Abstract

We advocate the inference of qualitative information about 3D human pose, called posebits, from images. Posebits represent boolean geometric relationships between body parts (e.g., left-leg in front of right-leg or hands close to each other). The advantages of posebits as a mid-level representation are 1) for many tasks of interest, such qualitative pose information may be sufficient (e.g., semantic image retrieval), 2) it is relatively easy to annotate large image corpora with posebits, as it simply requires answers to yes/no questions; and 3) they help resolve challenging pose ambiguities and therefore facilitate the difficult task of image-based 3D pose estimation. We introduce posebits, a posebit database, a method for selecting useful posebits for pose estimation and a structural SVM model for posebit inference. Experiments show the use of posebits for semantic image retrieval and for improving 3D pose estimation.

1. Introduction

While tremendous effort has focused on the extraction of quantitative 3D human pose from images and video, in this paper we consider the estimation of qualitative pose information, called posebits. Posebits are attributes of pose that specify the relative positions or orientations of body parts. They have the advantage that one can easily collect posebit image annotations for training purposes, and they can be reliably inferred from images. Further, they are useful for resolving 3D pose ambiguities and for myriad other tasks where quantitative pose is not required.

The effective use of both generative and discriminative approaches to pose estimation require training data comprising image features and corresponding 3D poses (e.g., see [?, 4, 14, 20, 22]). In practice these data are difficult to obtain, either from high-fidelity commercial marker-based MoCap (Motion Capture) systems [21] or from RGB-D systems such as Microsoft Kinect. They are limited primarily to indoor lab environments, and require significant data curation. Manual annotation of 3D pose from images is not an effective alternative as it is very time-consuming and prone to errors [5, 6].

By contrast, it is relatively simple to obtain training data for posebits from human annotations. Indeed, people often perceive and express pose in terms of the relative positions between body parts (see Fig. 1), rather than absolute 3D position or joint angle representations that are commonly used in pose estimation tasks. For example, common human verbalizations of pose are: the left leg in front of right leg, left hand in front of the torso, etc. It is therefore quite natural to explore the design, inference and use of mid-level, qualitative pose representations.
Posebits can also be viewed as attributes of pose. Advantages of attributes have been demonstrated for object categorization [9, 26], and human action recognition [13, 29] with emphasis on transfer learning between classes. Pose attributes have been used for content-based MoCap retrieval [16]. Attributes have also been used to retrieve action-specific priors to stabilize tracking [2]. But none of these approaches infer pose attributes directly from images.

Finally, our work is inspired by work on poselets [6], a new notion of parts, which do not necessarily correspond to intuitive body parts (e.g. as in [1, 28]). It is argued that the detection of configurations of body is often easier than single parts. However, whereas poselets have shown good performance for people detection, here we focus on estimating the 3D pose information from single images. Unlike poselets, we do not require 3D annotations for training.

3. Posebits

Posebits represent binary, geometric relationships between body parts. They may be useful for myriad tasks, in and of themselves, or as an intermediate representation, e.g., toward 3D pose estimation.

Broadly speaking, we consider three types of posebits which appear relevant to 3D pose inference. But we do not rule out other types that might be relevant to other tasks. The three types, depicted in Fig. 2, are:

1. **Joints distance**: Posebits are activated when two joints in the body are closer or further than a given threshold.
2. **Articulation angle**: Posebits are activated when a given joint angle is bent more than \( \alpha \) degrees.
3. **Relative position**: Posebits are activated when a body part A is to the left, right, above, below, in front or behind relative to a second body part B. To determine such posebits, the signed distance between body part A and a plane centered at body part B is computed.

Further, while one might identify hundreds of useful specific posebits, initial exploration of the concept focused only on a relatively small set of 30 candidates, chosen at random from among the three types listed above.

3.1. Posebits database

For selecting, learning and inferring posebits, we exploit a MoCap corpus and a collection of annotated images, which we call the Posebit Database (PbDb). As discussed above, to date we have only annotated images with 30 posebits, but ideally one might want to have annotations with many more than 30 posebits.

