Tracking by Natural Language Specification

Zhenyang Li, Ran Tao, Efstratios Gavves, Cees G. M. Snoek, Arnold W.M. Smeulders
QUVA Lab, University of Amsterdam

Abstract

This paper strives to track a target object in a video. Rather than specifying the target in the first frame of a video by a bounding box, we propose to track the object based on a natural language specification of the target, which provides a more natural human-machine interaction as well as a means to improve tracking results. We define three variants of tracking by language specification: one relying on lingual target specification only, one relying on visual target specification based on language, and one leveraging their joint capacity. To show the potential of tracking by natural language specification we extend two popular tracking datasets with lingual descriptions and report experiments. Finally, we also sketch new tracking scenarios in surveillance and other live video streams that become feasible with a lingual specification of the target.

1. Introduction

The goal of this paper is to track an object in video, a long-standing challenge in computer vision. The common approach is to specify a target by means of a bounding box around the object and to track this target as it moves throughout the video [38, 33, 20]. The paradigm has proven to be effective and considerable progress has been achieved [17, 37, 34, 3, 11]. Yet, the fundamental assumption of having a bounding box target specification available has never been challenged. In this paper, we propose a new approach to object tracking in video, in addition to or in contrast to target specification by means of a bounding box.

We are inspired by recent progress in object retrieval [15, 14, 26]. Both Hu *et al*. [15, 14] and Mao *et al*. [26] present a recurrent neural network able to localize an object in an image by means of a natural language query only, either returning a bounding box [15] or a free-form segment [14]. To cope with language ambiguity, Mao *et al*. introduce referring expressions that uniquely describe an object in an image. However, unlike in still images, in videos objects may change their appearance and position, also the background can vary dramatically from frame to frame. Even if the video frames contain the same object category, that object might have a different appearance, be in different location or in a different background, thus rendering any description challenging. Unlike [15, 14, 26] we do not retrieve but track the object of interest in video from a natural language specification.

Figure 1: **Tracking by natural language specification** allows for a novel type of human-machine interaction in object tracking. It enhances standard trackers by helping them against drift. It also opens up opportunities for new applications like random start tracking, for example when the target is lost, or simultaneous multiple-video tracking.

First and foremost, the contribution of this paper is tracking by natural language specification, which allows for a novel type of human-machine interaction in tracking, see Figure 1. In several real-life applications, such as robotics or autonomous driving, defining the target by a description is more natural, *e.g*., "track the red car in the middle lane". As a second contribution, we define three variants of tracking by language specification, that are dominated by lingual target specification, visual target specification, or leverages their joint capacity. As third novelty we enrich standard

tracking from a human-provided bounding box with our language specification. To show the potential of tracking by natural language specification we extend the OTB100 [38] object tracking dataset and ImageNet Video Object Detection dataset [31] with lingual descriptions and report experiments. Finally, we also sketch new tracking application scenarios for surveillance and other live video streams that become feasible with a lingual specification of the target.

2. Related Work

Tracking. The breadth and depth of single object tracking is covered by recent reviews [38, 33] including an overview of the many diverse and dynamic factors to overcome conditions general to the scene: uneven illumination, shadow casting, reflection, as well as to the target: shape changes, and in relation to other objects: occlusion, similar close objects, clutter with the background, and the camera: fast motion and zooming. Diverse benchmarks like OTB [38], ALOV [33], and VOT [20], have accelerated the performance of trackers in general, and they have caused a convergence in tracking methods.

Many modern trackers rely on discriminative correlation filters [4, 12, 7]. While originally selected for the Fast Fourier Transform to compute one channel quickly, Danelljan *et al*. [10] use multiple channels to augment the discrimination of the correlation filters. Henriques *et al*. [12] introduce kernelized filters to further refine the trimming of features to the tracking situation at hand. Ma *et al*. [25] enrich the model with long-term memory, while Danelljan *et al*. [8] proposes a scale-invariant version and Liu *et al*. [24] use structured correlation filters. They all aim to enhance the robustness of discriminative correlation filters against the diverse circumstances of tracking as well as to enhance the generality of discriminating the target from the background.

Neural networks have aided in focusing the tracking on the target by attention-based tracking in [6, 5]. Wang *et al*. [36] transform deep network optimization into sequential ensemble learning by online training. In [28] Nam *et al*. enable fine-tuning to more than one domain, where each domain is represented by a single training sequence. In [34, 3], tracking is cast as instance search for which a Siamese network architecture is used. The original window of the target is compared with candidates windows from the current frame by a similarity function, learned from many examples before the tracking starts. As they function on the similarity to a stable original, and they are not updated during the tracking, Siamese trackers achieve state-of-the-art performance and recovery from loss, while being robust against variations in the query definition [34].

