

MDL Patch Correspondences on Unlabeled Images with Occlusions

Johan Karlsson, Kalle Åström

Centre for Mathematical Sciences, Lund University, Sweden

johank@maths.lth.se, kalle@maths.lth.se

Abstract

Automatic construction of Shape and Appearance Models from examples via establishing correspondences across the training set has been successful in the last decades. One successful measure for establishing correspondences of high quality is minimum description length (MDL). In other approaches it has been shown that parts+geometry models which model the appearance of parts of the object and the geometric relation between the parts have been successful for automatic model building. In this paper it is shown how to fuse the above approaches and use MDL to fully automatically build optimal parts+geometry models from unlabeled images.

1. Introduction

MDL [1, 3, 16] has been applied both to establish corresponding parameterizations of manually pre-segmented object outlines and in order to group-wise register the entire training images [18]. In other approaches parts+geometry models have been successful for fully automatic model building ([11, 10, 2, 5]). In this paper it is shown how to fuse the above approaches and use MDL to fully automatically build optimal parts+geometry models from unlabeled (i.e. different images in the training set can be of objects from different object classes) images.

MDL is here used as a framework for a number of decisions, not only to determine correspondences. Decisions about classification (what object class the object in each image belongs to), model complexity selection (e.g. number of parts), what to include in the modeling and what to consider as outliers and missing data are done using MDL.

The work most related to ours is [14]. In that work correspondences and decisions about missing data are handled in a similar way as in this paper. However, we make further important decisions in the same MDL framework. Most importantly this includes handling unlabeled images (i.e. mixed object classes), deciding what in the images to model and what to consider as irrelevant background, and deciding on model complexity (e.g. number of parts).

2. MDL Framework

From a set of images a number of parts are extracted by calculating corners/edge-points and sampling patches around these points on a number of scales.

The main problem that we formulate and solve in this paper is the following. Assume that a number of examples $S = \{S_1, \dots, S_n\}$ are given, where each example S_i is a set of unordered feature points $z_{i,j}$ (with surrounding patches) $S_i = \{z_{i,1}, \dots, z_{i,k_i}\}$, and each point $z_{i,j}$ lies in \mathbf{R}^p for some dimension p , typically $p = 2$. An example of such sets is the output of a feature point detector, e.g. [12], with patches extracted around the points.

We pose the problem as that of selecting: (i) image classification (ii) inliers, (iii) correspondences and (iv) model complexity, so that the description length is minimized. As we shall see this becomes a mixed combinatorial and continuous optimization problem, where for each of a discrete set of possible inliers and correspondences, there is a continuous optimization problem which has to be solved. These continuous optimization problems involve both the problem of missing data Procrustes and missing data principal component analysis.

To begin with, assume that all the images are of objects from the same object class. We have a set S of n unordered point sets, $S = \{S_1, \dots, S_n\}$ with surrounding patches is given, with the k_i points in each set S_i ordered arbitrarily. The object is now to find a reordering of such points. Assuming for the moment that the model contains N points with patches, such a reordering can be represented as a matrix O (for order) of size $n \times N$. The entries in O are either 0 - representing that a model point is not visible (missing) in an image or an identity number between 1 and k_i signifying which image point is to be considered as a representative of the model point, i.e. $O_{i,j} = 0$ if model point j is not visible (missing) in image i or $O_{i,j} = k$ if model point j in image i is represented by $z_{i,k}$. Also introduce the set I consisting of pairs of indices (i, j) such that model point j is visible in image i , i.e.

$$I = \{(i, j) \mid O_{i,j} \neq 0\} . \quad (1)$$

The matrix O then settles both questions about out-

liers/inliers, missing data and correspondences. Outliers/clutter are all image points not represented in O . Missing data are signified by zeros. Image points whose indices are in the same column of O are corresponding.

Given an ordering matrix O the data can be reordered, possibly with missing data into a structure T of N points in n images, i.e.

$$T_{i,j} = \begin{cases} z_{i,O_{i,j}} & \text{if } (i,j) \in I \\ \text{undefined} & \text{if } (i,j) \notin I. \end{cases} \quad (2)$$

For such a ordered point set T with missing data one can align the shapes from the different images with a Procrustes analysis with respect to a transformation group G . The aim is to find a mean shape m and a number of transformations $\{g_1, \dots, g_n\}$ with $g_i \in G$ such that $g_i(m) \approx T_i$ or rather $g_i^{-1}(T_i) \approx m$.

