

Recognizing Proxemics in Personal Photos

Yi Yang
UC Irvine

yyang@ics.uci.edu

Simon Baker

Microsoft Research

{sbaker, ankannan}@microsoft.com

Anitha Kannan

Deva Ramanan

UC Irvine

dramanan@ics.uci.edu

Abstract

Proxemics is the study of how people interact. We present a computational formulation of visual proxemics by attempting to label each pair of people in an image with a subset of physically-based “touch codes.” A baseline approach would be to first perform pose estimation and then detect the touch codes based on the estimated joint locations. We found that this sequential approach does not perform well because pose estimation step is too unreliable for images of interacting people, due to difficulties with occlusion and limb ambiguities. Instead, we propose a direct approach where we build an articulated model tuned for each touch code. Each such model contains two people, connected in an appropriate manner for the touch code in question. We fit this model to the image and then base classification on the fitting error. Experiments show that this approach significantly outperforms the sequential baseline as well as other related approaches.

1. Introduction

People interact in interesting ways; Figure 1 shows a few images of two people interacting. Even a seemingly simple interaction such as two people holding hands exhibits a large amount of variability. See Figures 1(g)-(j). Anthropological research on understanding interpersonal behavior can be traced back to the pioneering works of Hall [9, 10] and Argyle and Foss [1]. In his seminal work [9], Hall coined the term *Proxemics* for this field of study.

Inspired by these anthropological papers, we present a computational theory of proxemics. This area of research is relatively unexplored in computer vision, often limited to the use of video [16]. To the best of our knowledge, we are the first to study this topic in the context of consumer photographs. Besides the scientific motivation, proxemics has a number of applications. Most notably, in the area of personal photo organization we may wish to find all photographs of two specific people holding hands, hugging, etc.

Hall [9] defines interaction types as an *unknown* “function” over combinations of various factors including



Figure 1: People interact in a wide variety of different ways. (a)-(f) The six specific touch codes that we study in this paper. (g)-(j) An illustration of the wide variation in appearance for the hand-hand proxemic. (g) Also illustrates that multiple touch codes may appear at the same time.

postural-sex identifiers, sociofugal - sociopetal orientation, kinesthetic factors and temporally measured touch codes. Many of these factors are often not measurable in static photographs, and there is no existing approach to combine them, computationally. Hence, we took a pragmatic approach and characterized proxemics as the problem of recognizing how people physically touch each other. This enabled us to enumerate the types of interactions, which we call *touch codes*. (In this paper, we use the terms touch code and proxemics interchangeably.) We define *touch codes*¹ as the pairs of body parts (each element of the pair comes from a different person) that are in physical contact.

¹An alternative way to formulate the problem might have been to define proxemics classes for “hugging,” “holding a baby,” “holding hands,” etc. We explored this option, however, found the labeling process to be far more subjective than labeling our physically based touch codes.

Through an annotation study conducted over a large collection of images from multiple sources (see Section 2), we identified that there are six dominant touch codes, namely, hand-hand, hand-shoulder, shoulder-shoulder, hand-elbow, elbow-shoulder, and hand-torso. Figures 1(a)-(f) show example images corresponding to each of these touch codes.

It is often the case that the same two people can exhibit multiple touch codes, simultaneously. For instance, in Figure 1(g), the two people are engaged in, both, hand-hand and hand-shoulder. Hence, for each pair of people, proxemics recognition amounts to correctly identifying the correct subset of touch codes that they exhibit.

The problem we address in this paper is touch code recognition. Since touch codes reflect body parts in physical contacts, it is natural to use pose estimation algorithms for touch code recognition. However, it is difficult [4, 18] to reliably estimate pose when there are multiple people interacting due to various factors such as occlusion, body part ambiguities, etc. Another approach may be to forego pose estimation altogether, and simply train different visual templates tuned for each proxemic interaction. The difficulty here is that a single touch code can exhibit large variability due to articulation. Instead, we propose a joint approach that *simultaneously* recognizes touch codes (and hence proxemics) and estimates multi-body articulated pose. We show this joint approach produces significantly better results for both proxemic recognition and pose estimation, and is far superior to approaches that apply these steps sequentially or employ non-articulated templates.

