Talking Heads: Detecting Humans and Recognizing Their Interactions

Minh Hoai **Andrew Zisserman**
Department of Engineering Science, University of Oxford, Oxford, UK

Abstract

The objective of this work is to accurately and efficiently detect configurations of one or more people in edited TV material. Such configurations often appear in standard arrangements due to cinematic style, and we take advantage of this to provide scene context.

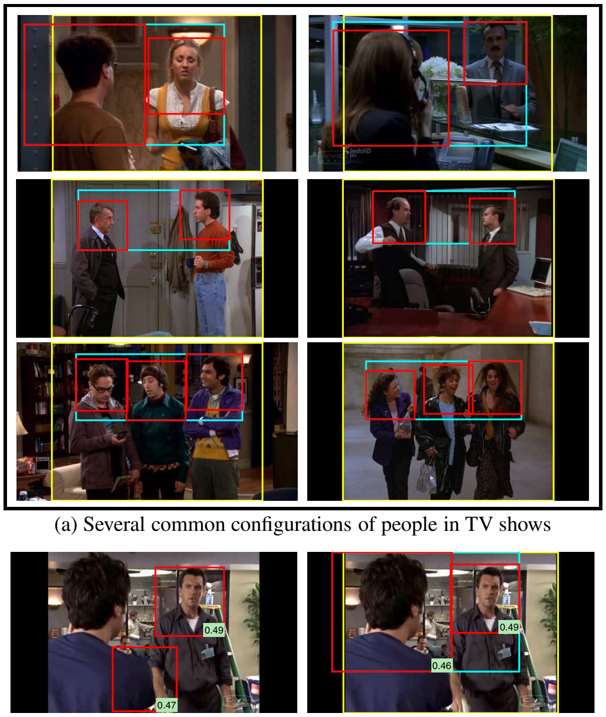
We make the following contributions: first, we introduce a new learnable context aware *configuration model for detecting sets of people in TV material that predicts the scale and location of each upper body in the configuration; second, we show that inference of the model can be solved globally and efficiently using dynamic programming, and implement a maximum margin learning framework; and third, we show that the configuration model substantially outperforms a Deformable Part Model (DPM) for predicting upper body locations in video frames, even when the DPM is equipped with the context of other upper bodies.*

Experiments are performed over two datasets: the TV Human Interaction dataset, and 150 episodes from four different TV shows. We also demonstrate the benefits of the model in recognizing interactions in TV shows.

1. Introduction

Humans are ubiquitous in TV shows, and consequently detecting their presence, location, posture and interactions is important for automated semantic analysis of TV material. This importance is well recognized, and a variety of techniques for detecting humans and their spatial layout have been developed [1, 8, 21, 28]. Similarly, the importance of context in aiding detection, whether from scene geometry or from other objects in the scene, is also well recognized [3, 4, 12, 23].

To this end, we propose a novel *context aware* configuration model for detecting sets of people in TV material. The key idea is to exploit the fact that the locations, scales, and configurations of people in TV video are constrained: first, they are restricted by the frame which requires important content to mostly stay within it; and, second, TV shows are made by professional production teams who employ standard techniques of cinematography in composing shots and choosing camera angles. These constraints lead to a relatively small number of commonly occurring human configurations in TV shows, especially when the number of people involved is not too large. Previous human detection methods have not, for the most part, benefited from this type of context, and certainly have not developed a model that is able to efficiently and optimally choose when to use it. Fig. 1 shows the benefit of using this type of learnt context.



(a) Several common configurations of people in TV shows

(b) Sliding window detection

(c) Configuration-aware detection

Figure 1. **Advantage of configuration-aware detection**. (a) several common configurations for two and three people in edited TV material. We explore the commonality and learn a model of human configurations that can be used for detection and categorization. (b): output of a sliding window detector; it misses an Upper Body (UB), while producing an unlikely formation of two UBs. (c): a configuration-aware detector can remove false positive and resolve ambiguity. In each image, red squares are UBs, which are enclosed by the cyan box. The yellow rectangle delineates the center part of the frame that has 4:3 aspect ratio.

Specifically, we explore the commonly occurring human configurations in TV shows and build a set of exemplar Upper Body (UB) configurations by clustering thousands of frames with annotated UBs. The exemplars are used at test time to aid in detecting people and their configurations, acting as a spatial prior. We introduce an Upper Body Configuration (UBC) detector that proceeds in two stages: first, a sliding window detector, such as a DPM [7], is used to obtain dense UB detection scores at multiple locations and scales. Second, configurations of UBs that have high detection scores and high similarity with an exemplar are obtained using an efficient and globally-optimal inference algorithm that searches across multiple locations and scales.

As would be expected, for a small number of people (e.g., two or three) there are a limited number of exemplar configurations, but as the number of people increase so do the number of configurations. For this reason it is important to make judicious use of the exemplar configurations, and complement detections from the UBC model with 'singleton' detections that are not part of the dominant configuration (using an UB detector with individual context).