After this a Principal Component Analysis handling missing data can be performed on the residuals between $g_i^{-1}(T_i)$ and m . From this a number of shape variational modes, denoted v_l , can be determined. New shapes can then be synthesized as $g(m + \sum_{l=1}^d \lambda_l v_l)$, where λ_l are scalar coordinates and $g \in G$.

This takes care of the geometry variation, i.e. the shape variation as defined by the corresponding points. Next, the appearance of the parts as defined by the corresponding patches is dealt with in a similar fashion. For each N model points all the patches belonging to that model point are analyzed with a PCA. Note that for these PCA's there is no missing data but the number of patches differ if some points are missing. These PCA's then determine mean patches $\{m_i^p\}_{i=1}^N$, numbers of modes used $\{d_i^p\}_{i=1}^N$, patch-modes $\{v_{ij}^p\}_{i=1\dots N, j=1\dots d_i^p}$ and patch-mode-parameters $\{\lambda_{ijk}^p\}_{i=1\dots N, j=1\dots d_i^p, k=1\dots n}$.

So we need to assess a number of different choices: the number of model points N , the ordering O , the mean shape m , the transformation group G , the transformations g_i , the number of shape variation modes d , the shape variation modes v_l , the shape coordinates λ_{li} , mean patches $\{m_i^p\}_{i=1}^N$, numbers of patch-modes $\{d_i^p\}_{i=1}^N$, patch-modes $\{v_{ij}^p\}_{i=1\dots N, j=1\dots d_i^p}$ and the patch-mode-parameters $\{\lambda_{ijk}^p\}_{i=1\dots N, j=1\dots d_i^p, k=1\dots n}$. The idea here is that a common framework such as the minimum description length framework could be used to determine all of these choices. This would put the whole chain of difficult modeling choices on an equal footing and would make it possible to use simple heuristics for making reasonable choices, while at the same time have a common criterion for evaluating different alternatives.

The whole process can thus be seen as an optimization problem

$$\min_{\mathcal{M}} \text{dl}(S, \mathcal{M}), \quad (3)$$

over the unknowns

$$\mathcal{M} = (O, m, \{g_i\}, d, \{v_l\}, \{\lambda_{li}\}, \{m_i^p\}, \{d_i^p\}, \{v_{ij}^p\}, \{\lambda_{ijk}^p\}) \quad (4)$$

given data S .

In principle these could now all be optimized simultaneously using MDL. In practice however this becomes impossibly cumbersome. Also for some of the parameters to be determined, more easy methods not using MDL will give very close to the same result. This includes for example the Procrustes analysis and the PCA's. Therefore these parameters are determined without MDL and we focus the use of MDL where it really makes a difference, namely in determining image classifications, number of model points, inliers/outliers/missing data and for establishing correspondences.

3. Calculating the Description Length

Introduction A number of sets of points with surrounding patches are given. In order to determine a model that explains points that can be seen in many of the images, the goal is to minimize the description length that is needed to transmit all the interesting points of all views, in hope that a model will be able to make a cheaper description than simply sending the data bit by bit. Here we will derive the description length for the data and the model. For the outliers one must simply send the information bit by bit. For the points and patches that are included as inliers in the model, the idea is that it is cheaper to send the model with parameters and residuals etc. to explain the data. For the modeled points and patches one must send: the model, the model parameters, information if a certain point is missing, the transformation and the residuals.

Preliminaries on Information Theory To transmit a continuum value α it is necessary to quantify the value. The continuum value α quantified to a resolution of Δ is here denoted $\hat{\alpha}$, $\alpha_{min} \leq \hat{\alpha} \leq \alpha_{max}$, $\hat{\alpha} = m\Delta$, $m \in \mathbf{Z}$.

The ideal coding codeword length for a value $\hat{\alpha}$, encoded using a statistical model $\mathcal{P}(\hat{\alpha})$ is given by the Shannon codeword length [17]. Using Shannon's codeword length the description length of a given value, $\hat{\alpha}$, encoded using a probabilistic model, is $-\log(\mathcal{P}(\hat{\alpha}))$, where \mathcal{P} is the probability-density function.

Coding Data with Uniform Distribution Assume α is uniformly distributed and quantified to $\alpha_{min} \leq \hat{\alpha} \leq \alpha_{max}$, $\hat{\alpha} = m\Delta$, $m \in \mathbf{Z}$. Then α can take $\frac{\alpha_{max}-\alpha_{min}}{\Delta}$ different values. Since uniform distribution is assumed, the probability for a certain value of $\hat{\alpha}$ is $\mathcal{P}(\hat{\alpha}) = \frac{\Delta}{\alpha_{max}-\alpha_{min}}$. This gives Shannon's codeword length for $\hat{\alpha}$, $-\log(\mathcal{P}(\hat{\alpha})) = -\log(\frac{\Delta}{\alpha_{max}-\alpha_{min}})$. If the parameters

α_{min} , α_{max} and Δ are unknown to the receiver, these need to be coded as well.

