(t)}$ of $A_i^{(t)}$ for each particle.

3. **Decision Making**. This module measures the likelihood of each candidate subspace $A_i^{(t)}$ to the stored object models in the bag $\mathcal{M}$. Since object models are encoded by affine subspaces as well, this module determines the similarity between affine subspaces. The most similar candidate subspace to the bag $\mathcal{M}$ is selected as the result of tracking.

4. **Bag of Models**. This module keeps a history of previously seen objects in a bag. This is primarily driven by the fact that a more robust and flexible tracker can be attained if a history of variations in the object appearance is kept [15]. To understand the benefit of the bag of models, assume a person tracking is desired where the appearance of whole body is encoded as an object model. Moreover, assume at some point of time only the upper body of person is visible (due to partial occlusion) and the tracker has successfully learned the new appearance. If the tracking system is only aware of the very last seen appearance (upper-body in our example), upon termination of occlusion, the tracker is likely to lose the object. Keeping a set of models (in our example both upper-body and whole body) can help the tracking system to cope with drastic changes.

Each of the components is elucidated in the following subsections.

2.1. Motion Estimation

In the proposed framework, we are aiming to obtain the location $x \in \mathcal{X}$, $y \in \mathcal{Y}$ and the scale $s \in \mathcal{S}$ of an object in frame t based on prior knowledge about previous frames.

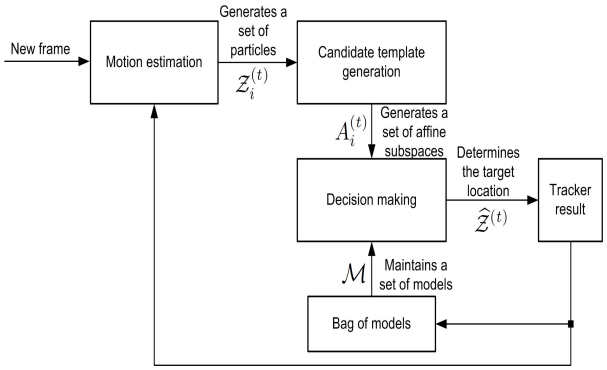

Figure 2. Block diagram for the proposed Affine Subspace Tracker (AST).

A blind search in the space of $\mathcal{X}-\mathcal{Y}-\mathcal{S}$ is obviously inefficient, since not all possible combinations of x, y and s are plausible. To efficiently search the $\mathcal{X}-\mathcal{Y}-\mathcal{S}$ space, we use a sequential Monte Carlo method known as the Condensation algorithm [14] to determine which combinations in the $\mathcal{X}-\mathcal{Y}-\mathcal{S}$ space are most probable at time t. The key idea is to represent the $\mathcal{X}-\mathcal{Y}-\mathcal{S}$ space by a density function and estimate it through a set of random samples (also known as particles). As the number of particles becomes large, the condensation method approaches the optimal Bayesian estimate of density function (*i.e.* combinations in the $\mathcal{X}-\mathcal{Y}-\mathcal{S}$ space). Below, we briefly describe how the condensation algorithm is used within the proposed tracking approach.

Let $\mathcal{Z}^{(t)} = \left(x^{(t)}, y^{(t)}, s^{(t)}\right)$ denote a particle at time t. By the virtue of the principle of importance sampling [2], the density of $\mathcal{X}-\mathcal{Y}-\mathcal{S}$ space (or most probable candidates) at time t is estimated as a set of N particles $\{\mathcal{Z}_i^{(t)}\}_{i=1}^N$ using previous particles $\{\mathcal{Z}_i^{(t-1)}\}_{i=1}^N$ and their associated weights $\{w_i^{(t-1)}\}_{i=1}^N$ with $\sum_{i=1}^N w_i^{(t-1)} = 1$. For now we assume the associated weights of particles are known and later discuss how they can be determined.

