# Regression-based Hand Pose Estimation from Multiple Cameras

Teófilo E. de Campos and David W. Murray
Department of Engineering Science, University of Oxford
Oxford OX1 3PJ, UK
{teo, dwm}@robots.ox.ac.uk

## Abstract

*The RVM-based learning method for whole body pose estimation proposed by Agarwal and Triggs is adapted to hand pose recovery. To help overcome the difficulties presented by the greater degree of self-occlusion and the wider range of poses exhibited in hand imagery, the adaptation proposes a method for combining multiple views. Comparisons of performance using single versus multiple views are reported for both synthesized and real imagery, and the effects of the number of image measurements and the number of training samples on performance are explored.*[1]

## 1 Introduction

Discriminative approaches have recently became widely explored for the challenging problem of estimating the 3D pose of articulated objects from 2D images. The idea is to recover a direct, but not physically-based, mapping between a robust representation of appearance and the model parameters such as joint angles. The approach exploits the fact that the typically explored range of hand poses is much smaller than the potential range.

One approach to relating image measurements qualitatively to 3D poses is that of classification, where a discrete set of 3D poses constitutes the set of classes. Training samples are generated using synthetic images of a hand model at several poses and an efficient classifier, usually based on a decision tree, is used [2, 8, 9]. Although high accuracy can be obtained, these frameworks demand a large set of classes if a comprehensive range of recoverable poses is desired, which makes the computation time prohibitive.

An alternative is to use strong temporal priors as done in [4] for walking people. But unlike walking, hand movements are not typically cyclic, which presents a difficulty for HMM-based methods, leading to the need of more complex graphical models, as presented in [11].

[1]This work was supported by CAPES (Brazil) and EPSRC (UK).

Another category of discriminative methods for 3D pose estimation is that of regression, where a continuous map between image measurements and 3D poses is created. This expands the range of possible poses and leads to smoother estimates if the method is used for tracking. Rosales *et al*. [7] proposed a system that uses a non-linear supervised learning framework, the specialised mappings architecture, which are learnt using a set of training pairs of image measurements and 3D poses. The measurements used are seven invariant Hu moments.

Agarwal and Triggs [1] used richer image measurements, shape contexts [3], and a sparser mapping method, which is based on Relevance Vector Machine (RVM) [13]. Their method has been extended to include a dynamical model and also to consider multiple hypotheses to deal with ambiguities in a probabilistic manner. Sminchisescu *et al*. [10] contemporaneously proposed a similar method for pose estimation and tracking.

In this paper, a regression-based approach to hand pose recovery is taken, following in part Agarwal and Triggs' work on whole body pose estimation.

However, the hand pose recovery is in general a more difficult problem, not least because of the far greater degree of actual occlusion, and of "apparent" occlusion where finger bounding contours are lost. For this reason this paper proposes an extension of the single view method to multiple cameras, an approach which Erol *et al*. [5] point out has not been widely explored for this problem. An experimental comparison of single and multiple view performance is presented, taking into account variation in the number of image measurements and training samples needed.

## 2 Extracting Multiple View Image Descriptors

The initial step of the method (both in training and application phases) is the conversion of each image of a hand into a silhouette contour, and thence into a compact description using shape contexts [3]. Because of the wide variation in scale and orientation of hands in imagery, it is important

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

to incorporate invariance to these transformations within the context. A novel modification for rotation invariance is proposed. Its description is followed by the description of our method for combination of multiple view information.

Recovery of the silhouette of the hand, assumed ungloved, is achieved using a trained histogram-based skin colour classifier applied, for robustness, to the Cr and Cb channels of the YCbCr chromatic space. Images are subsampled to $90 \times 120$ pixels to reduce computation cost. In our database, hands occupied about 20% ($\pm 6.2\%$ STD) of the image pixels. The shape contexts are computed only from positions on the silhouette contour, which is easily derived by edge detection in the resulting skin/not-skin binary image (see Fig. 1).

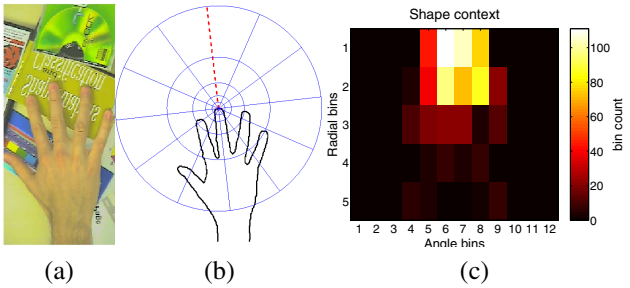


**Figure 1. A hand image (a), its extracted silhouette contour (b), and the shape context vector (c) obtained from the middle finger tip.**

A shape context [3] is a local non-parametric description of shape computed at each point on the silhouette contour. Neighbouring contour pixels are accumulated in 60 bins arranged in log-polar fashion, five along the radial direction and twelve around the polar angle, spaced equally in log-distance and angle, respectively. To provide a first layer of scale invariance, the inner radius is set to $\mu/8$, where $\mu$ is the mean of the distances between all pairs of points in the silhouette. The radius increases in octaves to $2\mu$, typically covering all of the hand silhouette. The resulting 60-bin histogram is normalised, providing again for scale-invariance. For image $i$ the complete image description is generated as the set of $n_i$ 60-bin histograms computed at $n_i$ points along the silhouette contour.