We postpone a detailed discussion on the differences between the UBC and DPM detectors until after we have defined the UBC model in Section 3. The empirical benefits of using the UBC detector are investigated in Section 4.2. In particular, it is shown that the UBC detector exceeds both the precision and the recall obtained by a DPM, even when the DPM is rescored using contextual information provided by other UBs (using the method of [7]) or after adding temporal information (tracking). Finally, in Sec. 4.4, it is shown that using the UBC detector advances the state-of-the-art performance for human interaction recognition [16].

2. Related Work

Most work that has investigated detecting groups of people has done so for overcoming partial occlusion [5, 17, 22, 27], rather than the objective being, as here, a configuration of potentially non-overlapping people. Others have studied people configurations in order to obtain further information, such as gender or types of interactions [10, 15], but do not use the configurations to improve detection. Methods for employing 3D scene context, such as Hoiem *et al.* [12], or local context and support regions, such as Divvala *et al.* [4], are largely irrelevant for TV material because camera shots often show close-ups, with no or little support regions and background. Similarly, methods for detecting multiple people in crowded scenes acquired by security videos [19, 26] are not directly applicable to TV material.

More relevant are algorithms for detecting groups of objects, such as [3, 13, 20], since these can be applied to human UBs. Desai et al. [3] re-score object detection outputs using binary relational attributes (e.g., a lamp is on top of a desk). However, in our case it is not beneficial to model the spatial relationship of human UBs using binary attributes because almost all spatial relations of this type (e.g., on-top and to-the-right) are probable. Sadeghi & Farhadi [20] and Li et al. [13] train a DPM for each configuration of close-by objects. This approach does not take advantage of prior location and scale of the object group, which are important cues for detecting people in TV shows. Furthermore, this approach can be two orders of magnitude slower than running an individual-person detector, because one needs to run a separate DPM for each configuration, and there are many people configurations. In contrast, our method utilizes shared computation; it efficiently recognizes people configurations with little additional processing time, relative to an individual-person detector.

The output of a human UB detector can be enhanced in a number of ways: Prest *et al.* [18] combine a face detector with two UB detectors. Patron-Perez *et al.* [16] use tracking to link upper bodies between consecutive frames, subsequently discarding some false positives. These approaches are practical and useful, and can be combined with the proposed method to further enhance its performance.

3. Upper-body Configuration Detector

Overview. An Upper-Body Configuration (UBC) detector takes an image frame as input and outputs a configuration of UBs (specifying their location and scale). The detector uses an 'ensemble' of UB Configuration Models, where each configuration model arises from an exemplar configuration that has been learnt in advance (from annotated TV material). The exemplar configuration is a set of deformable UB parts with parameters for: (i) the relative locations and scales of the constituent UBs; and (ii) the relative location and scale of the UB union w.r.t. the image frame. The configuration model is used to score a set of candidate UBs based on: (i) their unary potentials (obtained from a sliding window UB detector) and (ii) the similarity between the candidate UBs and its exemplar configuration (deformation cost). The importance of these two factors are specific to each configuration model, and is specified by a set of learnable parameters. For example, if an exemplar configuration has two UBs, then all possible configurations of two UBs would be scored by the configuration model.

To detect UBs in a test image, first the unaries are computed, and then *all* configuration models are used to evaluate all sets of candidate UBs, at all locations and scales. This inference algorithm (Sec. 3.2) is efficient and globally-optimal. To detect 'off-focus' UBs that are not part of the dominant configuration, we complement the UBC detector with a singleton detector that detects UBs individually.

Relation to DPMs. It can be seen that a Configuration Model (CM) bears some resemblance to a DPM [7] because the UB union is analogous to the root filter of a DPM and the UBs can be regarded as deformable parts. To this extent, CMs inherit the advantages of DPMs, including its ability to model spatial deformation. However, CMs advance DPMs in several ways. First, a CM has no root filter, reducing the computational complexity of the inference algorithm. Second, a CM maintains the prior location and scale of the configuration w.r.t. the image frame; this is an important cue for detecting UBs in TV material. Third, and most important, CMs allow deformation in scale, while DPMs do not. In a DPM, the scale of each part is determined by the scale of the root filter. This enforces fixed relative scales between the parts, limiting the ability for modeling multiple UBs where the relative scales vary. In contrast, CMs allow deformation in scale, while maintaining low computational cost.

3.1. Building configuration clusters

Configuration Clusters (CCs) of UBs in TV material are learnt by clustering frames with annotated UB locations (using k-means). Separate clusters are built for groups of one, two, three, and four or more UBs. The clustering is based on the relative locations and scales of the UBs. These clusters provide the exemplar configurations for the configuration models of the UBC detector.

Consider a group of frames each containing k UBs (for a group with four or more UBs, only the largest four UBs are considered). The UBs are sorted from left to right based on their centers, and the smallest enclosing bounding box is computed, which will be referred to as the *UB union*. Two sets of relative locations and scales are then computed and stored as two configuration vectors: the relative locations and scales of the UBs w.r.t. the UB union, and the relative location and scale of the UB union w.r.t. the reference frame. To account for different aspect ratios of video, the reference frame is taken as the center portion of the image that has 4:3 aspect ratio.