As we are anticipating a sloppy definition of the starting box, we rely on the last of the above mentioned methods. We adopt the Siamese tracker in [34] as our starting point for the visual object tracking in our model, for its robustness against errors in the starting box-definition. Furthermore, the Siamese tracking scheme has the advantage that it does not rely on model dynamics and hence it can be rebooted at any time as long as the lingual description is valid.

Natural language and images. What is common between all the aforementioned tracking methods is that they ask the user to provide a bounding box around the target in the first frame. In this work we do not require a bounding box in any frame. When provided with a target box, the method described here would still be valid to enhance the tracking. Instead, we track on the basis of a lingual description of the target.

Automatically describing the content of images has been an important challenge in computer vision for years, *e.g*., [27, 29]. The interest has increased since recurrent neural architectures have become available [35, 18, 16] with impressive qualitative and quantitative results. In these encoder-decoder models, the encoder typically is a CNN [32], while the decoder is composed of LSTM-cells [13], sequentially predicting the words in the caption. As evaluating captions is subjective, recently a turn has been made to question-answer tasks [1, 21, 40, 22]. In this paper, we also consider the interplay between visual appearance and a lingual description. In contrast to [21, 27, 29, 35, 18, 16, 40, 22], where natural language is the output of the system, for our tracker a natural language expression specifying one or more targets is the input.

Recently [15] proposes the task of object retrieval by natural language specification. A related topic is zero-shot object localization [23], where an object is localized from a verbal description of a previously unseen object. Where [23] learns a matching function between attributes and object segment appearances, [15, 14] learns a matching function between segment appearances and lingual queries. Since a sentence can match many patches in an image dataset, Hu *et al*. [15, 14] cast this as an object segment retrieval problem within an image. They rank image locations according to the estimated resemblance to the sentence description. In the references [23, 15, 14], a sentence is supposed to capture one complete image. It is difficult to generalize the approach beyond single, static images, as the sentence relevant for one frame is not necessarily relevant for all. We propose a model that localizes the target in the video and also learns how to attend to these parts of the query when they become more or less relevant over time. Our attention model conditions text on a video frame, differing from others, *e.g*. Bahdanau *et al*. who condition text on text [2], or Xu *et al*. who condition image on text [39]. Ultimately, we combine lingual specifications with tracking based on visual specifications over time.

3. Tracking by Natural Language Specification

Given a frame in a video and a natural language expression as query, the goal of our work is to track the target in the video as specified by the expression. To achieve this goal we present three models, illustrated in Figure 2.

Model I: Lingual Specification Only

The first model relies on the lingual specification only for tracking. Model I utilizes the *Lingual Specification Network* to analyze the textual description and localizes the target in an arbitrary video frame, as shown in Figure 2.

To analyze the lingual specification we first embed each word into a vector and use an LSTM network to scan the word embedding sequence. For an input sequence $W = (w_1, ..., w_K)$ with K words, at each time step i, an LSTM network takes the i-th word embedding w_i as input and outputs its hidden state h_i. In this manner, the lingual specification is encoded by the hidden states of the LSTM network. We choose the hidden state h_K at the final time step K as the representation of the whole expression. We employ a deep CNN to extract the visual feature map of an input frame. To enable the model to reason about the spatial relationships such as "car in the middle", the spatial coordinates (x, y) of each position are also added as extra channels to the feature maps. We use relative coordinates by normalizing them into $(-1, +1)$. The augmented feature map I_t for frame f_t now contains both local visual and spatial descriptors.

Dynamic convolutional layer. To localize the target in the video frame, we propose a dynamic convolutional layer. This layer generates new convolutional filters on the fly depending on the text query. In the first frame the only information we have about the target is its lingual specification encoded by the last hidden state of the LSTM, namely $s_t = LSTM(W) = h_K$. We, therefore, generate target-specific visual filters based on the language input only. A single layer perceptron is adopted to transform the semantic information in s_t into novel convolutional visual filters $v_t^{language}$:

$$v_t^{language} = \sigma(W_v s_t + b_v), \tag{1}$$

where σ is the sigmoid function, and $v_t^{language}$ has the same number of channels as the image feature map I_t. We use the same repertoire of filters for all frames in a video. Different from the general, static filters in a CNN, the dynamically generated filters can be thought of as filters specialized by and fine-tuned for the semantics of the lingual specification. For example, the target specification "a brown dog" will generate visual filters that are more specific to the "brownness" and the "dogness". This approach is, therefore, more flexible than [14], where a fixed repertoire of filters are learnt and then convolved with the concatenated linguistic and visual features.

