Action Recognition in Videos Acquired by a Moving Camera Using Motion Decomposition of Lagrangian Particle Trajectories

Shandong Wu
Computer Vision Lab
University of Central Florida
sdwu@eecs.ucf.edu

Omar Oreifej
Computer Vision Lab
University of Central Florida
oreifej@eecs.ucf.edu

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@eecs.ucf.edu

Abstract

Recognition of human actions in a video acquired by a moving camera typically requires standard preprocessing steps such as motion compensation, moving object detection and object tracking. The errors from the motion compensation step propagate to the object detection stage, resulting in miss-detections, which further complicates the tracking stage, resulting in cluttered and incorrect tracks. Therefore, action recognition from a moving camera is considered very challenging. In this paper, we propose a novel approach which does not follow the standard steps, and accordingly avoids the aforementioned difficulties. Our approach is based on Lagrangian particle trajectories which are a set of dense trajectories obtained by advecting optical flow over time, thus capturing the ensemble motions of a scene. This is done in frames of unaligned video, and no object detection is required. In order to handle the moving camera, we propose a novel approach based on low rank optimization, where we decompose the trajectories into their camera-induced and object-induced components. Having obtained the relevant object motion trajectories, we compute a compact set of chaotic invariant features which captures the characteristics of the trajectories. Consequently, a SVM is employed to learn and recognize the human actions using the computed motion features. We performed intensive experiments on multiple benchmark datasets and two new aerial datasets called ARG and APHill, and obtained promising results.

1. Introduction

Action recognition from videos is a very active research topic in computer vision with many important applications for surveillance, human-computer interaction, video retrieval, robot learning, etc. Various action detection approaches are reported in the literature; however, they mostly tackle stationary camera scenarios. Recently, there has been an increasing interest in studying action recognition from moving cameras such as in aerial videos recorded by UAVs [34]. Action recognition from moving cameras poses significant challenges since the field of view is constantly changing with the camera motion. More importantly, the global camera motion and the local object motion are mixed up in the acquired frames (see Figure 1). Therefore, action recognition in such scenarios imposes a critical demand to eliminate the often dominant camera motion, and to recover the independent motions merely resulting from the performing actors.

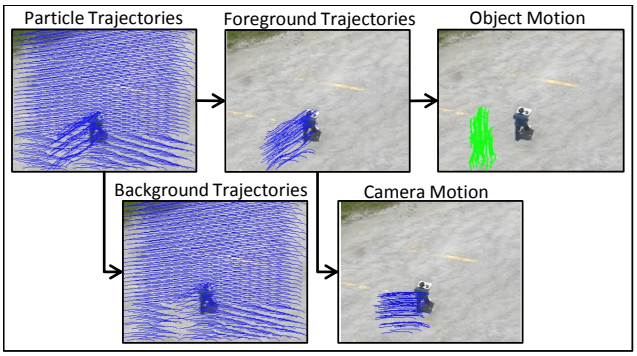

Figure 1. Motion decomposition for a moving camera sequence. The ensemble motions of a sequence is captured by particle trajectories, some of which solely correspond to the camera motion (background trajectories), and the others combine both the camera motion and the object motion (foreground trajectories). In this work, we show how to extract the object motion and employ it for action recognition. Note that the object motion component (green) appears displaced from the actor since the actor still carries the camera motion in the unaligned frame.

Traditional approaches dealing with moving cameras usually need to go through a motion compensation step by performing video alignment [33, 36]. Consequently, the moving objects are detected by background subtraction, followed by the tracking of the detected moving blobs in order to compute certain motion features from the tracks to be employed in action recognition. However, this approach

suffers from two inherent problems: First, video alignment is difficult and noisy due to the perspective distortions, and the errors in feature point detection and localization; second, the errors from alignment and moving object detection further propagate to the tracking stage.

In addition, a fundamental problem in action recognition is to extract good features to describe the actions. In this work, we focus on motion features (trajectories). Motion trajectories are informative, compact, and spatiotemporally continuous, which makes them useful for action recognition [1, 2, 3, 4, 5]. Automatic trajectory acquisition can be performed by tracking. Although it is relatively easier to track the whole body or a part of a moving object and obtain a single trajectory corresponding to its centroid, the single trajectory is not able to provide semantically rich motion information for depicting complex and articulated motions. Multiple-interest-point tracking using a tracker such as KLT as in [17] is possible yet very challenging due to three critical factors: First, good features for tracking (e.g. corners) need to be selected beforehand, which tends to be noisy in cluttered sequences. Second, the selected features may not be associated with the action of interest. Third, the obtained trajectories tend to be discontinuous due to the difficulty in maintaining consistent and correct point correspondence; therefore, the obtained tracks usually have variable lengths, which adds additional inconvenience for trajectory matching and alignment [35]. In addition, several methods employ explicit tracking markers attached to the objects to facilitate the tracking as in [1]; however, such invasive trajectory acquisition techniques are usually impractical.