Our main contributions are:

- We introduce the problem of image-based proxemics recognition to computer vision and provide a computational grounding of Hall’s work.
- We have created a new dataset, fully annotated with joint positions and touch code labels.
- We propose a joint model of body pose estimation and proxemics touch codes recognition that enables each of these tasks to help each other. We show experimental results that supports that joint estimation provides a richer model for not only proxemics recognition but also provides better body pose estimation.

1.1. Related Work

The focus of our work is modeling the physical contact between two people as they interact in a single image. To the best of our knowledge, we are the first to study this problem in a systematic manner. Human interactions are one component of human activity recognition. While activity recognition in single images is an active area, most activities studied involve a single person [21, 13]. In particular, Yang et al [22] use articulated poses as an intermediate representation for image-based action recognition.

Past work has analyzed people interacting in video. Often this is motivated from a surveillance perspective, focusing on events such as pick-pocketing and package exchanges [12]. Recent work has applied contextual models to the problem of recognizing group activities [2, 15, 11]. Most related to us is [16], who analyze people interactions in commercial content such as TV shows and movies. We differ in our focus on static image analysis, and so cannot make use of such temporal models.

Another related area is the modeling of human-object relationships. Gupta et al [8] and Yang and Fei Fei [24] analyze interactions between people and objects. Body pose plays a crucial role in such data, but is difficult to estimate because of occlusion due to the interacting object. While one could apply such techniques to our problem by treating the second human as just another object, a second articulated body considerably complicates analysis due to additional occlusions, *etc.* Moreover, while object interactions tend to be functionally defined, proxemic interactions are defined in part by culturally-dependant social norms [10].

The “Visual Phrase” technique [19] is closely related to our approach. It is argued in [19] that complex visual interactions, such as a person riding a horse, are better modeled as a single phenomenon rather than two separate objects. Such person-object composites are typically fed into a weakly-supervised recognition system [6]. We empirically demonstrate that multi-body articulation requires additional supervision and precise encoding of spatial structure.

We argue that directly modeling the interaction between two people is more practical than an approach that first estimates the pose of each independently, particularly when there are severe occlusions and body part ambiguities. While there has been much work on articulated pose estimation (see the survey in [18]) including methods that jointly estimate poses of multiple people [4], our joint approach is more robust.

2. Proxemics Dataset

There is huge variation in how people interact. One approach to assigning labels to these interactions is through phrases that correspond to abstract concepts such as “arm around shoulder”, “hugging”, “holding hands”, “holding a baby,” *etc.* However, these concepts are very subjective. For instance, it is hard to distinguish a “hug” from an “arm around shoulder”. In fact, our first set of annotations based on this approach had poor inter-annotator agreement. Hence, a better approach, and the one we use in this paper is an objective labeling scheme that relies on body parts. In particular, we define interactions through “touch codes”, where each touch code is a pair of body parts (each from a different person) that are physically in contact.

We restrict attention to the upper body where the interactions in personal photographs most commonly occur. There

(a) Image Statistics

No. Images	No. People	No. People Pairs
589	1207	1332

(b) Touch Code Statistics

Hand touch hand	340	25.5%
Hand touch shoulder	180	13.5%
Shoulder touch shoulder	210	15.8%
Hand touch elbow	96	7.2%
Elbow touch shoulder	106	8.0%
Hand touch torso	57	4.3%

(c) Co-occurrence Statistics

0 Codes	1 Code	2 Codes	3+ Codes
531	626	162	13

Figure 2: Statistics of our proxemics dataset. The data originates from a combination of personal photo collections and web searches. We labeled all possible body part pair touch codes. However, the occurrence frequency drops off rapidly and so we restrict attention to the top 6 codes. Between any pair of people, there maybe any number of touch codes. In most cases, there are 0 or just 1, however there are a significant number of cases with 2 or more touch codes.

are at least 5 major parts of the upper body (head, shoulders, elbows, hands, torso). With these parts, there are $5 \times 5 = 25$ possible ways that they can touch. See Section 3.2.1 for a description of how we handle the fact that each person has two shoulders, elbows, and hands.

