A Corpus-Guided Framework for Robotic Visual Perception

Ching L. Teo, Yezhou Yang, Hal Daumé III, Cornelia Fermüller, Yiannis Aloimonos
University of Maryland Institute for Advanced Computer Studies, College Park, MD 20742-3275
{*cteo, yzyang, hal, fer, yiannis*}*@umiacs.umd.edu*

Abstract

We present a framework that produces sentence-level summarizations of videos containing complex human activities that can be implemented as part of the Robot Perception Control Unit (RPCU). This is done via: 1) detection of pertinent objects in the scene: tools and direct-objects, 2) predicting actions guided by a large lexical corpus and 3) generating the most likely sentence description of the video given the detections. We pursue an active object detection approach by focusing on regions of high optical flow. Next, an iterative EM strategy, guided by language, is used to predict the possible actions. Finally, we model the sentence generation process as a HMM optimization problem, combining visual detections and a trained language model to produce a readable description of the video. Experimental results validate our approach and we discuss the implications of our approach to the RPCU in future applications.

Introduction

Robot perception has been a well researched problem both in Robotics and Artificial Intelligence. In this paper, we focus on the *visual perception* problem: how can one enable a robot to make sense of its visual environment. During the past few years, with the development of statistical machine learning techniques, several data-driven detection methods were used as a basic *Robot Perception Unit* (RPU) – a place in the robot's Operating System (OS) that performs visual processing such as detecting/recognizing objects, actions and scenes.

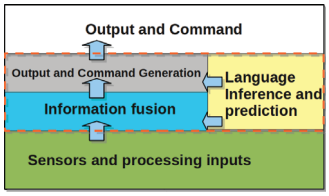

Figure 1: Overview of the proposed robot perception system, which includes the bottom-layer inputs module, RPCU (in the dashed box) and the top-layer output module.

This RPU, however, is lacking in some aspects, more specifically in providing high-level information concerning the visual input. A more sophisticated RPU should be able to 1) fuse (noisy) information from various sensors and processing inputs, 2) perform inference and predictions and 3) eventually generate a useful output or command that show that the robot has truly perceived the world with all its complexity and richness. From a systems point of view, this represents a further step of refinement over the basic RPU: where we endow it with a *control unit* which we will term as the *Robot Perception Control Unit* (RPCU) in the rest of the paper. From Fig. 1, we see that the RPCU is the central component of the overall Robotic Perception System, consisting of a bottom-layer that feeds it with input detections (visual, sonar or inertial etc.), and a top-layer that uses the outputs of the RPCU to perform other tasks (feedback or response). In this paper, we focus on the RPCU's design and implementation specifically for video inputs of human actions.

There are obviously numerous ways to realize the RPCU, all which entail numerous challenges. The most crucial challenge is *grounding* the visual inputs to semantically meaningful units: getting from a patch of pixel values to identifying the object itself. We show that by adding *language*, learned from a large generic corpus, we are able to produce a reasonable grounding that enables the RPCU to handle noisy detections and make reasonable predictions. Our proposed framework for the RPCU contains the following key novel elements (summarized in Fig. 1):

- Using Language: we use language as a prior in guiding the RPCU so as to handle noisy inputs and make reasonable predictions. We believe this approach mimics how we as humans perceive the world, where vast amounts of high-level knowledge acquired over our lives allows us to infer and recognize complex visual inputs. This knowledge is encoded in various forms, of which language is clearly the most predominant. Language manifests itself in the form of text which is also extremely accessible from various large research corpora.
- Information Fusion: we use current state of the art object detectors to detect hands, tools and direct-objects (objects that are manipulated by the tool) as initial hypothesis to predict actions in an iterative manner using an Expectation-Maximization (EM) formulation, with the noisy visual detections supporting the (equally noisy) proposed action model.

- Sentence (Output) Generation: the proposed RPCU is able to generate a human-readable sentence that summarizes the visual inputs that it has received, essentially grounding the visual inputs to a high-level semantic description. This is achieved by modeling the sentence generation process as a Hidden Markov Model (HMM) and performing Viterbi decoding to produce the most likely sentence, given the visual detections and a language model. Such summaries are extremely useful and can be used by further downstream modules for further processing, analysis, storage or even enable the robot to interact with us in a more natural way (via speech).

