Depth-Based Detection
with
Region Comparison Features

Ruud Mattheij*, Kim Groeneveld, Eric Postma, H. Jaap van den Herik

Tilburg center for Cognition and Communication (TiCC), Tilburg University, P.O. Box 90153, 5000 LE, Tilburg

Abstract

Most object detection approaches proposed over the years rely on visual features to segregate objects from their backgrounds. Segregation may be facilitated by depth features, because they provide direct access to the third dimension. This enables accurate object-background segregation. While providing a rich source of information, depth images are sensitive to background noise. This paper presents and evaluates the Region Comparison (RC) features for fast and accurate body part detection. RC features are depth features inspired by the well-known Viola-Jones detector. Their performances are compared to the recently proposed Pixel Comparison (PC) features, which were designed for fast and accurate object detection from Kinect-generated depth images. The results of our comparative evaluation reveal that RC features outperform PC features in detection accuracy and computational efficiency. We conclude that RC features are to be preferred over PC features to achieve accurate and fast object detection in noisy depth images.

Keywords: Face detection, Person detection, Depth data, Haar-like features, Random forest classifier, Integral Image Representation

*Corresponding author

Email addresses: R.J.H.Mattheij@tilburguniversity.edu (Ruud Mattheij), K.Groeneveld@tilburguniversity.edu (Kim Groeneveld), E.O.Postma@tilburguniversity.edu (Eric Postma), h.j.vandenherik@law.leidenuniv.nl (H. Jaap van den Herik)

1. Introduction

In the last few years, the automatic detection of objects from digital video and image sources has gained considerable attention within the field of image analysis and understanding [1, 2, 3]. Many object detection approaches focus on two-dimensional visual features [4, 5, 6] in order to segregate objects from their backgrounds. Well-known visual features for object detection are the Haar-like features [7] as proposed by Viola and Jones [8, 9].

Despite the widespread and successful use of two-dimensional (2D) visual features in visual detection tasks, they have some limitations. Their main limitation is that they typically respond to local visual transitions, without being sensitive to the larger spatial context [10]. As a consequence, they are sensitive to factors that may influence scene properties locally, such as illumination conditions [11, 12]. Bright lights, for example, may cause shadows (i.e., non-object contours) in the image. Local 2D visual features will respond to the contours of the shadows in the same way as to the contours of other, real objects.

Typical situations in which 2D visual features fail, are those where variations in the third dimension (depth) lead to shape deformations. In general, the failures are caused by object pose variations [1, 13]. A wide variety of methods attempts to overcome these sensitivities. The most frequently applied methods focus on extracting context-sensitive features (see, e.g., [4]). Although such approaches improve classification performance, they tend to be costly in terms of computational resources [13, 14].

1.1. From 2D Features to 3D Features

To overcome the limitations of 2D features, we can add a third dimension, yielding 3D features (which combine 2D spatial and 1D depth information) [15, 16, 17, 18, 19]. Depth cues will then provide contextual information for a scene, thereby facilitating image segmentation [20, 21, 22, 23]. Indeed, visual objects such as faces or persons are much easier to distinguish in a 3D space than from a 2D image [24, 25]. In recent years, the use of depth cues became feasible by the development of affordable depth sensors, such as Microsoft Kinect [26].
1.2. Capturing Depth with Microsoft Kinect

The Microsoft Kinect device generates its depth images by (1) illuminating a spatial area with the Kinect’s infrared laser, and (2) triangulating the corresponding depth using an infrared sensor [27]. Using an infrared laser that passes through a diffraction grating, a grid of infrared dots is created. Given the known spatial distance between the Kinect’s infrared laser and sensor, matching the dots observed in an image with the dots projected using the pattern from the diffraction grating allows for effective depth triangulation. The resulting depth images have a resolution of $640 \times 480$ pixels. The pixel values of the depth images encode for the distance between an object and the Kinect device. A large depth value indicates a large distance between the object and the Kinect device, while a small depth value encodes for a small distance. Figures 4, 5, and 6 show several examples of depth images that are created with a Kinect device.