In contrast to the traditional motion trajectory acquisition mechanisms, in this paper we propose to automatically extract a set of particle motion trajectories for action representation and recognition. Particle trajectories are a set of dense trajectories that capture the ensemble motions of a scene through the motion of an overlaid grid of particles. The basis of particle trajectory acquisition lies in advecting the particles using optical flow. The advection-based particle trajectory acquisition follows a bottom-up method where neither pre-definition of interest points nor point correspondence across frames is required; hence, it is inherently different from traditional object tracking, and does not suffer from any of the aforementioned problems faced by tracking-based approaches. To the best of our knowledge, particle trajectories have been mainly used for crowded flow analysis [18], but they have never been used for action recognition.

Furthermore, in contrast to the traditional approaches for dealing with moving cameras where video alignment is usually employed beforehand, we propose a novel algorithm based on robust sparse optimization that concurrently segments the trajectories corresponding to the moving object and eliminates their camera motion component; thus providing the relevant independent particle trajectories which correspond only to the object's motion.

Once we obtain the independent particle trajectories, we compute a compact set of motion features consisting of chaotic invariants and simple statistical features that describe the underlying motion properties. Consequently, a SVM is used for action learning and recognition. Figure 2 shows the overall workflow of the proposed framework.

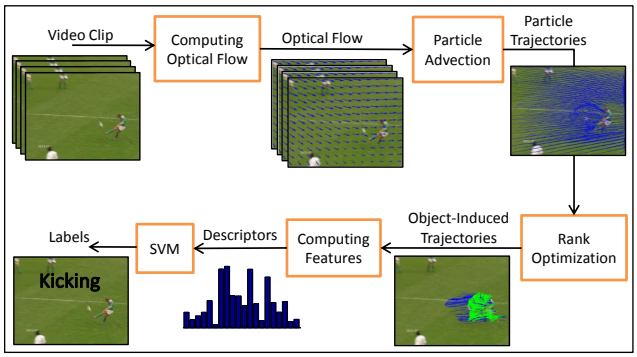

Figure 2. The various steps of our action recognition framework.

The main contributions of this paper are summarized as: First, our method is the first to utilize the dense particle trajectories of the objects for action recognition. Second, we propose a novel approach based on low rank optimization to robustly extract the trajectories which merely correspond to the object's motion from the whole set of particle trajectories obtained in a moving camera scenario, thus avoiding the standard approach which requires explicit video alignment and moving object detection.

2. Related Work

Motion trajectories have been employed in a variety of problems for human action representation and recognition [1, 2, 3, 4, 13, 16, 17, 23]. Many tracking-based methods are used or could be adapted for trajectory acquisition (for a comprehensive review on tracking techniques, readers may refer to relevant surveys [19]). Usually, a single trajectory can be acquired by simple techniques such as temporal filtering [3]. The tracking entity typically is either a human body part (head, hand, foot, etc.) or a person as a whole. For the simultaneous tracking of multiple points, KLT [20] is a popular choice [16, 17, 24]. Statistical mixture models are also developed for multi-trajectory tracking [2]. Some specific tracking strategies (e.g. SMP [1]) are designed to handle complicated and subtle full-body motions. A common drawback among tracking-based methods is that it is difficult to obtain reliable trajectories for the reasons discussed in the previous section. In addition, several studies assume that the motion trajectories are already available [5], or they rely on manual annotations [4], or the so-called semi-automatic manner [13]. In contrast, the particle ad-

vection in our work is fully automatic and is very easy to implement.

Particle trajectories have been previously used to model crowded scenes in [18], where the flows normally occupy the whole frame, and the camera is static; thus, such dense trajectories could be directly employed. We, in contrast, adopt the particle trajectories for recognizing actions in videos acquired from a moving camera, which imposes several challenges since the actions usually only cover a small part of the frame, and more importantly, the obtained trajectories combine both the camera motion and the object motion. Therefore, we propose a novel approach to detect the foreground trajectories and extract their object-induced component, which in principal requires estimating the background motion subspace. A large variety of subspace estimation methods exist in the literature such as PCA-based and RANSAC-based approaches. Such methods are, however, sensitive to noise which is considerably present in our scenario since a significant number of trajectories can be contaminated with the foreground motion. Fortunately, sparsity-based matrix decomposition methods such as [26, 27, 28] which have been primarily employed in image denoising domain, proved that a robust estimation of an underlying subspace can be obtained by decomposing the observations into a low rank matrix and a sparse error matrix. Therefore, in this work, we show how Robust PCA [27] can be adopted to extract the object motion relevant to the action of interest.

