Spatio-temporal Person Retrieval via Natural Language Queries

Masataka Yamaguchi[1], Kuniaki Saito[1], Yoshitaka Ushiku[1], Tatsuya Harada[1,2]

[1]Graduate School of Information Science and Technology, The University of Tokyo   [2]RIKEN

{yamaguchi, ksaito, ushiku, harada}@mi.t.u-tokyo.ac.jp

Abstract

In this paper, we address the problem of spatio-temporal person retrieval from videos using a natural language query, in which we output a tube (i.e., a sequence of bounding boxes) which encloses the person described by the query. For this problem, we introduce a novel dataset consisting of videos containing people annotated with bounding boxes for each second and with five natural language descriptions. To retrieve the tube of the person described by a given natural language query, we design a model that combines methods for spatio-temporal human detection and multimodal retrieval. We conduct comprehensive experiments to compare a variety of tube and text representations and multimodal retrieval methods, and present a strong baseline in this task as well as demonstrate the efficacy of our tube representation and multimodal feature embedding technique. Finally, we demonstrate the versatility of our model by applying it to two other important tasks.

1. Introduction

As the number of videos uploaded on the web or stored in personal devices continues to increase, systems capable of finding a person in videos are in greater demand. Although object instance search methods [32, 19] can be used to find a person in videos, such methods require an example image of that person. In terms of usability, it would be more desirable if the person could be retrieved using more easily available types of queries.

Motivated by this demand, we address the problem of retrieving a person from multiple video clips via a natural language description, which can be easily input and used even if no image of the person is available. Specifically, we output the target people in the form of tubes (i.e., sequences of bounding boxes), which provide rich information that can be used for various applications, such as video editing. The overview of our task is shown in Figure 1.

To address this problem, we constructed a novel dataset as a platform for evaluating this task. We first collected 5,293 video clips from the ActivityNet dataset [1], which
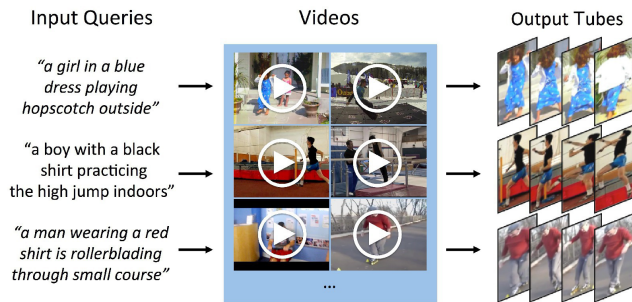

Figure 1. Overview of our task. Given a natural language query, we aim to retrieve a person described by it from the videos in the form of a spatio-temporal tube.

contains videos including diverse human actions. We then chose 6,073 people in those clips and annotated each of them with a bounding box for each second and also with five descriptions. The total duration of our dataset is 13.7 hours, the number of bounding boxes is 62,627, and the number of descriptions is 30,365.

To solve this problem, we design a model in which we first obtain candidate tubes and then retrieve the top K tubes suitable for a given natural language query. To obtain candidate tubes, we utilize a spatio-temporal human detection method based on convolutional neural networks (CNNs). To retrieve tubes via a given query, we utilize a multimodal retrieval method in combination with rich tube and description representations. We construct a strong baseline in this task by conducting comprehensive experiments to compare a variety of tube and text representations and multimodal feature embedding methods. Through the experiments we also demonstrate the efficacy of our novel tube representation and feature embedding technique.

Moreover, the model trained with our dataset can be used for various purposes. To verify the versatility of our model, we apply our model to two important tasks: (1) video-clip retrieval and (2) spatio-temporal action detection, and show our model is also effective in these tasks.

We will make the code, features, and annotations pub-

licly available[1].

2. Related work

Spatio-temporal action detection. Spatio-temporal action detection is the problem of localizing people performing an action in a set of pre-defined action categories. A number of approaches to this problem have been developed [37, 41, 2, 6, 44, 34, 29, 23]. The task we address is similar to spatio-temporal action detection in that we output the tubes of people suitable for the query, however it is more challenging than spatio-temporal action detection in that a query is given in the form of a nearly-free natural language description.

Locating objects in images via natural language queries. [24, 40, 7, 17, 47, 20, 28, 43] addressed the problem of localizing objects in images via natural language queries. The task we address can be seen as a video counterpart of localizing a target in an image based on text queries. However, our task can be considered more challenging than this task in the following two ways: (1) we retrieve targets from multiple videos rather than from an image, and (2) we output the results in the form of sequences of bounding boxes.

Natural language person search. Recently, [13] addressed the problem of retrieving person images with natural language queries. For similar reasons as above, our task can be considered more complex than the task addressed by [13].

Natural language video search. Several works such as [21, 14, 33, 16, 30] addressed the problem of natural language video search. Compared to this task in which one returns the clips as retrieval results, the task we address can be considered more difficult since one returns the results in the form of tubes. Accordingly, models for our task can be applied to more diverse purposes than those for video search.