The set of exemplar configurations for k (≥ 2) UBs, is obtained using two-level hierarchical clustering. The first level of clustering is on the configuration vector of the UBs w.r.t. the UB union; and the second level divides each cluster from the first level using the configuration vector of the UB union w.r.t. the reference frame. Fig. 2 displays exemplar configurations for two UBs. The benefits of the two-level configuration clustering are twofold. First, it emphasizes the importance of the two types of configurations. Second, the hierarchical model improves the detection speed (using shared computation) and increases training data (using shared data) for each configuration model, as will be explained in Secs. 3.2 and 3.3.

Exemplar models for one UB are obtained by clustering the configuration vector of the UB w.r.t. the image. To maximize the amount of training data, the flipped training images are included in the clustering process. However, k-means is non-deterministic, and a generated cluster may not have the exact mirrored version.

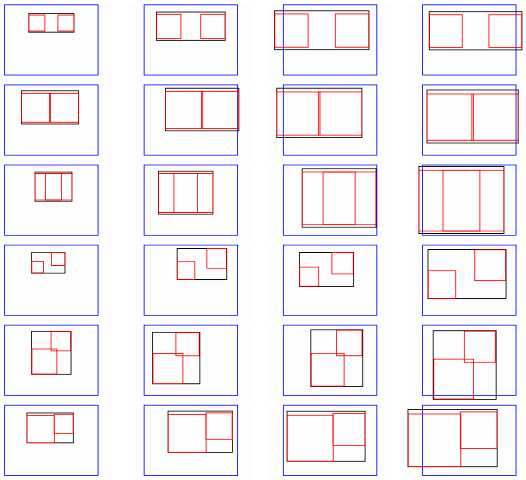


Figure 2. **Configuration Clusters (CCs) for two UBs**. Clustering is first based on the relative locations and scales of UBs w.r.t. its union (shown by rows), and second based on the relative location and scale of the UB union w.r.t. the image frame (shown by columns). Red squares are locations of UBs. Black rectangles are the UB unions. Note, each CC on the last three rows has a left-right mirror counterpart, which is not shown in the figure.

The total numbers of clusters for 1, 2, 3, and 4 UBs are 12, 36, 10, and 2, respectively. The settings of these numbers are influenced by: (i) the variation of upper-body formations, and (ii) the amount of training data. Unfortunately, data for 3 and 4 UBs is limited, and this constrains the number of configuration models for 3 and 4 UBs. The sensitivity to the number of CMs is evaluated in Sec. 4.3.

3.2. Energy and inference

For any video frame, our goal is to detect all UBs and recognize their configuration. We formulate this problem as an energy minimization problem, where we seek a set of UBs with high detection scores (unary potentials) and low deformation cost w.r.t. a CC (prior). This section defines the energy function and describes the inference algorithm.

Let Θ be the set of all CMs, the energy for a set of k candidate UBs $\mathbf{p}_1, \cdots, \mathbf{p}_k$ and their union $\mathbf{u}$ is defined as: $E(\{\mathbf{p}_i\}, \mathbf{u}) = \min_{\theta \in \Theta} E(\{\mathbf{p}_i\}, \mathbf{u}|\theta)$. The energy $E(\{\mathbf{p}_i\}, \mathbf{u}|\theta)$ for a set of candidate UBs and their union w.r.t. a CM is defined as the sum of UB detection energies and the deformation cost:

$$\sum_{i=1}^{k} \alpha_i \mathcal{U}(\mathbf{p}_i) + \sum_{i=1}^{k} \boldsymbol{\beta}_i^T \phi_1(\mathbf{p}_i|\mathbf{u}) + \boldsymbol{\gamma}^T \phi_2(\mathbf{u}) + b. \quad (1)$$

In the above formulation, $\mathcal{U}(\mathbf{p}_i)$ is the negative of the UB detection score at $\mathbf{p}_i$. We assume this information is available, e.g., from a DPM. $\phi_1(\mathbf{p}_i|\mathbf{u})$ is a vector for the relative location and scale of $\mathbf{p}_i$ w.r.t. $\mathbf{u}$ while $\phi_2(\mathbf{u})$ encodes the relative location and scale of $\mathbf{u}$ w.r.t. the image frame. The parameters $b, \alpha_i, \boldsymbol{\beta}_i, \boldsymbol{\gamma}$ are specific to the configuration model in consideration. This energy encourages candidate

UBs to have high detection scores while their arrangement is close to the UB configuration in consideration.

In the above formulation, $\mathbf{u}$ is an explicit variable for the UB union. By definition, $\mathbf{u}$ is the enclosing bounding box of the UBs, so it is completely determined by the UBs $\{\mathbf{p}_i\}$. However, the explicit introduction of $\mathbf{u}$ has several benefits. First, $\mathbf{u}$ acts as a factor variable that decorrelates the UBs, making the dependency between variables a tree structure. Second, by relaxing the hard constraint that $\mathbf{u}$ must be the enclosing bounding box of the UBs, we can derive a fast inference and moreover allow scale deformation.