The acquired motion trajectories can be represented by certain descriptors to identify the underlying spatio-temporal characteristics. Wu *et al.* [5] proposed a systematic signature descriptor that can provide advantages in generalization, invariants, and compactness, etc. Ali *et al.* [13] showed that the features based on chaotic invariants for time series analysis perform very well in modelling manually-annotated trajectories. Meanwhile, "trajecton" was proposed in a Bag-of-Words context [17] for trajectory-based action recognition. Messing *et al.* [24] investigated the temporal velocity histories of trajectories as a more representative feature for recognizing actions. In this work, we employ the particle trajectories and choose the chaotic invariants [13] as a trajectory descriptor. It should be noted that we adopted the algorithms in [18] for computing the chaotic features as they have been shown more robust than [13].

Aside from trajectory features, a variety of feature representations have been developed for action recognition such as appearance features [9], shape-based representation [10]), volumetric features (e.g. Poisson equation-based features [6], 3D Haar feature [7]), spatiotemporal interest points ([8, 11, 14]), motion history image (MHI) [15], and kinematic features [12].

3. Action Recognition Framework

We first employ particle advection to obtain particle trajectories, and then extract the independent trajectories that represent the object-induced motion. The extracted trajectories are then described by a set of chaotic invariants and simple statistical features, which are finally fed to a SVM.

3.1. Lagrangian particle advection

We use the concept of a "particle" to explain our Lagrangian particle trajectory acquisition approach. We assume that a grid of particles is overlaid on a scene where each particle corresponds to a single pixel (the granularity is controllable). The basic idea is to quantify the scene's motions in terms of the motions of the particles which are driven by dense optical flow. A so-called particle advection [18] procedure is applied to produce the particle trajectories. Given a video clip represented by a matrix of $T \times W \times H$, where T is the number of frames, and $W \times H$ denotes the frame resolution (width by height), we denote the corresponding optical flow by (U_w^t, V_h^t), where $w \in [1, W]$, $h \in [1, H]$, and $t \in [1, T-1]$. The position vector (X_w^t, Y_h^t) of the particle at grid point (w, h) at time t is estimated by solving the following equations:

$$\frac{dX_w^t}{dt} = U_w^t, \tag{1}$$

$$\frac{dY_h^t}{dt} = V_h^t. \tag{2}$$

We use Euler's method to solve them similar to [18]. By performing advection for the particles at all grid points with respect to each frame of the clip, we obtain the clip's particle trajectory set, denoted by $\{(X_w^t, Y_h^t) | w \in [1, W], h \in [1, H], t \in [1, T]\}$.

Figure 3 illustrates the obtained particle trajectories for three examples from each of our experimental datasets. The obtained particle trajectories almost occupy the full frame and therefore capture all the motions occurring in the scene. It is obviously unwise to use all of the particle trajectories for action recognition since the motion induced by the camera is irrelevant to the action of interest, and hence may significantly confuse the action recognition task. Therefore, in the coming subsection, we propose a robust method to extract the foreground trajectories and concurrently eliminate their camera motion component.

3.2. Independent Object Trajectory Extraction

The obtained particle trajectories are induced from two motion components: rigid camera motion, and object motion. When the action of interest includes global body motion (e.g. body translation in running action), the object motion can be further decomposed into two components: rigid body motion, and articulated motion. We employ the

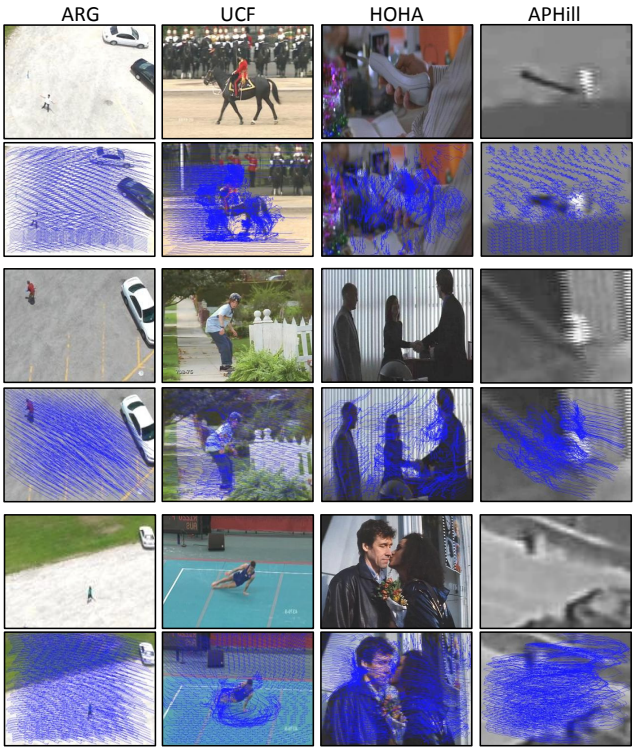

Figure 3. Three examples from each of our experimental datasets illustrating the obtained particle advection trajectories. Rows one, three, and five show the original frames, and rows two, four, and six show the corresponding overlaid trajectories respectively.