Locating objects in videos via natural language queries. The studies most related to this paper are [30, 16, 15], which addressed the problem of grounding objects described in a given sentence in a given video clip. [30] localized objects using their sentence-tracker, which matches nouns in a given sentence and objects in a given video based on pre-defined rules for visual attributes such as colors and actions. [30] used a dataset that comprises 94 short video clips, containing limited types of actors, objects, and actions, using similar scenes filmed from similar camera angles. [16] proposed a method that first parses a given sentence into a semantic graph, then matches noun nodes in the graph and detected objects using a generalized bipartite matching algorithm in combination with pre-defined scoring functions for visual attributes such as object appearance and motion. [16] used a dataset comprised of 21 videos filmed by an on-vehicle camera, of which the average frame length was 381.

Note that there are three key differences between the problem settings in existing works [30, 16] and our study:
(1) [30, 16] focused on localizing one or more nouns included in a given query, whereas we assume that a given query describes just one person and focus on localizing him/her.
(2) [30, 16] addressed the problem of localizing targets in a single video clip, whereas we address the problem of retrieving targets from multiple video clips.
(3) The videos used in [30, 16] were filmed under limited conditions or in limited environments, whereas the videos used in our work are derived from the ActivityNet dataset [1], of which videos are very diverse in terms of backgrounds, camera angles, actors, human actions, and human attributes.

It is also worth noting that the methods used in [30, 16] require a pre-defined rule or scoring function for each visual attribute. Contrarily, our model utilizes a multimodal retrieval method in combination with recently-proposed rich text and visual representations, and can retrieve a target described by a given query without any pre-defined rules or scoring functions for visual attributes.

More recently, [15] addressed the problem of tracking a target in a given clip using a query that describes the target in the first frame only. While queries used in [15] do not contain time-series information, queries used in our work do. To fully utilize information contained by queries in our task, models must have the ability to comprehend time-series information such as person's action and state change.

3. Dataset

To tackle the problem of retrieving a person in videos using a natural language query, we created a novel dataset, which consists of videos containing people annotated with bounding boxes every second and with five descriptions Dataset examples are shown in Figure 2. We first explain how our data was collected, and then show the statistics of our dataset.

3.1. Dataset Collection

3.1.1 Selection of Video Clips and People

First, we collected short video clips for the dataset. To diversify visual content in our dataset, we chose clips from videos in the ActivityNet dataset [1], which covers a wide range of complex human activities. We first split each of the activity instances in ActivityNet into shots based on the L1 distance between color-histograms of adjacent frames, then checked whether there are any people who satisfy all of the following conditions in each shot: (1) the head of the person can be seen, (2) the person is one of the main people in the shot, (3) the person exists for more than 5 seconds, and (4) the person is displaying any characteristic actions, or his
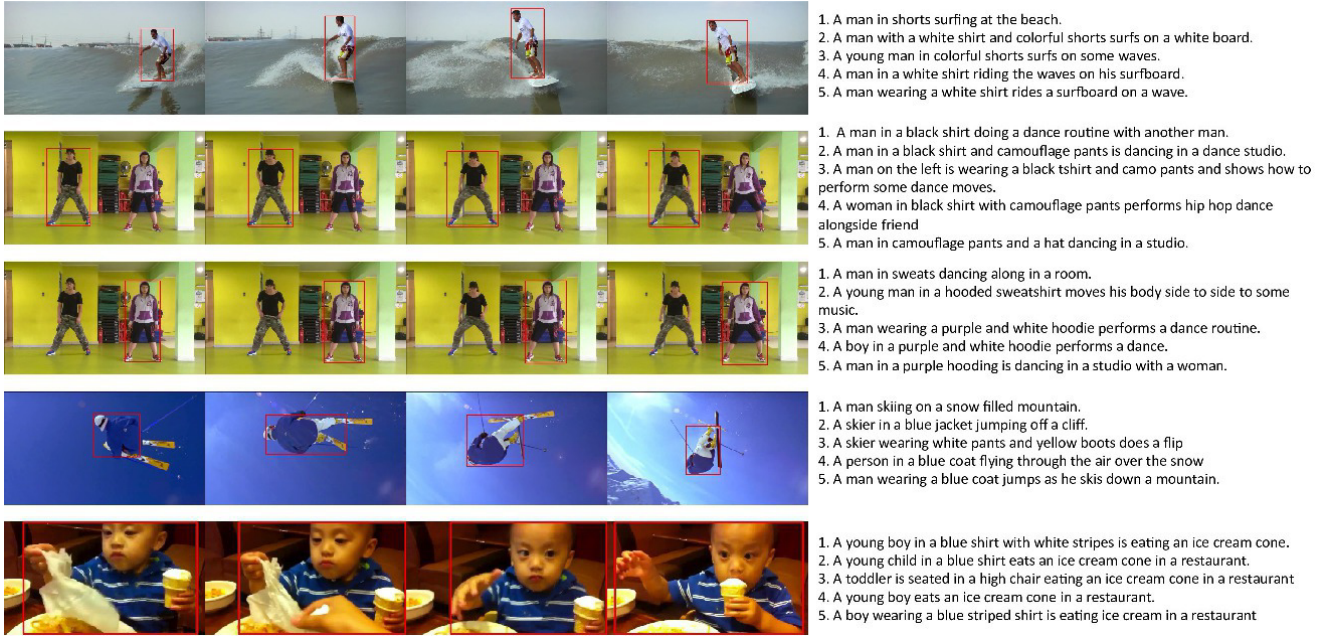

Figure 2. Examples of people and annotated descriptions in our dataset. We show only four frames for each of five samples.

