Statistical Pose Estimation with Local Dependencies

Josef Pösl *

Lehrstuhl für Mustererkennung (Informatik 5)
Universität Erlangen–Nürnberg
Martensstr. 3, D–91058 Erlangen, Germany
email: {poesl,niemann}@informatik.uni-erlangen.de

Abstract

This paper shows how the use of partial dependencies can improve statistical pose estimation. The pose estimation is hierarchically performed on different scale levels. We define a density for local features, which are extracted for all image positions. The presented theory allows arbitrary dependency structures in the context of Bayes nets. We show how the density parameters can be estimated and how the global pose search on the starting level of the hierarchy can be computed efficiently by a couple of filter banks. The paper presents the results of 2–D as well as 3–D experiments.

1 Introduction and Motivation

Statistical object modeling without segmentation is motivated by speech understanding. Recognition results were strongly enhanced by the idea of incorporating statistical methods in the recognition task [11]. Recent publications [2, 12, 7] show, that a statistical framework can also successfully be established in image object recognition.

Most of the publications construct statistical models based on segmentation results. They use the geometric information provided by results, like lines or vertices, as random variables. There are two major disadvantages of using solely segmentation results. When restricting the recognition process to this level of abstraction a lot of information contained in an image is lost. Another disadvantage are the errors made by the segmentation. Segmentation results may be incomplete or located incorrectly.

*The author is member of the Center of Excellence 3-D Image Analysis and Synthesis sponsored by the „Deutsche Forschungsgemeinschaft“.

One way to cope with the problem is to avoid segmentation. Instead of that the gray–level information of an image can be used. Various publications (see e.g. [8, 12, 5, 13, 7, 6]) deal with gray–level based object recognition. A short overview can be found in [9, 10]. Most of them either restrict the pose of the object, consider only part of the image information or are computationally expensive. We describe a pose estimation technique based on a functional basis decomposition of probability density parameters in [9, 10]. Local features are the basis for the recognition process. They can be extracted from wavelet decompositions [1] for example. In our previous papers the feature vectors were treated as independent random variables. This publication will show how the consideration of partial dependencies can enhance the recognition results and provide the necessary theory. The focus thereby is on the definition of local dependencies. Complete object dependencies as introduced for example by eigenvector decompositions lack the possibility to handle object occlusions easily and compute the density values efficiently.

2 System overview

The aim of the presented system is the pose estimation of a rigid 3–D object in a single 2–D gray–level image. We assume that the object does not vary in scale.

In a first step of the localization process a multiresolution analysis of the image is used to derive feature values on different scales $s \in \mathbb{Z}$ and resolutions (sampling rates) $r_s \in \mathbb{R}^+$ at the locations of rectangular sampling grids. Given an image $f(x, y)$ with $x \in \{0, 1, \ldots, D_x - 1\}$ and $y \in \{0, 1, \ldots, D_y - 1\}$, the observed feature val-

ues at scale level s are denoted by $\boldsymbol{c}_s(x,y) = (c_{s,0},\ldots,c_{s,N-1})^{\mathrm{T}}$ ($x \in \{0, r_s,\ldots, r_s D_x - 1\}$, $y \in \{0, r_s,\ldots, r_s D_y - 1\}$). In the experiments of this paper the features $\boldsymbol{c}_s$ are chosen as the logarithmic coefficients of the scaling functions — that are the low pass coefficients — of a discrete wavelet transform ($N = 1$). We use the symmetric Johnston 8–TAP wavelet.

With those features a statistical measure for their probability under the assumption of an object transformation can be defined. The complexity of the pose estimation is high if all features on the different scale levels are combined into one measure function. Therefore, a hierarchical solution is used (Figure 1). Measures are defined for each scale and the localization is performed for each level successively. Let $\tilde{\boldsymbol{c}}_s$ be the vector of the concatenated feature values detected in an image on scale s, $\boldsymbol{B}_s$ the model parameters of an object class and $\boldsymbol{R}, \boldsymbol{t}$ be the 3–D rotation matrix and translation vector. The rotation $\boldsymbol{R}$ is defined by the rotation angles ϕ_x, ϕ_y and ϕ_z round the x–, y– and z–axis respectively.

The model parameters $\boldsymbol{B}_s$ consist of geometric information like probability density locations and other density parameters. The density $p(\tilde{\boldsymbol{c}}_s|\boldsymbol{B}_s, \boldsymbol{R}, \boldsymbol{t})$ is then used for localization. The maximum likelihood estimation results in $(\widehat{\boldsymbol{R}}_s, \widehat{\boldsymbol{t}}_s) = \mathrm{argmax}_{(\boldsymbol{R},\boldsymbol{t})}\, p(c_s|\boldsymbol{B}_s, \boldsymbol{R}, \boldsymbol{t})$.

3 Statistical model

3.1 Model formulation

This section shows the definition of a probability density function on each of the scale levels of the analysis. To simplify the notation the index s is omitted.