latest advances in sparse optimization to estimate each of these components, and extract the object trajectories which solely correspond to the action of interest. Without loss of generality, we assume that the majority of the observed motion is induced by the camera motion (this is reasonable for most realistic datasets). Therefore, the trajectories should generally span a subspace determined by the scene structure and the camera's intrinsic and extrinsic parameters. In order to find the basis for the particle trajectory subspace, we first construct a $2T \times P$ (P is the number of particles, i.e. $P = W \times H$) measurement matrix M using the position vectors of the advected particle trajectories in a clip

$$M = \begin{bmatrix} X_1^1 & \cdots & X_P^1 \\ Y_1^1 & \cdots & Y_P^1 \\ \vdots & \vdots & \vdots \\ X_1^T & \cdots & X_P^T \\ Y_1^T & \cdots & Y_P^T \end{bmatrix}. \quad (3)$$

Through rank minimization, we can decompose M into two components: a low-rank matrix A, and the sparse error matrix E

$$\arg\min_{A,E} rank(A) \text{ s.t. } M = A + E, ||E||_0 \leq \beta, \quad (4)$$

where β is a constant that represents the maximum number of corrupted measurements expected across the sequence. Introducing the Lagrange multiplier λ, we get

$$\arg\min_{A,E} rank(A) + \lambda||E||_0 \text{ s.t. } M = A + E, \quad (5)$$

where λ trades off the rank of the solution versus the sparsity of the error, and we always set it to $1.1/\sqrt{(W \times H)}$ following the theoretical considerations in [27], and the results from our experiments. Consequently, we apply convex relaxation to the problem by replacing $rank(A)$ with the nuclear norm or sum of the singular values $||A||_* = \Sigma_i(\sigma_i)$, and replacing $||E||_0$ with its convex surrogate ℓ_1 norm $||E||_1$

$$\arg\min_{A,E} ||A||_* + \lambda||E||_1 \text{ s.t. } M = A + E. \quad (6)$$

Equation 6 is convex and can be solved with convex optimization methods such as the Augmented Lagrange Multiplier (ALM) algorithm [29] which we found robust and fast in our scenarios. The columns of the resulting low-rank matrix A define the basis of the low rank components in the trajectories. Since the camera motion is dominant, the subspace spanned by the major basis of A correspond to the desired background subspace which includes both the background trajectories and the camera motion component of the foreground trajectories. On the other hand, any rigid body motions in the scene will also contribute to A; therefore, the subspace spanned by the rest of the basis of A mostly correspond to rigid body motions. Since the camera motion subspace is approximately spanned by three basis [21, 25], the camera motion component can be estimated by $A_c = US^*V'$, where U and V are obtained by singular value decomposition $[U, S, V] = SVD(A)$, and S^* is equal to S except that all the singular values other than the most significant three are set to zero. Therefore, the rigid body motion component is expressed by $A - A_c$.

Moreover, the columns of the matrix E correspond to the deviation of each trajectory from the recovered low rank subspace, which captures the articulated motions. Therefore, the total object trajectories E_t which include the articulated and the rigid body motion is given by

$$E_t = E + A - A_c. \quad (7)$$

If the action of interest involves only articulated motions without a rigid motion component (e.g. boxing, waving, etc.), the object motion will be mostly captured in E while the rigid body component $A - A_c$ will be negligible. On the other hand, if the action of interest involves rigid body motion (e.g. running, walking, etc.), each of E and $A - A_c$ will contribute to the total object motion. Figure 4 illustrates the motion decomposition for two actions, "boxing" and "carrying".

Since additional noise is usually present, some object trajectories can correspond to noise. However, the motion in such trajectories is minor compared to the actual object's

motion; therefore, they are easily eliminated by a simple threshold. In out experiments, we compute the sum of squared value for the columns of E, and accordingly select only the trajectories which attain at least 10% of the maximum value.

It is worth mentioning that we discard the boundary trajectories before constructing the measurement matrix M. The boundary trajectories are the trajectories that exhibit particles hung-up in the scene boundaries during the advection. Therefore, the points from such trajectories will remain stationary during the hung-up, and the resulting trajectories will not follow the complete camera motion. Hence, including such trajectories in M could deteriorate the performance of the rank minimization. Normally only a very small set of trajectories are excluded. Figure 5 depicts example object motion detection results for four sequences taken from each of our experimental datasets. It is clear from the figures that our method is able to robustly extract the object trajectories relevant to the action of interest.

