SHOULD WE DISCARD SPARSE OR INCOMPLETE VIDEOS?

Chuan Sun, Hassan Foroosh

Department of EECS, Division of Computer Science
University of Central Florida, USA

ABSTRACT

In this paper, we determine whether incomplete videos that are often discarded carry useful information for action recognition, and if so, how one can represent such mixed collection of video data (complete versus incomplete, and labeled versus unlabeled) in a unified manner. We propose a novel framework to handle incomplete videos in action classification, and make three main contributions: (1) We cast the action classification problem for a mixture of complete and incomplete data as a semi-supervised learning problem of labeled and unlabeled data. (2) We introduce a two-step approach to convert the input mixed data into a uniform compact representation. (3) Exhaustively scrutinizing 280 configurations, we experimentally show on our two created benchmarks that, even the videos are extremely sparse and incomplete, it is still possible to recover useful information from them, and classify unknown actions by a graph based semi-supervised learning framework.

Index Terms— Action classification, sparse video, tensor decomposition, semi-supervised learning

1. INTRODUCTION

For action recognition, many approaches have been proposed and high recognition rates have been reported for various datasets in the literature. However, we noticed that many existing methods lack some level of realism, in the sense that they include only video data that are complete, i.e. have no missing parts, occlusions, or noise, and are densely sampled. In a world inundated with videos (e.g. Youtube), vast majority of data may be one way or another subject to some level of incompleteness (as defined shortly). The purpose of this paper is thus to determine to what extent incomplete data are useful and carry helpful information for action recognition.

For clarity, we use the word "complete" throughout this paper to refer to raw, uncontaminated, occlusion-free, or densely sampled video, and "incomplete" to refer to video that is contaminated, corrupted, occluded, sparsely sampled, or simply missing data (pixels or blocks). We observe that most existing recognition methods require action data to be complete and they would fail to achieve the same recognition performance if the video is incomplete. For instance, methods based on the spatiotemporal descriptors, such as the space time interest point (STIP) descriptor [1][2], the HOG3D descriptor [3], and the bag-of-words based approach [4], etc., would fail to detect sufficient features under high degree of missing data (e.g. when using a compressive sensing device). Some approaches have also been proposed to handle occlusions in *tracking* or *detection* [5][6][7], but they may still fail under severe occlusions.

Researchers prefer complete and clean video data rather than incomplete or corrupted data due to mere convenience. As a result, it is conceivable to assume that potentially useful information is lost when we discard incomplete data in a dataset. This leads to two important questions our study aims to answer: (1) Is it always wise to simply discard incomplete video just because they are incomplete? (2) Is it possible to make full use of incomplete data in action classification?

To study these two questions, we propose a framework including four steps. The first is the mixing of incomplete and complete videos. To generate a large amount of incomplete videos, we sparsely sample the complete videos under various sparsity settings. Secondly, we regard an incomplete video as a three-way incomplete tensor, and recover incomplete videos by a solid tensor completion algorithm. Thirdly, we build lower dimensional representation using a rank one tensor decomposition algorithm. Finally, we apply graph based semi-supervised learning for action classification. Our experiments show that the proposed framework is very effective, and to our best knowledge, is the first attempt to address this challenging problem in the literature.

2. SPARSE VIDEO RECOVERY

2.1. Incompleteness

We first summarize 3 types of incomplete video. (1) The *first* type is closely related to the sparseness and randomness in compressive sensing [8, 9, 10, 11, 12, 13, 14]. In sparsely sampled video, a number of pixels in random locations would be missing. As compressive sensing is becoming popular, it is likely that more and more sparsely sampled videos will be generated. (2) The *second* type is caused by partial occlusion lasting throughout all frames. The work in [7] explored this type by artificially placing occluders on some predefined parts in the human body. This method however needs to

This work was supported in part by the National Science Foundation under grants IIS-1212948 and IIS-091686.