After obtaining the generated dynamic filters, we convolve the augmented image feature map I_t:

$$A_t^{language} = v_t^{language} * I_t, \tag{2}$$

where $A_t^{language}$ is the response map for the frame f_t containing classification scores for each location in the feature map. The network is applied in a fully convolutional way over an input image.

To track the object over a sequence of frames the lingual specification network is applied repeatedly and for each frame independently, namely:

$$t = 0, \ldots, T : x_t^{language} = \arg\max_{r \in R} A_t^{language}(r), \tag{3}$$

where $A_t^{language}(r)$ is the output of the response map $A_t^{language}$ for region r and R are all the candidate locations for the target and T is the number of video frames. The tracking trajectory over time, therefore, is $x_t^{language}, t = 0, \ldots, T$.

Model I details. We employ the VGG-16 [32] as our fully convolutional network architecture for the input frame by treating `fc6`, `fc7` and `fc8` as convolutional layers. All the LSTM units have 1000-dimensional hidden states. Since there is no spatial extent encoded in the language expression, we generate 1×1 dynamic convolutional filters for $v_t^{language}$. The dynamic convolution is then performed on the feature maps from the `fc8` layer output. To enable training with a segmentation mask, we further upsample the response map $A_t^{language}$ to produce a response map which has the same size as the input image. The upsampling is implemented with a deconvolution layer using stride 32 [14]. During test, we also propose a bounding box location of the target in a video frame described by our language expression input. We use simple thresholding to first segment the regions of which the response value is above 50% of the max value in the response map. Then we take the bounding box that covers the largest connected component in the binary segmentation map.

Model II: Lingual First, then Visual Specification

The second model relies on the lingual specification for identifying the location of the target in the first frame. Then, the discovered target is used as the visual specification for a visual tracker, *e.g.*, [34, 9]. Thus, the first step for model I and model II is the same, by applying the *Lingual Specification Network* on the first frame, namely $x_{t=0}^{visual} = x_{t=0}^{language}$. Then, $x_{t=0}^{visual}$ is used to initialize a visual tracker, *i.e.*, the *Visual Specification Network* in Figure 2.

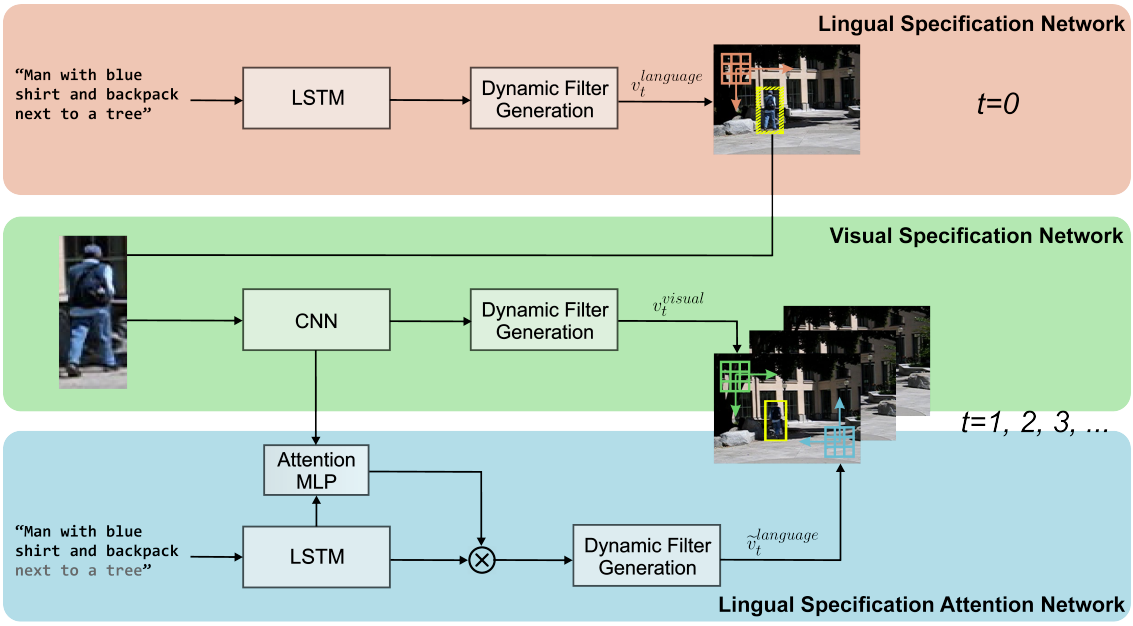

Figure 2: **Three models for our tracking by natural language specification**. In the first query frame ($t = 0$), all three models rely on the *Lingual Specification Network* to identify the target. An LSTM scans the text query and feeds a dynamic filter generation layer that produces novel visual filters to convolve the frame's feature map. In the following frames ($t = 1, 2, 3, ...$), ***Model I*** tracks the target by lingual specification only, independently applying the *Lingual Specification Network* on all frames. ***Model II*** takes the visual patch corresponding to the target identified from the first frame as input to the *Visual Specification Network*, which employs a CNN to dynamically generate the visual filters and convolves an input frame with the filters. ***Model III*** relies jointly on the lingual and visual specification. The visual specification utilizes the *Visual Specification Network*, while the lingual specification utilizes the *Lingual Specification Attention Network*, including an attention model that selectively focuses on parts of the lingual description.