Coding Data with Assumed Gaussian Distribution

Since the mean μ of our data will be zero, the 1-parameter Gaussian function can be used. The frequency function of a 1-parameter Gaussian function is $f(x; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$. The derivation for sending a one-dimensional dataset from a Gaussian 1-parameter distribution was done in [4] and later modified by different authors. The version used here can be found for example in [7].

Using a two-part coding scheme the description length is decomposed to

$$\mathcal{L}_{gaussian} = \mathcal{L}_{param} + \mathcal{L}_{data} , \quad (5)$$

where \mathcal{L}_{param} is the description length of the parameters to be used for the Gaussian encoding and \mathcal{L}_{data} is the description length of the data using the Gaussians.

Coding the Parameters The standard deviation s of the Gaussian needs to be coded. It must be discretized to \hat{s} using the accuracy Δ . Without prior knowledge, the standard deviation \hat{s} , is assumed to have the same possible range R as the data. If \hat{s} is further assumed to be uniformly distributed in this range, the description length is

$$\mathcal{L}_{param} = \mathcal{L}_{\hat{s}} = -\log \mathcal{P}(\hat{s}) = \log\left(\frac{R}{\Delta}\right) . \quad (6)$$

Coding the Data Assume there are n_s data-points in the set. Depending on the actual range and standard deviation \hat{s} of the dataset of coordinates from the training set there are three different encoding strategies. If \hat{s} is not too small, a simple approximation of the Shannon description length can be used. If $\hat{s} \leq s_{cut} = 2\Delta$, the approximation formula is not valid and \hat{s} is replaced with s_{cut} . If the range of the dataset $\mathcal{Y} \leq \Delta$ the data can simply be coded as zeros and the description length is zero.

Case 1 ($\hat{s} > s_{cut}$) The description length of the data set consisting of the parameters for the whole training set in one of the shape or patch-appearance modes is

$$\begin{aligned} \mathcal{L}_{data} &= -\sum_{i=1}^{n_s} \log(\mathcal{P}(\hat{y}_i)) = \\ &= -\sum_{i=1}^{n_s} \log\left(\frac{\Delta}{\hat{s}\sqrt{2\pi}} \exp\left(-\frac{\hat{y}_i^2}{2\hat{s}^2}\right)\right) \\ &= -n_s \log \Delta + \frac{n_s}{2} \log(2\pi\hat{s}^2) + \frac{1}{2\hat{s}^2} \sum_{i=1}^{n_s} \hat{y}_i^2 = \\ &= -n_s \log \Delta + \frac{n_s}{2} \log(2\pi\hat{s}^2) + \frac{n_s \hat{s}^2}{2\hat{s}^2} . \end{aligned} \quad (7)$$

Case 2 ($\hat{s} \leq s_{cut}$) Replacing \hat{s} with s_{cut} gives

$$\mathcal{L}_{data} = -n_s \log \Delta + \frac{n_s}{2} \log(2\pi s_{cut}^2) + \frac{n_s s^2}{2s_{cut}^2} . \quad (8)$$

Case 3 When the range of $\mathcal{Y} \leq \Delta$ we get

$$\mathcal{L}_{data} = 0 . \quad (9)$$

Assuming Δ is small, we can replace \hat{s} with s , giving the following descriptions lengths (the indexes (1) and (2) referring to case 1 and case 2 respectively).

$$\begin{aligned} \mathcal{L}_{data(1)} &= -n_s \log(\Delta) + \frac{n_s}{2} \log(2\pi) + \\ &= n_s \log(s) + \frac{n_s}{2} , \\ \mathcal{L}_{data(2)} &= -n_s \log(\Delta) + \frac{n_s}{2} \log(2\pi) + \\ &= n_s \log(s_{cut}) + \frac{n_s}{2} \left(\frac{s}{s_{cut}}\right)^2 . \end{aligned} \quad (10)$$

The Total Description Length of the Interesting Points in All Images

For things that are unknown we assume even distribution. First, the description length for a point \hat{x} equally distributed over the image is

$$dl_{rect} = -\log(\mathcal{P}(\hat{x})) = -2\log\left(\frac{dx}{X}\right) . \quad (11)$$

Here X is the possible coordinate range, and dx is the resolution. The factor 2 comes from that an image point is two-dimensional. Similarly, the description length for a general patch \hat{p} is

$$dl_{rect}^p = -\log(\mathcal{P}(\hat{p})) = -n_{patch} \log\left(\frac{dy}{Y}\right) . \quad (12)$$

Here n_{patch} is the number of pixels in the patch, Y is the possible intensity range, and dy is the resolution.