For the surfing sample:
1. A man in shorts surfing at the beach.
2. A man with a white shirt and colorful shorts surfs on a white board.
3. A young man in colorful shorts surfs on some waves.
4. A man in a white shirt riding the waves on his surfboard.
5. A man wearing a white shirt rides a surfboard on a wave.

For the dance sample:
1. A man in a black shirt doing a dance routine with another man.
2. A man in a black shirt and camouflage pants is dancing in a dance studio.
3. A man on the left is wearing a black tshirt and camo pants and shows how to perform some dance moves.
4. A woman in black shirt with camouflage pants performs hip hop dance alongside friend
5. A man in camouflage pants and a hat dancing in a studio.

For the hoodie dance sample:
1. A man in sweats dancing along in a room.
2. A young man in a hooded sweatshirt moves his body side to side to some music.
3. A man wearing a purple and white hoodie performs a dance routine.
4. A boy in a purple and white hoodie performs a dance.
5. A man in a purple hooding is dancing in a studio with a woman.

For the skiing sample:
1. A man skiing on a snow filled mountain.
2. A skier in a blue jacket jumping off a cliff.
3. A skier wearing white pants and yellow boots does a flip
4. A person in a blue coat flying through the air over the snow
5. A man wearing a blue coat jumps as he skis down a mountain.

For the ice cream sample:
1. A young boy in a blue shirt with white stripes is eating an ice cream cone.
2. A young child in a blue shirt eats an ice cream cone in a restaurant.
3. A toddler is seated in a high chair eating an ice cream cone in a restaurant
4. A young boy eats an ice cream cone in a restaurant.
5. A boy wearing a blue striped shirt is eating ice cream in a restaurant

or her appearance is characteristic. If there were any people who satisfied the above conditions in the shot, we chose a period of around 10 seconds from the period in which these people exist in the shot. To ensure the diversity of the dataset, for training data we chose at most two clips for each video, and for validation and testing data we chose at most one clip for each video.

3.1.2 Bounding Box Annotation

Next, we annotated each person who satisfied the conditions explained in Section 3.1.1 with a bounding box every second, with the help of in-house annotators. In the case where the target person could not be seen because of occlusion or framing out, we asked them not to annotate the person with a bounding box. To improve the quality of the annotation, we next checked all the annotated bounding boxes and modified inappropriate ones.

In addition to bounding boxes, we also annotated each clip with a binary label every second, to indicate whether the frame of the timestep includes any people who are not annotated with bounding boxes. Namely, in the timestep of which the binary label is true, it is guaranteed that there are one or more people not annotated with bounding boxes. These labels are useful when training human detectors using this dataset since we can remove images that include unlabeled humans using these labels.

3.1.3 Description Annotation

Next, we annotated each person with five natural language descriptions using Amazon Mechanical Turk. Following [25], we used a brief qualification test to prevent workers whose English skills were insufficient from joining the task. For each target person, we asked five workers to describe the person under the following conditions: (1) the description must contain at least eight words, (2) the description must contain all the important parts of the person such as their clothes and actions, (3) the description must be unique to the target person in case multiple people are present, and (4) the description should also include information about the scene around the person if it can be described.

In cases where multiple people were present, it was necessary to prevent workers from describing people other than the target person. We solved this problem by using a tool we developed to indicate the target person by repeatedly redrawing a bounding box enclosing him/her on the embedded video element.

To improve the quality of the annotated descriptions, we finally asked workers to correct each description if it contains any spelling or grammatical errors, or does not satisfy any of the above conditions.

3.2. Dataset Statistics

Statistics of the training, validation, and testing datasets are included in Table 1. We collected 5,293 clips and annotated 6,073 people in those clips with 30,365 descriptions and 62,627 bounding boxes in total. The number of unique

	#Clip	Duration	#Person	#Description	#Box
train	4,734	732 min.	5,437	27,185	55,875
val	276	44 min.	313	1,565	3,311
test	283	46 min.	323	1,615	3,441

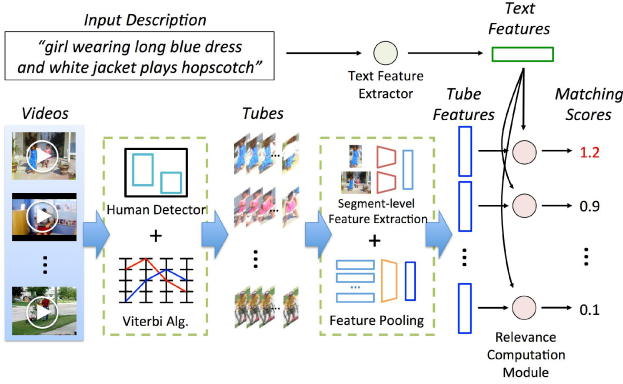Table 1. Dataset statistics. We discuss other statistics further on the supplementary.