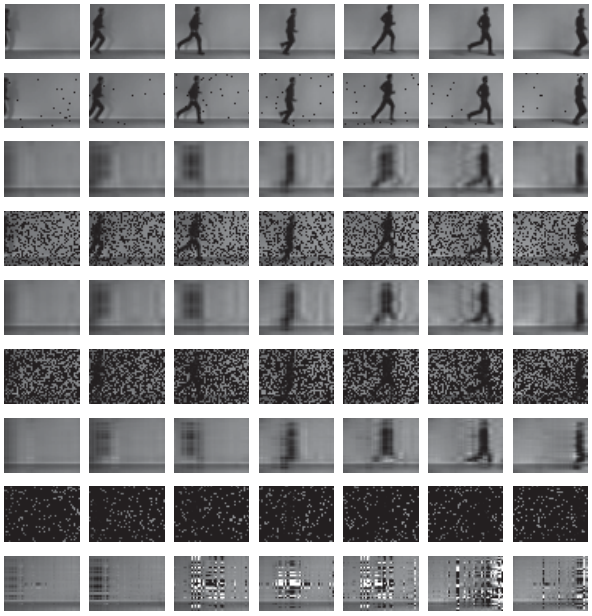

Fig. 1. Occluded frames under different occlusion settings for the KTH *running* action. The $1st$ row shows the original complete frames; The $2nd, 4th, 6th, 8th$ rows shows $1\%, 31\%, 61\%$, and 91% pixels are occluded; The $3rd, 5th, 7th, 9th$ row shows corresponding recovered frames. We fix the rank value $R = 16$.

calculate the overlapping amount with the occluding object, and complex local partitioning and hierarchical classification is required to ensure robustness. (3) The *third* type is the partial occlusion with short duration. This behaves like spatiotemporal "holes" and is directly related to the video completion problem, which is the process of filling in missing pixels or replacing undesirable pixels [15].

The first type is closely related to compressive sensing. The last two are special cases of the first. Since the second and the third type are well studied in inpainting and video completion, we concentrate on the first throughout this paper.

2.2. Recovery procedure

Regarding the sparse video $\mathcal{V}$ under sparse sampling mask $\mathcal{W}$ as an incomplete tensor $\mathcal{X}$, the sparse video recovery is equivalent to sparse tensor completion. Let $\mathcal{V} \in \mathbb{R}^{I\times J\times T}$ be an incomplete video in gray scale, and T is the frame number. Let $\mathcal{W} \in \mathbb{R}^{I\times J\times T}$ be a sparse weight tensor represent $\mathcal{V}$'s missing entries. $\mathcal{W}$ acts as a binary mask filtering out certain portion of $\mathcal{V}$. To specify video sparsity, we follow the approach in [16]. Let $\mathcal{X}$ be a three-way tensor of size $I \times J \times K$ with rank R. For the missing entries in the sparse tensor, tensor recovery can be defined as minimizing the error function:

$$f_w(A, B, C) = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\{w_{ijk}(x_{ijk} - \sum_{r=1}^{R} a_{ir}b_{jr}c_{kr})\}^2,$$

where $x_{ijk} = \sum_{r=1}^{R} a_{ir}b_{jr}c_{kr}$ for all $i \in [1, I], j \in [1, J]$, and $k \in [1, K]$. The factor matrices A, B, and C are of size $I \times R$, $J \times R$, and $K \times R$, respectively. It leads to a weighted least square Canonical Polyadic (CP) decomposition problem [16], which is a generalization of the SVD to tensors [16] in multilinear algebra. $\mathcal{W}$ is a nonnegative weight sparse tensor defined as $w_{ijk} = 1$ if x_{ijk} is known, and $w_{ijk} = 0$ if x_{ijk} is missing, and is of the same dimension as $\mathcal{X}$. The objective function can further be generalized as:

$f_{\mathcal{W}}(A^{(1)}, A^{(2)}, A^{(3)}) = \|\mathcal{W} * (\mathcal{X} - \langle A^{(1)}, A^{(2)}, A^{(3)}\rangle)\|^2$, where the $\langle A^{(1)}, A^{(2)}, A^{(3)}\rangle$ defines a $I_1 \times I_2 \times I_3$ tensor whose elements are given by: $(\langle A^{(1)}, A^{(2)}, A^{(3)}\rangle)_{i_1 i_2 i_3} = \sum_{r=1}^{R} a^{(1)}_{i_1 r}a^{(2)}_{i_2 r}a^{(3)}_{i_3 r}$. Let $\mathcal{Y} = \mathcal{W} * \mathcal{X}$ and $\mathcal{Z} = \mathcal{W} * \langle A^{(1)}, A^{(2)}, A^{(3)}\rangle$, the gradient of the objective function can be computed by the partial derivatives of $f_{\mathcal{W}}$ with respect to each element of the factor matrices, namely $\frac{\partial f_{\mathcal{W}}}{\partial A^{(n)}} = 2(\mathcal{Z}_{(n)} - \mathcal{Y}_{(n)})A^{(-n)}$, where $A^{(-n)} = A^{(3)} \odot A^{(2)} \odot A^{(1)}$ for $n = 1, 2, 3$ and $\odot$ is the Khatri-Rao product. The optimal factor matrices can then be computed iteratively using the gradient descent method.

To recover an incomplete video, we need to consider three critical parameters. The *first* one is the sparsity of $\mathcal{W}$. It can be denoted by the percentage of missing entries. The smaller the percentage, the less sparse the $\mathcal{W}$ will be. Meanwhile, the larger the percentage, the more costly to recover the incomplete video. The *second* is the randomness of $\mathcal{W}$. The sparse element locations are generated from a standard normal distribution with small perturbation noises. The *third* is the rank of $\mathcal{V}$, which is defined as the smallest number of rank-1 tensors that generate $\mathcal{V}$ as their sum [17].

Fig.1 illustrates the frames from the recovered actions in KTH dataset under a fix rank $R = 16$. We set different occlusion percentages ranging from 1% to 90%. We observe that the frames can be nicely recovered if the occlusion percentage is under 80%. Severe occlusion occurs above this percentage, making the recovered frames full of noise, or even meaningless (see the $9th$ row in Fig.1). This is because the original video structure is heavily corrupted, and the recovery algorithm fails to factorize its factor matrices. Note that, although under 1% occlusion, the recovered frames in the $3rd$ row look as blurred as the $5th$ row, this stems from the nature of the recovery procedure, which is not intended to generate element-wise (pixel-wise) recovery.

3. COMPACT REPRESENTATIONS

After incomplete video recovery, we are facing two issues: (1) The three-way tensor is highly costly to be directly used in classification due to high dimensionality. (2) It is hard to extract dominant features due to heavy contamination.

To overcome both issues, we extend the rank-1 tensor decomposition method proposed in [18] to generate the low-dimensional compact representations. The outputs are one scalar value and three compact one-dimensional vectors, which discriminately represent the decomposed video. In the context of [18], the tensors degenerate to symmetric tensor due to frame symmetry. In our scenario, however, we handle

Algorithm 1: Tensor rank-1 decomposition

input : A 3-order tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$, and an iteration termination threshold ϵ

output: Three vectors α, β, and γ that minimize $\|\mathcal{A} - \lambda\alpha \circ \beta \circ \gamma\|_2$, where $\alpha \in \mathbb{R}^{I \times 1}$, $\beta \in \mathbb{R}^{J \times 1}$, $\gamma \in \mathbb{R}^{K \times 1}$, and $\|\alpha\|_2 = \|\beta\|_2 = \|\gamma\|_2 = 1$