Inference. For a given CM, the inference for finding the set of UBs and their union that minimize the above energy is efficient. First, the most time consuming procedure is to compute the UB detection scores at all locations and scales. However, for an UB detector that uses the sliding window approach such as DPM [7], dense detection scores can be obtained without additional cost. Second, the dependency between the UBs and their union and between the union and the image frame is a tree structure. This enables the use of dynamic programming and generalized distance transforms to efficiently search over all possible configurations in an image, without restricting the possible locations and scales of each part. The spatial deformation of this inference algorithm resembles DPM [7] and Mixture-of-Parts [28]. However, unlike [7, 28] where the relative scales of parts are constant, our model and inference allow scale deformation. This is important for modeling multiple UBs where the relative scales vary.

Timings and implementation details. The run time complexity of this inference is linear in the number of UB parts and quadratic in the number of scales considered. On a 2.3GHz Intel Core i7 machine, for an image of size 352×624, it takes 945ms to compute dense detection scores. It takes 4ms, 12ms, 22ms, 30ms to run the inference algorithm for a configuration model with 1, 2, 3, 4 UBs, respectively.

The detection algorithm runs the inference for all configuration models. This procedure scales sub-linearly thanks to shared computation. As mentioned above, the computational bottleneck is to compute dense UB detection scores but this can be done by running a sliding window detector once. This differs from the approach for detecting groups of heterogeneous or occluding objects [13, 17, 20, 22], which requires running a different sliding window detector for every configuration. The second most time consuming procedure is to perform a generalized distance transform for every pair of UB and UB union. However, the output of generalized distance transform can be shared between group of CMs that have the same parameters $\alpha_i, \boldsymbol{\beta}_i$. We distribute the same set of these parameters to CMs in the same level-1 cluster (Sec. 3.1). This reduces computational cost and, furthermore, increases training data for each configuration model. Computing the deformation potential for a given UB union w.r.t. the image does not involve generalized distance transform, and this can be done with a few matrix multiplications. Notably, the CMs for one UB are different from the CMs for two or more UBs. The one-UB models have no UB union, and there is no need to perform a generalized distance transform.

The complete inference algorithm for all models is relatively fast. For all 60 configuration models, it takes 610ms to run the inference algorithm for an image of size 352×624 pixels. This is reasonable, given the time for computing dense detection scores (945ms) and performing non-maximal suppression on the dense detection score (99ms). Nevertheless, it is important to emphasize that this inference algorithm outputs not only a set of UBs, but also predicts the formation type (e.g., two people standing side-by-side) and the location and scale of their formation (e.g., medium close-up shot). This information provides informative cues towards semantic understanding of TV shows.

3.3. Learning the parameters of a UBC detector

The parameters of a UBC detector can be learned with the max-margin framework. This framework has a convex quadratic objective, which can be effectively optimized. The parameters of CMs are jointly learned, eliminating the need for post-calibration.

We assume the availability of video frames $\mathbf{I}_1, \cdots, \mathbf{I}_n$ with annotated UBs $\mathbf{P}_1, \cdots \mathbf{P}_n$. Let the configuration models of these training frames be $y_1, \cdots, y_n \in \{1, \cdots, m\}$. The energy of a configuration model for a particular set of parts defined in Eq. 1 is linear in terms of its parameters. Consider the configuration model k, we rewrite the energy function for an image $\mathbf{I}$ and a set of UBs $\mathbf{P}$ as:

$$E_k(\mathbf{I}, \mathbf{P}) = -(\mathbf{w}_k^T \varphi_k(\mathbf{I}, \mathbf{P}) + b_k). \tag{2}$$

Here $\varphi_k(\mathbf{I}, \mathbf{P})$ is the feature vector that depends on the unary potentials and the deformation vector. $\mathbf{w}_k$ and b_k are the weight vector and bias term that need to be learned. We train these parameters using maximum-margin learning:

$$\underset{\{\mathbf{w}_k, b_k, \xi_i \geq 0\}}{\text{minimize}} \; \frac{1}{2m} \sum_{1}^{m} ||\mathbf{w}_k||^2 + C \sum_{i=1}^{n} \xi_i \tag{3}$$

$$\text{s.t. } \mathbf{w}_{y_i}^T \varphi_{y_i}(\mathbf{I}_i, \mathbf{P}_i) + b_{y_i} \geq \mathbf{w}_y^T \varphi_y(\mathbf{I}_i, \mathbf{P}) + b_y + 1 - \xi_i$$
$$\forall i, \mathbf{P}, y : n_y \neq n_{y_i}. \tag{4}$$

Here, n_y denotes the number of UBs for configuration model y. This learning formulation requires the SVM score of the correct configuration model for the annotated set of UBs to be higher than the score of any other set of UB parts with any other CM that has a different number of UBs. This formulation resembles multi-class SVM [2], but has two key differences. First, we only compare CMs that have

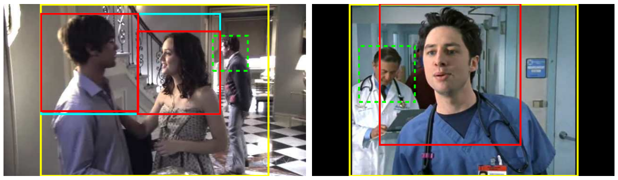


Figure 3. **Out-of-configuration UBs**. People shown in dashed green boxes are not part of the main video focus.

different numbers of UBs; this is to avoid the competition between similar CMs. Second, the negative examples (i.e., the right hand side of Constraint (4)) cover all sets of UBs at different locations and scales; this effectively increases the amount of training data.