Belongie *et al*. [3] ensured rotational invariance by aligning the fiducial $0^\circ$ line of the shape context with the tangent to the silhouette contour at each point. While this works well if the contour is smooth (which in our experience requires either large images or fitting parametrised curves to the edges), the result in low resolution images, and using pixel contour points, was found to be noisy. A more robust alternative is to use the geometric centre of the silhouette and set the fiducial line to be orthogonal to the line from the centre to the contour point. The rotation invariance of both tangent-based and centroid-based methods is obtained at the cost of reducing the amount of global information about the shape of the silhouettes.

The solution adopted in this paper is to orient the shape contexts with the hand axis. For simplicity, it is assumed that two points of the silhouette contour lie on the image borders, and these points are taken to be either side of the forearm. The silhouette point that is the furthest away from the end of the forearm is classified as the hand tip, so the vector between the forearm and the hand tip is taken as the hand axis. The results in Fig. 2 show that this maintains the discrimination power of non-rotation invariant shape contexts and adds robustness to planar rotations. Note that using the principal axis, the fingers ambiguity is avoided (row a). In row (b) tangent-based and principal axis-based rotation invariant shape contexts provided better results than the shape contexts without rotation invariance.

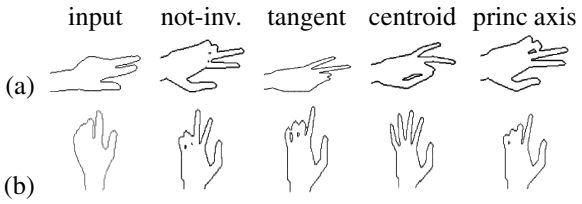


**Figure 2. Nearest-neighbour classification or the two silhouettes on the left using different methods to orient shape contexts.**

The dimensionality required to describe an image is reduced by quantising the shape context manifold into a codebook using $K$-means ($K = 90$ in our experiments). Since shape contexts are histograms, the natural dissimilarity measure for histograms is the $\chi^2$ test statistic [3]. To soften the effects of spatial quantisation, histograms are built by allowing context vectors to vote with Gaussian weights into the few centres nearest to them [1]. These histograms are normalised w.r.t. the number of points in the silhouette contour, again for scale invariance. The obtained vectors are robust to small shape variations and to noise in segmentation.

Three ways to combine multiple view information have been considered. The low level approach is to group all the shape contexts from all the images together before clustering to build the histograms. The problem of this approach is that the improvement obtained by using multiple views may not be very significant, as one set of measurements can be associated with more than one global orientation.

An alternative is to estimate the pose from each view individually and combine the results at a high level using, for example, a graphical model. If global pose parameters can be estimated using triangulation, and if regressors can be

trained with a comprehensive samples set, then the same regressor can be applied for all the cameras, and the setup of cameras may not need to be the same as in training. However, as discussed later, it is not realistic to use comprehensive training sets.

The approach proposed here is to combine the information at an intermediate level, by generating description vectors $\mathbf{x}$ for each camera individually and concatenating them into a higher dimensional vector that describes the current measurements from all the cameras. The regressor is then trained using these concatenated vectors. This provides the best trade-off between complexity and robustness: it is robust to planar rotations and to translations along the cameras axes, but retraining is necessary if other modifications of the camera pose happen. In our implementation, $K$ was set to 30 for each of three views, so the concatenated vector $\mathbf{x}$ has length 90. The projection onto the two principal axes of the 90-d manifold for the training data is shown in Fig. 3, using hand axis-oriented shape contexts.

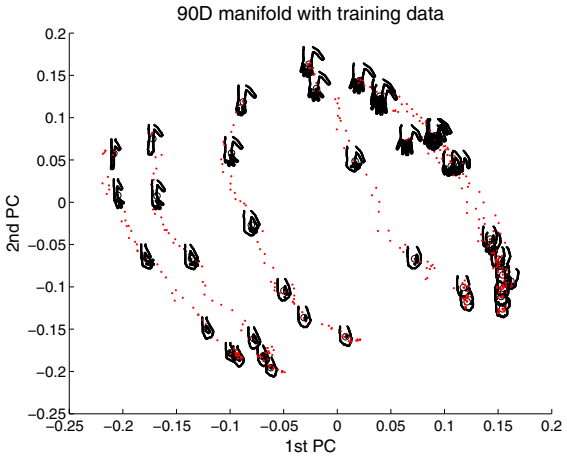


**Figure 3. 90-d manifold of multiple view $\mathrm{x}$ vectors obtained from the training data.**

Note that the first and second principal components are roughly aligned with the variation in $\theta_Z$ and with the overall degree of flexion of the fingers, respectively. This hints that this dataset of hand appearances can roughly be represented with two degrees of freedom. This effect cannot be observed for single view descriptors $\mathbf{x}$. In that case, the manifold seems to need at least three dimensions to show more separability between hands poses.