The model object is covered with a rectangular grid of local feature vectors (see Figure 2). The grid resolution is the same as the image resolution on the actual scale. Let $A \subset \mathbb{R}^2$ be a small region (e.g. rectangular) which contains the object projection to the image plane for all possible rotations $\boldsymbol{\phi}_{ext} = (\phi_y, \phi_x)$ outside the image plane and constant $\boldsymbol{\phi}_{int}$ and $\boldsymbol{t}$ (see Figure 2). Let $X = \{\boldsymbol{x}_m\}_{m=0,\ldots,M-1}$, $\boldsymbol{x}_m \in \mathbb{R}^2$ denote the grid locations and $\boldsymbol{c}(\boldsymbol{x})$ the feature vector at location $\boldsymbol{x}$. In this paper we choose the grid locations as the sampling locations of the image transformation: $X = \left\{\boldsymbol{x}_m = (kr_s, lr_s)^{\mathrm{T}}\right\}$. The local features $c_n(\boldsymbol{x}_m)$ are the components of the image feature vector $\tilde{\boldsymbol{c}}$ if the object is not transformed in the image plane: $c_n(\boldsymbol{x}_m) = c_n(kr_s, lr_s) = c_{k,l,n} = c_{m,n}$. The local features $c_{m,n}$ are interpreted as random variables. The randomness thereby is, among others, the consequence of noise in the image sampling process and complex changes in environment (e.g. lighting) conditions. Assuming the background features as independent of themselves and independent of the object features leads to

$$p(\tilde{\boldsymbol{c}}|\boldsymbol{B}) = p(\boldsymbol{c}_A|\boldsymbol{B}) \prod_{\boldsymbol{x}_m \notin A} p(\boldsymbol{c}(\boldsymbol{x})|\boldsymbol{x} = \boldsymbol{x}_m)$$

where $\boldsymbol{c}_A$ is the subset of $\boldsymbol{c}$ which is covered by A. If a uniform distribution for the features outside the model area A (which belong to background) is assumed, the second product in the above equation is constant. So it is sufficient to consider $p(\boldsymbol{c}_A|\boldsymbol{B})$.

The grid positions and the model area A are part of the model parameters $\boldsymbol{B}$.

The feature vectors are assumed to be normally distributed. Let $\mathcal{N}(\boldsymbol{c}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the normal densities, where $\boldsymbol{\mu}$ is the mean vector with concatenated local feature mean vectors $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}$ is the covariance matrix with elements $\sigma_{m,\bar{m},n} = \mathrm{cov}(\boldsymbol{c}_m, \boldsymbol{c}_{\bar{m}})$.

The density parameters are a function of the rotation parameters ϕ_y, ϕ_x for 3–D object rotations perpendicular to the image plane., so that:

$$\begin{aligned} & p(\boldsymbol{c}_A|\boldsymbol{B}, \boldsymbol{R}, \boldsymbol{t}) \\ &= p(\boldsymbol{c}_A|(\boldsymbol{\mu}(\phi_y, \phi_x), \boldsymbol{\Sigma}(\phi_y, \phi_x)), \boldsymbol{R}, \boldsymbol{t}) \\ &= \mathcal{N}(\boldsymbol{c}_A(\phi_z, \boldsymbol{t}_{2D})|\boldsymbol{\mu}(\phi_y, \phi_x), \boldsymbol{\Sigma}(\phi_y, \phi_x)), \end{aligned}$$

with $\boldsymbol{c}_A(\boldsymbol{R}(\phi_z), \boldsymbol{t}_{2D})$ as the concatenated feature vectors $\boldsymbol{c}(\boldsymbol{R}(\phi_z)\,\boldsymbol{x}_m + \boldsymbol{t}_{2D})$, the 2–D rotation matrix $\boldsymbol{R}(\phi_z)$ for the rotation and the translation $\boldsymbol{t}_{2D}$ in the image plane. The image feature vectors at the transformed 2–D locations are calculated by linear interpolation. Assuming continuous functions $\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$ they can be rewritten using a basis set for the domain of two-dimensional functions $\{v_r\}_{r=0,\ldots,\infty}$ with coordinates $a_{m,n,r}, b_{m,n,r} \in \mathbb{R}$ ($r = 0,\ldots$) and the elements $\tilde{\sigma}_{m,\bar{m},n}$ of the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$:

$$\mu_{m,n} = \sum_{r=0}^{\infty} a_{m,n,r} v_r, \quad \tilde{\sigma}_{m,\bar{m},n} = \sum_{r=0}^{\infty} b_{m,\bar{m},n,r} v_r.$$

The functions are approximated by using only part of the complete basis set $\{v_r\}_{r=0,\ldots,L-1}$. The