Figure 3. Diagram of our model. We first propose candidate tubes, and then extract features of them. Given a description, we extract its text features, and then compute the matching score between the description and each tube using extracted features.

words is 6,730, the total number of words is 398,397 and the average length of descriptions is 13.1. We discuss the statistics of our dataset further in the supplementary.

4. Approach

4.1. Overview

We aim to retrieve a tube suitable for a person described by a given natural language description from a set of video clips. To this end, inspired by the recent method for object detection [5], we design a model in which we first propose candidate tubes, rank them based on a given query, and then output the top K scoring tubes. Figure 3 shows an overview of our method. For obtaining candidate tubes, we utilize a method for spatio-temporal person detection. For ranking candidate tubes based on a given query, we extract features from tubes and the text query, and then retrieve tubes suitable for the query based on extracted features using a multimodal retrieval method.

In the following, we present details of the modules.

4.2. Candidate Tube Proposal

Some researchers previously proposed methods [46, 39, 22, 9, 45, 18] for obtaining candidate tubes of human actions, however, many of these utilize motion information for obtaining candidate tubes and such methods may fail to propose candidate tubes for people who move subtly or infre-

quently. Alternatively, we directly obtain candidate tubes by using spatio-temporal human detection. More specifically, we detect human tubes with a method inspired by the action detection method proposed in [6], in which we first detect humans in each frame, and then obtain candidate tubes by linking the detected bounding boxes. In the following, we explain the details of those two steps.

4.2.1 Human Detection in each frame

First, we detect humans in each frame. For human detection, we use Faster R-CNN [26] as a detector for static images and employ VGG 16-layer net [31] pretrained on the ImageNet dataset [3] as a base network within Faster R-CNN. For training and early stopping, we use our training and validation datasets, respectively. The approximate joint training strategy introduced in [4] is used to train the model.

4.2.2 Candidate tube generation

To create spatio-temporal tubes, we next link up the human detections between frames. As with [6], we define the linking score $s_{link}(B_t, B_{t+1})$ between the i_t-th bounding box B_{t,i_t} at time t and the i_{t+1}-th bounding box $B_{t+1,i_{t+1}}$ at time $t+1$ as follows:

$$s_{link}(B_{t,i_t}, B_{t+1,i_{t+1}}) = s_{det}(B_{t,i_t}) + s_{det}(B_{t+1,i_{t+1}}) + \lambda \mathrm{OV}(B_{t,i_t}, B_{t+1,i_{t+1}}), \tag{1}$$

where $s_{det}(B)$ is the human score of the bounding box B obtained by Faster R-CNN, $\mathrm{OV}(B_{t,i_t}, B_{t+1,i_{t+1}})$ is the intersection over union between those two bounding boxes, and λ is a hyperparameter.

As with [6], we next define the energy function $E(\overline{B})$ of the tube $\overline{B} = [B_{1,i_1}, B_{2,i_2}, ..., B_{T,i_T}]$ as follows:

$$E(\overline{B}) = \frac{1}{T} \sum_{t=1}^{T-1} s_{link}(B_{t,i_t}, B_{t+1,i_{t+1}}). \tag{2}$$

We can obtain the optimal path $\overline{B}^* = \arg\max E(\overline{B})$ by using the Viterbi algorithm. We repeatedly obtain the optimal path by maximizing Equation (2) and removing the bounding boxes included in the obtained path until at least one set of bounding boxes included in the same timestep is empty. We finally choose the top N_c scoring tubes from those obtained from all the clips in the target dataset, and use them as the candidate tubes.

4.3. Ranking tubes via a description

We next explain the approach to ranking tubes given a natural language description. Inspired by the state-of-the-art method [40] in the image-sentence retrieval task, we employ the approach in which we project features of both
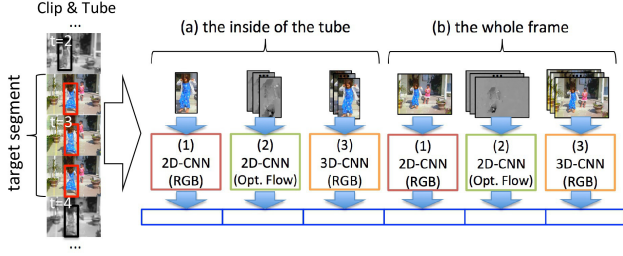
Figure 4. Diagram of extracting the segment-level features. For each tube, we compute and concatenate the above six types of features every second.

modalities into the common space and retrieve tubes in the space. We first extract the tube features from candidate tubes and the text features from a given description. We then project the features of those tubes and the given description into the common space, and rank the candidate tubes based on the distance to the description in that space.