In the condensation algorithm, to generate $\{\mathcal{Z}_i^{(t)}\}_{i=1}^N$, $\{\mathcal{Z}_i^{(t-1)}\}_{i=1}^N$ is first sampled (with replacement) N times. The probability of choosing a given element $\mathcal{Z}_i^{(t-1)}$ is equal to the associated weight $w_i^{(t-1)}$. Therefore, the particles with high weights might be selected several times, leading to identical copies of elements in the new set. Others with relatively low weights may not be chosen at all. Next, each chosen element undergoes an independent Brownian motion step. Here, the Brownian motion of a particle is modelled by a Gaussian distribution with a diagonal covariance matrix. As a result, for a chosen particle $\mathcal{Z}_*^{(t-1)}$ from the first step of condensation algorithm, a new particle $\mathcal{Z}_*^{(t)}$ is obtained as a random sample of $\mathcal{N}\left(\mathcal{Z}_*^{(t-1)}, \Sigma\right)$ where $\mathcal{N}(\mu, \Sigma)$ denotes a Gaussian distribution with mean μ and covariance Σ. The covariance Σ governs the speed of motion, and is a constant parameter over time in our framework.

2.2. Candidate Templates

To accommodate variations in object appearance, this module models the appearance of particles[1] by affine subspaces (see Fig. 3 for a conceptual example). An affine subspace is a subset of Euclidean space [23], formally described by a 2-tuple $\{\mu, U\}$ as:

$$\boldsymbol{A} = \left\{\boldsymbol{z} \in \mathbb{R}^D : \boldsymbol{z} = \mu + \boldsymbol{U}\boldsymbol{y}\right\} \tag{1}$$

where $\mu \in \mathbb{R}^D$ and $\boldsymbol{U} \in \mathbb{R}^{D\times n}$ are origin and basis of the subspace, respectively. Let $\boldsymbol{I}(\mathcal{Z}_*^{(t)}, t)$ denote the vector representation of an $N_1 \times N_2$ patch extracted from frame t by considering the values of particle $\mathcal{Z}_*^{(t)}$. That is, frame t is first scaled appropriately based on the value $s_*^{(t)}$ and then a patch of $N_1 \times N_2$ pixels with the top left corner located at $\left(x_*^{(t)}, y_*^{(t)}\right)$ is extracted.

The appearance model for $\mathcal{Z}_*^{(t)}$ is generated from a set of $P+1$ images by considering P previous results of tracking. More specifically, let $\widehat{\mathcal{Z}}^{(t)}$ denote the result of tracking at time t, *i.e.* $\widehat{\mathcal{Z}}^{(t)}$ is the most similar particle to the bag of models at time t. Then set $\mathbb{B}_{\mathcal{Z}_*}^{(t)} = \left\{\boldsymbol{I}(\widehat{\mathcal{Z}}^{(t-P)}, t-P), \boldsymbol{I}(\widehat{\mathcal{Z}}^{(t-P+1)}, t-P+1), \cdots, \boldsymbol{I}(\mathcal{Z}_*^{(t)}, t)\right\}$ is used to obtain the appearance model for particle $\mathcal{Z}_*^{(t)}$. More specifically, the origin of affine subspace associated to $\mathcal{Z}_*^{(t)}$ is the mean of $\mathbb{B}_{\mathcal{Z}_*}^{(t)}$. The basis is obtained by computing the Singular Value Decomposition (SVD) of $\mathbb{B}_{\mathcal{Z}_*}^{(t)}$ and choosing the n dominant left-singular vectors.

2.3. Bag of Models

Although affine subspaces accommodate object changes along with a set of images, to produce a robust tracker, the object's model should be able to reflect the appearance changes during the tracking process. Accordingly, we propose to keep a set of object models $m_j = \{\mu_j, U_j\}$ for coping with deformations, pose variations, occlusions, and other variations of the object during tracking.

[1] We loosely use "particle appearance" to mean the appearance of a candidate template described by a particle.