At present, PbDb comprises 1) a MoCap database comprising 10000 poses taken from Human-Eva [21] and HMODB [17], and 2) a set of 4000 images, each annotated with 30 posebits. Images were collected from four
publicly available databases. There are 1500 images from Human-Eva [21], 1500 images from HMODB [17], 685 images from Fashion [27] and 305 from Parse [28].

Human-Eva and HMODB come with 3D pose annotations, so it is trivial to compute the corresponding poses using simple geometric tests, such as point to point, or point to plane, distances, or by thresholding joint angles. Fashion and Parse images do not have 3D pose annotations. However it is straightforward to obtain pose annotations using Amazon Mechanical Turk, where takers simply answer yes/no questions about each image. Indeed, based on our initial data collection this is an effective way to gather annotations for a much larger image corpus and for many more than 30 poses. The PbDb image dataset is split into two subsets of 1995 images for training and testing. Fig. 3 shows the result of querying PbDb with a small subset of poses to obtain semantically similar images.

### 3.2. Selection

Posebits may be effective in different ways. They may be sufficient for some tasks directly. Or they may be useful as an intermediate representation. Here we focus on their use as a mid-level encoding to facilitate 3D pose inference. It is also clear that different posebits may be useful for different tasks, or redundant. Hence choosing a good set of posebits is essential. To this end we advocate the use of a simple selection mechanism, inspired by decision trees, to choose subsets of posebits from PbDb.

For a given task (e.g., 3D pose estimation), we aim to select a subset of posebits \( S_m \) from a pool of candidates \( S_C \) (i.e., PbDb). To this end we use two criteria: Useful posebits are those that can be reliably inferred from image features \( \mathbf{r} \), and that help reduce uncertainty in the hidden variable of interest, \( \mathbf{x} \). Selection makes use of small set of training pairs of image features and 3D poses \( L = \{ \mathbf{r}_i, \mathbf{x}_i \}_{i=1}^L \), and a larger set of 3D poses \( \mathcal{U} = \{ \mathbf{x}_j \}_{j=1}^P \).

To make the problem tractable, we select posebits greedily, one bit at a time, using a forward selection mechanism.

### 3.2.1 Clustering

Let a vector of \( m \) posebits, called a posebyte, be denoted by \( \mathbf{a} = (a_1, \ldots, a_m) \in \mathcal{A}^m \), where \( \mathcal{A} = \{-1, 1\} \). To reduce pose ambiguity, we want posebits that minimize the entropy of \( p(\mathbf{x}|\mathbf{a}) \), i.e., at each leaf of the binary tree. Thus, when adding the \( j \)-th posebit, the clustering information gain is computed as \( I_j^C = H_j - H_j \), where \( H_j \) is a weighted sum of entropies at each node of the \( j \)-th level of the tree:

\[
H_j = \sum_{c=1}^{2^j} \frac{|S^c|}{|S^x|} H(S^c).
\]

Here, \( S^x = \mathcal{U} \) is the set of MoCap poses, \( S^c \subseteq (\mathcal{U}) \) is the subset of poses \( \mathbf{x}_j \), in posebyte class \( c \), and \( H(S) \) is the differential entropy of the pose density for the cluster.
Entrophy is difficult estimate with high-dimensional data, so we use the cluster variance as a surrogate for entropy. While a crude assumption, variance provides a measure of cluster compactness and works well in practice. Fig. 4 shows how the conditional pose distribution becomes more concentrated as one travels down from the root to the leaves.

3.2.2 Reliability

A good posebit should also be inferred reliably from image features, and provide as much information about pose as possible. As a simple measure of the extent to which they provide information about pose we consider a information measure in which posebits constitute the only intermediate information available from which one can infer pose.

In more detail, let $x \in \mathcal{X}^D$ be a target variable, such as 3D pose, let $r \in \mathcal{R}^d$ denote image features. Marginalizing over the posebytes $a$, and supposing that all information about pose is mediated by the posebytes, we consider an approximation to the posterior $p(x|r)$, i.e.,

$$Q(x|r,m) = \sum_{a \in \mathcal{A}^m} p(x|a) p(a|r).$$

(3)

Here, $p(x|a)$ is the conditional pose distribution, and $p(a|r)$ is posterior posebyte distribution.