## 3 Obtaining and Testing the Training Data

An essential input to the later regression process is, of course, the association of each $\mathbf{x}_i$ with a set of known joint angles $\mathbf{y}_i$. For this paper, training pairs $(\mathbf{x}_i, \mathbf{y}_i)$ were obtained by generating imagery synthesised from a hand model using joint angle data from the hand trajectory database prepared by Stenger *et al*. [11].

A hand model including forearm, palm, thumb and fingers was created using generalised cylinders and spheres. The palm is rigid, and each finger is modelled as a planar mechanism with 3 dof for flexion and 1 dof for abduction and adduction with the palm. The same model is used for the thumb, but its plane is not parallel to the fingers' planes. This gives a total of 20 internal dof plus 2 inactive dof for the wrist, and 6 dof of global pose parameters. Thus the hand pose is described by vector $\mathbf{y} \in \mathbb{R}^{28}$.

In this paper, we use two training sets. The first set, dubbed *open-close*, consists of a trajectory that starts with all the fingers stretched and a grasping gesture is performed in 78 frames. The glove used to generate this data did not have a global position and orientation sensor, so the trajectory was duplicated seven times for $15°$ spaced values $0° \leq \theta_Z \leq 90°$, giving a total of 546 poses. For desktop tasks the variation of the other orientation parameters ($\theta_X$ and $\theta_Y$) is usually small enough to enable us to rely on the invariance properties of the modified shape contexts. A more accurate global orientation can be obtained by triangulation when multiple views are used. For a fair comparison between single and multiple view, $\theta_X$ and $\theta_Y$ are not taken into account. For multi-camera application, the hands were rendered from three different viewpoints.

The second training set, dubbed *complex*, was generated from a sequence of 239 internal poses in which fingers move independently. As before, the trajectory was reproduced for seven instances of $\theta_Z$, giving a total of 1673 three-dimensional poses.

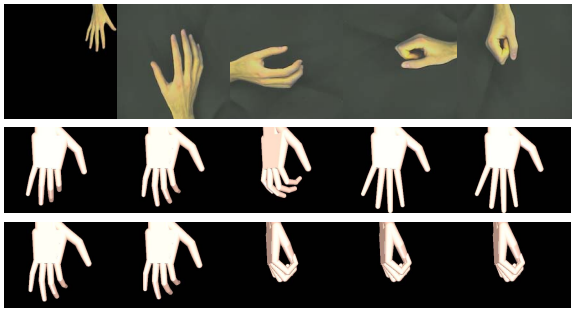

**Figure 4. $1^{st}$ row: sample images from the top view with modifications in orientation, translation and scale. The nearest-neighbour classification results using single and multiple views are shown in the $2^{nd}$ and $3^{rd}$ rows.**

In order to assess the discriminatory power of the image descriptors $\mathbf{x}_i$, a nearest neighbour classification exper-

iment was performed with 36 hand images – 9 hand poses taken from 4 orientations. The results, shown in Fig. 4, suggest that the image descriptor is robust enough to provide a good qualitative description of the hand shape from images that are not in the training set, even though the hand model is not accurate. The same figure also shows that the use of multiple views can improve the nearest neighbour classification result.

## 4 Learning to Relate Descriptors to 3D Poses

To relate the image descriptors $\mathbf{x}_i$ to the 3D joint and pose settings $\mathbf{y}_i$, Agarwal and Triggs [1] proposed the use of a regression method that learns the relation between $I$ pairs of vectors $(\mathbf{x}_i, \mathbf{y}_i)$ by estimating the coefficients or weights of a linear combination of basis functions $\phi_k$. The problem is described as:

$$\mathbf{y}_i = \sum_{k=1}^{p} \mathbf{a}_k \phi_k(\mathbf{x}_i) + \boldsymbol{\epsilon} \equiv \mathtt{A}\mathbf{f}(\mathbf{x}_i) + \boldsymbol{\epsilon} \qquad (1)$$

where $\boldsymbol{\epsilon}$ is a residual error vector, $\mathbf{y}_i \in \mathbb{R}^m$ $(i = 1, 2, \cdots, I)$, and $\mathbf{a}_k \in \mathbb{R}^m$ $(k = 1, 2, \cdots, p)$. For compactness, the weight vectors can be gathered into an $m \times p$ matrix $\mathtt{A} \equiv (\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_p)$ and the basis functions into a $\mathbb{R}^p$-valued function $\mathbf{f}(\mathbf{x}) = (\phi_1(\mathbf{x})\ \phi_2(\mathbf{x})\ \cdots \phi_p(\mathbf{x}))^\top$. As discussed later, $p = K$ for linear kernels, and $p = I$ for Gaussian kernels.