The outliers are assumed to be uniformly distributed over all the image and their patches are similarly considered to be uniformly distributed, so with n_o number of outliers

$$dl_{outliers} = n_o(dl_{rect} + dl_{rect}^p) . \quad (13)$$

For each model point we need to know if the point is missing in an image or not. This means one bit for each n_p model points in all n_v images. Conversion to nats gives $dl_{index} = \log(2)n_v n_p$, where n_p is the number of landmarks in the model and n_v is the number of images.

For each image the transformation g of the model has to be encoded. Translations are assumed equally distributed within the size of the image and other transformation parameters are coded with equal accuracy. This gives the following expression $dl_{trans} = n_v n_{dof} dl_{rect}$, where n_v is

the number of images and n_{dof} is the degrees of freedom in the transformation group.

The coordinates of the mean shape and the coordinates of the shape modes are also assumed equally distributed within the size of the image, thus the cost is

$$dl_{meanshape} = n_p dl_{rect} \quad (14)$$

$$dl_{shapemodes} = n_p n_m dl_{rect} \quad , \quad (15)$$

where n_m is the number of shape modes used by the model. Similarly, for each model point i , the cost for the mean patch and the patch-modes is

$$dl_{meanpatch} = dl_{rect}^p \quad (16)$$

$$dl_{patchmodes}^i = n_m^i dl_{rect}^p \quad , \quad (17)$$

where n_m^i is the number of patch appearance modes used by the model for model-point i . We put these together as

$$dl_{model} = dl_{meanshape} + dl_{shapemodes} + n_p dl_{meanpatch} + \sum_{i=1}^{n_p} dl_{patchmodes}^i \quad . \quad (18)$$

Coordinates for shape modes and patch appearance modes as well as residuals for point coordinates and patch intensities are assumed Gaussian. Different possible range and different accuracy for point coordinates and for patch intensities gives two different costs for Gaussians, $\mathcal{L}_{gaussian}^{point}$ and $\mathcal{L}_{gaussian}^{patch}$. In both cases the description length is a function of the standard deviation s and the number of data-points n_s . The total cost for these becomes

$$dl_{coordinates} = \sum_{k=1}^{n_m} \mathcal{L}_{gaussian}^{point}(s_k, n_v) + \sum_{i=1}^{n_p} \sum_{k=1}^{n_m^i} \mathcal{L}_{gaussian}^{patch}(s_{ik}, n_v^i) \quad (19)$$

$$dl_{residuals} = \mathcal{L}_{gaussian}^{point}(s, 2 \sum_{i=1}^{n_p} n_v^i) + \sum_{i=1}^{n_p} \mathcal{L}_{gaussian}^{patch}(s_i, n_v^i n_{patch}) \quad (20)$$

$$dl_G = dl_{coordinates} + dl_{residuals} \quad , \quad (21)$$

where n_v^i is the number of images that model point i is not missing in.

So the full cost for sending the data is

$$DL_{tot} = dl_{trans} + dl_{index} + dl_G + dl_{model} + dl_{outliers} \quad , \quad (22)$$

where we note that dl_{trans} actually does not depend on any decisions once the transformation group is decided, so it does not need to be included.

Note on Total Description Length Often when using MDL to establish correspondence, the formula for the total description length is simplified further by always using all the available modes. Since we want to simultaneously make decisions about model complexity and about correspondence that is not done here. The gain in reduced residuals is weighed against the cost of more modes.

4. Optimizing DL

Given a current set of parameters \mathcal{M} as in Equation (4), a model of global shape and part appearance can be built and the description length for the model and for the data described using the model can be calculated. For each suggested model one needs to calculate the description length of sending all the outliers and all the data modeled with that particular model. The number of model modes can vary between zero to $n_s - 1$ and all these models must be evaluated. Note here that since the modes calculated when using missing data PCA depends on the number of modes used, the model needs to be calculated over and over as the different numbers of modes are tested. In the optimization procedure the tested model with least description length is then compared to previous best solutions.