When posebits are reliably inferred from the image features, we expect $Q(x|r)$ to approach the ideal case in which the ground truth pose is known, i.e., $Q^{opt}(x|r) = p(x|a_{\hat{n}})$. To this end, we express the reliability information gain in terms of the average KL-divergence between $Q(x|r,j)$ and $Q^{opt}(x|r,j)$ at level $j$ of the binary tree. The reliability information gain for adding a posebit to the set is defined as $I^j = D^j_{KL} - D^j_{KL}$, where $D^j_{KL}$ the discrete KL-divergence based on training pairs in the labeled set $L = \{r_i, a_i, x_i\}_{i=1}^n$,

$$D^j_{KL} = \sum_i Q^{opt}(x_i|r_i,j) \log \left( \frac{Q^{opt}(x_i|r_i,j)}{Q(x_i|r_i,j)} \right).$$

(4)

Note that although $Q^{opt}(x|r)$ is harder to approximate as we go down the tree, the information gain increases as $Q^{opt}(x|r)$ carries much more information about the pose.

In what follows we consider simple ways to approximate the two key elements of $Q$ that are required for this selection criterion, namely, $p(x|a)$ and $p(a|r)$.

Modeling $p(a|r)$: We model $p(a|r)$ in terms of the score functions for discriminative classifiers for each posebit in PbDb\textsuperscript{5}. To each posebit we train an SVM classifier with a linear score function, e.g., $F_j(r)$ for posebit $j$. The score function for a given posebyte, denoted $\hat{G}(a, r)$, is then used to define the density for $p(a|r)$.

In particular, for computational convenience, avoiding summations over $2^m$ posebytes (e.g., in (3)), we consider only the top $N$ ranked posebytes,\textsuperscript{3} $a_n$, $n = (1 \ldots N)$, thereby approximating $p(a|r)$ as a multinomial distribution $p(a|r) = \sum_{n=1}^N \pi_n \delta(a - a_n)$ where $\pi_n$ are computed from the scores using soft-max

$$\pi_n = \frac{\exp \left( \hat{G}(a_n, r)/\tau \right)}{\sum_{s=1}^N \exp \left( \hat{G}(a_s, r)/\tau \right)},$$

(5)

where $\tau$ is a temperature parameter set to 0.5. In this setting, Eq. (3) becomes a class conditional mixture model with weights proportional to the posebyte probabilities

$$Q(x|r) = \sum_{n=1}^N p(x|a_n) p(a|r) \approx \sum_{n=1}^N \pi_n p(x|a_n),$$

(6)

And therefore the KL-divergence in Eq.(4) above becomes

$$D^j_{KL} \approx \sum_i p(x_i|a_n) \log \left( \frac{p(x_i|a_n)}{\sum_{n=1}^N \pi_n p(x_i|a_n)} \right).$$

(7)

Modeling $p(x|a)$: To model $p(x|a)$ we bin the poses in a MoCap database in $2^m$ classes, one for each posebyte. Recall that, given a pose $x$ its corresponding posebyte description is easily obtained with simple geometric computations.

We then represent each class distribution by computing $k$-medoids obtaining $K$ representatives $\{x_{k,a_n}\}_{k=1}^K$ for class $a_n$. To avoid unwanted bias we assume the $K$ poses are equally probable, i.e., $p(x|a_n) = \frac{1}{K} \delta(x - x_{k,a_n})$. In this way, we always sample a fixed set of $K$ codeposes per class, see Fig. 7.

With this approximation to $p(x|a)$, and the multinomial approximation to $p(a|r)$ we obtain a model for $Q$ in terms of a weighted set of samples; i.e., $Q(x|r) = \sum_{k=1}^K \frac{1}{K} \delta(x - x_{k,a_n})$, with $k \in \{1 \ldots K\}$ and $n \in \{1 \ldots N\}$, with weights $w_{k,a_n} = \frac{1}{K} \pi_n$.