For $I$ training pairs, the estimation problem takes the form

$$\mathtt{A} := \arg\min_{\mathtt{A}} \left\{ \sum_{i=1}^{I} ||\mathtt{A}\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i||^2 + R(\mathtt{A}) \right\} \qquad (2)$$

where $R(\cdot)$ is a regulariser on $\mathtt{A}$. Gathering the training vectors into an $m \times I$ matrix $\mathtt{Y} \equiv (\mathbf{y}_1\ \mathbf{y}_2\ \cdots\ \mathbf{y}_I)$ and a $p \times I$ feature matrix $\mathtt{F} \equiv (\mathbf{f}(\mathbf{x}_1)\ \mathbf{f}(\mathbf{x}_2)\ \cdots\ \mathbf{f}(\mathbf{x}_I))$, equation (2) can be rewritten as:

$$\mathtt{A} := \arg\min_{\mathtt{A}} \left\{ ||\mathtt{A}\mathtt{F} - \mathtt{Y}||^2 + R(\mathtt{A}) \right\} \ . \qquad (3)$$

For unidimensional signals $y$, Tipping [13] proposed the use of Relevance Vector Machine (RVM), a method based on sparse Bayesian learning to estimate efficiently a good approximation of $\mathtt{A}_{(1\times p)}$ with large sparsity. A straightforward extension for multidimensional patterns can be achieved by regressing input vectors $\mathbf{x}$ against each individual parameter $y_j$ (of vector $\mathbf{y}$). The obtained row vectors of weights can be concatenated into matrix $\mathtt{A}_{(m\times p)}$.

With the *open-close* data set, using $K = 90$ (i.e. $K = 30$ for each view) and linear kernel functions ($\mathbf{f}(\mathbf{x}) = \mathbf{x}$), the resulting $\mathtt{A}$ matrix is shown in Fig. 5 (top row). For samples in the training set, this resulted in the mean absolute error (computed by $\sum_i^I |\mathtt{A}\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i|/I$) of $2.8°$, and mean standard deviation of $2.4°$. The maximum average error and standard deviation were $34.0°$ and $28.8°$ respectively, but both occurred for the interphalangeal joint of the thumb, which is occluded in many of the training images.

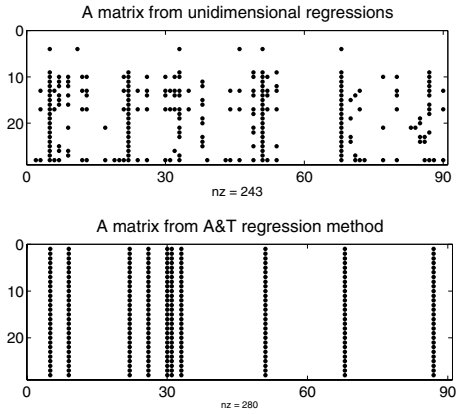


**Figure 5. Map of non-zero elements of matrix $\mathtt{A}_{(m\times p)}$ resulting from: (top) RVM regression of individual parameters separately, and (bottom) linear regression using [1].**

A problem with regressing parameters independently is that noisy data can potentially provide impossible output poses. For example, a regressor trained to recover 3D pose of walking humans might output poses having both legs to the front. Furthermore, training each row of $\mathtt{A}$ individually can cost too much computational time.

A method that estimates the whole matrix $\mathtt{A}$ in a single process, creating a linear combination of relations with multi-dimensional output is described in [1]. This regressor is estimated by direct optimisation of the weights keeping the hyperprior $\nu$ fixed.

The first step is the initialisation of $\mathtt{A}$ with *ridge regression*. The regulariser is chosen to be $R(\mathtt{A}) \equiv \lambda||\mathtt{A}||^2$, where $\lambda$ is a regularisation parameter. The problem can be described as the minimisation of

$$||\mathtt{A}\tilde{\mathtt{F}} - \tilde{\mathtt{Y}}||^2 := ||\mathtt{A}\mathtt{F} - \mathtt{Y}||^2 + \lambda||\mathtt{A}||^2 \,, \qquad (4)$$

where $\tilde{\mathtt{F}} \equiv (\mathtt{F}\ \lambda\mathtt{I})$ and $\tilde{\mathtt{Y}} \equiv (\mathtt{Y}\ \mathtt{0})$. $\mathtt{A}$ can be estimated by solving the linear system $\mathtt{A}\tilde{\mathtt{F}} = \tilde{\mathtt{Y}}$ in least squares. Ridge solutions are not equivariant under scaling of inputs, so both $\mathbf{x}$ and $\mathbf{y}$ vectors are scaled to have zero mean and unit variance before solving.

The next step is to successively approximate the penalty terms with "quadratic bridges". Therefore, with $a$ an element of $\mathtt{A}$, the regularisers $R(a) = \nu \log ||a||$ are approximated by $\nu(||a||/a_{scale})^2 + const$, which has the same gradient as the original function. If $const$ is set to

$\nu(log||a_{scale}|| - \frac{1}{2})$ the regularising function values match at $a_{scale}$. Quadratic bridges approximation allows parameters to pass through zero if they need to, with less risk of premature trapping and over-fitting.