The proposed learning formulation is convex, but it contains a large number of constraints. We use constraint generation in optimization, i.e., we maintain a smaller subset of constraints and iteratively add the most violated ones. Constraint generation is guaranteed to converge to the global minimum [24]. In our experiments, this converges within 50 iterations. Each iteration requires minimizing a convex quadratic objective, which is solved using Cplex.

In practice, the above formulation is a simplified version of our learning algorithm. As discussed in the previous section, we enforce parameter sharing among members of certain configuration groups for faster inference and data sharing. This can be incorporated in the learning formulation while keeping it as a convex quadratic objective.

3.4. Singleton UB detector

The default output of a UBC detector is a single set of UB configuration that best explains a video frame. This can effectively detect *foreground* people, who are the main focus of the video frame in consideration. However, there might be other people in the background, as illustrated in Fig. 3. The presence or absence of these people has little effect on the main content of the video, and they are not part of the dominant configuration. To detect these people, we use a UBC that only incorporates 1-UB configuration models, which will be referred to as the singleton detector. For simplicity, we will refer to the full UBC detector as UBC and the UBC with a singleton detector as UBC+S.

An alternative approach for detecting background people is to consider additional UBs from a DPM. However, this approach is empirically worse than using UBC with only 1-UB CMs. The 1-UB CMs encode the prior locations and scales of UBs in video frames containing a single UB. This contextual prior is different from the true prior for off-focused UBs. Using this supposedly-wrong prior is still better than not using it, perhaps because of the similarity between the off-focus and on-focus priors (e.g., it is uncommon to find people at the bottom of an image, whether they are in focus or not). The true prior for off-focused people can be learned, but this requires additional data annotation.

Number of UBs	0	1	2	3	≥4	total
TVHI train data	0	118	370	79	32	599
TVHI test data	0	100	464	121	29	714
Combined train data	143	448	740	291	32	1654
Combined test data	128	406	726	227	29	1566

Table 1. **Numbers of frames with a specific number of UBs**. Train and test data are disjoint. For the TVHI data, we maintain the train/test split of the data [16]. For TV episodes, we split based on seasons.

4. Experiments

Considering the tasks of UB detection and counting, we compare UBC and a publicly available[1] DPM UB detector [11]. Subsequently, we use the detected UBs to assist recognition of human interaction.

4.1. Datasets

The data for these experiments is collected from two sources: the TV Human Interaction (TVHI) dataset [16] and 150 episodes of four different TV shows. The TVHI dataset consists of 300 video clips compiled from 23 different TV shows. We select three key frames (the first, middle, and last) for each shot of each video clip. This yields 1313 frames, each comes with annotated UB locations.

The 150 TV episodes are from 8 seasons of 4 different TV shows (two seasons each). The TV shows are: The Big Bang Theory (BBT) (season 1–2), Frasier (season 1–2), Scrubs (season 1–2), and Seinfeld (season 3–4). For these videos, shots are detected automatically and frames sampled from the middle of each shot. Similar frames are detected using SIFT matching, and excluded. This provides 1907 frames which we annotate with their UBs.

The two datasets are disjoint. In terms of TV series, there is some negligible overlapping: TVHI has no video clips from Seinfeld or Frasier, and less than 12% of the clips are from Scrubs and The Big Bang Theory. Furthermore, there is no overlap at the season level (for example, our dataset contains Scrubs seasons 1–2, while the Scrubs videos in TVHID are from seasons 3–8).

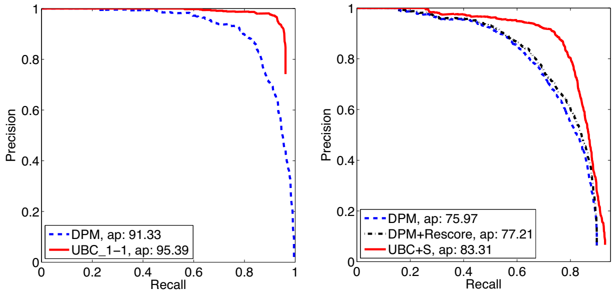
Tab. 1 gives the train/test splits used for the TVHI and combined datasets, and the distribution of the number of ground truth UBs over the frames. In the implementation the amount of data is doubled by left-right mirroring.