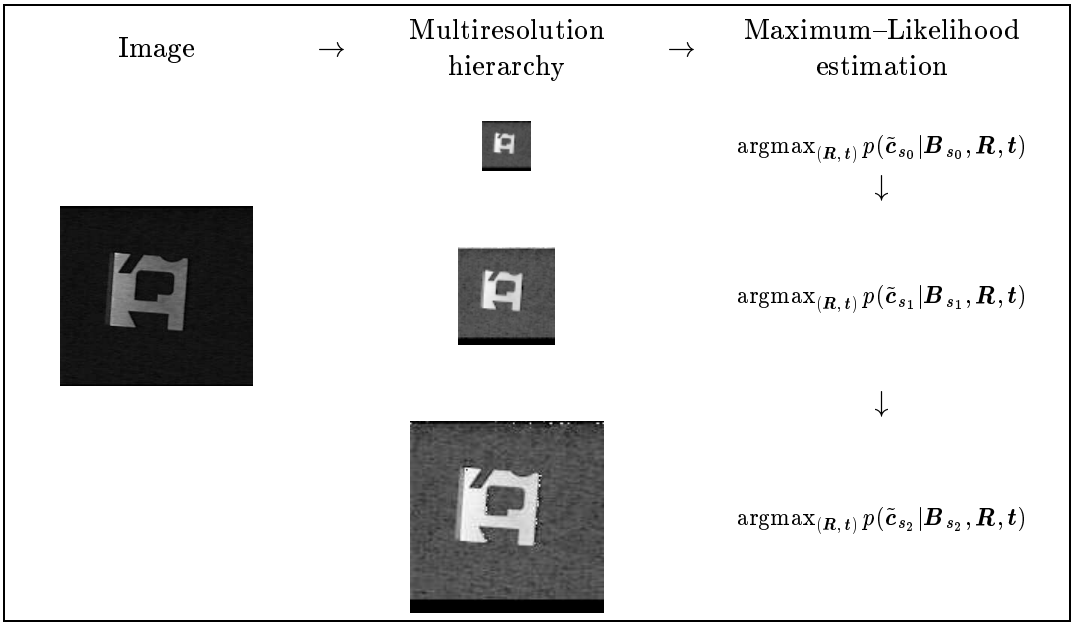

Figure 1: System overview.

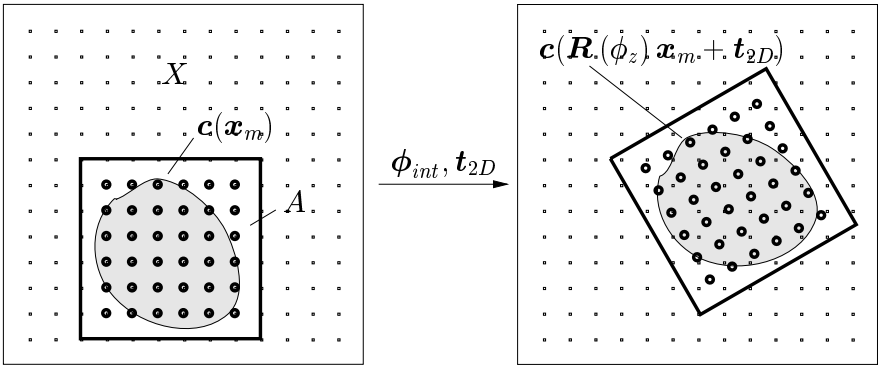

Figure 2: Object covered with grid for feature extraction.

Taylor decomposition shows, that the approximation error can be made as small as possible by choosing L large enough. With this approximation a fast computation of the density function and a maximum likelihood estimation of the basis coefficients is possible. The estimation results in closed estimation terms if $\sigma_{m,\bar{m}}$ is assumed as constant ($\sigma_{m,\bar{m},n} = b_{m,\bar{m},n,0}$, see also Sect. 3.2) as in the rest of this paper. The value of L is limited mainly by the computation time for the density and the size of the training set for estimation.

3.2 Parameter estimation

The model parameters are estimated by a maximum likelihood estimation. Under the assumption of N_ρ independent observations ${}^{\rho}\boldsymbol{c}_A$ this leads to the estimation

$$\left\{(\widehat{\boldsymbol{a}}_{m,n}, \widehat{\boldsymbol{b}}_{m,\bar{m},n})\right\} = \underset{\{(\boldsymbol{a}_{m,n},\boldsymbol{b}_{m,\bar{m},n})\}}{\text{argmax}} \prod_{\rho} p({}^{\rho}\boldsymbol{c}_A|\boldsymbol{x}_m, \{(\boldsymbol{a}_{m,n}, \boldsymbol{b}_{m,\bar{m},n})\}, {}^{\rho}\boldsymbol{R}, {}^{\rho}\boldsymbol{t}),$$

with the assumption of known transformation parameters ${}^{\rho}\boldsymbol{R}, {}^{\rho}\boldsymbol{t}$ and a predefined number L of basis functions.

The optimization of this function is rather complex. In order to reduce the complexity by providing an analytical solution, $\sigma_{m,n}$ is assumed to be constant.

Solving the equations for the parameters to be estimated results in:

$$\widehat{\boldsymbol{a}}_{m,n} = \boldsymbol{Q}^{-1}\left(\sum_{\rho} {}^{\rho}c_{m,n}\boldsymbol{v}({}^{\rho}\boldsymbol{\phi}_{ext})\right),$$

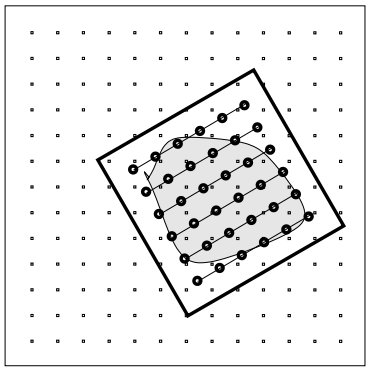

Figure 3: Location row dependencies.

$$\widehat{\sigma}_{m,\bar{m},n} = \frac{1}{N_\rho}\sum_\rho \left({}^\rho c_{m,n} - \widehat{\mu}_{m,n}({}^\rho\boldsymbol{\phi}_{ext})\right) \left({}^\rho c_{\bar{m},n} - \widehat{\mu}_{\bar{m},n}({}^\rho\boldsymbol{\phi}_{ext})\right),$$

with

$$\boldsymbol{Q} = \sum_\rho \boldsymbol{v}({}^\rho\boldsymbol{\phi}_{ext})\boldsymbol{v}^{\mathrm{T}}({}^\rho\boldsymbol{\phi}_{ext}),$$

${}^\rho c_{m,n} = {}^\rho c_n(\boldsymbol{R}({}^\rho\phi_z)\,\boldsymbol{x}_m + {}^\rho\boldsymbol{t}_{int})$ and
$\boldsymbol{v} = (v_0, \ldots, v_{L-1})^{\mathrm{T}}$.