Agarwal and Triggs proposed the use of column-wise set of priors in the regulariser $R(\mathtt{A})$: with $\mathbf{a}$ a column of $\mathtt{A}$, $R(\mathbf{a}) \approx \nu(||\mathbf{a}||/a_{scale})^2 + const$, implying that the estimated matrix $\mathtt{A}$ has some whole columns $\mathbf{a}_k \to \mathbf{0}$. Depending on the kernel functions used, two different aspects of cost reduction for pose estimation can be achieved. If *linear basis functions* are used, i.e., $\mathbf{f}(\mathbf{x}) = \mathbf{x}$, the nil vectors $\mathbf{a}_k$ indicate which components of vectors $\mathbf{x}$ can be removed without compromising the regression result. Therefore this method can be used as a *feature selector*, resulting in a reduction in the number of shape descriptors needed.

Alternatively, *kernel basis functions* can be used. They are expressed by $\phi_i(\mathbf{x}) = \mathcal{K}(\mathbf{x}, \mathbf{x}_i)$, making $\mathbf{f}(\mathbf{x}) = [\mathcal{K}(\mathbf{x}, \mathbf{x}_1), \mathcal{K}(\mathbf{x}, \mathbf{x}_2), \cdots, \mathcal{K}(\mathbf{x}, \mathbf{x}_n)]^\top$, where $\mathcal{K}(\mathbf{x}, \mathbf{x}_i)$ is a function that relates $\mathbf{x}$ with the training sample $\mathbf{x}_i$. For example (as used in this paper), one can use Gaussian kernels $\mathcal{K}(\mathbf{x}, \mathbf{x}_i) = e^{\beta||\mathbf{x}-\mathbf{x}_i||^2}$, with $\beta$ estimated from the scatter matrix of the training data. In this case, the column-wise sparsity of $\mathtt{A}$ acts as a method to select relevant training samples.

The estimation of $\mathtt{A}$ is then performed in a similar fashion to Equation 4, by iteratively solving the linear system: $\mathtt{A}(\mathtt{F}\ \mathtt{R}) = (\mathtt{Y}\ \mathtt{0})$, where $\mathtt{0}$ is a $m \times p$ matrix of zeros and $\mathtt{R}$ is a $p \times p$ matrix whose columns are defined by $\nu/a_{scale}$, and $a_{scale}$ is the norm $||\mathbf{a}||$ of each column vector of $\mathtt{A}$ from the previous iteration. To reinforce sparsity, the columns of $\mathtt{A}$ whose norms are smaller than a threshold $\mathcal{T}_\mathbf{a}$ are set to zero. This process is repeated until convergence of $\mathtt{A}$.

Fig. 5 (bottom row) shows the $\mathtt{A}$ matrix obtained by this method using linear kernel functions in the *open-close* data set, with multiple view 90-d descriptors $\mathbf{x}$. The threshold $\mathcal{T}_\mathbf{a}$ was tuned to lead to the selection of 10 relevant features, resulting on the selection of 5 features from camera 1 (side view), 3 features from camera 2 (top view), and 2 features from camera 3 (another side view). For samples in the training set, regression with this matrix resulted in the mean absolute error of $2.7°$, and mean standard deviation of $2.0°$. The maximum average error and standard deviation were $11.8°$ and $8.2°$ respectively, both for the interphalangeal joint of the thumb. This represents an improvement in comparison to the results obtained by regressing the dofs individually, with a simplification of matrix $\mathtt{A}$, allowing feature and sample selection. It is interesting to note that many of the vectors selected using Tipping's method coincide with rows selected by Agarwal and Trigg's method, confirming a consistency between these methods.

It has been observed that Gaussian kernel functions can provide better results at the expense of being slower than linear kernel functions [1]. Indeed, the results showed later suggest that linear functions are less stable to noise than Gaussian kernel functions. The alternative proposed here is to combine both by first reducing the dimensionality of the descriptors $\mathbf{x}$ with feature selection and then using regression with Gaussian kernel functions to select the most relevant samples. Since the dimension of the vectors $\mathbf{x}$ is reduced in the first stage, all the distance calculations required to compute $\mathbf{f}(\mathbf{x})$ with Gaussian kernels are sped up.

## 5 Experiments and Results

### 5.1 Number of Relevance Vectors

The graphs of Fig. 6 show the number of selected relevance vectors as a function of the threshold $\mathcal{T}_\mathbf{a}$. Note that the same threshold leads to the selection of more relevance vectors for a single view. This hints that even though the same number of training samples (and of the same dimensionality) is used in both cases, fewer relevance vectors are selected for multiple views, indicating that their measurements are more discriminative. Fewer samples and fewer features are needed to achieve the same relevance for multiple views.