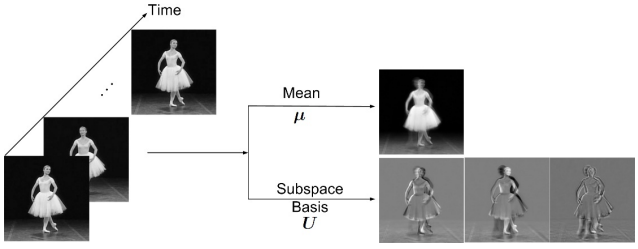

Figure 3. In the proposed approach, object appearance is modelled by an affine subspace. An affine subspace is uniquely described by its origin μ and basis U. Here, μ and basis U are obtained by computing mean and eigenbasis of a set of object images.

Fig. 4 shows two frames with a tracked object, the bag models used to localise the object, and the recent images of the image set used to generate each bag model.

A bag $\mathcal{M} = \{m_1, \cdots, m_k\}$ is defined as a set of k object models, *i.e.* each m_j is an affine subspace learned during the tracking process. The bag is updated every W frames (see Fig. 5) by replacing the oldest model with the latest learned model (*i.e.* latest result of tracking specified by $\widehat{\mathcal{Z}}^{(t)}$). The size of bag k determines the memory of the tracking system. Thus, a large bag with several models might be required to track an object in a challenging scenario. In our experiments, a bag of size 10 with the updating rate $W = 5$ is used in all experiments.

Having a set of models at our disposal, we will next address how the similarity between a particle's appearance and the bag can be determined.

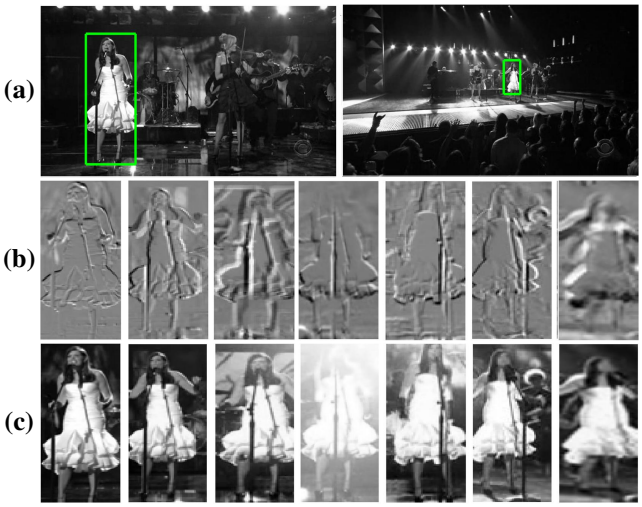

Figure 4. **(a)** Two examples of a frame with a tracked object. **(b)** The first eigenbasis of ten sample template bags. **(c)** The recent frame in each of the 10 image sets used to generate the templates.

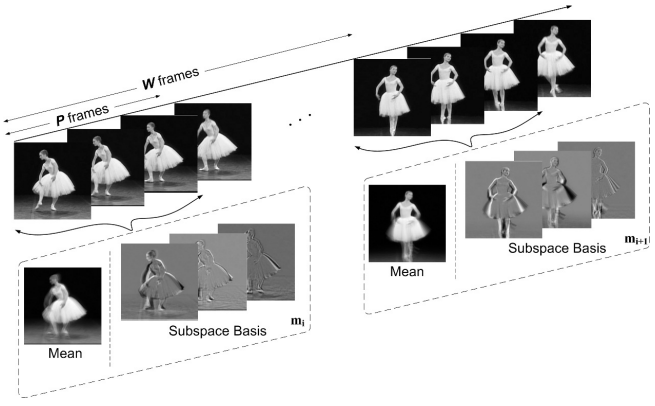

Figure 5. The model extraction procedure involves a sliding window update scheme. The template is learned from a set of *P* consecutive frames. Template update occurs every *W* frames.

2.4. Decision Making

Given the previously learned affine subspaces as the input to this module, the aim is to find the nearest affine subspace to the bag templates. Although the minimal Euclidean distance is the simplest distance measure between two affine subspaces (*i.e.* the minimum distance of any pair of points of the two subspaces), this measure does not form a metric [5] and it does not consider the angular distance between affine subspaces, which can be a useful discriminator [16]. However, the angular distance ignores the origin of affine subspaces and simplifies the problem to a linear subspace case, which we wish to avoid.