3.3 Partial covariances

For each of the N feature vector components there are $M_A = |A|$ feature vector locations which have to be considered when calculating the probability of an observation. If the complete covariance matrix is used for this calculation the time complexity of one probability calculation is of order $\mathrm{O}(M_A^2 LN)$. Compared with a complete independence assumption the time complexity is M_A times higher. In the experiments, M_A is greater than 50 already on the first scale level and greater than 200 on the second. Pose estimation with the complete covariance therefore would be too time consuming.

The complexity of the normal density computation is mainly determined by the argument of the exponential term: $(\boldsymbol{c}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{c}-\boldsymbol{\mu})$. If $\boldsymbol{\Sigma}^{-1}$ is chosen as sparse matrix the complexity can be reduced significantly. One possibility are three–band matrices, which, for example, only have nonzero elements $\tilde{\sigma}_{m,\bar{m},n}$ for neighbour locations $\boldsymbol{x}_m = \boldsymbol{x}_{k,l}$ and $\boldsymbol{x}_{\bar{m}} = \boldsymbol{x}_{k,l+1}$ of one row (see Figure 3).

The parameter estimation is performed by a maximum likelihood estimation with the additional restriction, that some of the matrix elements have to be zero. Let X_{00} be the set of location pairs $(\boldsymbol{x}_m, \boldsymbol{x}_{\bar{m}})$ with zero matrix entries in $\boldsymbol{\Sigma}^{-1}$: $\tilde{\sigma}_{m,\bar{m},n} = 0$.
Then we get the term

$$\sum_\rho log(p({}^\rho\boldsymbol{c}_A|\boldsymbol{x}_m, \{(\boldsymbol{a}_{m,n}, \sigma_{m,\bar{m},n})\}, {}^\rho\boldsymbol{R}, {}^\rho\boldsymbol{t}) + \sum_{(\boldsymbol{x}_m,\boldsymbol{x}_{\bar{m}})\in X_{00}} \sum_n \lambda_{m,\bar{m},n}\tilde{\sigma}_{m,\bar{m},n}$$

to be maximized with respect to $\mu_{m,n}$ and $\tilde{\sigma}_{m,\bar{m},n}$ under the additional restriction $\tilde{\sigma}_{m,\bar{m},n} = 0$ for $(\boldsymbol{x}_m, \boldsymbol{x}_{\bar{m}}) \in X_{00}$. $\lambda_{m,\bar{m},n}$ are Lagrange Multipliers. It is obvious, that the solution to this optimization is the same as for the general case described in Sect.3.2 except for the covariance matrix elements $\{\sigma_{m,\bar{m},n}|(\boldsymbol{x}_m, \boldsymbol{x}_{\bar{m}}) \in X_{00}\}$ which are determined solely by the additional restriction of zero matrix elements in $\boldsymbol{\Sigma}^{-1}$.

As a consequence, in the parameter estimation the mean feature vectors and the covariances $\sigma_{m,\bar{m},n}$ $((\boldsymbol{x}_m, \boldsymbol{x}_{\bar{m}}) \in X_{00})$ have to be computed out of the training data first. Then the remaining covariance elements have to be chosen in a way, that the inverse covariance matrix becomes the sparse matrix as desired.

For three–band matrices there exists a recurrence relation to calculate the remaining elements beginning with a known three–band of the covariance:

$$\sigma_{m,\bar{m},n} = \sigma_{\bar{m},m,n} = \frac{\sigma_{m+1,\bar{m},n}\sigma_{m,\bar{m}-1,n}}{\sigma_{m+1,\bar{m}-1,n}} \quad (\bar{m} > m+1)$$

This allows the parameter estimation for row neighbourhood dependencies if the locations $\boldsymbol{x}_m$ are ordered by m in each row: $\boldsymbol{x}_{m+1} = \boldsymbol{x}_{k,l+1}$.

We will not prove this relation as we will show a more general estimation method in the next section. Furthermore the inverse covariance has to be calculated here. This is time consuming and prone to numerical errors for large matrices.

3.4 Bayes nets

The formalism of Bayes nets (see [4]) provides a way to represent the dependencies of random variables in graph structures and convert those graphs to a formula for the overall density composed of the conditional densities of the variables. Without loosing generality we consider only one-dimensional feature vectors $\boldsymbol{c}_m = c_{m,0}$ and omit the component index $n = 0$ in this section. We furthermore omit rotation and translation parameters.