We first describe how the hand, tool and direct-objects are detected actively. We then describe how the three key elements of the proposed RPCU described above are implemented and combined. We validate the proposed RPCU by generating sentence-level summarizations of input test videos containing daily human activities. Finally, we conclude with a discussion of the results and its implications for robotic perception in general.

Hand, Tool and Object Detections

Active object detection strategy

We pursue the following active strategy as illustrated in Fig. 2 for detecting relevant tools and direct-objects $n_i \in N$ from the input video, $m_d \in M$. M is the set of all videos considered and $\mathcal{N}_1 \subset N$ and $\mathcal{N}_2 \subset N$ are the sets of possible tools and direct-objects respectively with N as the set of all objects considered. First, a trained person detector (Felzenszwalb et al. 2010) is used to determine the location of the human actor in the video frame. The location of the face is also detected using (Viola and Jones 2004). Optical flow is then computed (Brox, Bregler, and Malik 2009) and we focus on human regions which have the highest flow, indicating the potential locations of the hands. We then apply a variant of a CRF-based color segmentation (Rother, Kolmogorov, and Blake 2004) using a trained skin color+flow model to segment the hand-like regions which are moving. This is justified by the fact that the moving hand is in contact with the tool that we want to identify. In some cases, the face may be detected (since it may be moving) but they are removed using the face detector results. We then apply a trained object detector (Schwartz et al. 2009) near the detected active hand region that returns a detection score at each video frame. In order to detect the direct-object, we compute the midpoint of the two hand regions where the direct-object is likely to be found and applied same trained object detector as above to determine the direct-object's identity. Averaging out the detection yields $\mathcal{P}_I(n_i|m_d)$, the probability that a tool or direct-object n_i exists given the video m_d. Specially for tools, we denote $\mathcal{P}_I(\mathcal{N}_1|M)$ as the set of likelihood scores over all tools in $\mathcal{N}_1$ and all videos in M.

This active approach has two important benefits. By focusing our processing only on the relevant regions of the video frame, we dramatically reduce the chance that the object detector will misfire. At the same time, by detecting the hand locations, we obtain immediately the action trajectory,

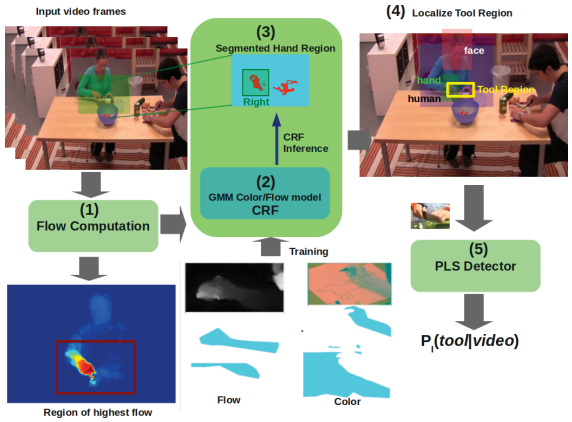

Figure 2: Overview of the tool detection strategy: (1) Optical flow is first computed from the input video frames. (2) We train a CRF segmentation model based on optical flow + skin color. (3) Guided by the flow computations, we segment out hand-like regions (and removed faces if necessary) to obtain the hand regions that are moving (the active hand that is holding the tool). (4) The active hand region is where the tool is localized. Using the PLS detector (5), we compute a detection score for the presence of a tool. Direct object detection follows a similar strategy (see text).

which is used to describe the action as shown in the next section.

Action features

Tracking the hand regions in the video provides us with two sets of (left and right) hand trajectories as shown in Fig. 3. We then construct for every video a feature vector F_d that encodes the hand trajectories. F_d encodes the frequency and velocity components. Frequency is encoded by using the first 4 real components of the 1D Fourier Transform in both the x and y directions, f_x, f_y, which gives a 16-dim vector over both hands. Velocity is encoded by averaging the difference in hand positions between two adjacent frames $\langle \delta x \rangle$, $\langle \delta y \rangle$ which gives a 4-dim vector. These features are then combined to yield a 20-dim vector F_d.