The whole optimization process over all unknowns to find

$$\min_{\mathcal{M}} dl(S, \mathcal{M}) \quad , \quad (23)$$

where S is the data, is divided into three parts: (1.) Optimization over the discrete ordering matrix O , (2.) optimization over the transformations $\{g_i\}$ and (3.) optimization over the remaining parameters

$$\tilde{\mathcal{M}} = (m, d, \{v_l\}, \{\lambda_{li}\}, \{m_i^p\}, \{d_i^p\}, \{v_{ij}^p\}, \{\lambda_{ijk}^p\}) \quad . \quad (24)$$

First of all consider the choice of the transformation group G . It is possible to include the choice of G as an unknown and determine it using MDL. You may simply try some different groups and see which one gives the lowest description length. This would mean weighing the higher cost of coding transformations with more degrees of freedom against the gain of smaller residuals left to encode using the model. However it is just as likely that you want to choose G before starting the optimization. In the experiments in this paper G was chosen to be the group of similarity transformations.

Assume that a ordering O is given. Then it is straightforward to reorder the inlier points into the data structure T as described above. Each ordering determines number of model points, the n_{inlier} inliers, the $n_{outlier}$ outliers and the correspondences. The description length for the outliers is independent of $\{g_i\}$ and $\tilde{\mathcal{M}}$.

Given T , $\{g_i\}$ is to be determined using some Procrustes analysis handling missing data. Ideally $\{g_i\}$ should

be evaluated using MDL. However, this considerably increases time complexity and the results of alignments based on euclidean distance is normally very close to the same as alignments based on MDL (although sometimes the difference can be significant) [13]. Therefore MDL is not used to determine $\{g_i\}$ given T . Instead the alignments are based on minimizing euclidean distances. If there are no missing data this can be done optimally without iteration, but this requires a singular value decomposition which with missing data is itself an iterative procedure. So instead this is done in a number of steps where the process can be terminated without doing all the steps if precision is less important than speed. By a shape will be meant the set of points from one image corresponding to the points in the model. First all shapes are translated to the origin and scaled so that the mean distance of points in the shape from the origin is 1. Next all shapes are aligned to the first shape $shape_1$. If the shapes are represented as complex column vectors this alignment is achieved by multiplying with $shape_i^* shape_1 / (shape_i^* shape_i)$ [6]. Next follows a loop of estimating a mean-shape and aligning all shapes to it, estimating a new mean-shape etc. Finally the error remaining is minimized to find

$$\min_{m, \{g_i\}} \sum_{(i,j) \in I} |T_{i,j} - g_i(m_j)|^2 \quad (25)$$

using Gauss-Newton while keeping the size of the mean-shape fixed to avoid shrinking everything.

Assume next also that the number of shape modes d and the numbers of part appearance modes $\{d_i^p\}$ are given. What remains to determine are the modes and the parameters for the data in those modes. As for the alignments this should ideally be done using MDL but again more direct approaches will give very close to the same result at less cost. For $\{v_{ij}^p\}$ and $\{\lambda_{ijk}^p\}$ there are no problems with missing data since if a point is missing that only means that the number of examples of that part is smaller. So $\{v_{ij}^p\}$ and $\{\lambda_{ijk}^p\}$ are simply determined by the standard procedure of doing a singular value decomposition. For the shape modes the missing data is a problem. What is needed then is a principal component analysis (via a singular value decomposition $X = USV^T$) of the residuals X after subtracting the mean-shape from all the aligned shapes and this PCA must handle missing data. There are different approaches for this but since if we could afford very time consuming approaches we would use MDL anyway, here a relatively simple (but not too simple) approach is used. Since the typical value of the residuals is zero first all missing entries are set to zero. Then a standard PCA is performed via $X = USV^T$. Next the missing entries in X are updated with the corresponding values of the rank d approximation $U_d S_d V_d^T$ of X . Here U_d is the first d columns of U , S_d is top left $d \times d$ part of S and V_d is the first d columns of V . Then a new PCA is done and

the missing entries are updated etc. The number of loops are chosen as deemed fit.

To sum up, given O , $(m, \{m_i^p\}, \{g_i\})$ are determined as described above. Then each valid value of the number of nodes d and $\{d_i^p\}$ are examined. These are independent so all combinations need not be considered. First d can be determined and then $\{d_i^p\}$ one at a time. For each value of d , missing value PCA is performed and then the part of the total description length depending on d is calculated in order to pick the d that gives the lowest description length. For the patch appearance models only one PCA per part is needed, not one for every value of d_i^p . For each value, the part of the total description length depending on d_i^p is calculated in order to pick the d_i^p that gives the lowest description length. Thus, the minimal description length can be seen as a function of the ordering O .

4.1. Optimization

Optimizing description length with respect to O is a combinatorial optimization problem. We suggest the following algorithm that (1) finds a reasonable initial guess and (2) searches for a local minima in a combinatorial optimization sense by attempting the following six types of changes and evaluating using description length. The optimization terminates when none of the operations below give an improvement in description length.