Dynamic convolutional layer. Similar to the *Lingual Specification Network*, we also rely on a dynamic convolutional layer to generate filters regarding the visual target. However, instead of employing the target's language specification to generate the convolutional filters, following [34] we adopt a CNN to generate the visual features of the target as our filters, namely:

$$v_t^{visual} = CNN(B), \tag{4}$$

where $B = f_{t=0}(x_{t=0}^{language})$ is the image patch that corresponds to the location retrieved in the first frame by the lingual specification network. We choose not to update the visual model v_t^{visual} while the target may still appear differently over time. We rely on off-line training without any online-updates to handle visual changes of the target [34, 3]. After obtaining the filters we convolve the feature map I_t of the input frame f_t as follows:

$$A_t^{visual} = v_t^{visual} * I_t, \tag{5}$$

where A_t^{visual} is the response map for the frame f_t regarding to the visual target.

Model II details. The *Visual Specification Network* is also implemented as a fully convolutional network [3]. We use the VGG-16 [32] as our CNN architecture for both the input frame and the visual target B. They share the parameters in all the layers. We concatenate the feature maps from `conv3` and `conv4` outputs to generate the dynamic filters v_t^{viusal} and produce the features I_t for the input frame. A pooling layer is used after `conv3` to ensure the same feature map size. In the end, we compute the tracking trajectory of the object by:

$$\begin{aligned} t = 0 &: x_t^{visual} = x_t^{language} \\ t > 0 &: x_t^{visual} = \arg\max_{r \in R} A_t^{visual}(r) \end{aligned} \tag{6}$$

where $A_t^{visual}(r)$ is the output of the visual tracker's response map A_t^{visual} for region r.

Model III: Lingual and Visual Specification

The third model relies jointly on lingual and visual specification for tracking. The *Visual Specification Network* is once more initialized with the visual target identified in the first frame by the lingual specification. However, different

from model II, the lingual specification is also employed for the rest of the frames. In particular, the *Lingual Specification Attention Network* is utilized with an attention model that selectively focuses on parts of the lingual description, as illustrated in Figure 2.

Attention model. We start from the network architectures presented in the previous sections. We note, however, that the lingual specification originally describes the visual target in the first video frame only. Therefore, the lingual specification must be adapted over time, as the target text potentially has words that are not relevant for subsequent frames. For example, in the lingual specification "man with blue shirt and backpack next to a tree", see Figure 2, the specification "next to a tree" is irrelevant after the man has walked away. Therefore, we develop an attention model in the language tracking network to selectively focus on parts of the lingual specification about the visual target.

The attention model aims to attend the parts of the target's lingual specification that are more likely to be consistent throughout the video. Again we embed each word into a vector and use an LSTM network to generate the hidden states $h_i, i = 1, ..., K$ from the word sequence $W = (w_1, ..., w_K)$. Instead of using the hidden state at the final time step, we compute the representation of the lingual specification as a weighted sum of these hidden states:

$$\widetilde{s}_t = \sum_{i=1}^{K} \widetilde{\alpha}_i * h_i, \tag{7}$$

where the weights $\widetilde{\alpha}_i, i = 1, ..., K$ indicate the word importance. The weights are computed by a multi-layer perceptron conditioned on the hidden state h_i at each word position and the visual features z of the target B through CNN,

$$\begin{aligned} \alpha_i &= W_\alpha \phi(W_h h_i + W_z z + b) + b_\alpha \\ \widetilde{\alpha}_i &= P(i|h_i, z) = \frac{\exp(\alpha_i)}{\sum_{l=1}^{K} \exp(\alpha_l)} \end{aligned} \tag{8}$$

where ϕ is a rectified linear unit (ReLU) and the attention weights are also normalized using softmax. The attention weights are basically generated by matching the visual target with the word sequence at each word position. As a result, the words that relate to the target object properties rather than the context are more likely to be emphasized. For example, in Figure 2, over time the attention weights will focus more on the "man with blue shirt and backpack", while the "next to a tree" part of the target text query will be suppressed. Once we have the attention weighted representation $\widetilde{s}_t$ we generate target-specific filters $\widetilde{v}_t^{language}$ and produce the response map $\widetilde{A}_t^{language}$ by convolving the input image feature map I_t as in eq. 1, eq. 2.