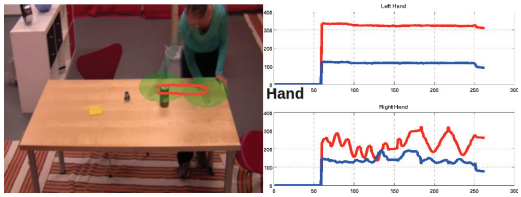

Figure 3: Detected hand trajectories. x and y coordinates are denoted as red and blue curves respectively.

We denote F_M as the set of of action features F_d over all videos in M.

The Robot Perception Control Unit

The Language Model

The first key component of the RPCU is the language model that predicts the most likely verb (action) that is associated

with a noun (tool or direct-object) trained from a large text corpus: the English Gigaword (Graff 2003). We view the Gigaword Corpus as a large text resource that contains the information we need to make correct predictions of actions given the detected tools from the video and the associated direct-object given the action. Denoting $v_j \in V$ as an action label from the set of admissible actions V, we train two related language models. The first model returns the maximum likelihood estimates of an action v_j given the tool $n_i \in \mathcal{N}_1$: $\mathcal{P}_L(v_j|n_i)$, and the second model returns the most-likely direct-object $n_i \in \mathcal{N}_2$ given the action v_j: $\mathcal{P}_L(n_i|v_j)$. This can be done by counting the number of times v_j co-occurs with n_i in a sentence:

$$\mathcal{P}_L(v_j|n_i) = \frac{\#(v_j, n_i)}{\sum_i \#(n_i)}, \mathcal{P}_L(n_i|v_j) = \frac{\#(v_j, n_i)}{\sum_j \#(v_j)} \quad (1)$$

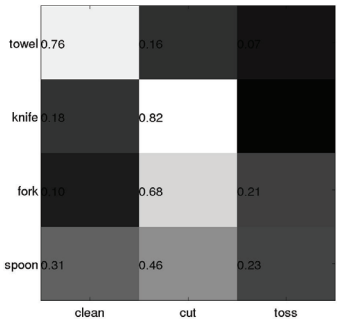

Figure 4: The Gigaword co-occurrence matrix for tools and predicted actions.

Fig. 4 shows the set of the $|\mathcal{N}_1| \times |V|$ co-occurrence matrix of likelihood scores over all tools and actions considered in the experiments, denoted as $\mathcal{P}_L(V|\mathcal{N}_1)$.

Information Fusion: Predicting Actions

Given the noisy video detections described in the previous section, the second key capability of the proposed RPCU is to combine them in some reasonable manner, using the trained language model to predict the action that is occurring. Formally, our goal is to label each video with their most likely action, along with the tool[1] that is associated with the action using an EM formulation. That is, we want to maximize the likelihood:

$$\begin{aligned} L(\mathcal{D}; A) &= \mathbb{E}_{\mathcal{P}(A)}[L(\mathcal{D}|A)] \\ &= \mathbb{E}_{\mathcal{P}(A)}[log\mathcal{P}(F_M, \mathcal{P}_I(\cdot), \mathcal{P}_L(\cdot)|A)] \end{aligned} \quad (2)$$

where A is the current (binary) action label assignments of the videos (see eq. (3)). $\mathcal{D}$ is the data computed from the video that consists of: 1) the language model $\mathcal{P}_L(\cdot)$ that predicts an action given the detected tool, 2) the tool detection model $\mathcal{P}_I(\cdot)$ and 3) the action features, F_M, associated with the video.

We first define the latent assignment variable A. To simplify our notations, we will use subscripts to denote tools $i = n_i$, actions $j = v_j$ and videos $d = m_d$. For each $i \in \mathcal{N}_1$, $j \in V$, $d \in M$, A_{ijd} indicates whether an action j is performed using tool i during video clip d.

$$A_{ijd} = \begin{cases} 1 & j \text{ is performed using } i \text{ during } d \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

and A is a 3D indicator matrix over all tools, actions and videos. Denoting the parameters of the model as $\mathcal{C} = \{\mathcal{C}_j\}$ which specifies the grounding of each action j, we seek to determine from eq. (2) the maximum likelihood parameter:

$$\mathcal{C}^* = \arg\max_{\mathcal{C}} \sum_A L(\mathcal{D}, A|\mathcal{C}) \quad (4)$$

Where,

$$\begin{aligned} L(\mathcal{D}, A|\mathcal{C}) &= \log P(\mathcal{D}, A|\mathcal{C}) \\ &= \log P(A|\mathcal{D}, \mathcal{C}) P(\mathcal{D}|\mathcal{C}) \end{aligned} \quad (5)$$

with the data $\mathcal{D}$ comprised of the tool detection likelihoods $\mathcal{P}_I(\mathcal{N}_1|M)$, the tool-action likelihoods $\mathcal{P}_L(V|\mathcal{N}_1)$ and action features F_M under the current model parameters $\mathcal{C}$. Geometrically, we can view $\mathcal{C}$ as the superset of the $|V|$ action label centers that defines our current grounding of each action j in the action feature space.