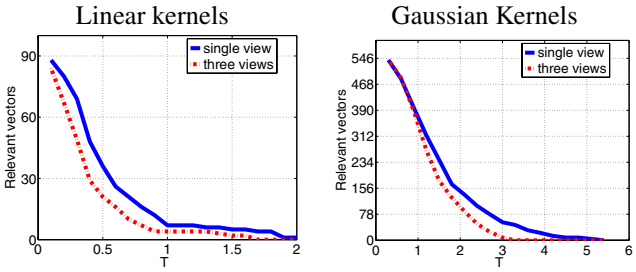


**Figure 6. Number of selected relevance vectors for linear and Gaussian kernels for single and multiple views as a function of the threshold $\mathcal{T}_\mathbf{a}$ evaluated for the *open-close* set.**

### 5.2 Synthetic Images

For the experiments with synthetic images, ground truth is available. The data set was evenly split in a training set and a testing set (with no intersection) from the same sequence of movements. Although this practice makes training and testing data very similar, it is enough to distinguish the performance between single and multiple view methods.

Table 1 shows a quantitative evaluation of the results for both data sets using synthetic images. The columns 'sel. ftrs.' and 'smpls.' indicate how many relevance vectors were selected with linear and Gaussian kernels, respectively. The column 'worst result' shows the average error

| # Views | Data Set | Kernels | Sel. Ftrs. | Sel. Smpls. | Avg. Error | STD | Worst Result | Which dof |
|---|---|---|---|---|---|---|---|---|
| 1 | *open-close* | lin. | 3 | 273 | 8.6° | 6.9° | 23.2° | $\theta_Z$ |
| | | gauss. | 90 | 10 | 5.6° | 4.3° | 14.5° | T IP |
| | | both | 13 | 29 | 2.3° | 2.0° | 6.0° | $\theta_Z$ |
| | *complex* | lin. | 31 | 839 | 3.0° | 2.7° | 11.4° | M DIP |
| | | gauss. | 90 | 42 | 2.9° | 2.5° | 9.9° | M DIP |
| | | both | 35 | 36 | 2.9° | 2.6° | 10.7° | M DIP |
| 3 | *open-close* | lin. | 2 | 273 | 5.4° | 4.4° | 17.0° | T IP |
| | | gauss. | 90 | 10 | 3.6° | 2.7° | 14.9° | T IP |
| | | both | 12 | 29 | 1.6° | 1.2° | 7.0° | T IP |
| | *complex* | lin. | 31 | 839 | 2.5° | 2.1° | 8.9° | M DIP |
| | | gauss. | 90 | 41 | 2.4° | 2.0° | 8.3° | M DIP |
| | | both | 34 | 36 | 2.4° | 2.0° | 9.0° | M DIP |

**Table 1. Results with synthetic data obtained using 273 and 839 training samples for *open-close* and *complex* data sets, respectively.**

for the parameter (dof) whose estimate was the worst, indicated in the column 'which dof'. The abbreviation T IP refers to the thumb's inter-phalangeal joint, and M DIP to middle finger's distal inter-phalangeal joint.

As expected, the worst estimates occurred in two cases: (i) for dofs related to parts of the hand whose contour was occluded in many of the images, and (ii) for the rotation $\theta_Z$ when a single view is used, as this is not a rotation parallel to the top view image plane.

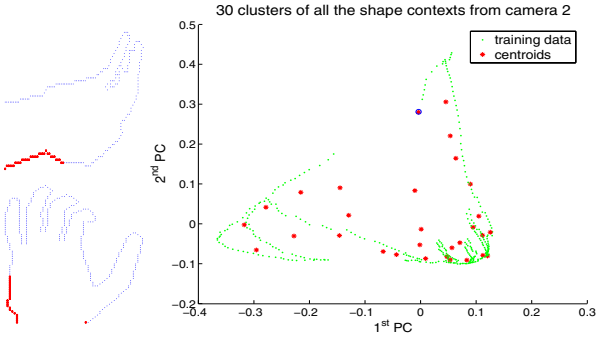


**Figure 7. Left: Silhouettes obtained from a sample pose in the training set from camera 1 (top) and 2 (bottom), highlighting (with red '*') the points whose shape context is taken into account after the selection of two relevant features. Right: Shape contexts manifold with the centroid of the selected cluster from camera 2 indicated by a blue circle.**

The sequence of movements in the *open-close* dataset can roughly be described by two degrees of freedom: flexion of the all joints and twisting movement of the hand about the forearm axis ($\theta_Z$). In order to verify the ability of the regressor to identify this, a feature selection experiment was performed, tuning the threshold $\mathcal{T}_\mathbf{a}$ to select only two relevance vectors. But for a *single-view*, three features were selected, because any greater threshold resulted in only one feature. For *three-views*, one vector from the top view and another vector from one of the side views (camera 1) were selected, as shown in Fig. 7.

Note that, for both views, the centroids selected are close to the wrist rather than the finger tips. A possible reason for that is that features closer to the finger tips present too much variation between samples and they are not present in some of the samples, *e.g.* those with the hand in fist pose. This has also been observed for single view.

The obtained regression results (see Fig. 8 and table 1) show that the regressor is able to give a rough approximation of the pose using a minimal set of selected vectors (in this case, image features). Even using less features for multiple views it is possible to achieve higher accuracy than with a single view. It was also observed that, for single view, as $\theta_Z$ grows, the pose estimate gets poorer because the top view does not offer enough distinct features on its own when the fingers get nearly aligned to the camera axis.

