Integrating spatial layout of object parts into classification without pairwise terms: application to fast body parts estimation from depth images

Mingyuan Jiu, Christian Wolf and Atilla Baskurt
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR CNRS 5205, F-69621, France
{*mingyuan.jiu, christian.wolf, atilla.baskurt*}*@liris.cnrs.fr*

Keywords: Object detection, pose estimation, spatial layout, unary terms, randomized decision forest, Kinect

Abstract: Object recognition or human pose estimation methods often resort to a decomposition into a collection of parts. This local representation has significant advantages, especially in case of occlusions and when the "object" is non-rigid. Detection and recognition requires modelling the appearance of the different object parts as well as their spatial layout. The latter can be complex and requires the minimization of complex energy functions, which is prohibitive in most real world applications and therefore often omitted. However, ignoring the spatial layout puts all the burden on the classifier, whose only available information is local appearance. We propose a new method to integrate the spatial layout into the parts classification without costly pairwise terms. We present an application to body parts classification for human pose estimation. As a second contribution, we introduce edge features from RGB images as a complement to the well known depth features used for body parts classification from Kinect data.

1 Introduction

Object recognition is one of the fundamental problems in computer vision, as well as related problems like face detection and recognition, person detection, and associated pose estimation. Local representations as collections of descriptors extracted from local image patches are very popular. This representation allows robustness against occlusions and permits non-rigid matching of articulated objects, like humans and animals. The representation is inherently structural and is therefore difficult to use in a statistical learning framework.

For object recognition tasks, the known methods in the literature vary in their degree of usage of spatial relationships, between methods not using them at all, as for instance the bags of visual words model (Sivic and Zisserman, 2003), and rigid matching methods using all available information, e.g. based on RANSAC (Fischler and Bolles, 1981). The former suffer from low discriminative power, whereas the latter only work for rigid transformations and cannot be used for articulated objects. Methods for non-rigid matching exist. Graph-matching and hypergraph matching, for instance, restricts the verification of spatial constraints to neighbors in the graph. However, non trivial formulations require minimizing a complex energy functions and are NP-complete (Torresani et al., 2008; Duchenne et al., 2009).

Pictorial structures, deformable parts based models, have been introduced as early as in 1973 (Fischler and Elschlager, 1973). The more recent seminal work creates a Bayesian parts based model of the object and its parts, where a prior is put on the possible relative parts locations (Felzenszwalb and Huttenlocher, 2005). The underlying discrete optimization problem is similar to the one in graph matching: an energy function coding the spatial relationships in pairwise terms needs to be minimized. However, here his prior is a tree structured Markov random field, whose absence of cycles makes minimization of the underlying energy function relatively fast — of course much slower than a model without pairwise terms. Apart from the tree structure, the energy minimization problem is similar as the one in (Fischler and Elschlager, 1973). The geometric configuration between parts is learned automatically. In (Felzenszwalb et al., 2010) the Bayesian model is replaced with a more powerful discriminative model, where scale and relative position of each part are treated as latent variables and searched by Latent SVM. This model obtained the best performance in the 2006 PASCAL person detec-

tion challenge.

A similar problem occurs in tasks where joint object recognition and segmentation is required. Layout CRFs and extensions model the object as a collection of local parts (patches or even individual pixels), which are related through an energy function (Winn and Shotton, 2006). However, unlike pictorial structures, the energy function here contains cycles which makes minimization more complex, for instance through graph cuts techniques. Furthermore, the large number of labels makes the expansion move algorithms inefficient (Kolmogorov and Zabih, 2004). In the original paper (Winn and Shotton, 2006), and as in our proposed work, the unary terms are based on randomized decision forests. Another related application which could benefit from this contribution is full scene labelling (Farabet et al., 2012).

Pose estimation methods are also often naturally solved through a decomposition into body parts. A preliminary pixel classification step segments the object into body parts, from which the joint positions can be estimated in a second step. The well known method used for the MS Kinect system completely ignores the spatial relationships between the objects parts and puts all the classification burden on the pixel wise working classifier (Shotton et al., 2011). The decision function to be learned by the classifier is complex and therefore requires a learning machine with complex architecture, which is difficult to learn. The good performance of the system has been obtained with an extremely large training set of $2 \cdot 10^9$ training vectors extracted from 1 million images and training on a computation cluster with 1000 nodes.

In this paper we propose to a method which segments an object into parts through pixelwise classification and which integrates the spatial layout of the part labels. Like the methods ignoring the spatial layout, it is extremely fast as no additional step needs to be added to pixelwise classification and no energy minimization is necessary. The (slight) additional computational load only concerns learning at an offline stage. The goal is not to compete with methods based on energy minimization, which is impossible through pixelwise classification only. The objective is to improve the performance of pixelwise classification by using all available information during learning.

Classical learning machines working on data embedded in a vectors space, like neural networks, SVM, randomized decision trees, Adaboost etc., are in principal capable of learning arbitrary complex decision functions, if the underyling prediction model (architecture) is complex enough. In reality the available amount of training data and computational complexity limit the complexity which can be learned. In most cases only few data are available with respect to the complexity of the problem. It is therefore often useful to impose some structure on the model. We already mentioned structured models based on energy minization and their computational disadvantages. Manifold learning is another technique which assumes that the data, although embedded in a high dimensional space, is distributed according to lower dimensional manifold in that space. Semi-supervised learning uses a large amount of additional training data, which is unlabeled, to help the learning machine to better infer the structure of the decision function. In this work we propose to use prior knowledge in the form of the spatial layout of the labels to add structure to the decision function learned by the learning machine.