Using these centers, we can write the assignment given *each* video d, tool i and action j, $P(A_{ijd}|\mathcal{D}, \mathcal{C})$ as:

$$\mathcal{P}(A_{ijd} = 1|\mathcal{D}, \mathcal{C}) = \mathcal{P}_I(i|d)\mathcal{P}_L(j|i)Pen(d|j) \quad (6)$$

where $Pen(d|j)$ is an exemplar-based likelihood function defined between the associated action feature of video d, F_d and the current model parameter for action j, $\mathcal{C}_j$ as:

$$Pen(d|j) = \frac{1}{Z} \exp^{-||F_d - \mathcal{C}_j||^2} \quad (7)$$

where Z is a normalization factor. What eq. (7) encodes is the penalty that we score against the assignment when there is a large mismatch between F_d and $\mathcal{C}_j$, the cluster center of action j.

Rewriting eq. (6) over *all* videos M, tools $\mathcal{N}_1$ and actions V we have:

$$\mathcal{P}(A = 1|\mathcal{D}, \mathcal{C}) = \mathcal{P}_I(\mathcal{N}_1|M)\mathcal{P}_L(V|\mathcal{N}_1)Pen(F_M|C) \quad (8)$$

where we use the set variables to represent the full data and assignment model parameters. In the derivation that follows, we will simplify $\mathcal{P}(A = 1|\mathcal{D}, \mathcal{C})$ as $\mathcal{P}(A = 1)$ and $\mathcal{P}(A = 0) = 1 - \mathcal{P}(A = 1)$. We detail the Expectation and Maximization steps in the following sections.

Expectation step We compute the expectation of the latent variable A, denoted by $\mathcal{W}$, according to the probability distribution of A given our current model parameters $\mathcal{C}$ and data ($\mathcal{P}_I$, $\mathcal{P}_L$, and F_M):

$$\begin{aligned} \mathcal{W} &= \mathbb{E}_{\mathcal{P}(A)}[A] \\ &= \mathcal{P}(A = 1) \times 1 + (1 - \mathcal{P}(A = 1)) \times 0 \\ &= \mathcal{P}(A = 1) \end{aligned} \quad (9)$$

According to Eq. 6, the expectation of $\mathcal{A}$ is:

$$\mathcal{W} = \mathcal{P}(A = 1) \propto \mathcal{P}_I(\mathcal{N}_1|M)\mathcal{P}_L(V|\mathcal{N}_1)Pen(F_M|\mathcal{C}) \quad (10)$$

Specifically, for each $i \in \mathcal{N}_1, j \in V, d \in M$:

$$\mathcal{W}_{ijd} \propto \mathcal{P}_I(i)\mathcal{P}_L(j|i)Pen(d|j) \quad (11)$$

Here, $\mathcal{W}$ is a $|\mathcal{N}_1| \times |V| \times |M|$ matrix. Note that the constant of proportionality does not matter because it cancels out in the Maximization step.

[1] Direct-objects can be used as well but are not considered here.

Maximization step The maximization step seeks to find the updated parameters $\hat{\mathcal{C}}$ that maximize eq. (5) with respect to $\mathcal{P}(A)$:

$$\hat{\mathcal{C}} = \arg\max_{\mathcal{C}} \mathbb{E}_{\mathcal{P}(A)}[\log \mathcal{P}(A|\mathcal{D},\mathcal{C})\mathcal{P}(\mathcal{D}|\mathcal{C})] \tag{12}$$