Initialize $\alpha^{(0)}, \beta^{(0)}, and\ \gamma^{(0)}$;

while $\|\mathcal{A} - \lambda^{(t)}\alpha^{(t)} \circ \beta^{(t)} \circ \gamma^{(t)}\|_2 \geq \epsilon$ **do**

$\widetilde{\alpha}^{(t+1)} = \mathcal{A}\overline{\times}_2\beta^{(t)}\overline{\times}_3\gamma^{(t)}$;

$\widetilde{\beta}^{(t+1)} = \mathcal{A}\overline{\times}_1\alpha^{(t)}\overline{\times}_3\gamma^{(t)}$;

$\widetilde{\gamma}^{(t+1)} = \mathcal{A}\overline{\times}_1\alpha^{(t)}\overline{\times}_2\beta^{(t)}$;

$\alpha^{(t+1)} = \widetilde{\alpha}^{(t+1)}/\|\widetilde{\alpha}^{(t+1)}\|$;

$\beta^{(t+1)} = \widetilde{\beta}^{(t+1)}/\|\widetilde{\beta}^{(t+1)}\|$;

$\gamma^{(t+1)} = \widetilde{\gamma}^{(t+1)}/\|\widetilde{\gamma}^{(t+1)}\|$;

$\lambda^{(t+1)} = \mathcal{A}\overline{\times}_1\alpha^{(t+1)}\overline{\times}_2\beta^{(t+1)}\overline{\times}_3\gamma^{(t+1)}$;

end

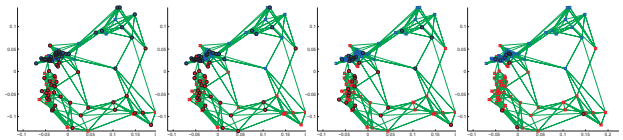

Fig. 2. The constructed k-NN ($k = 5$) sparse weighted graph after manifold denoising for the KTH dataset. For a total of 100 videos, there are two classes involved: 49 *running* actions (Red) and 51 *handwaving* actions (Blue). All circle dots denote complete videos, and all square dots denote sparse incomplete videos. The percentage of sparse unlabeled videos, from left to right, are 30%, 50%, 70%, and 90%, respectively. (Zoom in for better view)

instead the general asymmetric tensors due the randomness and sparseness of the weight tensor $\mathcal{W}$.

Starting with random initial values for α, β, and γ, the algorithm alternately updates one variable while fixing the other two and iteratively achieves the optimal decomposition. By Algorithm 1, we transform any 3D tensor $\mathcal{A}$ into three compact vectors α, β, and γ, such that $\|\mathcal{A} - \lambda\alpha \circ \beta \circ \gamma\|_2$ is less than a predefined sufficiently small threshold value ϵ, and that $\|\alpha\|_2 = \|\beta\|_2 = \|\gamma\|_2 = 1$. We linearly concatenate those three vectors to form a single feature vector f such that $f \in \mathbb{R}^{(I+K+J)\times 1}$. Therefore, the dimensionality of the feature vector is now reduced from $I \times K \times J$ to $I + K + J$. The details can be referred in [18].

4. CLASSIFICATION

We make three assumptions before classification: (1) There are many sparse videos available. (2) It might be expensive or impractical to label the sparse videos due to heavy occlusions. (3) Complete videos are labeled, whereas sparse ones are unlabeled. The reasons for those three are straightforward. On the one hand, the essence of our framework lies in the conjecture that using unlabeled incomplete videos would be helpful for classification. We believe that it might be hard to directly use sparse videos in classification if the occluders are dense, but their recovered versions could make this possible. On the other hand, the semi-supervised learning (SSL) uses readily available unlabeled data to improve classification rate given labeled data is scarce or limited. If sparse videos dominate the dataset and they happen to be unlabeled, the SSL will be our best choice.