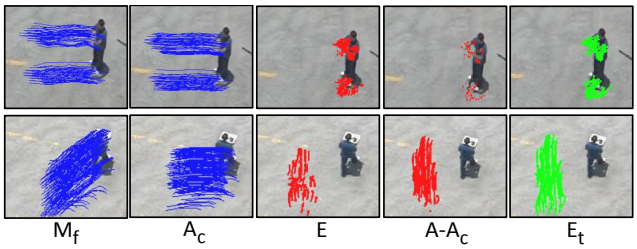

Figure 4. Two examples illustrating our proposed motion decomposition. From left to right: the detected foreground trajectories M_f, camera motion component A_c, articulated object motion component E, rigid object motion component $A - A_c$, total object motion E_t. The first row shows boxing action which only contains articulated motion component; thus, $A - A_c$ is negligible, and E and E_t are similar. The second row shows carrying action which contains both articulated and rigid body components; thus, E does not fully represent the motion, but E_t rather does. Note that the original foreground trajectories M_f is equal to $A_c + E_t$.

3.3. Action Description and Recognition

We use the extracted object trajectories to describe and recognize actions. Since a particle is typically placed on each pixel, we obtain a large number of particle trajectories. In order to get a more compact representation, we cluster the obtained trajectories into 100 clusters using k-means, and accordingly select the cluster's centroid as the representative trajectory for each cluster. Consequently, we characterize an action by computing a compact set of descriptors of the trajectories for training and recognition. In that, we use the chaotic invariants features [18, 13] augmented with a simple statistical feature for each of the x and y time series of a trajectory

$$F = \{\sigma, L, C\}, \tag{8}$$

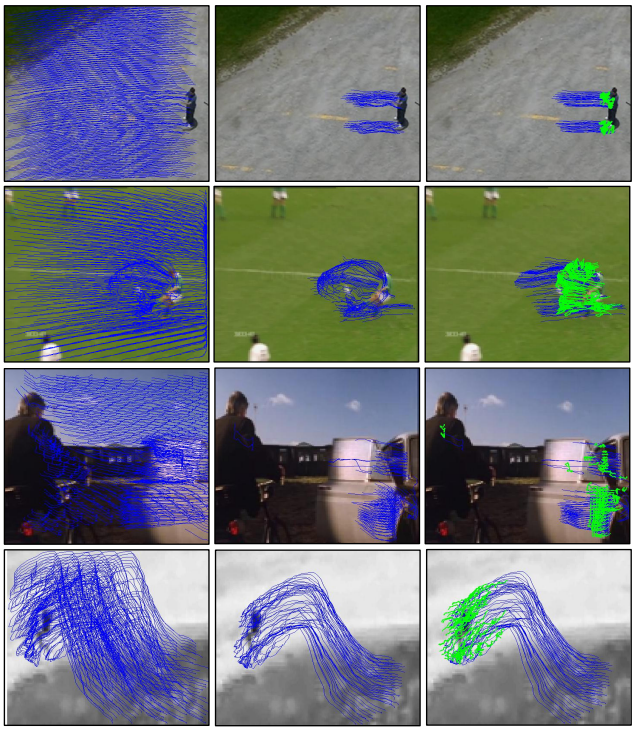

Figure 5. Each row shows an example illustrating our proposed object motion detection method. The examples are taken from four datasets, from top to down: ARG, UCF sports, HOHA, and APHill. The first column shows all the particle trajectories (excluding boundary trajectories). The second column shows the detected object trajectories. The third column shows the camera motion component (A_c) in blue, and the object motion component (E_t) in green. Please refer to our website for videos of the results.

where σ denotes the variance, L denotes the Largest Lyapunov Exponent (LLE), and C denotes the correlation dimension. σ has been proved to be a useful feature for time series description [13]. Meanwhile, L and C are typical chaotic invariants which are able to identify the underlying dynamics properties of a system (i.e., an action here). In the embedded phase space, L provides quantitative information about the orbits that start close together but diverge over time, and C measures the size of an attractor. We follow the algorithms described in [18] to calculate L and C. In order to estimate L for a time series X, we first locate all of the nearest neighbors ($\tilde{X}$) within the orbit in the embedding space. The nearest neighbors are assumed to diverge approximately at a rate L. We therefore have

$$\ln d_j(t_i) \approx \ln k_j + L t_i. \tag{9}$$

where j is the index of the pair of nearest neighbors, $t_i = i\Delta t$, Δt is the sampling period, k_j is the initial separation, and $d_j(t_i)$ denotes the distance between the jth pair of the nearest neighbors after i discrete time steps. Equation (9) represents a set of approximately parallel lines and thus L can be approximated by the slope of a fitted average line

denoted by $y(t_i) = \frac{\langle \ln d_j(t_i) \rangle}{\Delta t}$.