In the context of Bayes nets the dependencies of row neighbours, column neighbours or both are depicted as shown in Figure 4. Let $\mathcal{P}(\boldsymbol{x}_m)$ denote the ordered set of predecessors of $\boldsymbol{x}_m$ in this dependency graph. Then the overall density is defined by the following formula:

$$\begin{aligned} p(\boldsymbol{c}_A) &= \prod_{\boldsymbol{x}_m \in A} p\left(c_m | (c_{\tilde{m}})_{\boldsymbol{x}_{\tilde{m}} \in \mathcal{P}(\boldsymbol{x}_m)}\right) \\ &= \prod_{\boldsymbol{x}_m \in A} \frac{p\left(c_m, (c_{\tilde{m}})_{\boldsymbol{x}_{\tilde{m}} \in \mathcal{P}(\boldsymbol{x}_m)}\right)}{p\left((c_{\tilde{m}})_{\boldsymbol{x}_{\tilde{m}} \in \mathcal{P}(\boldsymbol{x}_m)}\right)} \end{aligned}$$

Because the overall density is assumed as normal, the feature vector parts are also normally distributed. Substituting the definition of the densities yields the following relation of the normal density parameters:

$$\begin{aligned} \tilde{\sigma}_{m,\bar{m}} = & \sum_{\{\tilde{m} | \{m,\bar{m}\} \subset \mathcal{P}(\boldsymbol{x}_{\tilde{m}}) \cup \{\boldsymbol{x}_{\tilde{m}}\}\}} \tilde{\sigma}_{\mathcal{P}(\boldsymbol{x}_{\tilde{m}}) \cup \{\boldsymbol{x}_{\tilde{m}}\}, m, \bar{m}} \\ & - \sum_{\{\tilde{m} | \{m,\bar{m}\} \subset \mathcal{P}(\boldsymbol{x}_{\tilde{m}})\}} \tilde{\sigma}_{\mathcal{P}(\boldsymbol{x}_{\tilde{m}}), m, \bar{m}}, \end{aligned}$$

where $\tilde{\sigma}_{\mathcal{M},m,\bar{m}}$ are the elements of the inverse covariance matrix of $p(\mathcal{M})$.

In the case of row dependencies (see Figure 4) only 2–D matrices have to be inverted. The calculation is, on the other hand, not restricted to such simple dependencies. Structures, which have dependency edges only between strongly correlated variables are also possible.

3.5 Efficient pose estimation

The pose estimation consists — at least on the roughest resolution level — of a global pose search and succeeding local search. The global pose search evaluates the function $p(\boldsymbol{c}_A | \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{t})$ for grid positions covering the possible parameter range. Let $D_{\boldsymbol{R},\boldsymbol{t}}$ denote the number of evaluated transformation parameters. The time complexity of the grid search is then of order $\mathrm{O}(|A|\, LND_{\boldsymbol{R},\boldsymbol{t}})$ if only neighbour dependencies are considered.

For pose estimation the function

$$\begin{aligned} p(\boldsymbol{c}_A | \boldsymbol{B}, \boldsymbol{R}, \boldsymbol{t}) = & \frac{1}{\sqrt{\det(2\pi \boldsymbol{\Sigma})}} \\ & \exp\left(\frac{-1}{2} \sum_{m,\bar{m},n} \tilde{\sigma}_{m,\bar{m},n} (c_{m,n}(\phi_z, \boldsymbol{t}_{2D}) - \mu_{m,n}(\boldsymbol{\phi}_{ext})) \right. \\ & \left. (c_{\bar{m},n}(\phi_z, \boldsymbol{t}_{2D}) - \mu_{\bar{m},n}(\boldsymbol{\phi}_{ext}))\right), \end{aligned}$$

with $\mu_{m,n}(\boldsymbol{\phi}_{ext}) = \boldsymbol{a}_{m,n}^{\mathrm{T}} \boldsymbol{v}(\boldsymbol{\phi}_{ext})$ and $c_{m,n}(\phi_z, \boldsymbol{t}_{2D})$ $= c_n(\boldsymbol{R}(\phi_z)\, \boldsymbol{x}_m + \boldsymbol{t}_{2D})$ has to be maximized with respect to $\boldsymbol{R}, \boldsymbol{t}$.
Applying the logarithm yields the following function to be minimized:

$$\begin{aligned} h(\boldsymbol{\phi}, \boldsymbol{t}) = & \sum_{m,\bar{m},n} \tilde{\sigma}_{m,\bar{m},n} \left(c_{m,n}(\phi_z, \boldsymbol{t}_{2D}) - \boldsymbol{a}_{m,n}^{\mathrm{T}} \boldsymbol{v}(\boldsymbol{\phi}_{ext})\right) \\ & \left(c_{\bar{m},n}(\phi_z, \boldsymbol{t}_{2D}) - \boldsymbol{a}_{\bar{m},n}^{\mathrm{T}} \boldsymbol{v}(\boldsymbol{\phi}_{ext})\right). \end{aligned}$$

With $\boldsymbol{\phi} = (\phi_x, \phi_y, \phi_z)$ and the functions

$$\begin{aligned} h_1(\boldsymbol{\phi}, \boldsymbol{t}) = & \sum_{m,\bar{m},n} c_{m,n}(\phi_z, \boldsymbol{t}_{2D}) c_{\bar{m},n}(\phi_z, \boldsymbol{t}_{2D}) \tilde{\sigma}_{m,\bar{m},n} \\ h_{2,r}(\boldsymbol{\phi}, \boldsymbol{t}) = & \sum_{m,\bar{m},n} c_{m,n}(\phi_z, \boldsymbol{t}_{2D}) a_{\bar{m},n,r} \tilde{\sigma}_{m,\bar{m},n} \\ h_3(\boldsymbol{\phi}, \boldsymbol{t}) = & \sum_{m,\bar{m},n} \left(\boldsymbol{a}_{m,n}^{\mathrm{T}} \boldsymbol{v}(\boldsymbol{\phi}_{ext})\right) \left(\boldsymbol{a}_{\bar{m},n}^{\mathrm{T}} \boldsymbol{v}(\boldsymbol{\phi}_{ext})\right) \tilde{\sigma}_{m,\bar{m},n}, \end{aligned}$$

the sum $(h_1 - \boldsymbol{v}(\boldsymbol{\phi}_{ext})^{\mathrm{T}} \boldsymbol{h}_2 + h_3)(\boldsymbol{\phi}, \boldsymbol{t})$ has to be minimized. The global search has to calculate the function values for all grid positions which results in the mentioned complexity, depending on the number of object feature positions.