In the following, we explain the tube and description features and the multimodal retrieval methods we use.

4.3.1 Tube features

To extract tube features from a given tube, we first extract the segment-level features once per second, and then construct the tube's feature vector by aggregating those features.

Segment-level features. As shown in Figure 4, we extract:
(1) the CNN features extracted from the RGB images,
(2) the CNN features extracted from the sequences of the optical flow maps, and
(3) the 3D-CNN (or C3D) features [38] extracted from the sequences of RGB images
from (a) the inside of tubes and (b) the whole frames for each. We then construct one feature vector by concatenating all of six types of the above features and use it as the segment-level features.

To extract the CNN features from the RGB images, we use the outputs from the fc7 layer in the VGG-16 layer net pretrained on ImageNet. To extract the CNN features from the optical flow images, we use the outputs from the fc7 layer in the VGG-16 layer net first pretrained on ImageNet and then re-trained on the first split of the UCF101 dataset [35]. We used the model provided by the CUHK team on the web[2] [42]. Following the CUHK team, we extract the optical flow maps by using the TV-L1 optical flow algorithm [48] and input the optical flow maps corresponding to the five frames preceding the target timestep and the five frames following the target timestep into the model. To extract the C3D features, we use the outputs from the fc7 layer

---

in the model pretrained on the Sports-1M dataset [10][3]. We input the RGB images corresponding to the frame at the target timestep, the 7 frames preceding it, and the 8 frames following it into the model.

The proposed segment-level features have the following ideal properties:

- From the features extracted from (a), the features include information for the inside of the tube.
- From the features extracted from (b), the features include context information.
- From (1) and (3), the features include appearance information.
- From (2) and (3), the features include motion information.
- From (2) and (3), the features include multi-frames' information-integrated information.

Feature aggregation. We next aggregate the set of the segment-level features. We compared three aggregation strategies (mean/max pooling and encoding with an RNN) and combinations of them, and found that even simple mean pooling works well. Therefore, in all experiments in Section 5, we use mean pooling for feature aggregation. We show the results of the comparison of the aggregation strategies in the supplementary.

4.3.2 Description features

We compared the following three methods for extracting text features and combinations of them: the Fisher Vectors (FVs) based on a hybrid Gaussian-Laplacian mixture model (HGLMM) [12], skip-thought vectors [11] and one method based on an RNN trained from scratch. We found that using the FVs based on HGLMM as is works best. Therefore, in all experiments in Section 5, we use the FVs based on HGLMM as text features. We explain the details of three types of text features and the results of comparison of them in the supplementary.

4.3.3 Multimodal Retrieval Methods

We compare the three following methods for multimodal retrieval:

Canonical Correlation Analysis. The first method is Canonical Correlation Analysis (CCA), which is much simpler than complex neural networks yet achieves high accuracy in various retrieval tasks[12, 24]. When using CCA, we retrieve tubes using the matching score $S_{CCA}(i,j)$ between the i-th tube features f_i^t and the j-th description features f_j^d, which is computed as follows:

$$S_{CCA}(i,j) = \cos(\mathrm{diag}(r) \cdot W^t \cdot f_i^t, \mathrm{diag}(r) \cdot W^d \cdot f_j^d), \quad (3)$$

Layer type	# Filters
(Input)	
Inner Product	2048
ReLU	
Inner Product	512
Batch Normalization [8]	
ReLU	
L2 Normalization	
(Output)	

Table 2. The network architecture of DSPE used in this paper for each modality. We also use Dropout [36] ($p = 0.5$) after the first Inner Product layers.

where $\cos(x_1, x_2)$ is the cosine similarity between x_1 and x_2, $\mathrm{diag}(r)$ is a diagonal matrix whose diagonal elements are eigenvalues learned by CCA, and W^t and W^d are the projection matrices of the tube and the description features into the common latent space, respectively.

Deep Structure-Preserving Embedding. Deep Structure-Preserving Embedding (DSPE) [40] is the state-of-the-art method for image-sentence retrieval. DSPE embeds features of two different modalities into the common space using two neural networks trained with the following loss function:

$$
\begin{aligned}
L_{DSPE} = & \sum_{i,j,k} max \left[0, m + \mathrm{d}(x_i, y_j) - \mathrm{d}(x_i, y_k)\right] \\
& + \alpha_1 \sum_{i',j',k'} max \left[0, m + \mathrm{d}(x'_j, y'_i) - \mathrm{d}(x'_k, y'_i)\right] \\
& + \alpha_2 \sum_{i,j,k} max \left[0, m + \mathrm{d}(x_i, x_j) - \mathrm{d}(x_i, x_k)\right] \\
& + \alpha_3 \sum_{i',j',k'} max \left[0, m + \mathrm{d}(y'_i, y'_j) - \mathrm{d}(y'_i, y'_k)\right]
\end{aligned}
\tag{4}
$$