Model III details. Again we produce the response map A_t^{visual} based on the visual target B derived from the lingual specification in the first frame. To obtain the final prediction, we first concatenate the response map from language $\widetilde{A}_t^{language}$ and visual target A_t^{visual}. Then the final response map is obtained by applying a 1×1 convolution on the stacked response maps, namely

$$A_t^{linguovisual} = \beta * [A_t^{visual}, \widetilde{A}_t^{language}], \tag{9}$$

which is essentially a weighted average of the stacked response maps. For model III we compute the tracking trajectory of the object by:

$$\begin{aligned} t = 0 &: x_t^{linguovisual} = x_t^{language} \\ t > 0 &: x_t^{linguovisual} = \arg\max_{r \in R} A_t^{linguovisual}(r) \end{aligned} \tag{10}$$

where $A_t^{linguovisual}(r)$ is the output of the visual tracker's response map $A_t^{linguovisual}$ for region r.

End-to-end Learning. All the network architectures presented are trained end-to-end with video frames. Suppose we obtain the final response map A, where A can be either of the $\{A^{language}, A^{visual}, A^{linguovisual}\}$, and the binary ground truth label Y for an input frame. The loss function for a training sample is defined as the average over all the response map locations:

$$\mathcal{L} = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} \mathcal{L}_{ij}, \tag{11}$$

where W and H are the width and height of the response map. $\mathcal{L}$ is the logistic loss defined as follows:

$$\mathcal{L}_{ij} = \log(1 + \exp(-A_{ij} Y_{ij})). \tag{12}$$

When we have the ground truth segmentation mask, the binary label Y_{ij} indicates a binary label at pixel (i, j). For tracking, however, where we do not have pixel-labels at our disposal, we calculate the logistic loss over the response map where each entry corresponds to a bounding box in the original image. An entry is considered positive if the intersection over union between its corresponding box and ground truth box is larger than a threshold τ ($\tau = 0.7$).

4. Experiments

4.1. Datasets

Lingual OTB99. The popular OTB100 [37] object tracking dataset contains 100 videos of a target being tracked, with 51 originating from OTB51 [38]. As we are interested in tracking by natural language specification, we augment the videos in OTB100 with natural language descriptions of the target object. Following the guidelines in [19] we ask annotators for a discriminative referring description of the target. For fairness the annotators describe the target based on the first frame only. We extend the OTB100 set with one sentence description of each target per video. As the annotators

	P@0.5	*P@0.6*	*P@0.7*	*P@0.8*	*P@0.9*	*overall IoU*
Hu *et al*. [14]	34.0	26.7	19.3	11.6	3.9	48.0
Hu *et al*. [15] (from [14])	11.9	7.7	4.3	1.5	0.3	17.8
Rohrbach *et al*. [30] (from [14])	14.1	9.6	5.8	2.7	0.6	20.1
This paper: Lingual specification network	**38.6**	**31.3**	**23.3**	**14.8**	**5.9**	**54.3**

Table 1: **Target identification by lingual specification.** We evaluate our lingual specification network in the task of segmentation by natural language expressions [14] with the precision at different overlap thresholds and the overall IoU.

could not describe one video, we arrive at Lingual OTB99. The OTB51 videos are kept for fine-tuning and the other 48 for testing the results.

Lingual ImageNet Videos. We start from the recently introduced ImageNet Video Object Detection dataset [31] by selecting 4 videos for each of the 25 object categories. We then augment the 100 videos following similar steps as for Lingual OTB99. We ask annotators to return a query description of the target object on the first frame in the video. Again we use 50 videos for fine-tuning and the other 50 for reporting results.

ReferIt [19]. The ReferIt dataset is proposed in [19] for the task of object localization and segmentation by natural language expression. It is the largest publicly available dataset that contains natural language expressions annotated on segmented regions. It contains about 20,000 images and 130,525 expressions annotated on 96,654 segmented image regions. We follow [14] and use 10,000 images for training and validation and 10,000 images for testing.

4.2. Implementation Details

Training. To train the lingual specification network, we first pre-train the network on the ReferIt [19] dataset using segmentation masks, since language queries from Lingual OTB99 and Lingual ImageNet Videos are still limited. For the visual specification network, instead of using the full image as input, we follow [3] to crop a large search region around the center of the target box location. The network is initialized from the pre-trained model on the ImageNet classification task [31]. We fine-tune it using the training videos from Lingual OTB99 or Lingual ImageNet Videos. Similarly, our joint model is also fine-tuned based on pre-trained networks using the ReferIt [19] and ImageNet classification datasets. The parameters of all the networks are all trained with a standard SGD solver with momentum.