h_1 and $h_{2,r}$ are of the form

$$\tilde{h}(\boldsymbol{t}_{2D}) = \sum_n \sum_{m,\bar{m}} f_n(\boldsymbol{x}'_m + \boldsymbol{t}_{2D}, \boldsymbol{x}'_{\bar{m}} + \boldsymbol{t}_{2D}) w_{m,\bar{m},n},$$

with $\boldsymbol{x}'_m = \boldsymbol{R}(\phi_z)\, \boldsymbol{x}_m$ and for fixed $\boldsymbol{\phi}$. Let the successors of the grid locations be uniformly defined by the set S of offset vectors, so that $\mathcal{P}(\boldsymbol{x}_m) = \left\{\boldsymbol{x}_{(k,l)-s} | s \in S\right\}$ with $\boldsymbol{x}'_m = \boldsymbol{x}'_{k,l}$. For single row dependencies S is: $S = \{(1,0)\}$ (see Figure 4).

The second sum in the above equation therefore can be written as:

$$\bar{h}(\boldsymbol{t}_{2D}) = \sum_{s \in S_0} \sum_m \bar{f}_{s,n}(\boldsymbol{x}'_m + \boldsymbol{t}_{2D}) w_{m,s(m),n},$$

for $s(\boldsymbol{x}_m) = \boldsymbol{x}_{(k,l)+s}$. Of course the summation has to be performed only for the valid ranges of neigbour locations. If the evaluation grid $(\boldsymbol{\phi}, \boldsymbol{t}_{2D}) \in \left\{(\boldsymbol{\phi}_{i,j,q}, \boldsymbol{t}_{2D,q,k,l})\right\}$ for the transformation parameters is chosen as extension of $X' = -\boldsymbol{R}(\phi_{z,q})\, X$ to the possible parameter range, so that

$$\begin{aligned} \boldsymbol{\phi}_{i,j,q} &= (\phi_{x,0} + i\Delta\phi_x, \phi_{y,0} + j\Delta\phi_y, \phi_{z,0} + q\Delta\phi_z) \\ &= (\phi_{x,i}, \phi_{y,j}, \phi_{z,q}) \\ \boldsymbol{t}_{2D,q,k,l} &= -\boldsymbol{R}(\phi_{z,q}) \left(t_{x,0} + k\Delta t_x, t_{y,0} + l\Delta t_y\right)^{\mathrm{T}} \\ &= -\boldsymbol{R}(\phi_{z,q}) \left(t'_{x,k}, t'_{y,l}\right)^{\mathrm{T}} \\ X' &= \{-\boldsymbol{R}(\phi_{z,q})\, \boldsymbol{x}_m\} \subset \{\boldsymbol{t}_{2D,q,k,l}\}, \end{aligned}$$

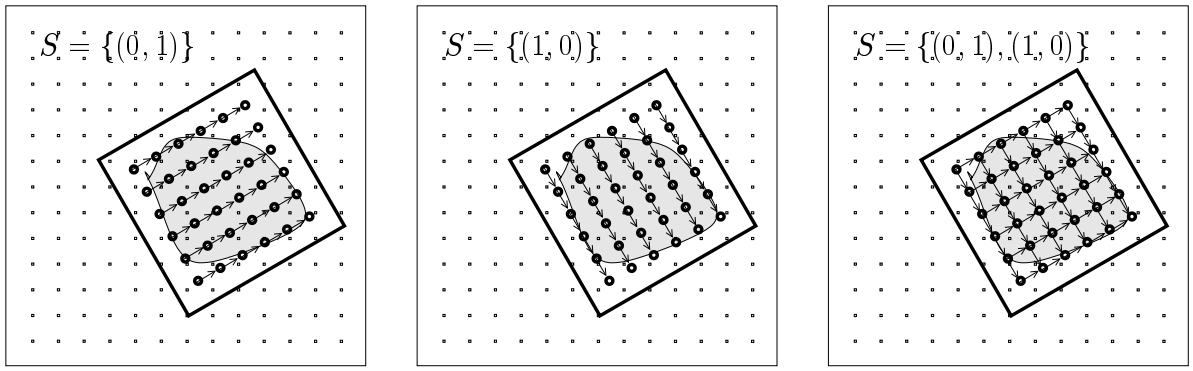

Figure 4: Bayes net with row dependencies, column dependencies and both of local feature vectors.