4.2. Upper-body detection and counting

4.2.1 Upper body detection

This section describes experiments on detecting UBs. An UB is deemed to be detected if the overlap between the predicted UB and annotated UB is more than 50% (where the overlap is the ratio of the area of their intersection to the

[1] www.robots.ox.ac.uk/~vgg/software/discrim_subcat/



(a) Combined dataset: one-vs-none (b) TVHI dataset: multiple UBs

Figure 4. **Precision-recall curves**. (a) detecting UBs in frames with at most one UB. UBC_1-1 is a UBC detector that only uses configuration models for one UB; it significantly outperforms the DPM, proving the benefits of knowing the prior locations and scales of UBs. (b) detecting UBs in the TVHI test data.

area of their union); i.e., the standard PASCAL VOC requirement [6]. While finding correspondence between predicted and annotated UBs, we ensure no detection or annotated UB are counted twice using the Hungarian algorithm. Performance is measured using precision-recall curves.

Detecting one UB: the first experiment is to detect UBs in video frames that contain at most one UB. For this experiment, it suffices to use a UBC detector with 1-UB configuration models. As previously discussed, configuration models for 1-UB are different from models for two or more UBs. The models for 1-UB does not have the UB union, and therefore, what they capture are the prior locations and scales of a single UB in TV shows. Thus the purpose of this experiment is to analyze the benefits of knowing the prior locations and scales of UBs when detecting them.

Fig. 4(a) plots the precision-recall curves for detecting UBs in frames that contain at most one UB. We compare the performance of a UBC detector with a DPM [7], which filters dense detection scores through non-maxima suppression. Fig. 4(a) shows the advantage of UBC over DPM, especially at high recall. The average precision of UBC and DPM are 95.38 and 91.33 respectively.

Detecting multiple UBs: the second experiment is to detect multiple UBs in video frames of the TVHI dataset. For this experiment, we use all configuration models, which are jointly trained as discussed in Sec. 3.3. Fig. 4(b) plots the precision-recall curves of UBC, DPM, and DPM+Rescore. DPM+Rescore [7] is the method that combines DPM and a post processing step. This post processing step rescores a detection using contextual information, which is the detection score, the scale and location of the detection, and the maximum score of other detections in the same image. This simple approach for incorporating contextual information slightly improves the performance, but the result remains inferior to UBC.

Fig. 5 displays the outputs of DPM and UBC for several images. For DPM, we use the threshold that attains highest F1-score. As can be seen, DPM relies on unary potentials, leading to missed detections and improbable false hits. UBC produces better detections thanks to the contextual cues from the configuration clusters. Fig. 6 displays some examples where the best configuration returned by

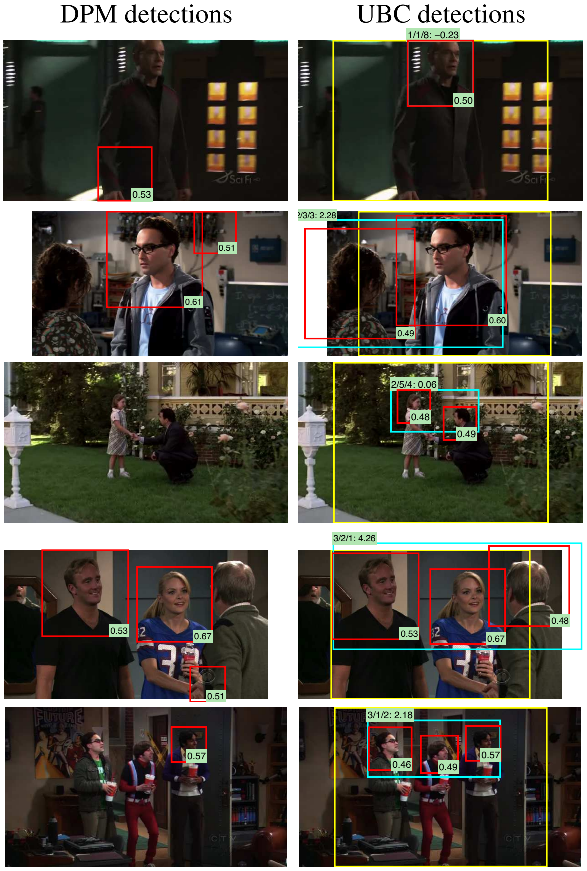


Figure 5. **Detection outputs of DPM (left column) and UBC (right column)**. The number at the bottom right of each red detection square is the unary potential. UBC uses configuration models to discard unary potentials found at improbable locations and boost up low unary potentials at the locations that are likely to contain an UB. The information displayed on the top left corner of each UB union are the number of people, the configuration type, the location-scale type, and the overall confidence score.

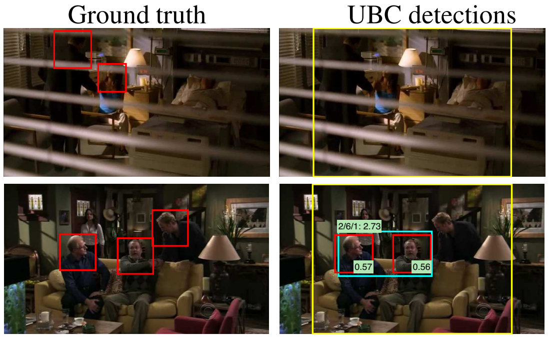


Figure 6. **Missed detections of UBC**, due to: low unary potentials and odd configuration of humans.