Evaluation criteria. Following the standard protocol in OTB51 [38] we report our tracking performance on all the datasets with the AUC (area under the curve) score metric.

4.3. Target Identification by Lingual Specification

We first assess the ability of the lingual specification network for target identification. As target identification in a single frame resembles the task of segmentation by natural language expression, we evaluate the task following the protocol of Hu *et al*. [14] on ReferIt [19] and compare with other state-of-the-art approaches, see Table 1.

The lingual specification network of our model results in $2.0 - 4.6\%$ higher precision for all overlap thresholds compared to [14]. It also obtains 6.3% more accurate overall IoU, which is defined as the intersection area divided by the union area, where both are summed over all the test samples. We observe that our lingual specification network that generates visual filters dynamically is stronger on visually and semantically richer images. This hints that the lingual specification network generalizes better than [14] from the examples seen in the training set. We conclude that our lingual specification network allows for state-of-the-art target localization based on natural language descriptions.

4.4. Tracking by Natural Language Specification

In this experiment we evaluate our three models for tracking by natural language specification from Section 3. We discard the user-specified bounding box in the first frame and all models rely only on the text query to track the target.

We present our results on Lingual OTB99 in Figure 3. In this plot the videos are ranked along the y-axis according to the accuracy of the target identification in the first frame. As shown in Figure 3, Model II (Lingual first, then visual specification) and Model III (Lingual and visual specification) generally perform better than Model I (Lingual specification only) in videos where the initial target identification is precise, whereas Model I is more accurate when the initial target identification is poor.

Model I, relying on lingual specification only, has difficulties in handling the scenarios where multiple semantically close objects are present or when some part of the language description, such as the spatial relationship, is no longer relevant. In Model II and Model III, the visual tracker, when well initialized given a precise target identification in the initial frame, enables to tackle the above scenarios better. See Figure 4 for some examples. Therefore, when the target identification in the initial frame is precise, Model II and Model III are often more accurate than Model I. However, when the target identification by the lingual specification in the first frame is not good enough, the

	Model I	*Model II*	*Model III*
Lingual ImageNet	26.3	23.3	23.4

Table 2: **Performance of our three models for tracking by language specification** on Lingual ImageNet Videos.

visual tracker in Model II and Model III initialized by the target specification also fails. Even worse, it has a negative, cumulative effect on subsequent frames. In contrast, model I that tracks by lingual specification only has no negative cumulative effect, as each frame is treated independently. Hence, Model I works better when the initial target specification is poor.

Model III is generally better than Model II. Model II may easily lose the target when the background is cluttered or the target initialization in the first frame contains extra background pixels. In contrast to Model II, the lingual tracking component in Model III can utilize the semantic information carried by the language expression to address, to some extent, the cases with clutter background and inaccurate target initialization.

Results for the Lingual ImageNet Videos are shown in Table 2. Note that ImageNet videos are visually more constrained than OTB100 videos with respect to tracking variations. Moreover, in ImageNet videos, the target of interest is often in the center of the camera view. As a result, the language description given based on the initial frame, including the spatial context information, often holds for a large portion of the sequence, and it makes tracking by lingual specification only usually suffice.

As a general conclusion, the joint tracking model by lingual and visual specification works better when the target identification by the lingual specification in the first frame is good. Otherwise, the tracking by lingual specification only is advantageous.

4.5. Tracking by Language and Box Specification

In the next experiment we update our model II and model III. Instead of inferring the target location in the first frame by lingual specification, we use a user-specified bounding box as our visual specification, as in a standard visual object tracking setting, namely $x^{user-box}$. For the remaining frames, we rely on this predefined visual specification to initialize the visual tracker in model II (box specification), as well as the joint tracking by lingual and visual specification in mode III (language and box specification). We also compare with our model I in which only the language specification is used.

We show the evaluations using AUC scores in Table 3. A user-specified sentence in combination with a user-specified bounding box brings an improvement from 56.1% to 57.8%