The contributions of this paper are threefold:

- The integration of the spatial layout of part labels into learning machines, in particular randomized decision forests;
- We introduce features extracted from edges calculated on the RGB image and show that they can provide valuable complementary information to the traditional depth features. We show that good performance in recognition of objects could be achieved, even when simple features are employed;
- As a third contribution we propose automatic ground truth creation for object detection with 2D markers.

The paper is organized as follows: section 2 presents the learning procedure which integrates the spatial layout of a part based model into the prediction model of a randomized decision forest. Section 3 introduces edge comparison features which can complement the classical depth features for pose estimation. Section 4 explains the experiments we performed on two different tasks (pose estimation and door detection), and section 5 finally concludes.

2 Learning object part classifiers from spatial layouts

We consider problems where the pixels i of an image are classified as belonging to one of L target labels by a learning machine whose alphabet is $\mathcal{L} = \{1 \dots L\}$. To this end, descriptors F_i are extracted on each pixel i and a local path around it, and the learning machine takes a decision $l_i \in \mathcal{L}$ for each pixel. If the target labels are parts in a spatial object, then a powerful prior can be defined over the set of possible labellings.

Beyond the classical Potts model known from image restoration (Geman and Geman, 1984), which favors equal labels for neighboring pixels over unequal labels, additional (soft) constraints can be imposed. Labels of neighboring pixels can be supposed to be equal, or at least compatible, i.e. belonging to parts which are neighbors in the spatial layout of the object. If the constraints are supposed to be hard, the resulting problem is a constraint satisfaction problem. In computer vision this kind of constraints is often modelled soft through the energy potentials of a global energy function:

$$E(l_1,\ldots,l_N) = \sum_i U(l_i,F_i) + \mu \sum_{i\sim j} D(l_i,l_j) \quad (1)$$

where the unary terms $U(\cdot)$ integrate decisions and confidence of a pixelwise employed learning machine and the pairwise terms $D(\cdot,\cdot)$ are over couples of neighbors $i \sim j$ and favor certain pair label configurations over others. In the case of certain simple models like the Potts model, the energy function is submodular and the exact solution can be calculated in polynomial time using graph cuts (Kolmogorov and Zabih, 2004). Taking the spatial layout of the object parts into account results in non-submodular energy functions which are difficult to solve. Let's note that even the complexity of the submodular problem (quadratic on the number of pixels in the worst case) is far beyond the complexity of pixelwise classification with unary terms only.

The goal of our work is to improve the learning machine in the case where it is the only source of information, i.e. no pairwise terms are used for classification. Traditional learning algorithm in this context are supervised and use as only input the training feature vectors f_i as well as the training labels l_i, where i is over the pixels of the training set. We propose to provide the learning machine with additional information, namely the spatial layout of the labels of the alphabet $\mathcal{L}$. Some pairs of labels are closer than other pairs in that they correspond to neighboring parts. The risk associated for misclassifying a label with a neighboring label should therefore be lower than the risk misclassifying a label with a not neighboring label.

For learning machines which directly minimize the loss associated with classification, this additional information can be integrated directly. Multi-class SVM with one-against-one strategy could be adapted to this extension by adding weights to the voting mechanism. We did not pursue this direction in this work. Multiple layer perceptrons learn by minimizing a loss function for each output unit. Layout information could be integrated by identifying by weighting the loss of each output unit according to the type of classification error made. We did not pursue this direction in this work.

Randomized decision forests

In this paper we focus on randomized decision forests (RDF) as learning machines, because they have shown to outperform other learning machines in this kind of problem and because they have become very popular in computer vision lately (Shotton et al., 2011). Decision trees, as simple tree structured classifiers with decision and terminal nodes, suffer from over-fitting. Randomized forests, on the other hand, overcome this drawback by integrating distributions over several trees. To classify a new vector, it is passed down each tree until a terminal node is reached which contains a distribution over labels.

A difficulty in this context is the classical learning algorithm for RDFs (Lepetit et al., 2004), which trains each tree separately, layer by layer. Each layer is also trained separately, which allows the training of deep trees with a complex prediction model. The drawback of this approach is the absence of any gradient on the error during training. Instead, training maximizes the gain in information based on Shannon entropy. In the following we give a short description of the classical training procedure.