To estimate the correlation dimension, we first calculate the correlation sum

$$S(\delta) = \frac{2}{Q(Q-1)} \sum_{i \neq j} H(\delta - \|\tilde{X}_i - \tilde{X}_j\|). \qquad (10)$$

where H denotes the Heaviside step function, δ is a threshold distance, Q is the number of points in the time series. Consequently, we can simply derive C by $S(\delta) \approx \delta^C$.

Finally, we use a radial basis SVM to learn action models from the feature set of training, and to recognize testing examples. It is worth mentioning that we experimented on several types of trajectory features, and found the selected set of features preferable.

4. Experiment Results

We extensively experimented on the proposed action recognition method using six datasets including four moving camera datasets (APHill, ARG, HOHA, and UCF sports), and two static camera datasets (KTH and Weizmann). APHill and ARG are two new aerial datasets which are available for download on our website. For all of the datasets, we use the algorithm described in [22] for computing optical flow. To reduce the computational cost, we associate each particle with a 2×2 grid window.

4.1. APHill action recognition

APHill is a newly formed dataset of aerial videos. It includes 6 actions with 200 instances for each, except for "gesturing" action which has 42. This dataset is very challenging due to the low resolution (as low as 50×50) and the large intra-class variations (refer to Figure 3 for action examples). Using 20-fold cross validation, we obtained 41.8% recognition rate. In order to evaluate the contribution of our independent object motion estimation technique, we repeated the experiment using all of the initially obtained trajectories instead of using only the object-induced trajectories. In such case, we observed a significant decrease in performance (only 31.1% achieved), which provides a clear evidence of the contribution of our object motion detection method in action recognition. Figure 6 shows the obtained confusion matrix. As can be seen, walking is mostly confused with running. Additionally, no actions were classified as gesturing which is mostly because the number of samples for this action is significantly less than the others. Moreover, standing is quite hard to distinguish as there are very minor motion features associated with such action. In general, given the difficulty of the dataset, the performance is quite promising.

4.2. ARG-aerial action recognition

ARG-aerial is a new multi-view dataset recorded from four viewpoints by a moving camera equipped in a freely

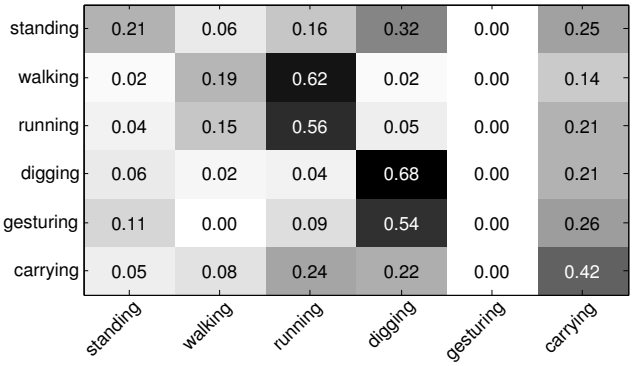

Figure 6. Confusion matrix for APHill dataset.

floating balloon. It includes 9 actions, each is performed by 12 actors. We use a subset sequences selected from one of the viewpoints. Each sequence ranges from $\sim 30 - 50$ seconds, with several repetitions of the action pattern; thus, we divide it into multiple shorter clips (50 frames for "digging" and "throwing", and 30 for the rest). We obtain a total of 112 clips for our experiments.

A major challenge in ARG dataset arises from the large, dramatic, and fast camera motions in most of the videos due to the free floating nature of the balloon. In addition, the actors are extremely small occupying approximately only $\sim 2 - 5\%$ of the full frame (frame size is 1920×1080). Such conditions are particularly challenging for articulated human action recognition.

We preprocess the clips by resizing them to $\sim 25\%$ of the original size, and cropping out a sub-window(ranging from $\sim 80 \times 80 - 300 \times 300$ pixels^2). Using 5-fold cross validation, we obtained an average recognition rate of 51.8%, and 30.8% when the independent motion estimation step is skipped, which provides additional support for the effectiveness of our method. Figure 7 shows the obtained confusion matrix. As can be seen from the matrix, both walking and carrying actions are mostly confused with running. In fact, it is indeed very difficult to distinguish such actions relying on only motion features. In view of the discussed challenges, such performance is promising.