To address the above limitations, we propose a distance measure with the following form:

$$\text{dist}(\boldsymbol{A}_i, \boldsymbol{A}_j) = \text{dist}_G(\boldsymbol{U}_i, \boldsymbol{U}_j) + \alpha(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \boldsymbol{M}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \tag{2}$$

where dist_G is the Geodesic distance between two points on a Grassmann manifold [7], $(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \boldsymbol{M}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$ is the Mahalanobis distance between origins of $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$, and α is a mixing weight. The components in the proposed distance are described below.

A Grassmann manifold (a special type of Riemannian manifold) is defined as the space of all n-dimensional linear subspaces of $\mathbb{R}^D$ for $0 < n < D$. A point on Grassmann manifold $\mathcal{G}_{D,n}$ is represented by an orthonormal basis through a $D \times n$ matrix. The length of the shortest smooth curve connecting two points on a manifold is known as the geodesic distance. For Grassmann manifolds, the geodesic distance is given by:

$$\text{dist}_G(\boldsymbol{X}, \boldsymbol{Y}) = \|\Theta\|_2 \tag{3}$$

where $\Theta = [\theta_1, \theta_2, \cdots, \theta_n]$ is the principal angle vector, *i.e.*

$$\cos(\theta_l) = \max_{\boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{y} \in \boldsymbol{Y}} \boldsymbol{x}^T \boldsymbol{y} = \boldsymbol{x}_l^T \boldsymbol{y}_l \tag{4}$$

subject to $\|\boldsymbol{x}\| = \|\boldsymbol{x}\| = 1$, $\boldsymbol{x}^T \boldsymbol{x}_i = \boldsymbol{y}^T \boldsymbol{y}_i = 0$, $i = 1, \ldots, l-1$. The principal angles have the property of $\theta_i \in [0, \pi/2]$ and can be computed through the SVD of $\boldsymbol{X}^T \boldsymbol{Y}$ [7].

We note that the linear combination of a Grassmann distance (distance between linear subspaces) and Mahalanobis distance (between origins) of two affine subspaces has roots in probabilistic subspace distances [9]. More specifically, consider two normal distributions $\mathcal{N}_1(\boldsymbol{\mu}_1, \boldsymbol{C}_1)$ and $\mathcal{N}_2(\boldsymbol{\mu}_2, \boldsymbol{C}_2)$ with $\boldsymbol{C}_i = \sigma^2 \mathbb{I} + \boldsymbol{U}_i \boldsymbol{U}_i^T$ as the covariance matrix, and $\boldsymbol{\mu}_i$ as the mean vector. The symmetric Kullback-Leibler (KL) distance between $\mathcal{N}_1$ and $\mathcal{N}_2$ under orthonormality condition (*i.e.* $\boldsymbol{U}_i^T \boldsymbol{U}_i = \mathbb{I}_n$) results in:

$$\begin{aligned} J_{KL} \;=\; & \frac{1}{2\sigma^2} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(2\mathbb{I}_D - \boldsymbol{U}_1 \boldsymbol{U}_1^T - \boldsymbol{U}_2 \boldsymbol{U}_2^T \right) (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\ & + \frac{1}{2\sigma^2(\sigma^2 + 1)} \left(2n - 2\text{tr}(\boldsymbol{U}_1^T \boldsymbol{U}_2 \boldsymbol{U}_2^T \boldsymbol{U}_1) \right) \end{aligned} \tag{5}$$

The term $\text{tr}(\boldsymbol{U}_1^T \boldsymbol{U}_2 \boldsymbol{U}_2^T \boldsymbol{U}_1)$ in J_{KL} is identified as the projection distance on Grassmann manifold $\mathcal{G}_{D,n}$ (defined

as $\text{dist}_{Proj}(\boldsymbol{U}_1, \boldsymbol{U}_2) = \|sin(\Theta)\|_2)$ [9], and the term $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(2\mathbb{I}_D - \boldsymbol{U}_1\boldsymbol{U}_1^T - \boldsymbol{U}_2\boldsymbol{U}_2^T\right)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ is the Mahalanobis distance with $\boldsymbol{M} = 2\mathbb{I}_D - \boldsymbol{U}_1\boldsymbol{U}_1^T - \boldsymbol{U}_2\boldsymbol{U}_2^T$.