The first key component in our SSL is to build a sparse weighted graph out of video data. Graph based models are ideally suited to represent data based on pairwise information such as similarities, distances, and relations [19]. Practically, instead of a fully connected dense graph, we prefer sparse graphs such as the k-NN graph and the ϵ-NN graph for video graph construction. We use the Gaussian kernel similarity function to measure the similarity between graph nodes, i.e., $s(x, y) = exp(-\|x - y\|^2/(2\sigma^2))$. Fig.2 show the constructed k-NN graphs for KTH dataset. Since many graph-based methods can be viewed as estimating a function f on the graph [19][20], we then follow the method in [20] and adopt *transductive learning* via regularized least squares to predict labels on the unlabeled videos.

5. EXPERIMENTS

To fully characterize how the incomplete videos affect the classification stage, we extensively evaluate a total of 280 possible parameter combinations. Here we emphasize that, we exhaustively examine all the 280 cases, primarily because we want to spot the true underlying factors behind our challenging classification-with-incomplete-data scenario.

However, to our best knowledge, there is no existing "incomplete video dataset" available for benchmark purpose, we thus create two special benchmarks from the KTH and the UCF Sports dataset. To decide how the SSL method behaves under different sparsity settings, we chose 2 action classes from the KTH dataset (*running* and *handwaving*). A total of 100 videos belong to these two actions, of which there are 49 running actions and 51 handwaving actions. The second benchmark stems from the UCF Sports dataset, in which a total of 40 actions are considered, 20 *walkingfront* actions and 20 *benchswing* actions. Our benchmarks are distinct from conventional ones because the many different sparsity and rank settings were generated from these videos.

5.1. 280 parameter combinations

Let n be its total video number for each benchmark. We randomly selected n_c out of n and regarded them as labeled complete observations, while for the rest $n_o = n - n_c$ videos, we generated sparse videos under various settings, and considered them as unlabeled observations. The sparsity and randomness of n_o videos were specified by tensor rank $R = \{4, 8, 12, 16\}$ and sparseness percentages $P = \{1\%, 20\%, 30\%, 40\%, 60\%, 80\%, 90\%\}$. The combination of both parameters yielded a total of $4 \times 7 = 28$ settings for each video, both in the complete and the sparse set.

For each of the n_o sparse videos, following the steps in Section 2, we recover its complete version under all 28 settings. Each recovered video yields a three-dimensional

Table 1. The classification error rates of our two benchmarks. The first and second table illustrate the error rates (%) for benchmarks stemming from KTH and UCF Sports, respectively. We extensively scrutinize 280 possible parameter combinations for our classification-with-incomplete-data scenario. Note that "r8" means the evaluation is under rank "R=8"

P	P=1%				P=20%				P=30%				P=40%				P=60%				P=80%				P=90%			
Q	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16
Q=90%	14.7	17.6	13.3	10.3	20.2	15.2	12.6	11.2	11.1	15.5	11.1	4.6	22.3	15.6	13.7	10.4	26.5	18.6	22.2	10.3	47.7	39.8	36.6	35.7	49.9	45.2	48.5	46.8
Q=70%	8.4	9.8	7.1	8.7	13.4	12.0	7.3	10.7	14.0	12.5	8.4	5.7	8.9	12.7	10.3	8.5	19.9	15.6	18.3	17.8	31.3	15.4	20.1	17.7	41.1	48.2	49.4	47.1
Q=50%	7.9	9.3	5.0	6.7	14.3	13.3	8.5	17.4	7.8	5.8	9.4	4.3	9.7	12.5	6.7	7.2	9.6	9.7	5.9	6.2	22.6	18.1	13.2	14.3	47.5	41.7	45.9	48.6
Q=30%	5.3	5.2	9.4	6.1	7.8	9.2	5.2	13.1	3.1	6.2	9.5	6.9	8.6	8.5	16.7	14.0	9.7	6.5	6.5	3.8	17.6	14.1	15.0	9.4	49.6	43.8	44.5	43.9
Q=10%	10.2	12.7	9.4	5.1	8.1	10.4	6.7	5.8	8.3	6.5	9.1	3.2	11.5	20.4	6.6	14.8	9.2	7.3	10.1	2.0	9.6	14.7	11.2	8.0	48.4	40.1	44.2	39.2