Where $\mathcal{D} = \mathcal{P}_I, \mathcal{P}_L, F_M$. EM replaces $\mathcal{P}(A)$ with its expectation $\mathcal{W}$. As $A, \mathcal{P}_I, \mathcal{P}_L$ are independent of the model parameters $\mathcal{C}$, we can simplify eq. (12) to:

$$\begin{aligned}\hat{\mathcal{C}} &= \arg\max_{\mathcal{C}} \mathcal{P}(F_M|\mathcal{C}) \\ &= \arg\max_{\mathcal{C}} \left(-\sum_{i,j,d} \mathcal{W}_{ijd} ||F_d - \mathcal{C}_j||^2 \right)\end{aligned} \tag{13}$$

where we had replaced $\mathcal{P}(F_M|\mathcal{C})$ with eq. (7) since the relationship between F_M and $\mathcal{C}$ is the penalty function $Pen(F_M|\mathcal{C})$. This enables us to define a target maximization function as $\mathbb{F}(\mathcal{C}) = \sum_{i,j,d} \mathcal{W}_{ijd} ||F_d - \mathcal{C}_j||^2$.

According to the Karush-Kuhn-Tucker conditions, we can solve the maximization problem by the following constraint:

$$\frac{\partial \mathbb{F}}{\partial \mathcal{C}} = -2 \sum_{i,j,d} (\mathcal{W}_{ijd}(F_d - \mathcal{C}_j)) = 0 \tag{14}$$

Thus, for each $j \in V$, we have:

$$\hat{\mathcal{C}}_j = \frac{\sum_{i\in\mathcal{N}_1, j\in V, d\in M} \mathcal{W}_{ijd} F_d}{\sum_{i\in\mathcal{N}_1, j\in V, d\in M} \mathcal{W}_{ijd}} \tag{15}$$

We then update $\mathcal{C} = \hat{\mathcal{C}}$ within each iteration until convergence.

Action Prediction Using the learned model $\mathcal{C}^*$, the conditional probability distribution of each action j given the input video d, $\mathcal{P}_I(j|d)$, can be computed by:

$$\begin{aligned}\mathcal{Z} &= \sum_{j\in V} \sum_{i\in\mathcal{N}_1} \left(\mathcal{P}_I(i|d)\mathcal{P}_L(j|i)Pen(F_t|\mathcal{C}_j^*)\right) \\ \mathcal{P}_I(j|d) &= \frac{\sum_{i\in\mathcal{N}_1} \left(\mathcal{P}_I(i|d)\mathcal{P}_L(j|i)Pen(F_t|\mathcal{C}_j^*)\right)}{\mathcal{Z}}\end{aligned} \tag{16}$$

where F_t is the action features extracted from d and $\mathcal{C}_j^*$ is the j^{th} action center from the learned model. Replacing our short-hand notations for $v_j = j$ and $d = m_d$, the pdf computed above is denoted as $\mathcal{P}_I(v_j|m_d)$.

Sentence Generation

The final key component of the proposed RPCU is to generate a reasonable sentence that summarizes the input video. In order to do this we define the sentence to be generated in terms of its *core components* by a triplet, $\mathcal{T} = \{n_1, v, n_2\}$ where $n_1 \in \mathcal{N}_1$, $n_2 \in \mathcal{N}_2$ refer to any tools and direct-objects detected previously from the input video m with v as the predicted action label from the EM formulation described above. We have dropped all subscripts i, j, d as we are not concerned with any particular object, action or video here. Using $\mathcal{T}$, we generate a sentence that summarizes the input video using a pre-defined sentence template.

Given the computed conditional probabilities: $\mathcal{P}_I(n_1|m)$, $\mathcal{P}_I(n_2|m)$ and $\mathcal{P}_I(v|m)$ (eq. (16)) which are observations from the input video with the parameters of the trained language model: $\mathcal{P}_L(v|n_1), \mathcal{P}_L(n_2|v)$ (eq. (1)), we seek to find the most likely sentence structure $\mathcal{T}^* = (n_1, v, n_2)$ by:

$$\begin{aligned}\mathcal{T}^* &= \arg\max_{n_1,v,n_2} \mathcal{P}(\mathcal{T}|n_1, v, n_2) \\ &= \arg\max_{n_1,v,n_2} \{\mathcal{P}_I(n_1|m)\mathcal{P}_I(n_2|m)\mathcal{P}_I(v|m)\times \\ &\qquad \mathcal{P}_L(v|n_1)\mathcal{P}_L(n_2|v)\}\end{aligned} \tag{17}$$

where the last equality holds by assuming independence between the visual detections and corpus predictions. Obviously a brute force approach to try all possible combinations to maximize eq. (17) will not be feasible due to the potentially large number of possible combinations. A better solution is needed.