1. Addition of a model point. In each image either the new point is set as missing or as corresponding to one of the points from the image.
2. Deletion of a model point. In each image the points corresponding to this model point are set as outliers.
3. Change of a point from outlier to inlier belonging to one of the model points.
4. Change of a point from inlier to outlier.
5. Changing correspondence by moving an inlier to a neighboring point, i.e. setting the old point as an outlier and changing the new one from being an outlier to representing the same model point that the old point represented.
6. Using the current model to restart the process for the points from one image.

The final algorithm for determining minimal description length is then

- I Make an initial guess on point ordering O .
- II Calculate optimal description length for that ordering.
- III See if any of the perturbations above lowers the description length.

- IV • If it does, make those changes and continue with step 3.
- If not, then we are at a local minima, stop.

4.2. Initial Guess

The initial guess of course can be important for the final result, but in this aspect the process is not very sensitive. Bad initial correspondences can be identified and removed since this can be a way to lower the description length. However the initial guess is very important for computation time since it takes a lot of time to identify and change or remove the bad correspondences. One way of picking a reasonable initial guess is the following. The initial guess is made by using the (for example 5) strongest corner points (for example Harris-corners) and their surrounding patches in one image as a model m_0 . For each of the other images $image_i$ matching is done as follows. Given n points in the model and m points extracted in $image_i$. Form an $(n+1) \times (m+1)$ cost matrix C whose top left $n \times m$ elements are $C_{ij} = d_E + d_P$, where d_E is the Euclidean distance between model point i and feature point j and d_P is the Euclidean difference in patch appearance. The last row $C_{n+1,j}$ is set to a constant representing the cost of not associating a feature point j with a model point. Similarly the last column $C_{i,m+1}$ is set to a constant representing the cost of not associating a model point j to any of the feature points. The matching is then done by solving the transport problem with supply of $s = [1, \dots, 1, m]$ and demand $t = [1, \dots, 1, n]$. Here we used standard algorithms for solving the transport problem, cf. [15]. A number of images from the training set can be tested for giving initial models m_0 in this way and DL can be used to pick the best initial guess for all the images.

5. Scale Space

The optimization is performed in a scale space framework, and the resulting model also has patch-information from several scales.

The original images are smoothed and down-sampled to create scale space representatives. Also the coordinates of the extracted feature points (extracted at the original level) are scaled to fit the down-sampled images and also rounded off to integers. In this way both gray level intensities (parts) are point coordinates (geometry) are represented with different detail at different scales. Next 7×7 patches are sampled around each point so that the same size patch (as measured in pixels) covers different relative amounts of the images at different scale.

In usual fashion the optimization is first run till convergence at the coarsest level and then the result is used as an initial estimate for the next level. As we move to the next level, point coordinates are simply scaled to the appropriate

range. For the patches however the patches from the new level do not replace the patches from the old level, but instead they are kept so that the new patch represents the local information at greater detail as well as the wider appearance information with less detail. Since the patch description becomes richer, this of course means that the description length increases, so before starting the optimization at the new level, a new best description length is calculated using the richer descriptions. For the point coordinates there is no such effect, since before calculating the description length, the shapes are temporarily scaled so that they have mean distance from the points to the center of the shape.

The resulting models then describe the geometry at the detail of the finest level and the parts at several levels of locality and detail.

6. Unlabeled Images

Assume now that not all the images are of objects from the same object class, but instead there are images of objects from different object classes mixed together and we want to automatically get a model for each object class.

Introduce a label for each image, denoting which object class it is currently assumed to belong to. Given such labels for all the images we get a set of problems as described in the previous sections, one for each object class. The total description length is simply the sum of the description lengths for the individual object classes.

Instead of just picking *one* random image to give the initial model as described in the initial guess section above we can pick $n_{classes}$ random images to give $n_{classes}$ initial models. For each image every initial model is tried and the initial label is set coresponding to the model giving the smallest cost in the transport problem solution. As above, we can try a number of pairs of initial models and use MDL to pick the best initial guess.

After the initial guess, the optimization terminates when none of the available operations give an improvement in description length. The available operations are the six described in the Optimization section above and also

7. Change the object class label for an image and fit the model for the new class to the image.

We can now use MDL to also get an automatic classification of the images as a natural part of the process.

7. Experimental Validation

7.1. Points Only

First we test the algorithm on images from only one object class. In the first experiments we only work with the points them self, i.e. not with the patches.