		DPM				
		0	1	2	3	≥4
Actual	0	.98	.02	.00	.00	.00
	1	.12	.67	.21	.00	.00
	2	.11	.31	.41	.14	.02
	3	.04	.10	.29	.36	.21
	≥4	.00	.22	.21	.34	.22

		UBC+S (ours)				
		0	1	2	3	≥4
Actual	0	.95	.02	.02	.01	.00
	1	.05	.87	.07	.00	.00
	2	.04	.21	.67	.07	.01
	3	.02	.03	.36	.53	.06
	≥4	.00	.00	.33	.33	.34

Table 2. **Confusion matrices for UB counting**, for DPM (left) and UBC+S (right). The mean accuracies of DPM and UBC+S are 52.84% and 67.09%, respectively.

UBC fails to capture all annotated UBs. There are several common sources of errors, which are discussed at the bottom of the figure.

UBC remains superior to DPM even with the use of temporal information (by tracking and linking UBs between consecutive frames [16]). For example, at 80% recall, the precision values of DPM and UBC+S after incorporating temporal cues are 74.45% and 84.18%, respectively.

4.2.2 Upper-body counting

This section describes experiments on categorizing frames based on the number of UBs they contain. Specifically, we consider the problem of classifying whether a frame contains 0, 1, 2, 3, or ≥ 4 UBs. This provides a useful cue for semantic understanding of TV shows.

We compare the performance of UBC+S with DPM. For both methods, we analyze the precision-recall tradeoff and pick the threshold that yields the highest F1 score. The F1 score is defined as the harmonic mean of precision and recall $F1 = \frac{2 \times prec \times rec}{prec + rec}$. Tab. 2 shows the confusion matrices of both methods. The average accuracy of UBC+S is 66.34%, compared with 52.84% of DPM. For categories of one and two UBs, the difference between the performances exceeds 20%. This proves the benefits of having configuration models in detecting UBs in TV shows. Our method performs poorly in recognizing frames with 4 or more UBs. This is perhaps due to the lack of training data with four UBs or more, as we only have 32 examples.

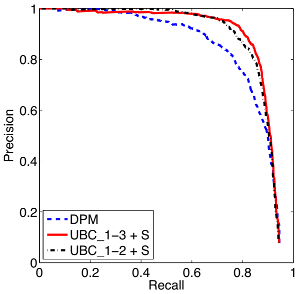
4.3. Contribution of configuration models

The UBC detector uses CMs for 1, 2, 3, and 4 UBs. This section analyzes their relative importance and the overall UB detection performance when the numbers of CMs vary.

Fig. 7.a shows the usage CMs by the UBC detector on test data of the combined dataset. Note that the singleton detector is not used, and the UBC only outputs the most probable configuration. CMs for 1 and 2 UBs are heavily used, and their usages correlate with the true number of UBs. CMs for 4 UBs have little importance because they are used only 8 times, mostly for the wrong frames. The importance of 3-UB CMs is harder to comprehend from Fig. 7.a alone. On the one hand, 3-UB CMs are used many times. On the other hand, UBC uses 2-UB CMs for a large proportion of frames with 3 UBs, and we can complement UBC with the singleton detector.

		#UBs of used CM				
		0	1	2	3	≥4
Actual #UBs	0	242	6	4	3	1
	1	43	732	33	4	0
	2	61	373	959	53	6
	3	10	29	305	210	0
	≥4	0	0	30	27	1

(a)



(b)

Figure 7. (a): confusion matrix for the usage of CMs: a number at row r and column c is the number of times a CM with c UBs is selected by the UBC for frames with r actual UBs. (b): precision-recall curves for detecting UBs on video frames with exactly 3 UBs. UBC_1-2 is the method that uses only CMs for 1 and 2 UBs, and UBC_1-3 is the method that uses CMs for 1, 2, and 3 UBs. Both methods are complemented with the singleton detector.

No. of 1-UB CMs	8	8	12	20	20
No. of 2-UB CMs	24	40	40	40	64
AP on TVHI data.	81.33	81.58	81.93	81.89	82.02
AP on Com. data.	85.39	85.96	86.56	86.58	86.83

Table 3. **Sensitivity analysis**. Average Precision (AP) for 5 settings of the numbers of CMs. Notably, the APs of UBC are not too sensitive to the numbers of models. For references, the APs of DPM on these two datasets are 75.97 and 81.88, respectively.

To further analyze the significance of 3-UB CMs, we compare the performance of UBC with and without 3-UB CMs. Fig. 7.b plots the precision-recall curves for detecting UBs on frames with exactly 3 UBs. As can be seen, using 3-UB CMs only yields marginal benefit at high recall. However, a high recall value might be what a method for semantic video analysis requires. Furthermore, another benefit of using 3-UB CMs is the information about the configuration type of detected UBs.