Since the geodesic distance is a more natural choice for measuring lengths on Grassmann manifolds (compared to the projection distance), we have elected to combine it with the Mahalanobis distance from (5), resulting in the following instantiation of the general form given in Eqn. (2):

$$\begin{aligned}\text{dist}(\boldsymbol{A}_i, \boldsymbol{A}_j) \;=\; & \text{dist}_G(\boldsymbol{U}_i, \boldsymbol{U}_j) \\ & + \alpha(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left(2\mathbb{I}_D - \boldsymbol{U}_i\boldsymbol{U}_i^T - \boldsymbol{U}_j\boldsymbol{U}_j^T\right)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)\end{aligned}$$

We measure the likelihood of a candidate subspace $A_i^{(t)}$, given template m_j, as follows:

$$p\left(A_i^{(t)}|m_j\right) = \exp\left(\frac{-\,\text{dist}(A_i^{(t)}, m_j)}{\sigma}\right) \tag{6}$$

where σ indicates the standard deviation of the likelihood function and is a parameter in the tracking framework. The likelihoods are normalised such that $\sum_{i=1}^{N} p(A_i^{(t)}|m_j) = 1$. To measure the likelihood between a candidate affine subspace $A_i^{(t)}$ and bag $\mathcal{M}$, the individual likelihoods between $A_i^{(t)}$ and bag templates m_j should be integrated. Based on [17], we opt for the sum rule:

$$p(A_i^{(t)}|\mathcal{M}) = \sum\nolimits_j^k p(A_i^{(t)}|m_j) \tag{7}$$

The object state is then estimated as:

$$\widehat{\mathcal{Z}}^{(t)} = \mathcal{Z}_j^{(t)}, \quad \text{where} \quad j = \underset{i}{\text{argmax}}\; p(A_i^{(t)}|\mathcal{M}) \tag{8}$$

2.5. Computational Complexity

The computational complexity of the proposed tracking framework can be associated with generating a new model and comparing a target candidate with a model. The model generation step requires $O(D^3 + 2Dn)$ operations. Computing the geodesic distance between two points on $G_{D,n}$ requires $O((D+1)n^2 + n^3)$ operations. Therefore, comparing an affine subspace candidate against each bag template needs $O((2n+3)D^2 + (n^2+1)D + n^3 + n^2)$ operations.

3. Experiments

In this section we evaluate and analyse the performance of the proposed AST method using eight publicly available videos[6] consisting of two main tracking tasks: face and object tracking. The sequences are: *Occluded Face* [1], *Occluded Face 2* [4], *Girl* [6], *Tiger 1* [4], *Tiger 2* [4], *Coke Can* [4], *Surfer* [4], and *Coupon Book* [4]. Example frames from several videos are shown in Fig. 6.

[6]The videos and the corresponding ground truth are available at *http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml*

Each video is composed of 8-bit grayscale images, resized to 320×240 pixels. We used the raw pixel values as image features. For the sake of computational efficiency in the affine subspace representation, we resized each candidate image region to 32×32, and the number of eigenvectors (n) used in all experiments is set to three. Furthermore, we only consider 2D translation and scaling in the motion modelling component. The batch size (W) for the template update is set to five as a trade-off between computational efficiency and effectiveness of modelling appearance change during fast motion.

We evaluated the proposed tracker based on **(i)** average center location error, and **(ii)** precision [4]. Precision shows the percentage of frames for which the estimated object location is within a threshold distance of the ground truth. Following [4], we use a fixed threshold of 20 pixels.

To contrast the effect of affine subspace modelling against linear subspaces, we assessed the performance of the AST tracker against a tracker that only exploits linear subspaces, *i.e.*, an AST where $\mu = 0$ for all models. The results, in terms of center location errors, are shown in Table 1. The proposed AST method significantly outperforms the linear subspaces approach, thereby confirming our idea of affine subspace modelling.