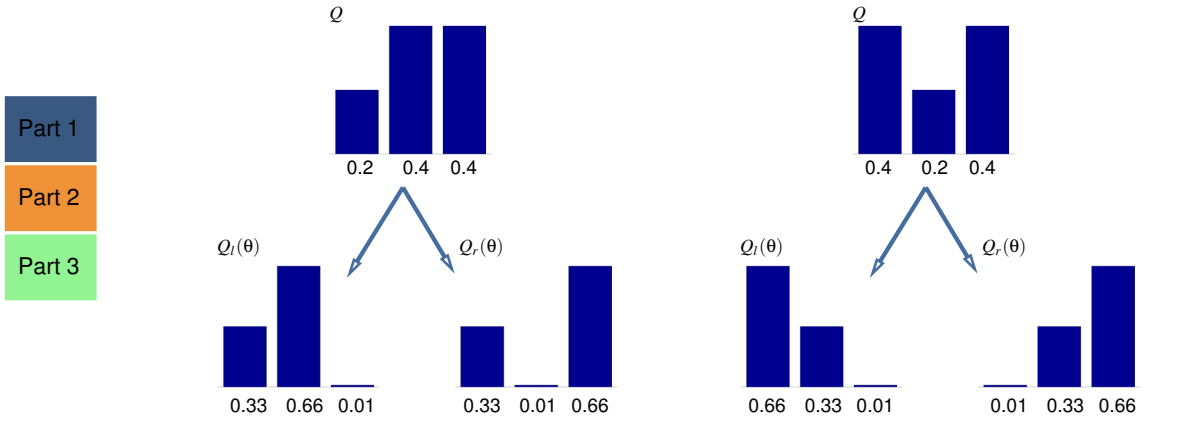
We describe the version of the learning algorithm from (Shotton et al., 2011) which jointly learns features and the parameters of the tree, i.e. the thresholds for each decision node. We denote by θ the set of all learned parameters (features and threshold) for each decision node.

For each tree, a subset of training instances is randomly sampled with replacement. The layers are trained in a top-down approach.

1. Randomly sample a set of candidates θ.
2. Partition the set of input vectors into two sets, one for the left child and one for the right child according to the threshold $\tau \in \theta$. Denote by Q the label distribution of the parent and by $Q_l(\theta)$ and $Q_r(\theta)$ the label distributions of the left and the right child node, respectively.
3. Choose θ with the largest gain in information:

$$\begin{aligned} \theta^* &= \arg\max_{\theta} \quad G(\theta) \\ &= \arg\max_{\theta} \quad H(Q) - \sum_{s\in\{l,r\}} \frac{|Q_s(\theta)|}{|Q|} H(Q_s(\theta)) \end{aligned} \quad (2)$$

   where $H(Q)$ is the Shannon entropy from class distribution of set Q.
4. Recurse the left and right child until the predefined level or largest gain is arrived.



	Classical Learning:	$G(\theta) = -0.32$	Classical learning :	$G(\theta) = -0.32$
	Spatial learning:	$G'(\theta) = -0.06$	Spatial learning :	$G'(\theta) = -0.01$
(a)		(b)		(c)

Figure 1: An example of three parts: (a) part layout; (b) a parent distribution and its two child distributions for a given θ; (c) a second more favorable case. The gain in information for the spatial learning cases are given with $\lambda = 0.3$ and equal number of pixels between the left and the right subtree.

Spatial learning for randomized decision forests

In what follows we will integrate the additional information on the spatial layout of the object parts into the training algorithm, which will be done without using any information on the training error. Let us first recall that the target alphabet of the learning machine is $\mathcal{L} = \{1 \ldots L\}$. Let us then imagine that we create groups of pairs of two labels, giving rise to a new alphabet $\mathcal{L}' = \{11, 12, \ldots 1L, 21, 22, \ldots 2L, \ldots, LL\}$. Each of the new labels is a combination of two original labels. Assuming independent and identically distributed (i.i.d.) original labels, the probability of a new label ij consisting of the pair of original labels i and j is the product of the original probabilities, i.e. $p(ij) = p(i)p(j)$. The Shannon entropy of a distribution Q' over the new alphabet is therefore

$$H(Q') = \sum_k -p(k) \log p(k) \qquad (3)$$

where k is over the new alphabet. This can be expressed in terms of the original distribution over the original alphabet:

$$H(Q') = \sum_{i,j} -p(i)p(j) \log[p(i)p(j)] \qquad (4)$$

We can now separate the new pairwise labels into two different subsets, the set of neighboring labels $\mathcal{L}'^1$, and the set of not neighboring labels $\mathcal{L}'^2$, with $\mathcal{L}' = \mathcal{L}'^1 \cup \mathcal{L}'^2$. We suppose that each original label is neighbor of itself. In the same way, a distribution Q' over the new alphabet can be split into two different distributions Q'^1 and Q'^2 from these two subsets.

Then a learning criterion can be defined which defines the gain in information obtained by parameters θ as a sum over two parts of the histogram Q', each part being calculated over one subset of the labels:

$$G'(\theta) = \lambda\, G'^1(\theta) \quad + \quad (1-\lambda)\, G'^2(\theta) \qquad (5)$$

where

$$G'^i(\theta) = H(Q'^i) - \sum_{s \in \{l,r\}} \frac{|Q'^i_s(\theta)|}{|Q'^i|} H(Q'^i_s(\theta)) \qquad (6)$$

Here, λ is a weight, and $\lambda < 0.5$ in order to give separation of non neighboring labels a higher priority.

A numerical example

Let's consider a simple parts based model with three parts numbered from 1 to 3. We suppose that part 1 is a neighbor of 2, that 2 is a neighbor of 3, but that 1 is not a neighbor of 3. Let's also consider two cases where a set of tree parameters $\theta = \{u, v, \tau\}$ splits a label distribution Q into two distributions, the left distribution $Q_l(\theta)$ and $Q_r(\theta)$.