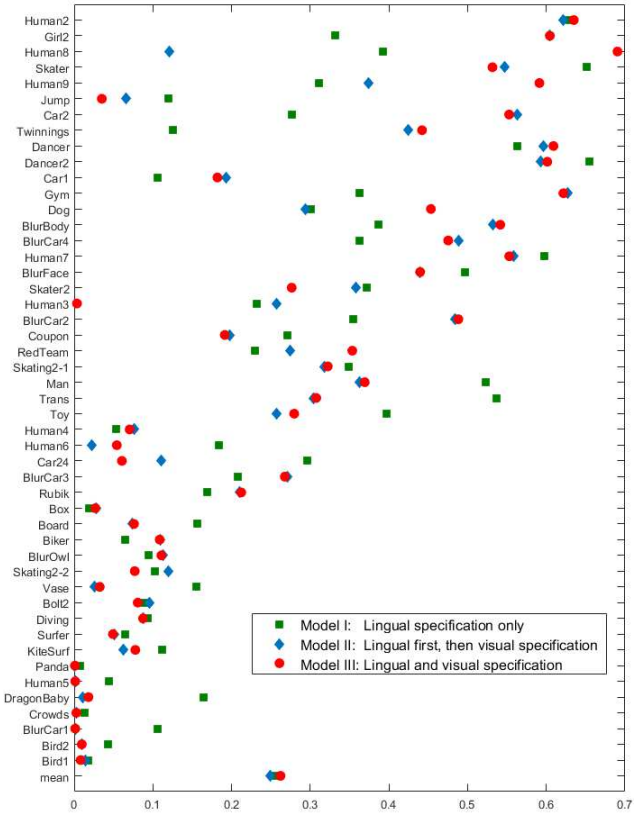

Figure 3: **Performance of our three models for tracking by language specification** on Lingual OTB99. Videos are ranked by target identification results in the first frame. When the target identification in the first frame is accurate (upper half), joint tracking by lingual and visual specification usually outperforms the other models. When the target identification is poor (bottom half), tracking by lingual specification only is better in general.

	Language specification	*Box specification*	*Language and box specification*
Lingual OTB99	25.9	56.1	57.8
Lingual ImageNet	26.3	47.9	49.4

Table 3: **Tracking by language and box specification**.

on Lingual OTB99 and from 47.9% to 49.4% on Lingual ImageNet. When inspecting qualitative results in Figure 5, we observe that the tracking by language component helps against drifting. In the top row the skater deforms her shape too fast for the bounding box to adapt, a problem that is alleviated when adding tracking by language. In the bottom row the bounding box confuses the target pedestrian with a white pole having as a consequence the target to be lost. Combining the bounding box with tracking by language, correctly grounds the target girl till the end, despite the ex-

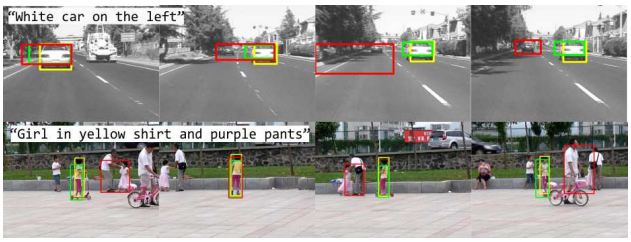

Figure 4: **Examples of tracking by natural language specification.** *Ground truth*, *Model I: Tracking by lingual specification only*, *Model III: Joint tracking by lingual and visual specification*. In the top row the language-only model gets confused because another vehicle is present and because the spatial description of the query, "on the left", is no longer valid. In the bottom row the language-only model confuses the target, a little girl, with other persons. The joint model is more robust in both cases.

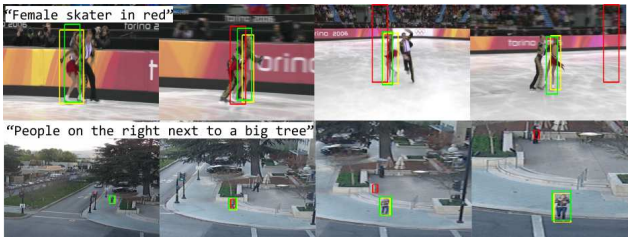

Figure 5: **Adding user-specified bounding box to tracking by natural language specification**. *Ground truth*, *Tracking by box specification only, similar to SINT [34]*, *Joint tracking by language and box specification*.

treme scale change. We conclude that when a user-specified bounding box is available adding language specification arrives at more robust tracking that better tackles accidental drifting.

4.6. Enabling Novel Tracking Scenarios

A unique property of tracking by language is that the same target text query may apply to multiple videos. This comes in stark contrast with standard tracking, where the user defines the target in each video separately. What is more, tracking by natural language specification does not need a "first frame" where the query is defined. This is relevant for live streaming videos where the user would otherwise need to attend all frames to set the target. We demonstrate qualitatively these two novel applications in two videos, in Figure 6. We use the same query for multiple videos and add irrelevant frames before they start. The algorithm is capable of tracking a man with blue pants in both videos, which appears for the first time at frame 25 for video 1 and frame 45 for video 2. Indeed, tracking by language specification allows to track in multiple videos, and