P	P=1%				P=20%				P=30%				P=40%				P=60%				P=80%				P=90%			
Q	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16	r4	r8	r12	r16
Q=90%	16.8	14.6	13.2	9.3	15.2	15.3	14.8	11.3	18.0	15.4	11.0	13.6	22.2	19.7	17.8	17.4	26.5	19.7	22.1	17.3	35.6	33.2	32.6	30.8	49.6	42.2	47.6	48.8
Q=70%	15.5	16.9	12.0	14.8	13.6	12.9	9.4	8.8	16.0	19.5	20.5	17.9	20.9	17.8	15.3	13.7	24.9	23.6	20.5	18.4	34.3	36.3	32.2	30.9	44.1	39.3	47.7	49.6
Q=50%	17.8	14.8	12.9	13.2	14.1	14.6	13.5	13.3	17.8	18.9	19.8	23.2	24.8	28.8	20.9	18.2	25.8	22.8	18.9	20.2	42.8	38.4	33.2	30.0	48.6	46.6	44.8	38.6
Q=30%	15.5	15.2	16.0	17.9	17.5	18.2	13.2	9.0	18.2	17.5	19.7	12.9	18.9	17.4	25.4	40.0	42.7	36.5	36.4	37.7	47.7	40.4	35.1	36.4	49.6	43.9	42.7	48.9
Q=10%	14.0	13.2	20.1	15.6	18.0	20.1	8.8	15.7	20.1	15.2	14.1	16.4	20.4	17.3	15.8	17.9	40.1	47.2	30.0	35.9	38.3	44.4	45.1	43.0	49.2	46.2	49.1	48.0

tensor. We then generated its rank-1 compact representations using the Algorithm 1, yielding its final feature vectors. It is worth noting that these final feature vectors are not necessarily of the same length due to different video sizes. Before the vectors are fed into the final graph based semi supervised learning stage, a z-normalization was conducted to make them of identical dimensions.

Another critical parameter $Q = \{10\%, 30\%, 50\%, 70\%, 90\%\}$ was also configured throughout our experiments to denote the percentage of the n_o sparse videos out of all videos, namely, $Q = n_o/(n_o + n_c)$. Q was very necessary in SSL learning, since practically, we cannot know in advance how many sparse videos should be used for better classification. Under each setting, we chose $n_o = Q \times n$ sparse videos randomly rather than deterministically (Fig.2).

To build the sparse weighted graph, we chose between the k-NN and the ϵ-NN graph. Note that as we increase the k or ϵ, the number of edges increase accordingly (Fig.2). Practically, we fixed $k = 5$ and $\epsilon = 0.3$ throughout our experiments. If under a certain setting the classification rate is larger for the k-NN graph structure than for the ϵ-NN structure, then we chose k-NN graph, and vice versa.

5.2. Results and discussions

There were totally 28 sparsity settings for each sparse video. If we take into consideration the 5 percentage ratios given by Q and the 2 graph structures in SSL framework, there will be totally $28 \times 5 \times 2 = 280$ experimental configurations for each benchmark. Note that every single configuration was involved with randomness: the sparsity itself was random, the selection of whether a video should be considered as sparse video was also made random. We thus performed classification on every single configuration for 20 times (10 for k-NN graph, the other 10 for ϵ-NN graph). We computed the average classification rate for each structure, and chose the larger value between these two. The classification rate comparisons for KTH and UCF Sports is shown in Table 1.

How does rank value affect error rates? The rank R is the primary factor that affects the incomplete video recovery. The gradient descent method benefits from a larger rank value, and hence increases recovery precision. Experimentally, we observed in most cases that larger rank does help reduce classification error rate (a.k.a, boosting classification rates). Also, it is worth noting that in Fig.1, we use fixed rank value $R = 16$, which makes the recovered frames more authentic than other ranks smaller than 16. Note that larger ranks also introduces higher computational burden.