We empirically explored the occurrence of these 25 touch codes in real photographs. We collected a large number of images consisting of personal photos of family and friends, and data assembled through a set of web-searches on Flickr, Getty Images and image searches on Google and Bing. For web searches, we used abstract concepts (described above) that are indicative of interactions as the key words to search on. In Figure 2(a), we present some basic statistics of the collected data. An image with n people can potentially create $n(n-1)/2$ pairs of people. As shown in Figure 2(a), on average, most images tend to have only 2 people in them. For each image in this collection, and for all people in the image, we labeled their body joint locations. Then, for each pair of people, we labeled all possible touch codes between pairs of body-parts. Figure 2(b) shows the frequency of occurrence of the touch codes. We can see that the frequency drops rapidly and is dominated by a small number of codes. We restricted our analysis to the top six codes. Next, we studied the co-occurrence of multiple touch codes between pairs of people. Figure 2(c) shows the statistics. While most pairs of people have 0 or 1 touch codes, there are a significant number of cases where there are 2 or more touch codes. For example, the elbow and hand of a single arm may both touch another person’s body.

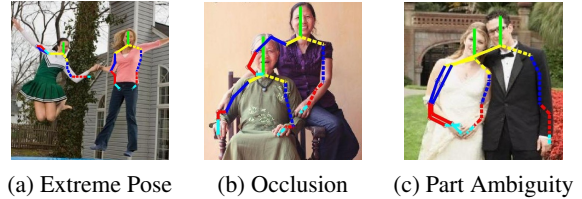


Figure 3: An illustration of why current articulated pose estimation algorithms such as [23] perform poorly on images containing two or more people interacting. The most common problems are: (a) extreme poses, (b) occlusion, and (c) body part ambiguity (for example, the algorithm may successfully fit an arm in the model to an arm in the image, but it may be the wrong person’s arm.)

3. Algorithms

We first describe a simple baseline sequential approach for performing proxemic recognition. The approach is based on using the outputs of pose estimation to perform recognition. We then present our joint approach which jointly models the estimation and proxemic recognition.

3.1. Baseline: Sequential Algorithm

A natural baseline is to first perform pose estimation and then use a measure of the distance between the appropriate body parts to identify the touch code. To perform pose estimation we use the “flexible mixture of parts” algorithm [23] which has shown state-of-the-art performance on various pose estimation benchmarks, and for which public code is available. The output of this algorithm is a set of 2D body part locations:

$$\{L_{He}^i, R_{He}^i, L_{Sh}^i, R_{Sh}^i, L_{El}^i, R_{El}^i, L_{Ha}^i, R_{Ha}^i, L_{To}^i, R_{To}^i\},$$

where $i = 1, 2$ is the index of the person, the subscript denotes the body part ($He = \text{head}$, $Sh = \text{shoulder}$, $El = \text{elbow}$, $Ha = \text{hand}$, $To = \text{torso}$), and R denotes the 2D image location of the right body part and L the location of the left.

In the second step, we compute the distance between the appropriate body parts. For notational simplicity in the definition of the distance below, we duplicate the head and torso locations into the $L_{He}^i = R_{He}^i$ and $L_{To}^i = R_{To}^i$ parameters. The distance between body parts $p1, p2 \in \{He, Sh, El, Ha, To\}$ is:

$$\text{Dist}(p1, p2) = \min_{M, N \in \{L, R\}} \|M_{p1}^1 - N_{p2}^2\|. \quad (1)$$

where distances are measured with respect to the average body scale, as given by the average size of both faces. We then perform classification using a simple threshold on this distance.

The performance of this algorithm is heavily dependent on the pose estimation. Unfortunately, existing pose estimation algorithms perform poorly on images containing two or

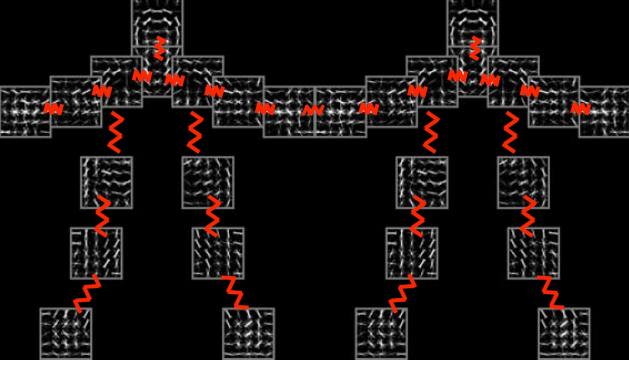


Figure 4: Our model for the hand-hand proxemic is a pictorial structure [6] consisting of two people plus a spring connecting their hands, shown here as a HOG template [3].

more people interacting because of the difficulties caused by extreme poses, occlusion and ambiguity of body parts. Figure 3 shows illustrative examples that depict the failure modes when [23] was used for pose estimation. We present quantitative results for such a baseline approach in our experimental results (Sec. 4.2).