Algorithm 1 : Affine Subspace Tracking

Input:
- New frame, a set of updated candidate object states from the last frame, and the previous $P-1$ estimated object states $\{\widehat{\mathcal{Z}}^{(\tau)}\}_{\tau=t-P+1}^{t-1}$

1: **Initialisation:**
- $t = 1 : P$
- Set the initial object state $\widehat{\mathcal{Z}}^{(t)}$ in the first P frames.
- Use a single state to indicate the location.

2: **Begin:**
- Select candidate object states according to the dynamic model $\{\mathcal{Z}_i^{(t)}\}_{i=1}^{N}$
- For each sample, extract the corresponding image patch
- For each $\mathcal{Z}_i^{(t)}$ do:
  - Generate the affine subspace $A_i^{(t)}\{\mu_i^{(t)}, U_i^{(t)}\}$ based on image regions corresponding to $\mathcal{Z}_i^{(t)}$ and $\{\widehat{\mathcal{Z}}^{(\tau)}\}_{\tau=t-P+1}^{t-1}$
  - Calculate the likelihoods given each template in the bag by Eqn. (6)
  - Compute the final likelihoods using Eqn. (7)
- Determine the object state $\widehat{\mathcal{Z}}^{(t)}$ by Maximum Likelihood (ML) estimation
- Update the existing candidate object states according to their probabilities [14]

Output: current object state $\widehat{\mathcal{Z}}^{(t)}$

Figure 6. Examples of bounding boxes resulting from tracking on several video sequences. For the sake of clarity, we only demonstrate the results of the overall top four trackers. **(a)** *Surfer* [4]: includes large pose variations, occlusion; **(b)** *Coupon Book* [4]: contains severe appearance change in addition to including an imposter to distract the tracker; **(c)** *Occluded Face 2* [4]: contains various occlusions; **(d)** *Girl* [6] involves partial and full occlusion, large pose changes.

Table 1. Performance comparison between tracking based on affine and linear subspaces, in terms of average center location errors (pixels).

Video	proposed AST	linear subspace
Surfer	8	39
Coke Can	9	31
Girl	19	29
Tiger 1	22	38
Tiger 2	15	42
Coupon Book	8	25
Occluded Face	14	27
Occluded Face 2	13	24
average error	**13.5**	**31.88**

3.1. Quantitative Comparison

To assess and contrast the performance of AST tracker against state-of-the-art methods, we consider six methods, here. The competitors are: fragment-based tracker (FragTrack) [1], multiple instance boosting-based tracker (MILTrack) [4, 3], online Adaboost (OAB) [8], tracking-learning-detection (TLD) [15], incremental visual tracking (IVT) [19], and Sparsity-based Collaborative Model tracker (SCM) [26]. We use the publicly available source codes for FragTrack[1], MILTrack[2], OAB[2], TLD[3], IVT[4] and SCM[5].

Tables 2 and 3 show the performance in terms of precision and location error, respectively, for the proposed AST method as well as the competing trackers. Fig. 6 shows resulting bounding boxes for several frames from the *Surfer*, *Coupon Book*, *Occluded Face 2* and *Girl* sequences. On average, the proposed AST method obtains notably better performance than the competing trackers, with TLD being the second best tracker.

[1] *http://www.cs.technion.ac.il/˜amita/fragtrack/fragtrack.htm*
[2] *http://vision.ucsd.edu/ bbabenko/project_miltrack.shtml*
[3] *http://info.ee.surrey.ac.uk/Personal/Z.Kalal/*
[4] *http://www.cs.toronto.edu/˜dross/ivt/*
[5] *http://ice.dlut.edu.cn/lu/Project/cvpr12_scm/cvpr12_scm.htm*

Table 2. Comparison of the proposed AST method against competing trackers, in terms of average center location errors (pixels). Best performance is indicated by $*$, while second best by $**$.