3.2.3 Selection Experiments

To demonstrate the efficacy of our selection method we compare the influence of the selected posebits on the performance of our pose estimation algorithm (Sec. 4.1). Our algorithm uses the general model in Eq. 3 to generate pose

sum of the $m$ single-posebit score functions. This allows us to evaluate the information gain without having to train a new structural SVM every time a new posebit is added to the set.

\textsuperscript{3}we use $N = 4$ in all experiments.
For learning we only require an image dataset with posebit values the posebit can take. The i-th posebit would then be a 3.3. Posebits Classifier

The set of m posebits generated and selected by our method (Sec. 3.2) form the posebyte a = (a1, . . . , am) ∈ Am. We infer the posebyte directly from raw image features r ∈ Rd using a model based on structural SVM [25]. For learning we only require an image dataset with posebit labels I = {r, a, b}i=1.

With a structural SVM the discriminant function for a single posebit F : Rd × A → R provides a score for the values the posebit can take. The i-th posebit would then be estimated by maximizing this function

\[ \hat{a}_i = \arg \max_{a_j} F(r, a_j, w_{a_j}) = w_{a_j}^T \phi(a_j, r) \] (8)

where \( \phi(r, a_j) = a_j r \) is the joint feature map of input r and output a_j, and \( w_{a_j}^T \) is the vector of weights to be learned.

While such score functions for a single posebit were used above for selecting good posebytes, they do not exploit the shared information among posebyte classes, i.e., classes with similar posebit strings will be semantically more similar in pose space. To this end, we learn a discriminant function \( G : Rd × Am \rightarrow R \) over input output pairs from which we can derive prediction by maximizing over the response variable a for a given input r. The joint SVM scoring function is expressed as

\[ \hat{a} = \arg \max_{a \in A^m} G(r, a, \beta_a) = a^T B r + b^T \psi(a) \] (9)

where \( \beta_a = [B(:, b)] \) is the vector of all weights to be learned, \( B \in \mathbb{R}^{d \times m} \) is a matrix the rows of which define separating hyperplanes for the posebits, \( \psi(a) \) is a potential that captures posebit co-occurrences. For efficiency and to prevent over-fitting we factorize the prior \( \psi(a) \) in pair-wise terms, so Eq. (9) becomes

\[ G(r, a, \beta_a) = \sum_{j} b_{a_j}^T \phi(r, a_j) + \sum_{j} \sum_{k} b_{a_j, a_k} \psi(a_j, a_k) \] (10)

where \( b_{a_j}^T \) is the i-th row of B providing the score of posebit \( a_j \) and \( \psi(a_j, a_k) \) is the pairwise potential capturing co-occurrences. The pair-wise potentials consist of normalized histograms learned from MoCap data. Since the prior only depends on the output variable it can be precomputed resulting in substantial computational savings.

Equation (10) can be written as a scalar product \( G(r, a, \beta_a) = \langle \beta_a, \Phi(r, a) \rangle \) between the vector of weights \( \beta_a \) and the joint feature map \( \Phi(r, a) \). Having zero training error means that the model scores better the correct posebytes than any other posebyte. Learning the weights \( \beta_a \) involves solving the quadratic optimization problem:

\[ \min_{\beta_a, \xi} \frac{1}{2} ||\beta_a||^2 + \frac{C}{M} \sum_{i=1}^M \xi_i \\
\text{s.t.} \quad \forall i, \quad \forall a \in A^m \setminus a_i, \xi_i > 0 \]

\[ \langle \beta_a, \Phi(r_i, a_i) - \Phi(r_i, a) \rangle \geq 1 - \frac{\xi_i}{\Delta(a_i, a)} \]

The above constraint states that the true output \( a_i \) should score at least a unit better (the margin) than the best runner-up. The objective function penalizes violations of these constraints using scaled slack variables \( \xi_i \). Intuitively, violation of a margin constraint associated with a high loss \( \Delta(a_i, a) \) is penalized severely. We do this by scaling the slack variables with the inverse loss \( \Delta(a_i, a) \). The loss is simply the Hamming distance between posebytes \( a_i \) and \( a \).