4.3. HOHA action recognition

HOHA (Hollywood Human Actions) dataset [30] includes 10 types of actions extracted from movies. Almost all of the sequences can be considered within the moving camera domain. HOHA is very challenging due to the complicated background of the realistic scenes, the large intra-class variation, and the existing changes of shots in a significant number of videos. The change of shots particularly is a considerable challenge for obtaining continuous particle trajectories; in fact, it raises the same challenge for any tracking-based method such as [17]. However, we found in our experiments that the shot change often slightly cor-

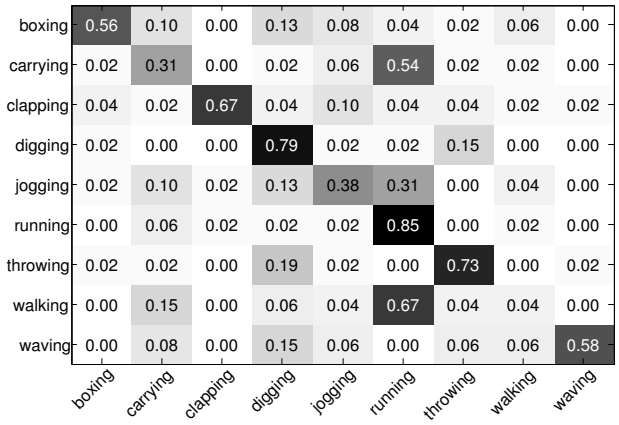

Figure 7. Confusion matrix for ARG-aerial dataset.

Table 1. Average Precision comparison for HOHA dataset.

Action	Ours	Ours(All Trajs)	STIP [30]	Trajecton [17]
Average	**47.6%**	**46.3%**	**38.4%**	**21.1%**
AnswerPhone	48.3%	46.9%	32.1%	4.5%
GetOutCar	43.2%	34.1%	41.5%	69.0%
HandShake	46.2%	49.7%	32.3%	71.4%
HugPerson	49.3%	49.6%	40.6%	0.0%
Kiss	63.6%	49.9%	53.3%	0.0%
SitDown	47.5%	50.0%	38.6%	5.3%
SitUp	35.1%	40.0%	18.2%	11.1%
StandUp	47.3%	50.0%	50.5%	7.7%

rupts the trajectories such that no major spurious effects are introduced.

We use the "clean" training set for training and the separate testing set for testing. We compare the performance of our method with Trajecton [17] and space-time interest points (STIP) [30] using the same experimental setup and performance measure (Average Precision). Table 1 shows the comparison results, from which it can be clearly observed that our method achieved a better performance than STIP. Additionally, our method significantly outperforms Trajecton which employs KLT to acquire trajectories; this particularly shows the advantage of our dense particle advection trajectories. Moreover, we repeated the same experiment except that we used all of the initially obtained particle trajectories (i.e., independent motion estimation step was skipped). The obtained performance, as can be seen from column 3 of the table, is still comparable to the case where only the object trajectories are employed. Such result is expected in this dataset since the camera motion is minor in many sequences, and more importantly, the actors occupy the majority of the frame such that most of particle trajectories are associated with the action of interest.

Table 2. Recognition rate comparison for UCF sports dataset.

Method	Recognition Rate (%)
Ours	**89.7**
Kovashka et. al. [32]	87.3
Wang et. al. [31]	85.6
Ours (All Trajs)	85.8

4.4. UCF sports action recognition

UCF sports is a challenging dataset with sequences mostly acquired by moving cameras. It includes 10 sports actions with a total of 150 sequences. We followed the same processing as in [31, 32] by adding a flipped version for all the videos in order to enlarge the dataset. Using 5-fold cross validation strategy we obtained the performance results summarized in Table 2 which demonstrates that our method outperforms the state-of-the-art.

4.5. Action recognition from static cameras

Though our proposed method is primarily designed for moving camera scenarios, we additionally experimented on KTH and Weizmann datasets which can generally be considered within the static camera domain though a small part of the videos are associated with a zoom-in and zoom-out operations in KTH. Each sequence is divided into multiple shorter clips ranging from $20 - 50$ frames per clip. We obtained an average recognition rate of 95.7% for KTH which is closely comparable to the state-of-the-art [16] with 96.7%. For Weizmann dataset, we obtained 92.8% which we particularly compare with the closely related work of [13] where a slightly inferior rate of 92.6% is achieved with manually obtained trajectories.

5. Conclusion

We proposed a novel method for recognizing human actions in videos acquired by moving cameras. To the best of our knowledge, this is the first work which employs Lagrangian particle trajectories for action recognition. Our method is able to extract a large number of particle trajectories corresponding to the motions; therefore, it better captures the articulation of human actions which improves the recognition performance. Particle trajectories are easily obtained by advecting pixel-wise optical flow; thus, representing the ensemble motions of a scene, from which we extract the independent object motion through a novel method using rank optimization. This enables our method to avoid traditional trajectory acquisition techniques which require video alignment, object detection, and tracking. Through experiments, we have demonstrated the robustness of the proposed approach while outperforming the state-of-the-art

on several benchmark datasets.