The distributions for the two different cases are given in figures 1a and 1b, respectively. Note that the parent distribution Q for the second case can be obtained by permutating the respective distribution Q from the first case. Similarly, the child distributions for the second case are permutated versions of the child distributions of the first case. Not surprisingly,

the classical measure on Shannon entropy is equal for both cases: the gain in information is $G(\theta) = -0.32$ for both of them.

If we take into account the spatial layout of the different parts, we can see that the gain in information is actually higher in the second case:

- In the first case, the highest gain in information is obtained for parts 2 and 3, which are equally probable in the parent distribution Q, whereas a high difference in probability is obtained for the child distributions $Q_l(\theta)$ and $Q_r(\theta)$. However, parts 2 and 3 are neighbors.
- In the second case, following a similar reasoning, the highest gain in information is obtained for parts 1 and 3 which are not neighbors.

The new information gain measure reflects this difference. Setting λ=0.3, we get $G'(\theta) = -0.17$ for the first case, whereas $G'(\theta) = -0.03$ for the second case.

3 Depth and RGB edge features

In (Shotton et al., 2011), depth features have been proposed for pose estimation from Kinect depth images. These features work well in different applications, from point matching in RGB images (Lepetit et al., 2004) to human pose estimation in depth images (Shotton et al., 2011). One of their main advantages is their simplicity and their computational efficiency. Briefly, at a given pixel x, the depth difference between two offsets centered at x is computed:

$$f_\theta(I, \mathbf{x}) = d_I(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}) - d_I(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}) \quad (7)$$

where $d_I(\mathbf{x})$ is the depth at pixel $\mathbf{x}$ in image I, parameters $\theta = (\mathbf{u}, \mathbf{v})$ are two offsets and are normalized by the current depth for depth-invariance. A single feature vector contains several differences, each comparison value being calculated from a different pair of offsets $\mathbf{u}$ and $\mathbf{v}$. These offsets are learned during training together with the prediction model, as described in section 2.

RGB edge features

Consumer depth cameras like Kinect deliver an RGB image together with the depth image. Combining both images may lead to richer information and better classification performance, especially in cases where texture is meaningful for the distinction of otherwise identical 3D objects. RGB comparison features have been introduced for keypoint matching (Lepetit et al., 2004). Here we extend this concept further by introducing edge comparison features extracted from the grayscale image.

Psychophysical studies show that we can recognize a object only with its contour, so contour is an important visual cue for object recognition. In (Shotton et al., 2008), contour is defined as the outline (silhouette) together with the internal edges of the object, which enable to represent the spatial structure of the object.

Now, contours of an object are usually sparsely distributed, which means that comparison features can not directly by applied to edge images. In the settings we are interested in, namely joint learning of features and of the prediction model with a RDF, we need features whose position can be sampled and tested by the training algorithm. Our solution to this problem is inspired by chamfer distance matching, which is a classical method to measure the similarity between contours (Barrow et al., 1977). We compute a distance transform on the edge image, where the value of each pixel is the distance to its nearest edge. Given a grayscale image I and its binary edge image E, the distance transform DT_E is computed as:

$$DT_E(\mathbf{x}) = \min_{\mathbf{x}':E(\mathbf{x}')=1} ||\mathbf{x} - \mathbf{x}'|| \quad (8)$$

The distance transform can be calculated in linear time using a two-pass algorithm.

We propose two different types of features based on edges, the first using edge magnitude, and the second edge orientation. The former is defined as:

$$f_\theta^{EM}(\mathbf{x}) = DT_E(\mathbf{x} + \mathbf{u}) + DT_E(\mathbf{x} + \mathbf{v}) \quad (9)$$

where $\mathbf{u}$ and $\mathbf{v}$ are the same in (7). Figure 2 shows the distance transform image for an input image showing a door (we will describe our experiments on doors in the experimental section). This feature indicates the existence of edges near two offsets.

Edge orientation features can be computed in a similar way. In the procedure of distance image, we can get another orientation image O_E, in which the value of each pixel is the orientation of its nearest edge:

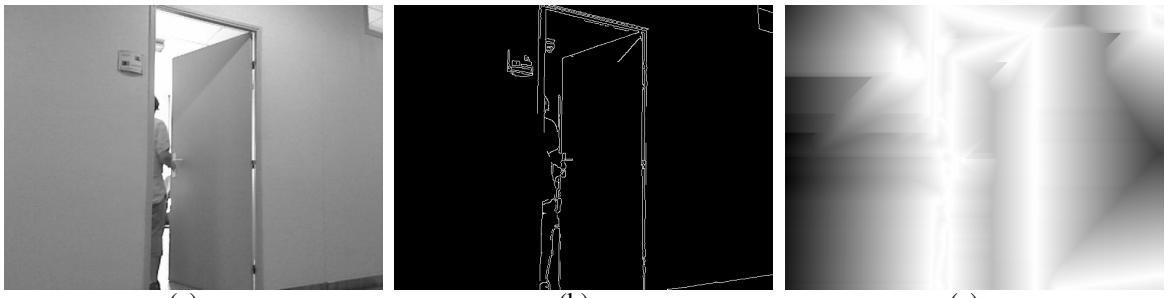
$$O_E(\mathbf{x}) = Orientation\left(\arg\min_{\mathbf{x}':E(\mathbf{x}')=1} ||\mathbf{x} - \mathbf{x}'||\right) \quad (10)$$

The feature is computed as the difference in orientation for two offsets:

$$f_\theta^{EO}(\mathbf{x}) = O_E(\mathbf{x} + \mathbf{u}) - O_E(\mathbf{x} + \mathbf{v}) \quad (11)$$

where the minus operator takes into account the circular nature of angles. We discretize the orientation to alleviate the effect of noise.