Video	AST (proposed)	TLD [15]	MILTrack [4]	SCM [26]	OAB [8]	IVT [19]	FragTrack [1]
Surfer	8 $*$	9 $**$	11	76	23	30	139
Coke Can	9 $*$	13 $**$	20	9 $*$	25	61	63
Girl	19 $**$	28	32	10 $*$	48	52	27
Tiger 1	22	10 $*$	16 $**$	37	35	59	39
Tiger 2	15 $*$	15 $*$	18 $**$	43	33	43	37
Coupon Book	8 $*$	37	15 $**$	36	25	17	56
Occluded Face	14	16	27	4 $*$	43	9	6 $**$
Occluded Face 2	13 $**$	28	20	8 $*$	21	17	45
average error	**13.5** $*$	**19.49** $**$	**19.87**	**27.87**	**31.62**	**36.00**	**51.5**

Table 3. Precision at a fixed threshold of 20, as per [4]. Best performance is indicated by $*$, while second best is indicated by $**$. The higher the precision, the better.

Video	AST (proposed)	TLD [15]	MILTrack [4]	SCM [26]	OAB [8]	IVT [19]	FragTrack [1]
Surfer	0.98 $*$	0.97 $**$	0.93	0.10	0.51	0.19	0.28
Coke Can	0.99 $*$	0.98 $**$	0.55	0.97	0.45	0.13	0.14
Girl	0.73 $**$	0.42	0.32	0.97 $*$	0.11	0.50	0.51
Tiger 1	0.54	0.92 $*$	0.81 $**$	0.35	0.48	0.32	0.28
Tiger 2	0.83 $*$	0.81 $**$	0.83 $*$	0.14	0.51	0.29	0.22
Coupon Book	0.94 $*$	0.66	0.69 $**$	0.52	0.67	0.57	0.41
Occluded Face	0.79	0.64	0.43	1.00 $*$	0.22	0.94	0.95 $**$
Occluded Face 2	0.75 $**$	0.18	0.60	0.95 $*$	0.61	0.72	0.44
average precision	**0.82** $*$	**0.69** $**$	**0.64**	**0.63**	**0.44**	**0.45**	**0.40**

3.2. Qualitative Comparison

Heavy occlusions. Occlusion is one of the major issues in object tracking. Trackers such as SCM, FragTrack and IVT are designed to resolve this problem. Other trackers, including TLD, MIL and OAB, are less successful in handling occlusions, especially at frames 271, 529 and 741 of the *Occluded Face* sequence, and frames 176, 432 and 607 of *Occluded Face 2*. SCM can obtain good performance mainly as it is capable of handling partial occlusions via a patch-based model. The proposed AST approach can tolerate occlusions to some extent, thanks to the properties of the appearance model. One prime example is *Occluded Face 2*, where AST accurately localised the severely occluded object at frame 730.

Pose Variations. On the *Tiger 2* sequence, most trackers, including SCM, IVT and FragTrack, fail to track the object from the early frames onwards. On *Tiger 2*, the proposed AST approach can accurately follow the object at frames 207 and 271 when all the other trackers have failed. In addition, compared to the other trackers, the proposed approach partly handles motion blurring (*e.g.* frame 344), where the blurring is a side-effect of rapid pose variations. On *Tiger 1*, although TLD obtains the best performance, AST can successfully locate (in contrast to the other trackers) the object at frames 204 and 249, which are subject to occlusion and severe illumination changes.

Rotations. The *Girl* and *Surfer* sequences include drastic out-of-plane and in-plane rotations. On *Surfer*, FragTrack and SCM fail to track from the start. The proposed AST approach consistently tracked the surfer and outperforms the other trackers. On *Girl*, the IVT, OAB, and FragTrack methods fail to track in many frames. While IVT is able to track in the beginning, it fails after frame 230. The AST approach manages to track the correct person throughout the whole sequence, especially towards the end where the other trackers fail due to heavy occlusion.

Illumination changes. The *Coke Can* sequence consists of dramatic illumination changes. FragTrack fails from frame 20 where the first signs of illumination changes appear. IVT and OAB fail from frame 40 where the frames include both severe illumination changes and slight motion blur. MILTrack fails after frame 179 where a part of the object is almost faded by the light. Since affine subspaces accommodate robustness to the illumination changes, the proposed AST approach can accurately locate the object throughout the whole sequence.