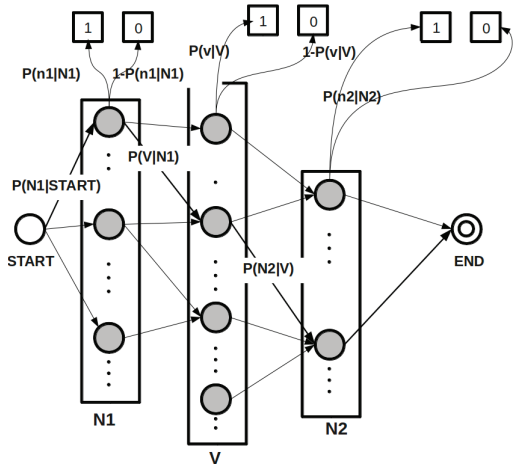

Figure 5: The HMM used for optimizing $\mathcal{T}$. The relevant transition and emission probabilities are also shown. See text for more details.

Our proposed strategy is to pose the optimization of $\mathcal{T}$ as a dynamic programming problem, akin to a Hidden Markov Model (HMM) where the hidden states are related to the sentence structure we seek: $\mathcal{T}$, and the emissions are related to the observed detections: $\{n_1, v, n_2\}$ in the video if they exist. The hidden states are therefore denoted as: $\{\texttt{N1}, \texttt{V}, \texttt{N2}\}$

Tools $n_1 \in \mathcal{N}_1$	Actions $v \in V$	Direct-objects $n_2 \in \mathcal{N}_2$
'towel' 'knife'	'clean' 'cut'	'table' 'cheese'
'fork' 'spoon'	'toss'	'tomato' 'salad'

Table 1: The set of tools, actions and direct-objects considered.

with values taken from their respective word classes from Table 1. The emission states are $\{\texttt{n1},\texttt{v},\texttt{n2}\}$ with binary values: 1 if the detections occur or 0 otherwise. The full HMM is summarized in Fig. 5. The rationale for using a HMM is that we can reuse all previous computation of the probabilities at each level to compute the required probabilities at the current level. From START, we assume all tool detections are equiprobable: $\mathcal{P}(\texttt{N1}|\texttt{START}) = \frac{1}{|\mathcal{N}_1|}$. At each N1, the

HMM emits a detection from the video and by independence we have: $\mathcal{P}(\texttt{n1}|\texttt{N1}) = \mathcal{P}_I(n_1|m)$. After N1, the HMM transits to the corresponding verb at state V with $\mathcal{P}(\texttt{V}|\texttt{N1}) = \mathcal{P}_L(v|n_1)$ obtained from the first language model. Similarly, V has emissions $\mathcal{P}(\texttt{v}|\texttt{V}) = \mathcal{P}_I(v|m)$. The HMM then transits from V to N2 with $\mathcal{P}(\texttt{N2}|\texttt{V}) = \mathcal{P}_L(n_2|v)$ computed from the second language model which emits the direct-object detection score from the video: $\mathcal{P}(\texttt{n2}|\texttt{N2}) = \mathcal{P}_I(n_2|m)$.

Comparing the HMM with eq. (17), one can see that all the corpus and detection probabilities are accounted for in the transition and emission probabilities respectively. Optimizing $\mathcal{T}$ is then equivalent to finding the best (most likely) path through the HMM given the video observations using the Viterbi algorithm which can be done significantly faster than the naive approach.

The computed $\mathcal{T}^*$ is then used to generate a sentence of the form V−N2−N1 which represents the generation template common in standard language generation work. To form readable sentences, standard English grammar rules and syntax are used to ensure that the words produced are grammatically and syntactically coherent – for example, we impose that V be of the present gerund form, and the preposition {with} is the only admissible preposition used with N1, the tool. We show in the experiment section that this simple approach is sufficient for the RPCU to generate concise sentences that summarizes the videos.

Related Works

The proposed RPCU combines two important computational problems: action recognition and sentence generation and solves them in an integrated manner guided by language. As such, related works spans both the Computer Vision (action recognition) and Computational Linguistics (sentence generation) domains.