In the first experiment we took a digital film recording of a persons face as it moves in the scene. A sequence of 944

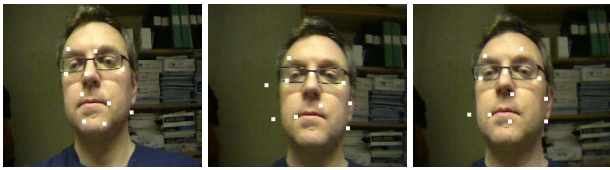


Figure 1. Three out of 100 frames used for testing. Detected feature points are shown as white squares

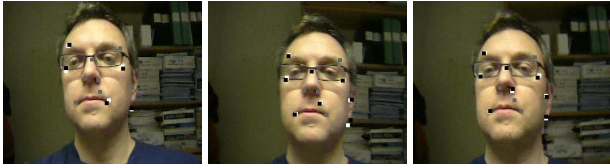


Figure 2. Three out of 100 frames used for testing. For measured points (in white) the fitted model points are shown in black. For missing data the fitted model points are shown in gray.

frames was captured and a standard interest point detector [12], was run on all of the frames. In each frame a face detector was run and those interest points that were within the rectangular frame of a face detector [8], was kept.

The first 100 frames were used for model estimation. This gave roughly 880 feature points (between 5 and 13 points in each frame). Three such frames are shown in Figure 1 together with the extracted feature point shown as small rectangular points.

The initial guess ordering resulted in 584 of the 880 feature points being associated with any of the 9 model points.

After local optimization to lower the description length we got a model with 12 model points. Here 740 of the 880 points were associated to a model point. In Figure 2 is shown three frames out of the 100 overlaid with feature points and best fit of the 12 model points obtained after minimizing description length. Notice that certain points in Figure 1 are classified as outliers and are not shown in Figure 2.

7.2. Patches

After these initial experiments we move on to using the patches as well as the points. Here the training set consists of (10-30) head shots of different persons and no face detector is used. Feature points (corners and edges) and surrounding patches are extracted and the optimization is performed as described in the previous sections. As can be seen in Figure 3 relevant model points are selected and correspondences are established.

7.3. Occlusions

Next the algorithm was tested on images with synthetic occlusions. For every image in the training set there was a 50% chance of putting in an occlusion. The occlusions were rectangles with random proportions, size, location and

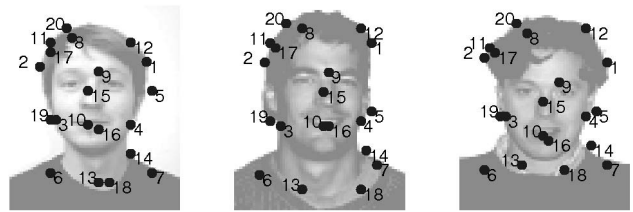
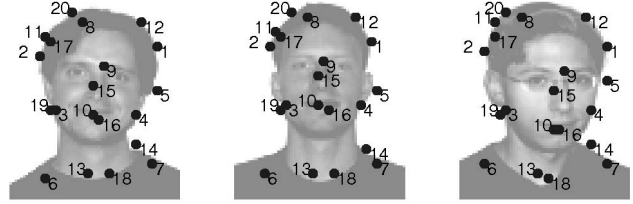


Figure 3. The selected model points on a subset of the images. Around each point patches at different scales are included and modeled.



gray level intensity. As can be seen in Figure 4 the algorithm handles the occlusions well. Occluded points are set as missing and the correspondences of remaining points are not destroyed by the occlusions.

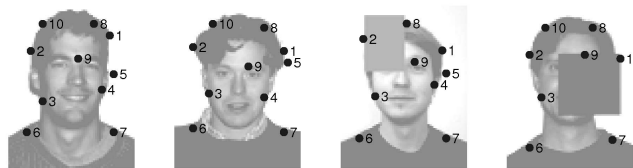


Figure 4. The selected model points on a subset of the images. For every image there was a 50% chance of a random occlusion.

7.4. Unlabeled Images

Next we test the algorithm on unlabeled images. We use the head shots from the above experiments together with some images of star fish from the Caltech101 database [9]. The images are mixed together in random order before they are fed to the algorithm. The result is that all images are classified correctly and we get two separate sets of correspondences and models as illustrated in Figure 5.

Finally we test the algorithm on unlabeled images with occlusions. The images are as in the previous experiment but for every image there was a 40% chance of putting in a random occlusion. All images are still classified correctly and the occlusions are handles well, see Figure 6.