the evaluation of the second sum on X' can be interpreted as convolution. This is, because

$$\begin{aligned}
&\bar{h}\,(\boldsymbol{t}_{2D,q,k,l}) \\
&= \sum_{s\in S_0} \sum_{m} f_{s,n}(\boldsymbol{R}\,(\phi_{z,q})\,\boldsymbol{x}_m + \boldsymbol{t}_{2D,q,k,l}) w_{m,s(m),n} \\
&= \sum_{s\in S_0} \sum_{m} f_{s,n}(\boldsymbol{R}\,(\phi_{z,q})\,(\boldsymbol{t}'_{2D,k,l} - \boldsymbol{x}_m)) w_{m,s(m),n} \\
&= \sum_{s\in S_0} \sum_{m} \tilde{f}_{q,s,n}(\boldsymbol{t}'_{2D,k,l} - \boldsymbol{x}_m) w_{m,s(m),n}
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{m} \tilde{f}_{q,n}\,(\boldsymbol{t}'_{2D,k,l} - \boldsymbol{x}_m) w_{m,n} \\
&\qquad = \sum_{\bar{k},\bar{l}} \tilde{f}_{q,n}(\boldsymbol{x}_{k,l} - \boldsymbol{x}_{\bar{k},\bar{l}}) w_{\bar{k},\bar{l},n} \\
&\qquad = \sum_{\bar{k},\bar{l}} \tilde{f}_{q,k-\bar{k},l-\bar{l},n} w_{\bar{k},\bar{l},n}
\end{aligned}$$

for $\boldsymbol{x}_m = \boldsymbol{x}_{\bar{k},\bar{l}} = \boldsymbol{t}'_{2D,\bar{k},\bar{l}}$.
Using FFT allows the computation of this convolution in $\mathrm{O}(D_{\boldsymbol{t}} \log(D_{t_x}) \log(D_{t_y}))$ time. The calculation of each of the $L+1$ functions h_1 and $h_{2,r}$ is therefore of complexity $\mathrm{O}(ND_{\boldsymbol{t}} \log(D_{t_x}) \log(D_{t_y}))$. h_3 has complexity $\mathrm{O}(D_{\phi_{ext}})$. This results in a complexity of order $\mathrm{O}(LND_{\boldsymbol{R},\boldsymbol{t}} \log(D_{t_x}) \log(D_{t_y}))$ for the complete search, where the computation of h out of the simpler functions can be performed very fast. With respect to the type of neighbourhood considered, the complexity for the global search based on the calculation of each indidual density is $\mathrm{O}(|S_0|\,|A|\,LND_{\boldsymbol{R},\boldsymbol{t}})$ and $\mathrm{O}(|S_0|\,LND_{\boldsymbol{R},\boldsymbol{t}} \log(D_{t_x}) \log(D_{t_y}))$ for the optimized search based on filter banks.

4 Results

Figure 5 and 6 show the objects used in this work. The images are 256 pixels in square. The localisation was performed on two scale levels s_0, s_1 with resolution $r_{s_0} = 8$ and $r_{s_1} = 4$ pixels for the 2–D objects. 3–D experiments were carried out only on the first scale level and with constant $\boldsymbol{\Sigma}$. Only the best localization result of level s_0 was used for further refinement at s_1. The Downhill Simplex algorithm was used for the local parameter search following the global grid search. The computation time on a SGI O2 (R10000) was about 10 seconds for feature extraction and localization of 2–D object *box* with complete feature indepencence and 11 seconds with row dependencies. The time for 3–D object *garfield* was 12 seconds (13 with dependencies) on one scale level with restricted search space for the parameters $\boldsymbol{\phi}_{ext}$ according to their availability for training as described below.

For the 2–D objects *box* and *car* 20 training images were available. 10 of them contain the object at the same position but different lighting conditions. In the other 10 the object position is arbitrary and the background is homogeneous but different. The density was first trained with the first ten images with known position. Then this density was used to iteratively locate the object in the remaining images and train together with the additional data. 10 other images with heterogeneous background (see Figure 5) were then used for experiments under bad conditions.

In the 2–D, experiments 5 images (of 10) of object *box* were located incorrectly with independence assumption and none with row dependencies. The pose of object *match* was correct for all images with independence and wrong for 2

images with dependencies. Figure 7 and Figure 8 show the logarithmic density values for the object images of the training set. It can be seen that the variance of the density is less, if dependencies are considered. Obviously the density better captures the object appearance in this case. The experiments for *box* confirm this observation. In the 2 images of *match*, which were located incorrectely when using depencencies, a small part of the object was occluded. This shows that locality of the density measurement is lost partially.

The 3–D object *garfield* was available in two image sequences with 256 images each. The transformation parameters were known. The external rotation parameters were restricted to $-\frac{1}{3}\pi < \phi_x < \frac{1}{3}\pi$ and $-\frac{1}{4}\pi < \phi_y < \frac{1}{4}\pi$. One sequence was used for training, the other for testing. The images in the training sequence were disturbed artificially by gaussian noise and the object was shifted, resulting in two addtional training sequences.

The position of 28 test images (out of 256) was determined incorrectely with independence assumption. Together with the training set, 29 images were handled wrong. With row dependencies the number of incorrect images is 20 and 24 respectively. Figure 9 shows the logarithmic density values for the undisturbed images of the training set. Though the experiments also show an advantage of using dependencies, it is not so obvious as for the 2–D objects. This may be a consequence of the restriction to a constant $\boldsymbol{\Sigma}$ with respect to the external rotation. The consequence of this constancy assumption is, that the estimated covariance parameters are only mean values of the real covariances over the complete parameter range. Future work will investigate the influence of the type of covariance estimation on the pose estimation results.