The objective of both features is to capture the edge distribution at specific locations in the image, which will be learned by the RDF.



(a) (b) (c)

Figure 2: The importance of edges: (a) original grayscale image of an open door; (b) edge image; (c) distance transform.

4 Experiments

We performed experiments for two different applications, described in the following two sub sections:

- the spatial learning algorithm was validated on a pose estimation application. We would like to point out that the learning method can be applied to any parts based model which integrates pixelwise classification with random forests, for instance also methods for joint object recognition and segmentation;
- the edge features have been additionally validated on an task requiring the detection of doors from images.

Body part estimation

The proposed learning algorithm has been evaluated on the *CDC4CV Poselets* dataset introduced in (Holt et al., 2011). Our goal was not to beat the state of the art in pose estimation, but to show that spatial learning is able to improve pixelwise classification of parts based models. The dataset contains upper body poses taken with Kinect and consists of 345 training and 347 test depth images. Along with the images, the authors also supplied corresponding annotation files which contain the locations of 10 articulated parts: head(H), neck(N), left shoulder(LS), right should(RS), left upper arm(LUA), left forearm(LFA), right upper arm(RUA), right forearm(RFA) left hip(LH), right hip(RH). A single position is provided with each part, we created groundtruth segmentations through nearest neighbor labeling. In our experiments we defined the part below the wrist as other (the black area in the Figure 3).

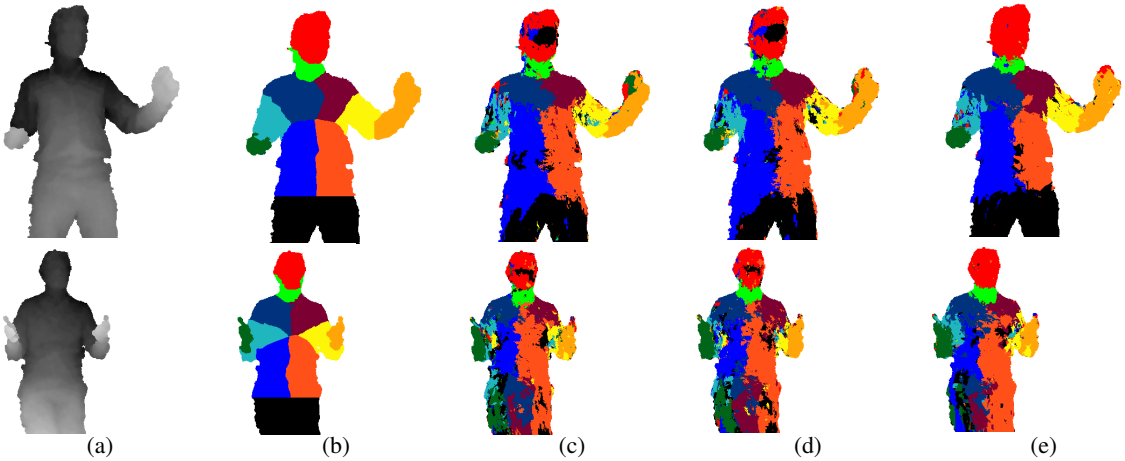
Unless otherwise specified, the following parameters have been used for RDF learning: 3 trees each with a depth of 9; 2000 randomly selected pixels per image, roughly distributed across the body; 4000 candidate pairs of offsets; 22 candidate thresholds; offsets and thresholds have been learned separately for each node in the forest. For spatial learning, 28 pairs of neighbors have been identified between the 10 parts based on a pose where the subject stretches his arms. The parameter λ was set to 0.4.

We evaluate our method at two result levels: pixelwise classification and part recognition. Pixelwise decisions are directly provided by the random forest. Part localisations are obtained from the pixelwise results through pixel pooling. For each pixel, the RDF provides a posterior probability for each part, which can be used to create posterior images for each part. After non-maximum suppression and low pass filtering, the location with largest response is used as an estimate of the part.

Table 1 shows the confusion matrices and the classification accuracies of the three settings. A baseline has created with classical RDF learning and depth features, shown in table 1a. Spatial learning with depth features is shown in 1b, and spatial learning with depth and edge magnitude features is shown in 1c. We can see that that spatial learning can obtain a performance gain, although the layout is used in the prediction model only and no pairwise terms have been used. Figure 3 shows some classification examples, which demonstrates that spatial learning makes the randomized forest more discriminative. The segmentation output is cleaner, especially at the borders.

At part level, we report our results according to the part estimation metric by (Ferrari et al., 2008): a groundtruth part is matched to a detected part if and only if the endpoints of the detected part lie within a circle of radius r=50% of the length of the groundtruth part and centered on it. Table 2 shows our results on part level using the three settings: classical randomized learning, spatial learning using depth features and spatial learning with depth and edge magnitude features. It demonstrates that spatial learning improves recognition performance for most of parts.