### 3.3.1 Classification Experiments

To learn the model in Eq. (10), we used the training images of the annotated image set, \( I = \{r, a, b\}_{i=1}^M \). Assuming a bounding box of the person, we construct the feature vector r by computing spatial pyramid features [10], which are spatially localized HOG (Histogram of Oriented Gradients) over increasing cells of sizes 8, 16, 32 and 64 pixels. Histogramming over larger windows adds robustness to misalignments in the training data.
Figure 6. **Classification accuracies** for top 10 posebits selected by our algorithm, when applied to test images from each of the four databases. Colored bars correspond to individual posebits. For instance the left-most two bars (blue and red) correspond to posebits *Right hand above the hips?* and *Right foot in front of the torso?* respectively. The black right-most bar is the average accuracy over the 10 posebits. Performance is very good for Human-Eva, HMODB and Fashion. On the Parse dataset some of the posebits are not reliably detected due to the high variability in the poses seen in the images. For example the posebit *Left hand to the left of the torso?* is not reliably estimated for Parse. This might be due to a bias in our dataset, i.e., we do not have enough positive examples for that posebit. Other posebits such as *Right hand above the neck?* (fifth bar from the left) is accurately classified in all datasets. Note however, that it is selected fifth because other posebits were deemed more informative, despite having lower test accuracies.

Figure 6 depicts the classification accuracies in the test sets of the four datasets, H-Eva, HMODB, Fashion and Parse, i.e., the fraction of test images for which the classifier was correct for a given posebit. Our model can predict posebits from images with remarkably high accuracies (70-90). The dataset where we perform more modestly is Parse. That is probably due to the high variability in pose and appearance and due to the fact that we only use 150 images for training (one order of magnitude less than for the other datasets). Since there is more redundancy in H-Eva and HMODB, better accuracies can be obtained. Notably, we obtain good accuracies across datasets even though a single model was trained using a joint dataset as opposed to training separate models.

4. Experiments

Here we report two more experiments with the use of posebits, one with monocular 3D pose estimation and one with pose-based image retrieval.

4.1. 3D Pose Estimation

We first consider the use of posebits for 3D pose estimation. The goal is to demonstrate a reduction in pose ambiguity that stems from the use of posebits. To that end, we use the \(Q(x|r)\) in Eq. (3) as a proposal distribution during inference. However, unlike it’s construction in the posebit selection algorithm, here we use the \(m\)-bit structural SVM, rather than single posebit classifiers, to obtain the classifier scores \(G(r, a)\) (see Sec.3.3).

To demonstrate a reduction in uncertainty we use a simple top-down generative model that uses of \(Q(x|r)\) to generate pose hypotheses. Given some image features \(z\) for pose estimation, we express the posterior as

\[
p(x|z, r) \propto p(z|x, r)p(x|r) \approx p(z|x)Q(x|r) \tag{11}
\]

where \(z\) and \(r\) are assumed to be conditionally independent given the pose \(x\).

**Image Likelihood:** Many research papers have focused on the design of high-fidelity likelihood models, such as [7], but while the likelihood is a key ingredient in pose estimation, it is not the primary focus of our work. Instead, here we assume that unlabeled 2D joint locations are available, perhaps obtained from a 2D pose estimation algorithm. Hence, the image features \(z = (m_1 \ldots m_J)\) consist of a collection of 2D points \(m_i \in \mathbb{R}^2\).

Let \(F(x; j) : \mathcal{A}^D \rightarrow \mathbb{R}^3\) be a function that maps a pose \(x\) to the \(j\)-th 3D joint position. We model \(p(z|x)\) as a product of isotropic 2D Gaussians centered at joint locations:

\[
p(z|x) = \frac{1}{C} \exp \left( - \sum_{i=1}^{p} e^2(m_i|x) \right) . \tag{12}
\]

where \(C\) is a normalization constant, and \(e(m_i|x)\) is the Euclidean distance between the 2D measurement and the closest 3D joint projected into the image:

\[
e(m_i|x) = \min_{j} \|m_i - \text{Proj}(F(x; j))\| . \tag{13}
\]