Figure 5: 2–D objects *box* and *match* for training (top) and test (bottom).

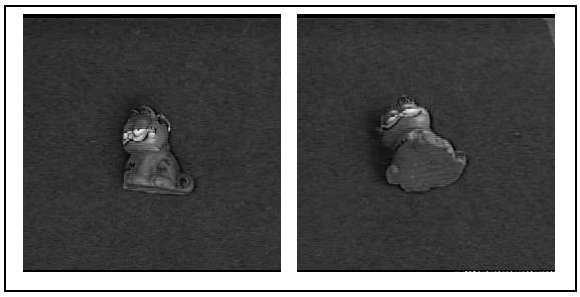

Figure 6: 3–D object *garfield*.

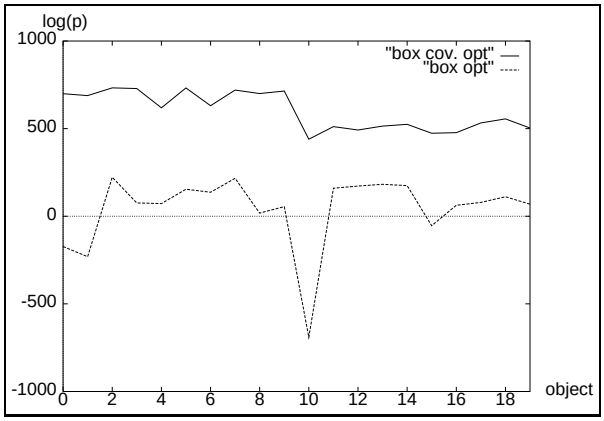

Figure 7: Optimum of logarithmic density for training set of *box*.

References

[1] A. Rieder A. K. Louis, P. Maass. *Wavelets*. Teubner, Stuttgart, 1994.

[2] J. Hornegger and H. Niemann. Statistical learning, localization, and identification of objects. In ICCV 95 [3], pages 914–919.

[3] *Fifth International Conference on Computer Vision (ICCV)*, Cambridge, MA, June 1995. IEEE Computer Society Press.

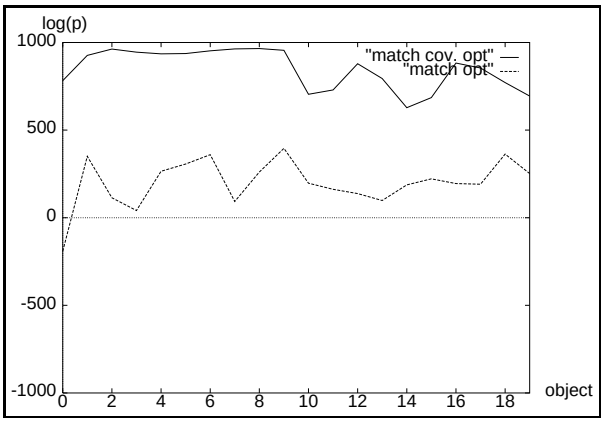

Figure 8: Optimum of logarithmic density for training set of *match*.

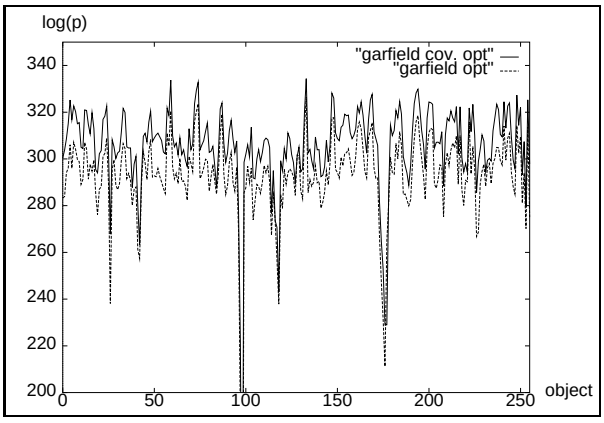

Figure 9: Optimum of logarithmic density for undistrubed part of training set of *garfield*.

[4] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, London, 1996.

[5] H. Kollnig and H.-H. Nagel. 3D pose estimation by fitting gradients directly to polyhedral models. In ICCV 95 [3], pages 569–574.

[6] N. Krüger, M. Pötzsch, and C. v.d. Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. Technical report, Ruhr-Universität Bochum, January 1996.

[7] V. Kumar and E. S. Manolakos. Unsupervised model–based object recognition by parameter estimation of hierarchical mixtures. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 967–970, Lausanne, Switzerland, September 1996. IEEE Computer Society Press.

[8] H. Murase and S. K. Nayar. Visual learning and recognition of 3–D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[9] J. Pösl and H. Niemann. Statistical 3–D object localization without segmentation using wavelet analysis. In *Proceedings of the* 7^{th} *International Conference on Computer Analysis of Images and Patterns (CAIP)*, Kiel, Germany, September 1997, to appear.

[10] J. Pösl and H. Niemann. Wavelet features for statistical object localization without segmentation. In *Proceedings of the International Conference on Image Processing (ICIP)*, Santa Barbara, California, USA, October 1997, to appear.

[11] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[12] P. Viola and W. Wells III. Alignment by maximization of mutual information. In ICCV 95 [3], pages 16–23.

[13] X. Wu and B. Bhanu. Gabor wavelets for 3D object recognition. In ICCV 95 [3], pages 537–542.