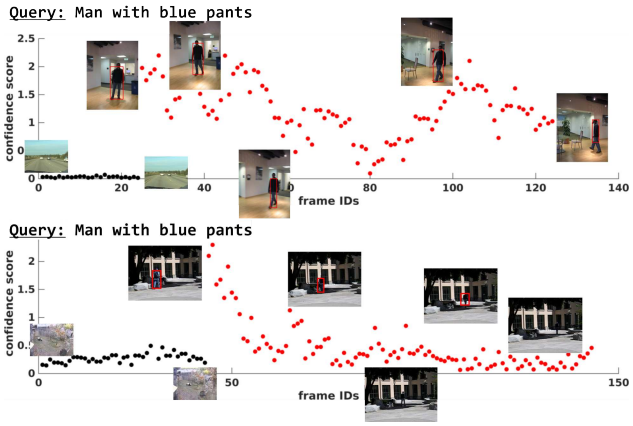

Figure 6: **Novel applications**: *I) Tracking targets in multiple videos simultaneously.* With standard tracking, each new video requires specification of the target. In tracking by language "Man with blue pants" applies to all relevant videos, certainly when running at the same time. *II) Start tracking at arbitrary timestamps.* A standard tracker cannot be directly employed on an arbitrary video, as the user needs to first browse through frames to find the target of interest. Our tracking can be initialized by a lingual description, triggering a process to start the tracking once a suitable target appears. Both applications are ideal for tracking in live-surveillance.

starts tracking from arbitrary frames. Both are scenarios where standard trackers cannot offer a natural solution. We conclude that tracking by language specification paves the way towards novel applications in visual object tracking.

5. Conclusion

We present tracking by natural language specification as an alternative to tracking by human-provided bounding box specification. We show how such tracking can be realized by presenting three models founded on a common neural network architecture. We extended two well-known tracking datasets with sentences describing the target of interest, to show the potential of the three models for tracking by natural language specification. Our experiments indicate the ability of the lingual target specification in determining the target location, investigate the trade-off between lingual, visual and joint specification of the target when the initial box prediction is less reliable, and we also show how traditional tracking with human-provided bounding box can be enhanced by the use of language. Finally, we sketch new tracking scenarios in surveillance and other live video streams that become feasible with a lingual specification of the target.

Acknowledgments This research is partly supported by the STW STORY project.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015. 2

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2

[3] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV workshop*, 2016. 1, 2, 4, 6

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 2

[5] J. Choi, H. Jin Chang, J. Jeong, Y. Demiris, and J. Young Choi. Visual tracking using attention-modulated disintegration and integration. In *CVPR*, 2016. 2

[6] Z. Cui, S. Xiao, J. Feng, and S. Yan. Recurrently target-attending tracking. In *CVPR*, 2016. 2

[7] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015. 2

[8] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014. 2

[9] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, 2016. 3

[10] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014. 2

[11] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, 2016. 1

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 2015. 2

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 1997. 2

[14] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. In *ECCV*, 2016. 1, 2, 3, 6

[15] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016. 1, 2, 6

[16] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guided long-short term memory for image caption generation. In *ICCV*, 2015. 2

[17] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *TPAMI*, 2012. 1

[18] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 2

[19] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 5, 6

[20] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernández, G. Nebehay, F. Porikli, and L. Cehovin. A novel performance evaluation methodology for single-target trackers. *TPAMI*, 2016. 1, 2

[21] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Deep learning for question answering. In *CVPR*, 2011. 2

[22] R. Li and J. Jia. Visual question answering with question representation update (QRU). In *NIPS*, 2016. 2

[23] Z. Li, E. Gavves, T. Mensink, and C. Snoek. Attributes make sense on segmented objects. In *ECCV*, 2014. 2

[24] S. Liu, T. Zhang, X. Cao, and C. Xu. Structural correlation filter for robust visual tracking. In *CVPR*, 2016. 2

[25] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *CVPR*, 2015. 2

[26] J. Mao, H. Jonathan, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 1

[27] M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. C. Berg, T. L. Berg, and H. Daumé III. Midge: Generating image descriptions from computer vision detections. In *EACL*, 2012. 2

[28] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 2

[29] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2

[30] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *ECCV*, 2016. 6

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2, 6

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 2, 3, 4

[33] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *TPAMI*, 2014. 1, 2

[34] R. Tao, E. Gavves, and A. W. M. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016. 1, 2, 3, 4, 8

[35] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2014. 2

[36] L. Wang, W. Ouyang, X. Wang, and H. Lu. STCT: Sequentially training convolutional networks for visual tracking. In *CVPR*, 2016. 2

[37] T. Wu, Y. Lu, and S. Zhu. Online object tracking, learning and parsing with and-or graphs. *TPAMI*, 2015. 1, 5

[38] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 1, 2, 5, 6

[39] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[40] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *arXiv*, 2015. 2