The experiments at both pixelwise and part level demonstrate that spatial learning makes randomized forest more discriminative by integrating the spatial



(a) (b) (c) (d) (e)

Figure 3: Examples of the pixelwise classification: each row is an example, each column is a kind of classification results, (a) test depth image; (b) part segmentation; (c) classical classification; (d) spatial learning with depth features; (e) spatial learning with depth features and edge magnitude features.

		head	neck	should	LUA	LFA	RUA	RFA	left wrist	right wrist	average
Classical E		46.69	0.29	34.58	2.02	0	21.90	77.81	1.15	19.88	22.70
Spatial E	$\lambda = 0.4$	52.45	0	39.48	1.44	0	14.41	82.13	0.86	22.48	23.70
Classical D		77.81	0	41.79	1.73	0	10.09	63.11	0.29	21.04	23.98
Spatial D	$\lambda = 0.4$	88.47	1.15	40.92	0.28	0.28	10.66	67.72	5.18	28.24	26.99
Spatial D+E	$\lambda = 0.4$	89.05	0	58.21	0.86	0	25.65	72.91	0	13.54	28.91

Table 2: Correct part rate(%) for different feature settings: D=deph features; E=edge magnitude features.

layout into its prediction model. This proposition is very simple and fast to implement, as the standard pipeline can be still used. Only the learning method has been changed, the testing code is unchanged. There is no additional computational burden whatsoever during testing; a slide increase in computational complexity can be observed for learning. No complex discrete optimization problems need to be solved.

Door detection and automatic groundtruth creation.

We additionally evaluated our edge features on a problem involving the detection of (open and closed) doors from images. This concept is interesting for robot navigation as well as activity recognition in an indoor environment, as the presence and position of a door is an important clue to classify certain activities. No parts based model was used in these experiments, whose sole goal was additional validation for the edge comparison features.

We selected images from two different sets, which we denote by D1 and D2:

D1 This subset has been taken from the LIRIS human activities dataset[1] used for the ICPR HARL 2012 *human activities recognition and localisation competition* (Wolf et al., 2012). We choose three activities associated with doors: (i) a person enters or leaves an office; (ii) a person tries to enter an office unsuccessfully, and (iii) a person unlocks an office and then enters it. The set contains 336 images of various open doors in various office locations. We manually annotated the positions of the doors.

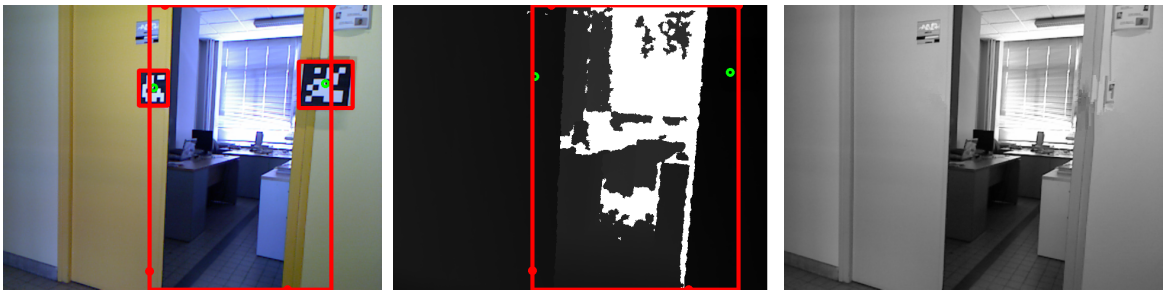
D2 We recorded another door dataset using a Kinect module, which contains open and closed doors. The dataset consists of approximately 9000 images. Instead of manually annotating the door positions we used an automatic groundtruth creation method described below. The subset has been split into two parts with open and closed doors, respectively.

Traditionally, groundtruth creation is performed manually by humans, which is a tedious and also time-consuming task. Of course, several tools has been developed to accelerate the annotations, such as the LabelMe image annotation tool [2] and the VATIC video annotation tool [3]. These tools help annotation but are

[1] http://liris.cnrs.fr/voir/activities-dataset

[2] http://labelme.csail.mit.edu

[3] http://mit.edu/vondrick/vatic



(a) (b) (c)

Figure 4: Automatic groundtruth annotation of doors: (a) the input RGB image showing the markers and superimposed bounding box; (b) the depth image; (c) the grayscale image with removed markers after inpainting.

still not fully automatic. Crowdsourcing may help to spread the work over a large number of people. Recently, K. Lai et al. proposed RGB-D scene annotation by means of 3D reconstruction (Lai et al., 2011). The object is manually located in the reconstructed 3D model, and then the 2D location is estimated in each frame from the estimated camera position using ego-motion estimation.

Here we describe a different method for automatic image annotation. Groundtruth for the D2 dataset has been created automatically from two 2D markers previously fixed at known locations beside the doors — see figure 4a for an example. The position of the markers is detected with ARToolKit[4]. The basic idea is simple: markers will change the RGB image but not the depth image, so depth features will not be altered by the presence of the markers. If features are calculated on the RGB image, then training the destination door detection algorithm on the markers needs to be avoided at all cost. We therefore remove the detected markers from the image and fill the uncovered area with a known inpainting algorithm (A. Criminisi, 2003).