Action recognition research spans a long history. Comprehensive reviews of recent state of the art can be found in (Turaga et al. 2008; Weinland, Ronfard, and Boyer 2010; Lopes et al.). Most of the focus was on studying human actions that were characterized by movement and change of posture, such as walking, running, jumping etc. Our approach is more closely related to the use of language for object detection and image annotation. With advances on textual processing and detection, several works recently focused on using sources of data readily available "in the wild" to analyze static images. The seminal work of (Duygulu et al. 2002) showed how nouns can provide constraints that improve image segmentation. (Gupta and Davis 2008) (and references herein) added prepositions to enforce spatial constraints in recognizing objects from segmented images. (Berg et al. 2004) processed news captions to discover names associated with faces in the images, and (Jie, Caputo, and Ferrari 2009) extended this work to associate poses detected from images with the verbs in the captions. Some studies also considered dynamic scenes. (Cour et al. 2008) studied the aligning of screen plays and videos, (Laptev et al. 2008) learned and recognized simple human movement actions in movies, and (Gupta, Kembhavi, and Davis 2009) studied how to automatically label videos using a compositional model based on AND-OR-graphs that was trained on the highly structured domain of baseball videos The work of (Farhadi et al. 2010) attempts to "generate" sentences by first learning from a set of human annotated examples, and producing the *same* sentence if both images and sentence share common properties in terms of their triplets: (Nouns-Verbs-Scenes). No attempt was made to generate *novel* sentences from images beyond what has been annotated by humans.

Natural language generation (NLG) is a long-standing problem. Classic approaches (Traum, Fleischman, and Hovy 2003) are based on three steps: selection, planning and realization. A common challenge in generation problems is the question of: what is the input? Recently, approaches for generation have focused on formal specification inputs, such as the output of theorem provers (McKeown 2009) or databases (Golland, Liang, and Klein 2010). Most of the effort in those approaches has focused on selection and realization. We address a tangential problem that has not received much attention in the generation literature: how to deal with *noisy inputs*. In our case, the inputs themselves are often uncertain (due to misrecognitions by object/scene detectors) and the content selection and realization needs to take this uncertainty into account.

Experiments

In order to validate the proposed RPCU, we extracted 24 video clips from a subset of the POETICON video dataset[2] of four everyday scenarios: 1) Clean table with towel, 2) Cut cheese with knife, 3) Cut tomato with knife, and 4) Toss salad with fork and spoon. Each scenario have 6 video clips of the action performed by 2 different pairs of actors (3 videos per pair), with intraclass variations in the location and appearance of the tools, and how the actions are performed by the actors in the videos. There are also multiple actions occurring at the same time, which makes this dataset extremely challenging with occlusions and constant human interactions. All the videos evaluated are taken from the same viewpoint.

We first detected the tools and direct-objects n_1 and n_2 actively and extracted the hand trajectories as action features as described previously. At the same time, the RPCU's language models are trained using the Gigawords corpus from the defined tool, objects and action sets (Table 1).

Next we invoke the RPCU's action prediction module that uses an EM formulation to predict the most-likely action given the tools and action trajectories. Among the 24 video clips, we are able to predict the correct action labels for 21 of them (87.5% accuracy). We normalize the EM output for each label using eq. (16) to obtain $\mathcal{P}_I(v|m)$.

Using the predicted action probability distribution and the language model, we use RPCU's sentence generation module to produce descriptive sentences for each of the 24 video clips. Sample results from each scenario are shown in Fig. 6. We are able to generate 22 correct sentences from the 24 videos (91.6% accuracy) by comparing the ground truth core sentence structure with each of the video's predicted $\mathcal{T}^*$.

[2] http://poeticoncorpus.kyb.mpg.de/

The observed improvement in the final accuracy of the generated sentence shows that the HMM is able to correct, given the video detections (emissions), initially wrong action predictions thanks to the trained language models.

Figure 6: Four frames (left) from the input videos and results. (Right-upper): Sentence structure $\mathcal{T}^*$ predicted using Viterbi and (Right-lower): Generated sentences.