3.2. Joint Estimation of Proxemics and Pose

We now propose a joint model for jointly recognizing proxemic classes and pose estimation for pairs of people. An illustration of our model for the hand-hand proxemic is included in Figure 4. The model is a pictorial structure [6] consisting of two people plus a spring connecting their hands. Suppose we have two people, each represented as a set of parts:

$$V = \{\text{head}_1, \text{neck}_1, \text{left shoulder}_1, \dots, \text{head}_2, \text{neck}_2, \text{left shoulder}_2, \dots\} \quad (2)$$

where the subscript denotes whether the part is for person 1 or person 2. We then formulate a part-based model similar to the one in [23], which models articulated body limbs using local mixtures of small, translating parts. We initially write out the equations for a single mixture model, but later explain how they can be generalized to multiple mixtures.

Let $l_i = (x_i, y_i)$ be the pixel location of part i . Given an image I , we score a collection of parts $L = \{l_i : i \in V\}$ as:

$$S(I, L) = \sum_{i \in V} \alpha_i \cdot \phi(I, l_i) + \sum_{ij \in E} \beta_{ij} \cdot \psi(l_i - l_j) \quad (3)$$

where $\phi(I, l_i)$ is a feature vector (e.g., HOG descriptor [3]) extracted from pixel location l_i in image I . The first sum in Equation (3) is an appearance model that computes the local score of placing a template α_i for part i at location l_i . We write $\psi(l_i - l_j) = [dx \quad dx^2 \quad dy \quad dy^2]^T$, where $dx = x_i - x_j$ and $dy = y_i - y_j$, the relative location of part i with respect to j . The second term is a deformation model

that controls the relative placement of part i and j . It can be interpreted as a spring, where the parameters β_{ij} specify the rest location and rigidity of the spring connecting part i and j . Finally, we write $G = (V, E)$ for a K -node graph which specifies which pairs of parts are constrained to have consistent spatial relations. When G is a tree, one can use dynamic programming algorithms to compute the best pose in an image $L^* = \text{argmax}_L S(I, L)$ [7].

To capture articulation of body parts, we follow the approach of [23] and model limbs using small parts centered at joints and their midpoints. We extend part domains to include position and orientation (x_i, y_i, θ_i) , where θ_i is a discrete variable that captures one of six orientation states. We augment Equation (3) to include orientation-dependant appearance models $\alpha_i^{\theta_i}$ and orientation dependant springs $\beta_{ij}^{\theta_i, \theta_j}$, and append a constant term to the pairwise feature $\psi(l_i - l_j)$ to allow the model to favor particular pairs of orientations (θ_i, θ_j) over others.

To enable the use dynamic programming for inference, we need the edge structure E to be a tree. Each person on their own can be naturally modeled as a tree. So long as we only connect one part from the first person to the second, the two-person model remains a tree. We simply add a single spring to the two parts which are touching as specified by the proxemic touch code. As shown in Figure 4, a hand-hand model will contain a spring connecting the two hands. A hand-shoulder model will contain a spring connecting the hand and shoulder, *etc.* For a finite set of proxemic models parameterized by the discrete variable $p \in \{1 \dots K\}$, we can combine the models into a single joint model over both proxemics and pose:

$$S(I, L, p) = \sum_{i \in V_p} \alpha_i^p \cdot \phi(I, l_i) + \sum_{ij \in E_p} \beta_{ij}^p \cdot \psi(l_i - l_j). \quad (4)$$

where $V_p \subseteq V$ is the set of parts modeled in proxemic p , $E_p \subseteq V$ is the set of edges/springs for proxemic p , and β_{ij}^p is the spring parameter connecting parts i and j for proxemic p . The 6 components of our model, one for each touch code, are illustrated in Figure 5.

While Equation (4) captures our model, there are a few details that we also encode:

Proxemic-Dependant Structure: Not every body part is essential to every proxemic. For example, in a hand-hand interaction, only one hand from each person is in contact. The other arms and legs are not so important and are better not modeled because their components of the fitting energy only add noise. We therefore allow the set of parts modeled to depend on the proxemic being modeled. As an illustration, in Figure 5(a), we show how the other arms and legs can be dropped in the hand-hand proxemic. In Equation (4), this detail appears in the use of V_p rather than V .