When using Gaussian kernels, it is harder to intuit the minimal set of samples needed to estimate the pose. $\mathcal{T}_\mathbf{a}$ was chosen so that 10 relevant samples were selected from the training set, and the results are shown in table 1.

Both for single and multiple views, the selected samples are mostly from 'near-fist' hand poses. This may seem odd, but it is usual in an RVM-based system for the the most relevant samples to be distant from the obtained pose estimates, and for them not to be the most comprehensive samples in terms of the variability of state (poses) [13].

Fig. 9 reports the application of feature selection followed by samples selection to combine speed and performance. Note that the superiority obtained for multiple views is more evident for $\theta_Z$. The pose of the hand was estimated individually for each frame, which explains the jittering trajectory.

In general, the improvement obtained by using multiple views is evident, particularly when the number of features used is small. However the improvement is view-dependent, and if a single view captures the most meaningful silhouette the improvement is diminished. A further reduction in improvement arises because the synthetic images used so far are noise free. As shown in next section, the rotation and scale improvement is restored when using real images.

### 5.3 Real Images

For real images, whole training sets were used, giving 1679 training pairs for the *complex* data set. For testing,

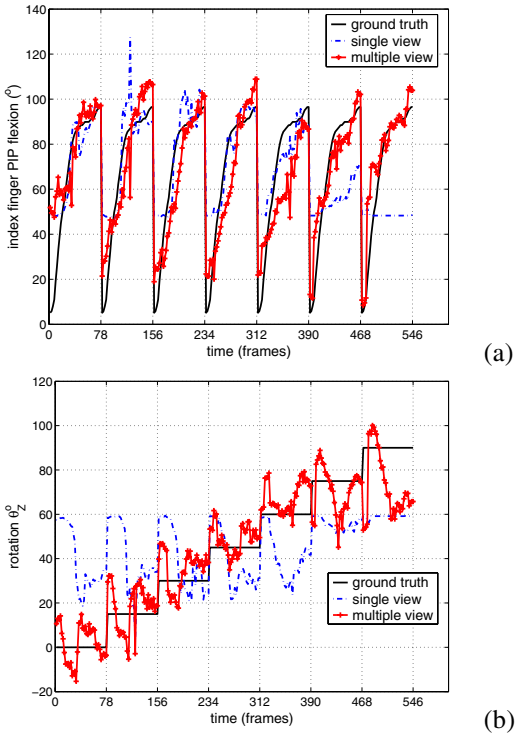


**Figure 8. Regression results using three (for single view) and two (for multiple views) relevance vectors (out of 90), with linear kernels: (a) estimated angle of the interphalangeal joint of the index finger; (b) estimated angle $\theta_Z$ of global rotation about the forearm.**

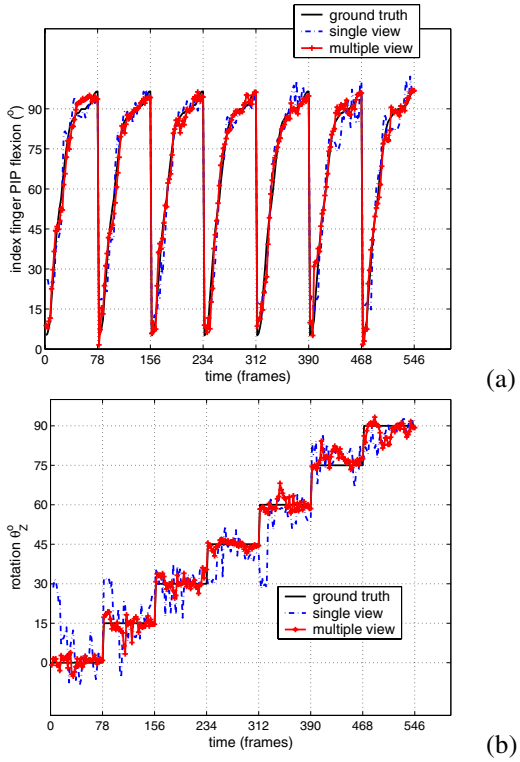


**Figure 9. Regression results combining both feature selection and samples selection. The parameters were tuned to select 13 or 12 features for single and multiple view, respectively; and 29 samples.**

images of the right hand of a single subject were used. Since there is no ground truth available for the real images, only qualitative results are shown.

Fig. 10 shows that multiple views provide a significant improvement over single view data. This improvement becomes more evident when a small selection of features and samples is used, as shown in Fig. 11. Note that, for a single view, the regressor seems to be unable to recover some of the poses, probably because the measurements generate poses that extrapolate the space of trained poses.

## 5.4 Computational Cost

As expected, the training phase, which is done off-line, is very demanding both in terms of memory and CPU usage, especially in the clustering for vector quantisation. However, once the histogram descriptors $\mathbf{x}$ are obtained, training the regressor is not so expensive: it takes between 7s for linear kernel functions using 32 features, and 328s for Gaussian kernels using all features and 38 samples. This is reduced to 305s if only 32 selected features are used. These measurements were obtained running a Matlab implementation on a 2.4GHz Pentium 4 computer, averaging the timing for the *complex* data set.