Conclusion and Future Work

In this paper, we have proposed a crucial enhancement to the standard Robot Perception Unit (RPU): the Robot Perception Control Unit (RPCU), which 1) Combines detection results in a iterative way to improve different sources of detection; 2) Uses language to help improve and combine the detection results and 3) Provide a concise output in the form a descriptive sentence. This is achieved by learning a language model from the Gigaword corpus, formulating an iterative EM approach to predict actions and generating a verbal description of the video using a HMM. Experimental results over 24 videos from a subset of the POETICON dataset shows that our approach is able to predict the associated action and generate a descriptive sentence with high accuracy.

The proposed RPCU has provided a viable framework that we believe is an important step forward for robotic perception to progress from simple detection based strategies of the RPU to a real cognitive agent – one that is able to perceive, reason and act accordingly to its various inputs. The key contribution is the semantic grounding afforded by language which our framework exploits. The proposed RPCU is also generalizable to different detection inputs: e.g. low to mid-level visual features or even other sensors (e.g. sonar). A more refined language model can also be used to handle a larger variety of inference tasks: e.g. predicting the next likely action given something as happened. A major limitation of the approach is that the set of tools, objects and actions (the vocabulary) must be pre-defined, future work will focus on discovering from language, the co-located set of such tools, objects and actions via *attributes*. Finally, we can also extend the language generation module to generate even more complicated sentences that involves, for example, adjectives and adverbs.

Acknowledgments

The support of the European Union under the Cognitive Systems program (project POETICON) and the National Science Foundation under the Cyberphysical Systems Program, is gratefully acknowledged.

References

Berg, T. L.; Berg, A. C.; Edwards, J.; and Forsyth, D. A. 2004. Who's in the picture? In *NIPS*.

Brox, T.; Bregler, C.; and Malik, J. 2009. Large displacement optical flow. In *CVPR*, 41–48. IEEE.

Cour, T.; Jordan, C.; Miltsakaki, E.; and Taskar, B. 2008. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*.

Duygulu, P.; Barnard, K.; de Freitas, J. F. G.; and Forsyth, D. A. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, volume 2353, 97–112.

Farhadi, A.; Hejrati, S. M. M.; Sadeghi, M. A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; and Forsyth, D. A. 2010. Every picture tells a story: Generating sentences from images. In Daniilidis, K.; Maragos, P.; and Paragios, N., eds., *ECCV (4)*, volume 6314 of *Lecture Notes in Computer Science*, 15–29. Springer.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D. A.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(9):1627–1645.

Golland, D.; Liang, P.; and Klein, D. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of EMNLP*.

Graff, D. 2003. English gigaword. In *Linguistic Data Consortium, Philadelphia, PA*.

Gupta, A., and Davis, L. S. 2008. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, volume 5302, 16–29.

Gupta, A.; Kembhavi, A.; and Davis, L. S. 2009. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans on PAMI* 31(10):1775–1789.

Jie, L.; Caputo, B.; and Ferrari, V. 2009. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In NIPS., ed., *Advances in Neural Information Processing Systems*, NIPS. NIPS.

Laptev, I.; Marszalek, M.; Schmid, C.; and Rozenfeld, B. 2008. Learning realistic human actions from movies. In *CVPR*.

Lopes, A. P. B.; do Valle Jr., E. A.; de Almeida, J. M.; and de Albuquerque Arajo, A. Action recognition in videos: from motion capture labs to the web. *CoRR*.

McKeown, K. 2009. Query-focused summarization using text-to-text generation: When information comes from multilingual sources. In *Proceedings of the 2009 Workshop on*

Language Generation and Summarisation (UCNLG+Sum 2009), 3. Suntec, Singapore: Association for Computational Linguistics.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23(3):309–314.

Schwartz, W.; Kembhavi, A.; Harwood, D.; and Davis, L. 2009. Human detection using partial least squares analysis. In *ICCV*.

Traum, D.; Fleischman, M.; and Hovy, E. 2003. Nl generation for virtual humans in a complex social environment. In *In Proceedings of the AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, 151–158.

Turaga, P. K.; Chellappa, R.; Subrahmanian, V. S.; and Udrea, O. 2008. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Techn.* 18(11):1473–1488.

Viola, P., and Jones, M. 2004. Robust real-time face detection. *ICCV* 57(2):137–154.

Weinland, D.; Ronfard, R.; and Boyer, E. 2010. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*. Article in Press, Accepted Manuscript.