Imposters/Distractors. The *Coupon Book* sequence contains a severe appearance change, as well as an imposter book to distract the tracker. FragTrack and TLD fail mainly where the imposter book appears. AST successfully tracks the correct book with notably better accuracy than the other methods.

4. Main Findings and Future Directions

In this paper we investigated the problem of object tracking in a video stream where object appearance can drastically change due to factors such as occlusions and/or variations in illumination and pose. The selection of subspaces for target representation purposes, in addition to a regular subspace update, are mainly driven by the need for an adaptive object template reflecting appearance changes. We argued that modelling the appearance by affine subspaces and applying this notion on both the object templates and the query data leads to more robustness. Furthermore, we maintain a record of k previously observed templates for a more robust tracker.

We also presented a novel subspace-to-subspace measurement approach by reformulating the problem over Grassmann manifolds, which provides the target representation with more robustness against intrinsic and extrinsic variations. Finally, the tracking problem was considered as an inference task in a Markov Chain Monte Carlo framework using particle filters to propagate sample distributions over time.

Comparative evaluation on challenging video sequences against several state-of-the-art trackers show that the proposed AST approach obtains superior accuracy, effectiveness and consistency, with respect to illumination changes, partial occlusions, and various appearance changes. Unlike the other methods, AST involves no training phase.

There are several challenges, such as drifts and motion blurring, that need to be addressed. A solution to drifts could be to formulate the update process in a semi-supervised fashion in addition to including a training stage for the detector. Future research directions also include an enhancement to the updating scheme by measuring the effectiveness of a new learned model before adding it to the bag of models. To resolve the motion blurring issues, we can enhance the framework by introducing blur-driven models and particle filter distributions. Furthermore, an interesting extension would be multi-object tracking and how to join multiple object models.

Acknowledgements

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 798–805, 2006.

[2] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50(2):174–188, 2002.

[3] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 983–990, 2009.

[4] B. Babenko, M. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[5] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):266–278, 2011.

[6] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 232–237, 1998.

[7] A. Edelman, T. Arias, and S. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[8] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *British Machine Vision Conference*, volume 1, pages 47–56, 2006.

[9] J. Hamm and D. Lee. Extended Grassmann kernels for subspace-based learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 601–608, 2009.

[10] M. Harandi, C. Sanderson, C. Shen, and B. C. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *Int. Conference on Computer Vision (ICCV)*, 2013.

[11] M. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2705–2712, 2011.

[12] M. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Kernel analysis on Grassmann manifolds for action recognition. *Pattern Recognition Letters*, 34(15):1906–1915, 2013.

[13] J. Ho, K. Lee, M. Yang, and D. Kriegman. Visual tracking using learned linear subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 782–789, 2004.

[14] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. *European Conference on Computer Vision (ECCV)*, pages 343–356, 1996.

[15] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[16] T. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1005–1018, 2007.

[17] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[18] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang. Incremental learning of 3D-DCT compact representations for robust visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):863–881, 2013.

[19] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *Int. Journal of Computer Vision (IJCV)*, 77(1):125–141, 2008.

[20] C. Sanderson, M. Harandi, Y. Wong, and B. C. Lovell. Combined learning of salient local descriptors and distance metrics for image set face verification. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 294–299, 2012.

[21] S. Shirazi, M. Harandi, C. Sanderson, A. Alavi, and B. C. Lovell. Clustering on Grassmann manifolds via kernel embedding with application to action analysis. In *Int. Conference on Image Processing (ICIP)*, pages 781–784, 2012.

[22] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[23] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[24] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *Int. Conference on Computer Vision (ICCV)*, pages 1323–1330, 2011.

[25] T. Wang, A. Backhouse, and I. Gu. Online subspace learning on Grassmann manifold for moving object tracking in video. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 969–972, 2008.

[26] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1838–1845, 2012.