8. Conclusions

In this paper we have studied the problem of automatic construction (i.e. no manual annotation of training data) of parts+geometry models (giving robustness as compared to modeling the entire image) from unlabeled images (i.e. mixed object classes) with occlusions using MDL (proven

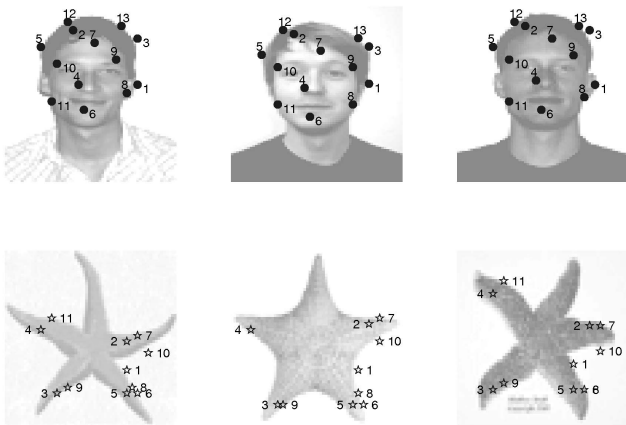


Figure 5. The selected model points on a subset of the images. Stars/dots denote that the image is classified as belonging to class 1 / class 2.

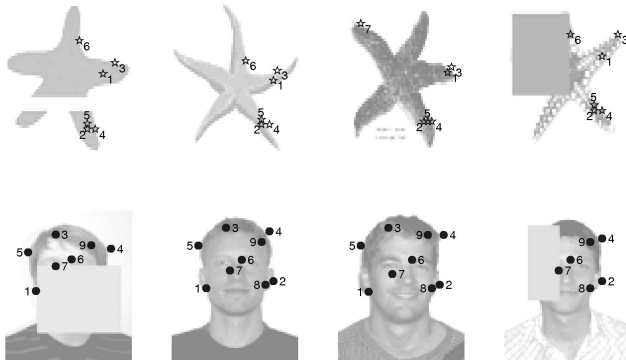


Figure 6. The selected model points on a subset of the images. Stars/dots denote that the image is classified as belonging to class 1 / class 2.

to be successful at establishing correspondence for model building).

The result is an algorithm that given a set of images of objects from different object classes extracts points and patches as different scales and from this determines among other things (i) which images are of objects from the same object class, (ii) the number of model points for each class, (iii) which image points are outliers/clutter and which are inliers, (iv) correspondences, (v) the mean shape and shape variational modes of the models (geometry) and (vi) for each model point (part) the mean patch and patch appearance variational modes.

The resulting models can be used for many different applications including for example recognition, pose determination, tracking and shape/appearance analysis.

References

[1] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE trans. on information theory*, 44(6):2743–2760, 1998.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[3] R. Davies, C. Twining, T. Cootes, J. Waterton, and C. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. medical imaging*, 21(5):525–537, 2002.

[4] R. H. Davies, T. F. Cootes, and C. J. Taylor. A minimum description length approach to statistical shape modeling. In *Information Processing in Medical Imaging*, 2001.

[5] A. Drobchenko, J. Ilonen, H. Kamarainen J, A. Sadovnikov, H. Kalviainen, and M. Hamouz. Object class detection using local image features and point pattern matching constellation search. In *SCIA*, volume 1, pages 273–282, 2007.

[6] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, Inc., 1998.

[7] A. Ericsson. *Automatic Shape Modelling with Applications in Medical Imaging*. PhD thesis, Lund University, Centre for Mathematical Sciences, Box 118, SE-22100, Lund, Sweden, 2006.

[8] A. P. Eriksson and K. Åström. Robustness and specificity in object detection. In *Proc. International Conference on Pattern Recognition, Cambridge, UK*, 2004.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples. In *CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[10] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples. *Computer Vision and Image Understanding*, 106:59–70, 2007.

[11] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71(3):273–303, 2007.

[12] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[13] J. Karlsson and A. Ericsson. Aligning shapes by minimising the description length. In *Scandinavian Conf. on Image Analysis, Juensuu, Finland*, 2005.

[14] G. Langs, R. Donner, P. Peloschek, and H. Bischof. Robust autonomous model learning from 2d and 3d data sets. In *Medical Image Computing and Computer-Assisted Intervention*, pages 968–976, 2007.

[15] D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[16] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[17] C. E. Shannon. Communication in the presence of noise. *Proc. IRE*, 37, 1949.

[18] C. Twining, T. Cootes, S. Marsland, V. Petrovic, S. R.S., and C. Taylor. Information-theoretic unification of group-wise non-rigid registration and model building. *Proceedings of Medical Image Understanding and Analysis*, 2:226–230, 2006.