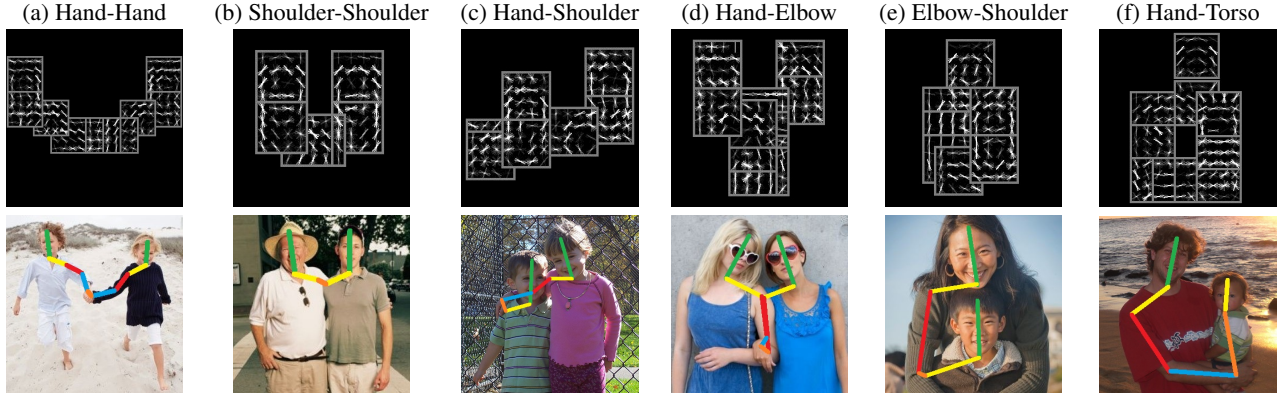


Figure 5: Illustration of the tree-structure of our proxemic-specific model in Equation (4). It is not important to consider the legs and other arms/torso parts to predict the proxemics. We crop out those regions out and build a chain connecting from one person’s head to the other person’s head through the touching body parts.

Proxemic-Dependant Geometry: The pose of a person depends on the proxemic; a person with a hand on a shoulder is posed different than a person holding hands. Hence it is natural to make the spring deformation parameters β_{ij}^p and the set of edges E_p depend on the proxemic interaction p .

Proxemic-Dependant Appearance: One crucial aspect of interactions is occlusion. Consider the hand-shoulder interaction in Figure 5(c); the arm of the hugging person is almost always occluded. One option is to drop occluded parts from the proxemic-specific graph V_p , however this would break the graph into two disjoint components, eliminating any geometric constraint between the two people. Another solution is to keep occluded parts in V_p , but force their associated appearance template β_i^p to be zero, ensuring that no image evidence is scored. We take the view that one can simply define a proxemic-dependant appearance which may or may not be zero (depending upon what parameters from learned from training data). Figure 5(c) suggests that we learn templates that looks for characteristic gradient features associated with partially occluded arms.

3.2.1 Proxemic Sub-Categories

Even a single proxemic category can be visually quite varied. One cause of this variation is the complexity arising from left/right ambiguities. For example, consider two people standing next to one another engaged in a hand-hand interaction. They look very different if the touching hands are facing each other, or on opposite sides of the body. See Figure 6 for an illustration. To resolve such issues we create a number of sub-categories for each proxemic class, obtained by considering all appropriate left/right permutations.

In particular, we augment the proxemic label with a mix-

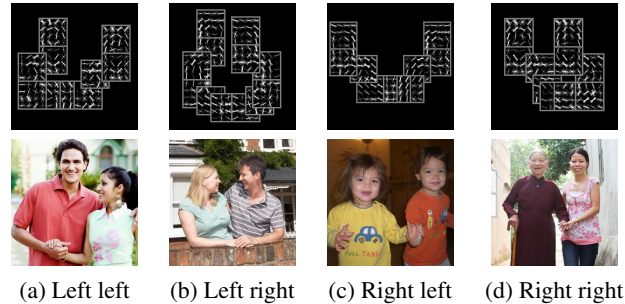


Figure 6: As each person has two arms, we use four sub-models to capture the different hand-hand appearances.