The RGB camera of the Kinect module has been calibrated, intrinsic parameters of the RGB camera are therefore known. ARToolKit provides the 3D position and orientation of each detected marker, which allows to obtain the bounding box of the door from a single marker position (provided the position of the marker with respect to the door is known). For higher robustness to large camera angles, we fixed two different markers to the door, using an average of the two bounding box positions if both of them are detected. To obtain positions even for frames where no marker has been found (which is rare), a simple tracking algorithm using template matching has been added. Figure 4 shows example images of this automatic annotation method.

Calibration data is known for both subsets, D1 and D2, so which allows to calculate transformations between the depth image and the corresponding edge images. The proposed features in this paper have been calculated on pixel pairs taking into account these transformation.

[4] http://www.hitl.washington.edu/artoolkit

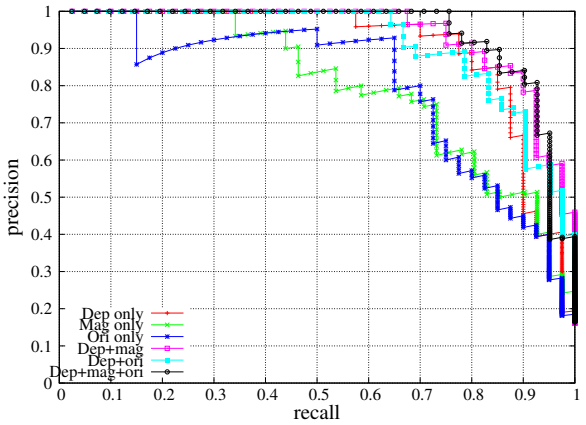


Figure 5: Results on the door detection problem on dataset D1: precision-recall curves for different feature settings.

When training a binary classifier, negative examples are also necessary and important to estimate the decision boundary. We employ bootstrapping to select negative examples for the classifier out of a large pool of available negative examples, in our case taken from all the non door images from the LIRIS human activities dataset. We found that one iteration of bootstrapping was enough.

For subset D1, two thirds of the positive examples were taken for training set and one third for testing. A randomized forest with 20 trees had been trained on a balanced dataset. All examples were scaled into a normalized block of 100×100 pixels. Figure 5 shows precision-recall curves for different settings with different feature types: depth features only, edge magnitude features only, edge orientation features only, depth and edge magnitude features, depth and edge orientation features, all three features together. We

	H	N	LS	RS	LUA	LFA	RUA	RFA	LH	RH
H	85	1	1	1	1	2	1	3	3	2
N	11	67	4	3	1	1	1	1	6	5
LS	2	2	56	6	4	2	0	1	18	9
RS	1	3	12	48	1	1	6	2	5	21
LUA	3	1	18	2	34	20	4	3	12	3
LFA	5	0	2	1	4	65	2	19	1	1
RUA	2	0	3	7	3	2	40	20	3	20
RFA	6	0	2	2	2	12	6	62	3	5
LH	2	1	15	1	4	4	0	0	67	6
RH	1	1	10	12	0	0	4	1	5	66

(a)

	H	N	LS	RS	LUA	LFA	RUA	RFA	LH	RH
H	84	1	1	1	1	1	0	4	4	3
N	12	70	4	2	1	1	0	2	6	2
LS	2	3	58	6	5	2	0	1	14	9
RS	2	4	14	49	0	0	5	3	4	19
LUA	3	1	17	2	35	19	5	3	13	2
LFA	4	0	2	1	4	63	2	22	1	1
RUA	3	0	4	9	2	1	37	21	3	20
RFA	5	1	2	3	1	10	5	66	2	5
LH	2	1	16	2	4	3	0	0	67	5
RH	1	1	9	15	0	0	2	1	4	67

(b)

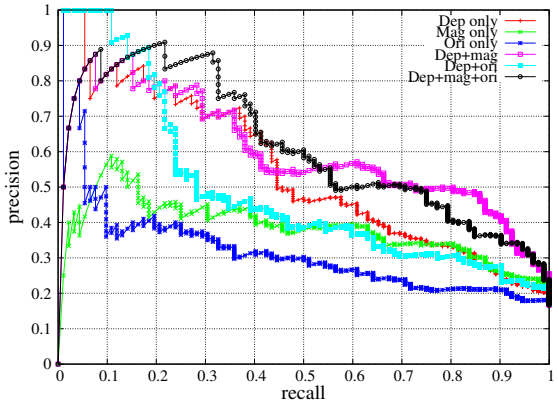
	H	N	LS	RS	LUA	LFA	RUA	RFA	LH	RH
H	94	1	0	0	0	2	0	2	0	1
N	24	60	5	3	1	1	0	2	2	2
LS	1	3	65	4	5	2	0	0	14	6
RS	1	3	12	57	1	0	5	3	3	15
LUA	2	1	18	1	44	22	1	1	9	1
LFA	6	0	3	1	4	73	1	9	2	1
RUA	3	0	2	15	1	1	41	21	2	14
RFA	5	0	2	3	1	5	5	67	3	9
LH	0	0	13	1	3	5	0	0	72	6
RH	1	1	5	11	0	0	2	1	4	75

(c)

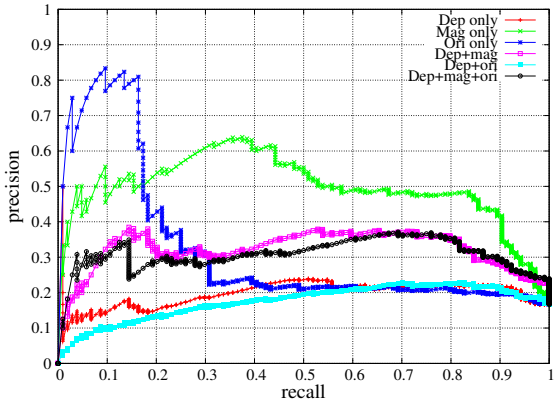
Table 1: Results on body part classification in pixelwise level: (a) classical RF learning; (b) spatial learning with only depth features; (c) spatial learning with depth and edge magnitude features. Respective accuracies: 60.30%, 61.05%, 67.66%.

can see that depth features are more discriminative than edge features extracted on the grayscale image. This is a predictable performance for a set containing open doors. However, even on open doors the addition of edge features can significantly improve detection. The combination outperforms any single feature type.