The performance of UBC is not very sensitive to the numbers of CMs. Tab. 3 shows the average precision for using a UBC detector that consists of 1-UB and 2-UB CMs, with five different settings.

4.4. Human interaction recognition

In this section the detected UBs are applied to the recognition of human interactions in TV shows. This experiment is performed on the TVHI dataset [16] which has annotation for human interaction. The dataset contains 300 video clips, with four interactions classes: Handshake, Highfive,

	Handshake	Highfive	Hug	Kiss	Mean
Patron *et al.* [16]	39.4	45.8	47.0	37.6	42.4
Marin *et al.* [14]	-	-	-	-	39.2
Yu *et al.* [29]	-	-	-	-	55.9
Gaidon *et al.* [9]	-	-	-	-	55.6
DTD [9, 25]	-	-	-	-	53.4
Ours	**55.8**	**60.2**	**60.8**	**48.2**	**56.3**

Table 4. **Average precision for human interaction recognition**.

Hug, and Kiss; and each of these interactions has 50 videos. There are 100 negative examples, which do not contain any of the above interactions. Each video clip has between 30 to 600 frames, and the interactions are not temporally aligned. The training/testing split are as suggested by the authors.

We use UBC+S to detect UBs and subsequently link them into tracks [16]. For each UB track, a track-focused descriptor is then computed based on Dense-Trajectory Descriptors (DTD) [25] which encode gradient and motion cues along the trajectories. We refer the reader to [25] for more details. Unlike [25], trajectories that lie outside an extended volume of the UB track (extending 50% to the left and right, and 100% to the bottom) are discarded. A human-focused descriptor is computed by averaging all UB track descriptors. We also compute a HOG-based scene descriptor, which is the average of HOG descriptors computed on key frames (3 frames per shot). Thus, a video is represented by a human-focused descriptor and a HOG-based scene descriptor.

A kernel SVM (using an exponential $\mathcal{X}^2$ kernel) classifier is trained for each interaction class. Tab. 4 shows the average precision for recognizing these human interaction classes. Our method outperforms Patron-Perez et al. [16] on all classes, and it achieves the best overall performance.

5. Discussion

We have demonstrated the benefits of UBC in detecting humans and in recognizing their interactions. Since detecting humans underpins so much analysis of video material, UBC can improve a number of existing areas of video analysis such as pose estimation and character identification . It also opens up new applications, such as retrieving frames based on the number of people or on cinematic composition (e.g., two-shot, over-the-shoulder shot). Indeed the configuration clustering suggests the possibility for unsupervised discovery of new cinematic configurations that don't even have a name.

Acknowledgements: This work was supported by EPSRC grant EP/I012001/1. The authors would like to thank Eric Sommerlade for providing code for linking UBs into tracks.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proc. CVPR*, 2009.

[2] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Machine Learning Research*, 2:265–292, 2001.

[3] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proc. ICCV*, 2009.

[4] S. K. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert. An empirical study of context in object detection. In *Proc. CVPR*, 2009.

[5] M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. In *Proc. ECCV*, 2010.

[6] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2009.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 32(9):1627–1645, 2010.

[8] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, 2008.

[9] A. Gaidon, Z. Harchaoui, and C. Schmid. Recognizing activities with cluster-trees of tracklets. In *Proc. BMVC.*, 2012.

[10] A. C. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[11] M. Hoai and A. Zisserman. Discriminative sub-categorization. In *Proc. CVPR*, 2013.

[12] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1):3–15, 2008.

[13] C. Li, D. Parikh, and T. Chen. Automatic discovery of groups of objects for scene understanding. In *Proc. CVPR*, 2012.

[14] M. J. Marin-Jimenez, E. Yeguas, and N. P. de la Blanca. Exploring STIP-based models for recognizing human interactions in TV videos. *PRL*, 34:1819–1828, 2013.

[15] M. J. Marin-Jiménez, A. Zisserman, and V. Ferrari. "Here's looking at you, kid." Detecting people looking at each other in videos. In *Proc. BMVC.*, 2011.

[16] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in TV shows. *IEEE PAMI*, 34(12):2441–2453, 2012.

[17] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *Proc. CVPR*, 2013.

[18] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE PAMI*, 34(3):601–614, 2012.

[19] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *Proc. CVPR*, 2011.

[20] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *Proc. CVPR*, 2011.

[21] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proc. ECCV*, 2010.

[22] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. In *Proc. BMVC.*, 2012.

[23] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003.

[24] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

[25] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc. CVPR*, 2011.

[26] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proc. ICCV*, 2005.

[27] Y. Yang, S. Baker, A. Kannan, and D. Ramanan. Recognizing proxemics in personal photos. In *Proc. CVPR*, 2012.

[28] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, 2011.

[29] G. Yu, J. Yuan, and Z. Liu. Propagative hough voting for human activity recognition. In *Proc. ECCV*, 2012.