How does the "incompleteness" affect error rates? We studied a whole range of possible occlusion percentages P (Table 1), and observed that as P increases, the classification error rates follow an overall subtle (but not monotonic) increasing pattern. Due to randomness, lower Q in some cases (e.g., 60% and 80% for UCF sports) leads to higher error rates. But the overall trend can be readily perceived. Firstly, under heavy occlusion ($P \geqslant 90\%$), a large portion of video pixels is missing and video's structure is heavily corrupted. The recovered incomplete videos no longer contribute positively to the classification rates that are slightly better than random binary selection (i.e. 50% probability). Secondly, within the range $1\% \sim 80\%$, the incomplete videos are under either mild or large occlusions. By properly combining occlusion percentage P and unlabeled data ratio Q, their classification rates can be acceptable, or even very high. To sum up, *incomplete videos, once properly recovered, indeed show merits in action classification. Especially, in the cases where only few complete videos are available, it is feasible (or even necessary) to include available incomplete videos, rather than simply ignore or discard them*.

6. CONCLUSIONS

In this paper, we study to what extent sparse unlabeled data can affect the action classification problem, and we propose a unified framework for handling the sparsity problem. We experimented with an exhaustive set of possible situations, and recover the complete version of the sparse videos by a CP-based decomposition algorithm. Then, with the help of a semi-supervised learning, we demonstrate the possibility of classifying actions under high or even severe sparsity. Our test results on two benchmarks show that it is feasible to include incomplete videos rather than simply discarding them. We plan to create benchmarks from other challenging datasets to better scrutinize this problem that has a direct application in the popular area of compressive sensing.

7. REFERENCES

[1] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2046–2053.

[2] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.

[3] Alexander Kläser, Marcin Marszałek, and Cordelia Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*, sep 2008, pp. 995–1004.

[4] J.C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[5] S. Kwak, W. Nam, B. Han, and J.H. Han, "Learning occlusion with likelihoods for visual tracking," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1551–1558.

[6] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse occlusion detection with optical flow," *International Journal of Computer Vision*, pp. 1–17, 2011.

[7] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," *Computer Vision–ECCV 2010*, pp. 635–648, 2010.

[8] J.Y. Park and M.B. Wakin, "A multiscale framework for compressive sensing of video," in *Picture Coding Symposium, 2009. PCS 2009*. IEEE, 2009, pp. 1–4.

[9] David L Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] Emmanuel J Candès, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

[11] V. Stankovic, L. Stankovic, and Cheng S., "Compressive video sampling," *16th European Signal Processing Conference*, 2008.

[12] Richard Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–212, 2007.

[13] Jason N Laska, Mark A Davenport, and Richard G Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*. IEEE, 2009, pp. 1556–1560.

[14] Aswin C Sankaranarayanan, Pavan K Turaga, Richard G Baraniuk, and Rama Chellappa, "Compressive acquisition of dynamic scenes," in *Computer Vision–ECCV 2010*, pp. 129–142. Springer, 2010.

[15] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 3, pp. 463–476, 2007.

[16] E. Acar, D.M. Dunlavy, T.G. Kolda, and M. Mørup, "Scalable tensor factorizations with missing data," *Siam Datamining 2010 (SDM 2010)*, 2010.

[17] J.B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[18] Sun Chuan, Imran N. Junejo, and Foroosh Hassan, "Action recognition using rank-1 approximation of joint self-similarity volume.," in *ICCV, Computer Vision, 2011 IEEE 13th International Conference on*. 2011, pp. 1007–1012, IEEE.

[19] X. Zhu, J. Lafferty, and R. Rosenfeld, *Semi-supervised learning with graphs*, Ph.D. thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2005.

[20] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, pp. 321–328, 2004.