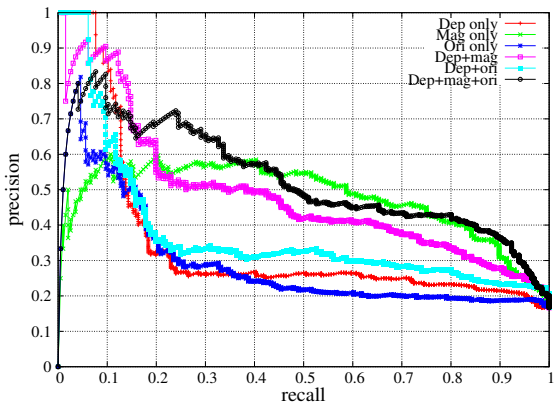
For dataset D2, we subsampled the images in the dataset D2 by taking every fifth frame. Two thirds of the dataset have been used for training and one third for testing. The results are shown in figure 6. As expected, we can see that for open doors the results are similar to the ones obtained for subset D1 (a combination working best, and depth features being the best



(a)



(b)



(c)

Figure 6: Results on the door detection problem on dataset D1; precision-recall curves for different feature settings: (a) open doors; (b) closed doors; (c) open and closed doors.

single feature). Also as expected, for closed doors the depth features do not work well, as doors are coplanar with the door frame and wall. Edge magnitude features give reasonable good performance. For the mixture of open and closed doors, multiple features perform better than the single features, similar to the setting of open doors.

5 Conclusion

In this paper, we proposed a novel learning algorithm for randomized decision forests which integrates information on the spatial layout of target labels. The classification algorithm is of exactly the same computational complexity, a slightly higher computational burden is put on the learning algorithm. We applied our algorithm on the body part classification, although any other application requiring the segmentation of an object into parts may benefit from the contribution. Results show that RDF indeed benefit from the integration of the information on the spatial layout of parts.

Another contribution extends the well known depth comparison features to include edge presence information obtained from grayscale images. The recognition of doors in RGB-D images was significantly improved by the combination of depth features and edge features.

REFERENCES

A. Criminisi, P. Prez, K. T. (2003). Object removal by exemplar-based inpainting. In *Proceeding of the conference on Computer Vision and Pattern Recognition(CVPR)*, volume 2, pages 721–728.

Barrow, H., tenenbaum, J., Bolles, R., and Wolf, H. (1977). Parametric correspondence and chamfer matching: two new techniques for image matching. In *Joint conference on artificial intelligence*, pages 659–663.

Duchenne, O., Bach, F. R., Kweon, I.-S., and Ponce, J. (2009). A tensor-based algorithm for high-order graph matching. In *CVPR*, pages 1980–1987.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2012). Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *International Conference on Machine Learning (ICML)*.

Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 61(1):55–79.

Ferrari, V., Marin-Jimenez, M., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, (figure 1):1–8.

Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communcations of the ACM*, 24:381395.

Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 22(1):6792.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Hoiem, D., Rother, C., and Winn, J. (2007). 3D layoutCRF for multi-view object class recognition and segmentation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8.

Holt, B., Ong, E.-J., Cooper, H., and Bowden, R. (2011). Putting the pieces together: Connected poselets for human pose estimation. In *ICCV Workshop on Consumer Depth Cameras for Computer Vision*.

Kolmogorov, V. and Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159.

Lai, K., Bo, L., Ren, X., and Fox., D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In *International Conference on Robotics and Automation*.

Lepetit, V., Pilet, J., and Fua, P. (2004). Point matching as a classification problem for fast and robust object pose estimation. In *CVPR*, pages 244–250.

Shotton, J., Blake, A., and Cipolla, R. (2008). Multiscale categorical object recognition using contour fragments. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1270–81.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proceedings of the International conference on computer vision (ICCV)*, volume 2, pages 1470–1477.

Torresani, L., Kolmogorov, V., and Rother, C. (2008). Feature correspondence via graph matching: Models and global optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 2, pages 596–609.

Winn, J. and Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceeding of the conference on Computer Vision and Pattern Recognition(CVPR)*, volume 1, pages 37–44.

Wolf, C., Mille, J., Lombardi, E., Celiktutan, O., Jiu, M., Baccouche, M., Dellandra, E., Bichot, C.-E., Garcia, C., and Sankur, B. (2012). The liris human activities dataset and the icpr 2012 human activities recognition and localization competition. Technical Report LIRIS RR-2012-004, Laboratoire d'Informatique en Images et Systmes d'Information, INSA de Lyon, France.