In the application phase, the extraction of the image descriptor $\mathbf{x}$ is the only step whose computational cost is $\mathcal{O}(C)$, where $C$ is the number of cameras. The average time for this step is 170ms per image.

The actual pose estimation process is extremely fast, taking between 7.2$\mu$s for linear kernel functions using 32 features, and 35.7$\mu$s for Gaussian kernels using all features and 38 samples. This is reduced to 25.4$\mu$s if a subset of 32 selected features are used. Therefore, using the most expensive parameters, the application of the algorithm takes 652ms per frame if three cameras are used.

In terms of memory usage, the computation of the histogram $\mathbf{x}$ of all the shape contexts is the most expensive part. If Gaussian kernel functions are used, matrix $\mathtt{A}$ is $\mathcal{O}(m \times I)$, which has shown to be not so demanding even using all the 1679 training samples.

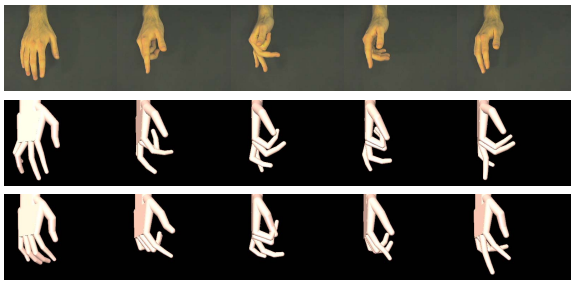

**Figure 10. Results obtained from real images (top row) for single (middle row) and multiple views (bottom row), using Gaussian kernels with all the samples and all the features.**

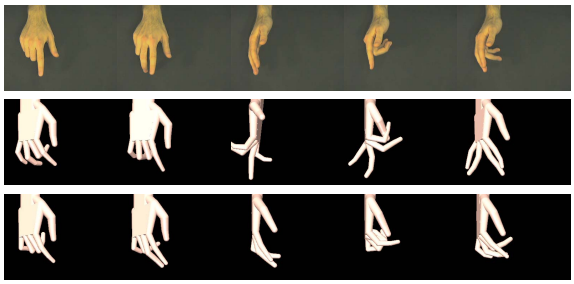

**Figure 11. Similar to fig. 10, but using a subset of 32 features and 38 selected samples.**

## 6 Conclusions

This paper has presented a regression-based method for estimation of hand pose in 3D from multiple view image descriptors, advancing the single-view method of Agarwal and Triggs [1] proposed for human pose estimation. We have shown that the use of rotation and scale invariant image descriptors can reduce the number of training samples needed, provided triangulation is first used to recover the global pose parameters. The images descriptors are combined at an intermediate level into a multiview descriptor by concatenation. The mapping between multiview descriptors and 3D poses is learnt using Agarwal and Triggs' [1] extension regressor based on RVM.

Our experiments have, inter alia, examined the effects of feature selection and sample selection both on the quality of pose determination and on the computational time, using both synthetic and real imagery. We have found that linear kernel functions have the advantage of computational cost independent on the amount of training data used. However, we have found Gaussian kernel functions to be more robust, so we have performed experiments combining both linear and Gaussian kernels for speed and robustness. Our experiments have also shown that, for general views, fewer relevance vectors are needed in the multiple view case. Their measurements are more discriminative, allowing correct pose estimates to be recovered in cases where a single view all but fails.

An obvious modification to the current image descriptor would involve the use of a better coding method, like Gaussian mixtures or Jurie and Triggs's method [6]. Another possibility is to explore the extension of RVM for multidimensional target spaces of Thayananthan *et al*. [12] which, like the original RVM, optimises the hyperparameters. But the main thrust of future work will be to evaluate how relevant is the use of multiple hypotheses if multiple views are employed. A more application-oriented direction of this work is the integration with a generative tracker for real-time results.

## References


[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE T-PAMI*, 28(1):44–58, 2006.

[2] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *Proc CVPR*. IEEE, 2004.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE T-PAMI*, 24(24):509–522, 2002.

[4] M. Brand. Shadow puppetry. In *Proc 7th ICCV*. IEEE, 1999.

[5] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. A review on vision-based full dof hand motion estimation. In *Workshop on Vision for HCI, in conjunction with CVPR*. IEEE, 2005.

[6] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc CVPR*. IEEE, 2005.

[7] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. 3D hand pose reconstruction using specialized mappings. In *Proc 8th ICCV*. IEEE, 2001.

[8] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc 9th ICCV*. IEEE, 2003.

[9] N. Shimada, K. Kimura, and Y. Shirai. Real-time 3-D hand pose estimation based on 2-D appearance retrieval using monocular camera. In *Proc Int WS RATFG-RTS*, 2001.

[10] C. Sminchisescu, A. Kanaujia, Z. Lio, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *Proc CVPR*. IEEE, 2005.

[11] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE T-PAMI*, 2006. In press.

[12] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. In *Proc ECCV*. Springer, 2006.

[13] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *JMLR*, 1:211–244, June 2001.