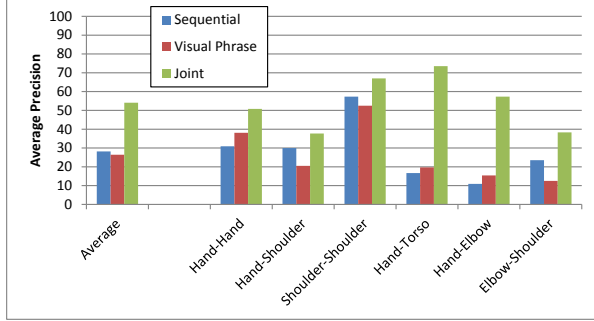
ture component, replacing p with $p' = (p, m)$ in Equation (4), where $m \in \{1 \dots 4\}$. Given an image I , the final score associated with a particular proxemic label p is a maximum score over all poses L and mixtures m associated with that proxemic:

$$S(I, p) = \max_m \left[\max_L S(I, L, p, m) \right] \quad (5)$$

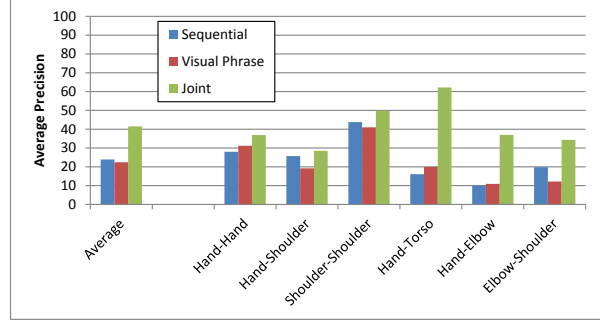
where the inner maximization is performed with dynamic programming, and the outer maximization is performed with a discrete search over the 4 mixture models. For hand-touch-torso and shoulder-touch-shoulder models, we only use 2 mixtures because the torso does not contain a left/right variant, and left-left (or right-right) shoulder touches are uncommon.

3.2.2 Learning

We assume a supervised learning paradigm, where we are given image of pairs of people with ground-truth part labels, proxemic labels, and proxemic sub-category labels $\{I_n, L_n, p_n, m_n\}$. We define a structured prediction objective function similar to the one proposed in [23]. We



(a) Using ground truth head locations



(b) Using face detection to identifying head location

Figure 7: Comparison between our proposed **Joint** algorithm, the **Sequential** algorithm (Section 3.1), and the **Visual Phrase** algorithm [19]. In (a) we use the ground-truth head positions. In (b) we use the faces obtained using a face detector. Our algorithm gives a very significant improvement in average precision in both cases, and across all six touch codes.

write $Z = (L, m)$ and note that the scoring function in Equation (4) is linear in the part appearance models and spring parameters $w_p = \{\alpha^p, \beta^p\}$. This means we can write $S(I, Z, p) = w_p \cdot \Phi(I, Z)$. We train a binary one-vs-all classifier for each p using positive examples of class and negative examples of all other classes:

$$\begin{aligned} \arg \min_{w_p, \xi_i \geq 0} \quad & \frac{1}{2} w_p \cdot w_p + C \sum_n \xi_n \quad (6) \\ \text{s.t.} \quad & \forall n \in \text{pos} \quad w_p \cdot \Phi(I_n, Z_n) \geq 1 - \xi_n \\ & \forall n \in \text{neg}, \forall Z \quad w_p \cdot \Phi(I_n, Z) \leq -1 + \xi_n \end{aligned}$$

The above constraint states that positive examples should score better than 1 (the margin), while negative examples, for all configurations of parts and mixtures, should score less than -1. The above optimization is a quadratic program (QP), and specifically an instance of a structural SVM [14], for which many solvers exist. We use the dual coordinate-descent QP solver in [23]. When selecting a sub-category mixture m in Equation (5), we found it useful to calibrate the scores returned from each mixture m using Platt rescaling [17].

4. Experiments

4.1. Performance Metric

Our dataset (Section 2) consists of annotated images, where for each image, all pairs of people are labeled with a set of active touch codes. We evenly partition our dataset into a train and test set. Given a test image, we consider two scenarios: (1) We assume we have ground-truth face locations or (2) We use face detections from a commercial face detector with removed false positives. Note we must deal with false negatives in the latter case, as we will later show. We score the ability of our system to correctly label touch codes for each pair of faces. Specifically, we evaluate a proxemic model p with its ability to retrieve “correct”

pairs from the test set: we score it on each pair with Equation (5), and generate a precision-recall curve by thresholding the score. We then compute average precision (AP) as in [5].