Here, \(\text{Proj}\) projects 3D points to the image plane. We a scaled orthographic projection, where the scale is set to match the person’s height in the image plane.
The pose estimate is given by the mode of the posterior \( p(\mathbf{x}|\mathbf{z}, \mathbf{r}) \), obtained by evaluating the \( K \times N \) poses from the proposal distribution \( Q(\mathbf{x}|\mathbf{r}) = \{x_k, a_n, w_k, a_n\} \). Recall, that \( Q(\mathbf{x}|\mathbf{r}) \) is represented by \( K \) poses for each of the \( N \) classes of the top ranked posebytes. To build the model for \( p(\mathbf{x}|a) \), the poses in the Posebit MoCap set are scaled to a unit pose, i.e., all bones are re-scaled by the size of a template pose. In addition, all poses are centered at the origin and the yaw angle\(^4\) is set to zero\(^5\).

The root orientation is estimated by uniformly sampling rotations \((\theta_{\text{root}})\) about the vertical axis at 32 equi-spaced angles. Let \( \mathcal{M}(\theta; \mathbf{x}) \) be a function that rotates a pose \( \mathbf{x} \) by \( \theta \) degrees. Then, the pose estimate is obtained by maximizing

\[
x^* = \arg \max_{\mathbf{x}_k, a_n} \left( \max_{\theta_{\text{root}}} (p(\mathbf{z}|\mathcal{M}(\theta_{\text{root}}; x_k, a_n)) w_k, a_n) \right)
\]

where \( x_k, a_n \) is the \( k \)-th pose of the class corresponding to posebyte \( a_n \), and \( w_k, a_n \propto p(\mathbf{x}|\mathbf{r}) \) are the importance sampling weights (see Sec. 3.2.2). In Fig. 7, we show an example of how \( Q(\mathbf{x}|\mathbf{r}) \) is used to reduce uncertainty about pose. A diagram of the approach is shown in Fig. 8 left.

**Validation:** We test the algorithm on the H-Eva sequences and report the mean pose error. Fig. 8 right shows mean pose error as a function of the number of posebits. As expected, with increasing numbers of posebits, the inference becomes less ambiguous and estimator accuracy thereby increases. The best results are obtained using 12 of the 30 random posebits currently in PbDb. That said, we think that 10 is a good trade-off between accuracy and annotation effort required to collect training data. Notice the big drop in pose error as we increase the number of posebits. We also show qualitative results in Fig. 9(a).

Our current unoptimized Matlab implementation runs at an average of 22 frames per second using 10 posebits, 4 mixtures and 10 code-poses per class.

\(^4\)The viewpoint w.r.t. the camera is arbitrary
\(^5\)For more implementation details see the supplemental material

4.2. Image Retrieval

Posebits may be useful for many applications beyond pose estimation. Here we consider image retrieval based on pose attributes. That is, posebits inferred from an image are used to retrieve other images in the DB with similar poses. We use the top ranked posebytes by the classifier to retrieve images with the similar posebyte strings. Qualitative results are shown in Fig. 9.

5. Conclusions

We introduced posebits, a semantically powerful pose descriptor. Experiments show that our selection method learns a good set of posebits, i.e., retains those that can be reliably inferred from images and are informative about the pose. We have also shown that using posebits as a mid-layer representation can improve monocular pose estimation. One advantage of the proposed method is that human annotation is easier and more intuitive. This enables easy collection of training data. Experiments reveal that posebits can resolve many of the monocular ambiguities and can be useful as basis for many potential applications. In particular, we do not see posebits as a competitor to existing approaches but rather as a powerful complementary feature. For future work, we plan on annotating more data, and to explore more posebit applications.

**Acknowledgments.** This work was also partly funded by the ERC grant DYNAMIC MINVP. D.J.F was funded in part by the Canadian Institute for Advanced Research (CIFAR), NSERC Canada, and GRAND NCE.

**References**

Figure 9. (a) Pose estimation results for images in PbDb. (b) Retrieval: We can use the inferred posebits from the image to retrieve images in our database with similar posebit annotations. In particular, here we retrieve images with posebyte annotations that match any of the top 2 ranked posebytes given by our model. We show the query images marked in red on the left column and the retrieved images on the right. Notice the semantic similarity in the images.