where d is the distance function between two embedded features, $\alpha_i (i = 1, 2, 3)$ and $m > 0$ are hyper-parameters, (x_i, y_j) and (x'_j, y'_i) are positive pairs, (x_i, y_k) and (x'_k, y'_i) are negative pairs, (x_i, x_j) and (y'_i, y'_j) are pairs of examples that are of the same modality and correspond to the same sample of the other modality, and (x_i, x_k) and (y'_i, y'_k) are pairs that do not. When using DSPE, we retrieve tubes using the matching score $S_{DSPE}(i, j) = -\mathrm{d}(f_i^t, f_j^d)$. Following [40], we use the network architecture shown in Table 2, and employ the Euclidian distance as the distance function d.

Modification of DSPE. Although DSPE is an effective retrieval method as is, to further improve accuracy we propose to add a simple yet effective term to the DSPE's loss. In addition to the original loss, which encourages the distance between positive pairs to be smaller than the distances between negative pairs, we propose to add to the loss the
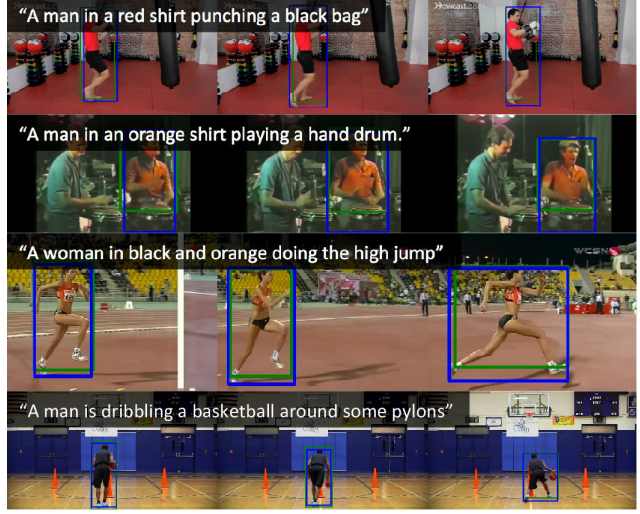

Figure 5. Correctly retrieved examples. The blue and green bounding boxes are the ground truth and the top 1 retrieved result, respectively. Note that the search space consists of 283 video clips with a total duration of 46 minutes.

summation of the distance between positive pairs, which directly encourage the model to reduce the distance between positive pairs. The proposed loss is as follows:

$$
L_{Proposed} = L_{DSPE} + \alpha_4 \sum_{i,j} \mathrm{d}(x_i, y_j),
\tag{5}
$$

where α_4 is a hyper-parameter. We use the same network architecture and distance metric as DSPE. We refer to the embeddings trained with this loss as DSPE++, and use it as the third choice.

5. Experiments

5.1. Settings

We address the problem of retrieving a tube for a person described by each query from video clips in the test dataset, of which the total duration is approximately 46 minutes. We use 1,615 descriptions from the test dataset as the queries.

Evaluation Metric. The retrieval task can be evaluated using R@K, that is, the recall rate of the ground truth included in the top K candidates retrieved using a given query. To use the R@K as metrics of our tasks, we must define the condition to judge whether a retrieved instance can be considered as a true positive of the ground truth. We compute R@K by assuming dt_{tube} is the true positive of gt_{tube} in the case that the following localization score $S_{loc}(gt_{tube}, dt_{tube})$ is over 0.5:

$$
S_{loc}(gt_{tube}, dt_{tube}) = \frac{1}{|\Gamma|} \sum_{f \in \Gamma} \mathrm{OV}(gt_{tube,f}, dt_{tube,f}),
\tag{6}
$$

VGG16 (RGB)		VGG16 (Opt. Flow)		C3D		R@1	R@5	R@10
Tube	Context	Tube	Context	Tube	Context			
✓	-	-	-	-	-	0.276	0.572	0.693
-	✓	-	-	-	-	0.228	0.568	0.713
✓	✓	-	-	-	-	0.279	0.623	0.751
✓	✓	✓	✓	-	-	0.285	0.620	0.760
✓	✓	-	-	✓	✓	**0.300**	0.654	0.788
✓	✓	✓	✓	✓	✓	**0.300**	**0.658**	**0.792**

Table 3. Performance comparison of tube representations. In these experiments, CCA is used for retrieval.
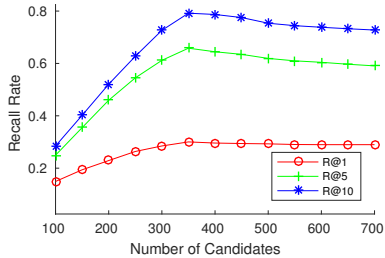
Figure 6. Influence of the number of candidate tubes on the retrieval accuracy.