4.2. Comparison with Baselines

We first compare our **Joint** model with two baselines. The first baseline is the **Sequential** algorithm described in Section 3.1. The second is the **Visual Phrases** algorithm [19] which directly models complex visual relationships involving two objects as a single phenomenon. Both baselines are trained (using publicly-available code [6, 23]) on the same training data as our models. In Figure 7 we plot the AP for each proxemic under two cases. In Figure 7(a) all three algorithms use the ground-truth head locations from our proxemics database. In Figure 7(b) none of the algorithms had access to true location of the heads, but estimated them using a face detector [20].

The first thing to note is that in both scenarios **Joint** outperforms both baselines across all six proxemics. For example, when using the ground-truth face locations, the average AP is 54.1% compared to 28.2% using **Sequential** and 26.4% using **Visual Phrases**. While these results illustrate how difficult the problem is, the improvement using **Joint** is huge.

As illustrated in Figure 3, **Sequential** fails mainly because the pose estimation algorithm is simply not robust enough. On a sample dataset consisting of two people interacting, we found that part localization accuracy dropped from 86.6% for the shoulders to 45.6% for the elbows and 24.4% for the hands. This dropoff in robustness with distance from the head is illustrated in Figure 3 where **Sequential** is competitive for the shoulder-shoulder touch code. Note that **Sequential** may fare better when trained with a mixture of articulated models, tuned for each proxemic category. We consider such a baseline below in Section 4.3.

Perhaps somewhat surprisingly, **Visual Phrases** do not

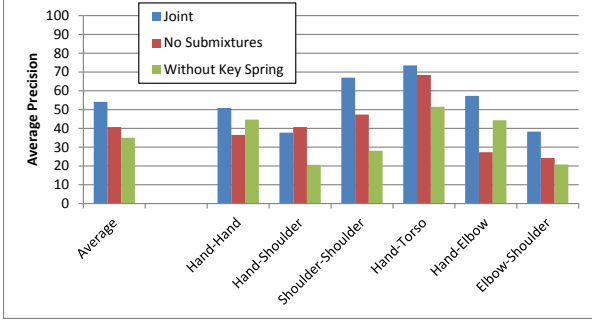


Figure 8: Ablation studies to show importance of using proxemic Sub-categories and Proxemic Dependent Geometry. Please see text for further descriptions.

perform any better than **Sequential**. While **Visual Phrases** do capture some deformation, they are limited because their parts are not articulated, and moreover part locations are estimated independently given the base location. This contrasts to our pictorial structure model where the positions of articulated parts are tightly constrained along the chain connecting the heads of the two people. See Figure 5. As we will show below in Section 4.3, the spring connecting the touching body parts is vital to our approach.

The result for all 3 algorithms when using the face detector in Figure 3(b) are worse across the board. The reason is quite simply that the face detector only detects around 80% of the faces. When either of the two faces is not detected, there is no way for any of the algorithms to classify the proxemic correctly. In such cases we assign all three algorithms a very large negative score which reduces the recall, while preserving the precision. The false negative rate for our face detector is somewhat high due to the large number of extreme poses with partial occlusions. While our approach would clearly benefit from a better face/person detector, relative performance of the three algorithms remains consistent in both ideal and realistic conditions.

Finally, though our joint model improves performance across all touch codes, we found less of an improvement for touch codes which tended to be confused with one another (e.g., “hand-shoulder” vs. “elbow-shoulder”).

4.3. Ablation Studies

In Figure 8 we present a comparison between **Joint** and two variants to help explain where the performance improvement is coming from. **No Submixtures** does not use the sub-categories described in Section 3.2.1. Otherwise the algorithm is identical. **Without Key Spring** is identical to **Joint**, and does use the sub-mixtures, except that the spring connecting the two people is removed. See Figure 5. The **Without Key Spring** model is two disconnected trees/chains.

The results in Figure 8 show these two components in

our model to be vitally important. **No Submixtures** shows that mixtures help capture the large amount of visual appearance variation within each proxemic class. **Without Key Spring** can be thought of as an augmented **Sequential** baseline. It is augmented in that (1) the articulated pose model uses submixtures and (2) two consistent submixtures are required to fire for each pair of people. Indeed, **Without Key Spring** does outperform **Sequential**. However, the additional spatial constraint encoded by the key spring is even more useful than these other factors.