Acknowledgments: This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-08-C-0135. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

References

[1] J. Min and R. Kasturi, Activity recognition based on multiple motion trajectories, IEEE ICPR, 2004, pp. 199-202. 2

[2] D. Meyer, J. Psl, and H. Niemann, Gait classification with HMMs for trajectories of body parts extracted by Mixture densities, BMVC, 1998, pp. 459-468. 2

[3] A. Psarrou, S. Gong, and M. Walter, Recognition of human gestures and behaviour based on motion trajectories, Image and Vision Computing 20(5-6) (2002) 349-358. 2

[4] C. Rao, A. Yilmaz, and M. Shah, View-invariant representation and recognition of actions, IJCV 50(2) (2002) 203-226. 2

[5] S. Wu and Y.F. Li, Flexible signature descriptions for adaptive motion trajectory representation, perception and recognition, Pattern Recognition, 42 (1): 194-214, 2009. 2, 3

[6] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, Actions as space-time shapes, IEEE ICCV, 2005. 3

[7] Y. Ke, R. Sukthankar, and M. Hebert, Efficient visual event detection using volumetric features, IEEE ICCV, 2005. 3

[8] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, Behavior recognition via sparse spatio-temporal features, IEEE VS-PETS, 2005. 3

[9] T. Darrell and A. Pentland, Classifying hand gestures with a view-based distributed representation, NIPS, 1993. 3

[10] K. M. Cheung, S. Baker, and T. Kanade, Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture, IEEE CVPR, 2003. 3

[11] C. Schuldt, I. Laptev, and B. Caputo, Recognizing human actions: a local SVM approach, IEEE ICPR, 2004. 3

[12] S. Ali and M. Shah, Human action recognition in videos using kinematic features and multiple instance learning, IEEE PAMI, 2010 3

[13] S. Ali, A. Basharat, and M. Shah, Chaotic invariants for human action recognition, IEEE ICCV, 2007. 2, 3, 5, 7

[14] V. Parameswaran and R. Chellappa, View invariance for human action recognition, IJCV, 66(1), 2006. 3

[15] A. F. Bobick and J. Davis, The recognition of human movement using temporal templates, IEEE PAMI 23(3), 2001. 3

[16] M. B. Kaaniche and F. Bremond, Gesture recognition by learning local motion signatures, IEEE CVPR 2010. 2, 7

[17] P. Matikainen, M. Hebert, and R. Sukthankar, Trajectons: action recognition through the motion analysis of tracked features, ICCV Workshop, 2009. 2, 3, 6, 7

[18] S. Wu, B. Moore, and M. Shah, Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes, IEEE CVPR 2010. 2, 3, 5

[19] A. Yilmaz, O. Javed, and M. Shah, Object tracking: a survey, ACM Journal of Computing Surveys 38 (4), 2006. 2

[20] C. Tomasi and J. Shi. Good features to track. IEEE CVPR 1994. 2

[21] Y. Sheikh, O. Javed, and T. Kanade, Background subtraction for freely moving cameras, IEEE ICCV 2009. 4

[22] C. Liu, Beyond pixels: exploring new representations and applications for motion analysis, Doctoral Thesis, MIT, 2009. 6

[23] N. Johnson and D. Hogg, Learning the distribution of object trajectories for event recognition, BMVC, 1995. 2

[24] R. Messing, C. Pal, and H.Kautz, Activity recognition using the velocity histories of tracked keypoints, IEEE ICCV 2009. 2, 3

[25] E. Elhamifar and R. Vidal, Sparse subspace clustering CVPR, 2009. 4

[26] G. Liu, Z. Lin and Y. Yu, Robust subspace segmentation by low-rank representation. International Conference on Machine Learning, 2010. 3

[27] E. J. Candes, X. Li , Y. Ma and J. Wright. Robust principal component analysis? In Preprint, 2009. 3, 4

[28] H. Ji, C. Liu, Z. Shen and Y. Xu. Robust video denoising using low rank matrix completion. CVPR, 2010. 3

[29] Z. Lin, M. Chen, L. Wu and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank completion. UIUC Technical Report, 2009. 4

[30] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. IEEE CVPR 2008. 6, 7

[31] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. BMVC 2009. 7

[32] A. Kovashka and K. Grauman. Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition. CVPR 2010. 7

[33] J. Xiao, H. Cheng, H. S. Sawhney, and F. Han. Vehicle detection and tracking in wide field-of-view aerial video. CVPR 2010. 1

[34] A.Hoogs, M. Chan, R. Bhotika, J. Schmiederer. Recognizing Complex Behaviors in Aerial Video. ICIA 2005. 1

[35] M.T. Chan, A. Hoogs, R. Bhotika, A.G.A. Perera, J. Schmiederer, G. Doretto, Joint Recognition of Complex Events and Track Matching. IEEE CVPR 2006. 2

[36] J. Xiao, H. Cheng, F. Han, H. S. Sawhney. Geo-spatial aerial video processing for scene understanding and object tracking. CVPR 2008. 1