Multimodal Retrieval Method	R@1	R@5	R@10
CCA	0.300	0.658	0.792
DSPE	0.347	0.687	0.783
DSPE++ (Proposed)	**0.357**	**0.702**	**0.795**

Table 4. Performance comparison of multimodal retrieval methods.

where Γ is the intersection of the set of frames to be annotated with bounding boxes and the set of frames in which gt_{tube} or dt_{tube} has any bounding box.

Training models. For training the retrieval models, we use 27,185 tube-description pairs in our training dataset. We show the details of training and hyper-parameters for them in the supplementary.

5.2. Results

In the following, we discuss three types of experiments. For more analysis, see the supplementary.

Influence of the number of candidate tubes. Our model first chooses the top N_c scoring tubes from tubes obtained by the candidate proposal module, and then ranks those based on a given description. We first discuss experiments conducted to investigate the influence of the number of candidate tubes N_c. In these experiments, we use CCA for retrieval and compute the recall rates by setting $N_c = 100, 150, ..., 700$.

The results for the validation dataset are shown in Figure 6. Considering all metrics, our method achieves the highest retrieval accuracy for $N_c = 350$, and retrieval accuracy decreases as the number of candidates becomes smaller

or larger than $N_c = 350$. These decreases can be explained by the following two reasons: (1) if N_c gets too small, the probability that the chosen tubes include any true positives of the ground truth gets too low, and (2) if N_c gets too large, the amount of noise in the chosen tubes increases.

Because of this result and the fact that the size of the testing dataset is almost the same as that of the validation dataset, we fix $N_c = 350$ in the following experiments.

Comparison of tube representations. Here we discuss the experiments conducted to compare tube representations. In these experiments, we investigate the efficacy of the context information and three types of features explained in Section 4.3. In this comparison, CCA is used for retrieval.

The results are shown in Table 3. In Table 3, the case in which the full range of features are used achieves the highest accuracy. These results suggest the importance of considering not only appearance, but motion, context, and multi-frame information simultaneously in this task.

Comparison of multimodal retrieval methods. Next, we discuss the experiments to compare the multimodal retrieval methods.

The results are presented in Table 4. We can see that while CCA and DSPE are comparable, DSPE++ outperforms them in all metrics. These results suggest the effectiveness of the proposed loss in this task.

Qualitative results. The correctly retrieved examples are shown in Figure 5. We stress that the search space consists of 283 video clips with a total duration of 46 minutes. These examples imply that our method can find a plausible person tube based on a complex query containing various types of information such as clothes and actions. The additional correctly retrieved examples and randomly chosen examples are also shown in the supplementary.

5.3. Applying our model to other tasks

Although our model is designed to retrieve a person tube based on a natural language description, our model can be used for various purposes. In this section, we verify its versatility by applying our trained model to two other important tasks: the clip retrieval task and the spatio-temporal action detection task.

	R@1	R@5	R@10
Baseline	0.373	0.749	0.879
Proposed Approach	**0.415**	**0.806**	**0.893**

Table 5. Retrieval accuracy for the clip retrieval task. Comparison of the baseline method and the method that leverages our method for spatio-temporal person retrieval task.

5.3.1 Clip Retrieval

For certain applications, such as video search, it is sufficient that the model just returns the clip containing the target person rather than the tube of the person. In the following, we demonstrate that our model is also useful for the clip retrieval tasks.

Setup. For this task, we address the problem of retrieving a clip including a person described by each query from the set of video clips in the test dataset. We use 1,615 descriptions in the test dataset as the queries. We use R@K as the evaluation metric.

Baseline. As a baseline, we employ a method that directly learns the common space between the description features and clip features using DSPE++. As clip features, we use the features obtained by concatenating the temporal means of the three types of features (i.e., the CNN features from RGB images, the CNN features from optical flow maps, and the C3D features), extracted from the whole frames. In the training phase, we consider the description and the clip containing the person described by the description as one training pair and train DSPE++ with 27,185 clip-description pairs.

Retrieval with our model. We propose to define the matching score between a description and a clip as the maximum of the relevance scores between the description and each candidate tube in the clip computed by our trained model. More specifically, we employ the following matching score between the i-th description and the j-th clip for retrieving a clip:

$$S^*_{proposed}(i,j) = \max_{k=k_{i,1},...,k_{i,N_i}} S(i,k), \qquad (7)$$

where $S(i,j)$ is the relevance score between i-th description and k-th tube computed by our model, $\left[k_{j,1},...,k_{j,N_i}\right]$ are indices of the candidate tubes in the j-th clip and N_j is the number of candidate tubes in the clip.

Results. Table 5 contains the results. All metrics indicated that higher retrieval accuracy can be achieved by the proposed approach than by using the baseline method. This result suggests that our model is also useful for clip retrieval.