4.4. Effect of our approach on Pose Estimation

Though we have focused on proxemic classification of touch codes, our joint model also provides better estimates of pose. We show qualitative results on Figure 9. Quantitatively, we evaluate pose estimation by computing the fraction of times a model correctly predicts the location of the two touching parts (using the criteria from [4]). Across all touch codes, **Joint** correctly predicts locations 73.6% of the time, **Without Key Spring** performs at 62.5%, while **Sequential** performs at 47.5%. **Joint** reports significantly more accurate poses because it better models occlusions and exploits multi-body spatial constraints.

5. Conclusions

In this paper, we introduced the problem of proxemic recognition in consumer photographs. To foster future research in the area, we have created a dataset that will be freely available for research purposes. We showed the importance of joint modeling of body pose estimation and proxemic recognition to enable synergies between the two problems. In the process, we also showed the serious failure modes of existing pose estimation, in the presence of multiple interacting people in the image.

This area of proxemic recognition is still in its infancy, and we have barely touched the surface. This means that we have a huge arena for future directions, both in the algorithm side and in the space of interactions that can be defined. Extensions along the algorithmic side can involve, for instance, enabling competition between the touchcodes that helps to leverage interesting relationships between body parts, and thereby serving as good indicators for recognition. In a parallel thread, currently, we have used touch codes for ease of labeling. However, it is a lot more interesting when we can map them to semantics such as “hugging”. Another potential direction is identifying a more broader type of activity (eg. a birthday party or a skiing retreat?) that the interacting people are engaged in, by possibly, making use of other cues in the image.

Acknowledgements: The research described in this paper was conducted while YY was an intern at Microsoft Research. DR was supported by NSF Grant 0954083 and ONR-MURI Grant N00014-10-1-0933.



(a) Hand hand (b) Shoulder shoulder (c) Hand shoulder (d) Hand elbow (e) Elbow shoulder (f) Hand torso

Figure 9: We show sample results for pose estimation. On the top, we show results of **Sequential**, which independently estimates poses of each person. In the middle, we show **Without Key Spring** where the spring connecting the two bodies is removed. In the bottom, we show our **Joint** algorithm. Our joint approach produces more reliable pose estimates because it better models occlusions and spatial constraints specific to each touch code.

References

- [1] M. Argyle and B. Foss. *The psychology of interpersonal behaviour*. Penguin Books Middlesex. England, 1967. 1
- [2] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *CVPR*, 2011. 2
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 4
- [4] M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. In *ECCV*, 2010. 2, 7
- [5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 6
- [6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE PAMI*, 32, 2009. 2, 4, 6
- [7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1), 2005. 4
- [8] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE PAMI*, 31(10), 2009. 2
- [9] E. Hall. A system for the notation of proxemic behavior. *American anthropologist*, 65(5), 1963. 1
- [10] E. Hall. *The hidden dimension*, volume 6. Doubleday New York, 1966. 1, 2
- [11] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *CVPR*, pages 1933–1940. IEEE, 2009. 2
- [12] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE PAMI*, 22, 2000. 2
- [13] N. Ikizler, R. Cinbis, S. Pehlivan, and P. Duygulu. Recognizing actions from still images. In *ICPR*, 2008. 2
- [14] T. Joachims, T. Finley, and C. Yu. Cutting plane training of structural SVMs. *Machine Learning*, 77, 2009. 6
- [15] T. Lan, Y. Wang, W. Yang, and G. Mori. Beyond actions: Discriminative models for contextual group activities. In *NIPS*, 2010. 2
- [16] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. Reid. High five: Recognising human interactions in tv shows. *BMVC*, 2010. 1, 2
- [17] J. Platt. Probabilistic outputs for support vector machines. *Bartlett P. Schoelkopf B. Schurmans D. Smola, AJ, editor, Advances in Large Margin Classifiers*, 1999. 6
- [18] D. Ramanan. Part-based models for finding people and estimating their pose. In T. Moeslund, A. Hilton, and L. Sigal, editors, *Visual Analysis of Humans*. Springer, 2011. 2
- [19] A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 2, 6
- [20] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS 18*. MIT Press, 2006. 6
- [21] Y. Wang, H. Jiang, M. Drew, Z. Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006. 2
- [22] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, pages 2030–2037. IEEE, 2010. 2
- [23] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *CVPR*, 2011. Code available at <http://phoenix.ics.uci.edu/software/pose/>. 3, 4, 5, 6
- [24] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 2