5.3.2 Action Detection

Spatio-temporal action detection is an important task in video analysis. Although our model is designed to retrieve
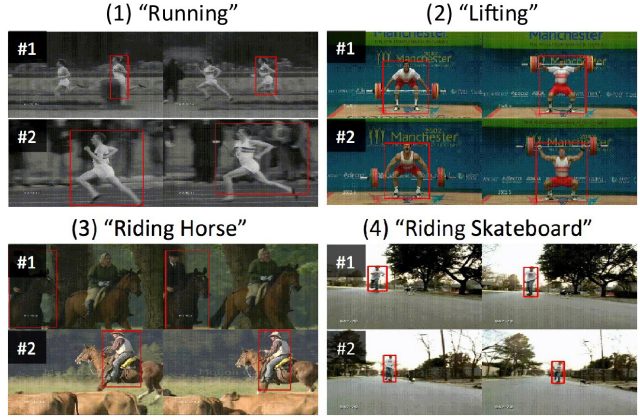

Figure 7. Top 2 tubes retrieved using each of four queries. Two frames are shown for each tube.

a person based on a natural language description, it can be also used for this task using an action category name as a query. To verify this, we conduct an experiment to test the efficacy of our model as a solution for spatio-temporal action detection on the UCF Sports dataset [27], which consists of 150 video clips with 10 action classes.

Figure 7 shows the examples retrieved from the dataset using each of the following four queries: "Running", "Lifting", "Riding Horse" and "Riding Skateboard". Even though our model is trained without the data in the UCF Sports dataset, these action instances can be correctly detected by just inputting the action name to our trained model. These results indicate that our model can be also used as a human-action detector. The quantitative results are also presented in the supplementary.

6. Conclusion

In this paper, we have addressed spatio-temporal person retrieval from multiple videos using natural language queries. We present a dataset of videos containing people annotated with bounding boxes and descriptions. We also present a model that combines methods for spatio-temporal human detection and multimodal retrieval. We optimize the model by conducting comprehensive experiments to compare tube and text representations and multimodal retrieval methods, and construct a strong baseline on this task. Finally, we validate our model's versatility by conducting experiments on two important additional tasks.

7. Acknowledgments

References

[1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 1, 2

[2] W. Chen and J. J. Corso. Action detection by implicit intentional motion clustering. In *ICCV*, 2015. 2

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[4] R. Girshick. Training r-cnns for vaious velocities. https://www.dropbox.com/s/xtr4yd4i5e0vw8g/iccv15_tutorial_training_rbg.pdf?dl=0. 4

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 4

[6] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015. 2, 4

[7] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016. 2

[8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 6

[9] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014. 4

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 5

[11] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015. 5

[12] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 5

[13] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang. Person search with natural language description. In *CVPR*, 2017. 2

[14] X. Li, D. Wang, J. Li, and B. Zhang. Video search in concept subspace: a text-like paradigm. In *CIVR*, 2007. 2

[15] Z. Li, R. Tao, E. Gavves, C. G. Snoek, and A. W. Smeulders. Tracking by natural language specification. In *CVPR*, 2017. 2

[16] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, 2014. 2

[17] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 2

[18] M. Marian Puscas, E. Sangineto, D. Culibrk, and N. Sebe. Unsupervised tube extraction using transductive learning and dense trajectories. In *ICCV*, 2015. 4

[19] J. Meng, J. Yuan, J. Yang, G. Wang, and Y.-P. Tan. Object instance search in videos via spatio-temporal trajectory discovery. *IEEE Transactions on Multimedia*, 18(1):116–127, 2016. 1

[20] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression. In *ECCV*, 2016. 2

[21] S.-Y. Neo, J. Zhao, M.-Y. Kan, and T.-S. Chua. Video retrieval using high level features: Exploiting query matching and confidence-based weighting. In *CIVR*, 2006. 2

[22] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014. 4

[23] X. Peng and C. Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016. 2

[24] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 2, 5

[25] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010. 3

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 4

[27] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 8

[28] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 2

[29] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016. 2

[30] N. Siddharth, A. Barbu, and J. Mark Siskind. Seeing what you're told: Sentence-guided activity recognition in video. In *CVPR*, 2014. 2

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[32] J. Sivic, A. Zisserman, et al. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1

[33] C. G. Snoek, B. Huurnink, L. Hollink, M. De Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia*, 9(5):975–986, 2007. 2

[34] K. Soomro, H. Idrees, and M. Shah. Action localization in videos through context walk. In *ICCV*, 2015. 2

[35] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014. 6

[37] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 2

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 5

[39] J. van Gemert, M. Jain, E. Gati, and C. Snoek. Apt: Action localization proposals from dense trajectories. In *BMVC*, 2015. 4

[40] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 2, 4, 6

[41] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*, 2014. 2

[42] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 5

[43] M. Wang, M. Azab, N. Kojima, R. Mihalcea, and J. Deng. Structured matching for phrase localization. In *ECCV*, 2016. 2

[44] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015. 2

[45] F. Xiao and Y. J. Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *CVPR*, 2016. 4

[46] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015. 4

[47] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV*, 2016. 2

[48] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l 1 optical flow. In *DAGM Symposium on Pattern Recognition*, 2007. 5