1.3. Shotton’s Pixel Comparison Features

Using the Microsoft Kinect, Shotton et al. [28, 29, 30] proposed a depth-based object detection algorithm that is able to classify individual pixel locations from single depth images as belonging to faces, body joints and body parts. To classify the pixel locations, Shotton et al. select a subset of random pixel locations from each depth image. For each pixel location $P$ from the subset, the depth difference is computed by comparing the depth values at two randomly chosen offset locations $Q$ and $R$. The offset locations are defined by the radius and angle with respect to $P$. The radius is defined to be inversely proportional to the depth value of $P$. A small depth value results in a larger radius for offset locations $P$ and $Q$, and vice versa. This way, a scale-invariant measure of depth is obtained. A single depth comparison between locations $Q$ and $R$ provides only a weak indication of the depth difference in a spatial area. Repeating the measurements for other random locations around point $P$, however, provides a fair description of the depth difference in an area around the location of point $P$. For the sake of readability, we refer to these pixel-based depth comparison features as the Pixel Comparison (PC) features.
There are two advantages of classifying individual pixel locations rather than image regions (e.g., by means of a sliding window): (1) the selection process allows for the detection of partially occluded objects, and (2) the classification process reduces the time required to process an entire depth image. Thus, using the PC features makes their detector computationally efficient. In addition to these qualities, the detector works directly on the raw input depth data, i.e., without an image preprocessing stage to reduce noise in the data [31]. The combination of efficient depth-comparison features and the raw input depth image results in a high detection speed, which allows for real-time operation.

The detection speed, however, comes at the cost of accuracy. The classification accuracy is hampered by two limitations [26, 32, 33]: (1) the limited quality of the depth images generated by the Kinect device, and (2) the limited resolution of the depth images. The first limitation arises from the triangulation sensor as used by the Kinect device. Depending on the image geometry, parts of a scene may not be illuminated by the sensor’s laser, i.e., the grid of infrared dots. These parts are therefore not captured by the infrared sensor, which results in empty regions in the depth image [32]. The second limitation is due to the point density of the Kinect device’s sensor. Using its laser and depth sensor, the Kinect device generates a point cloud of triangulated depth measurements. The dimensions of the spatial area that are covered by the point cloud increase quadratically with distance from the Kinect device. Hence, the resolution of the depth images generated by the Kinect device decreases with the distance [32]. These two limitations result in noisy depth measurements. This calls for feature computation methods that are able to efficiently deal with the noisy nature of depth images.

1.4. Improving Object Detection in Depth Images

Shotton et al. suggested that a larger computational budget may allow for the design of "potentially more powerful features based on, for example, depth integrals over regions, curvature, or more complex local descriptors" [29]. Alternatively, studies seeking to improve object detection in depth images [31]
can opt to use a larger computational budget to refine the input depth data itself, e.g., by including advanced depth image filter and refinement techniques [35, 36, 37, 38].

This paper proposes an improvement of Shotton et al.’s Pixel Comparison (PC) features by introducing advanced region-based descriptors, that do not require an increased computational budget: the Region Comparison (RC) features. Inspired by the work by Viola and Jones [8], Haar-like region features [7, 39] are combined with the integral image representation [39] to detect transitions in adjacent regions of depth images. The RC features provide an indication of the direction and the extent of depth transitions in an area of a depth image by averaging over regions, i.e., large groups of pixels. The additional computational cost to calculate the surfaces of the regions is negligible when integral images are employed [40, 41]. It is, however, unclear to what extent RC features enable fast and accurate body part detection in noisy depth images. To assess to what extent the RC features enable fast and effective body part detection in noisy depth images, we first define the region comparison detector which incorporates our RC features. Then, we compare its performance to a detector that deploys Shotton et al.’s PC features: the pixel comparison detector. In a comparative evaluation of the RC and PC features, both associated detectors are trained and evaluated on three challenging object detection experiments: two face detection tasks and a person detection task. There are two evaluation criteria. The first evaluation criterion is the classification performance, which is defined as the average per-class segmentation accuracy. The second evaluation criterion is computational efficiency, which is defined in terms of the time required to process an entire depth image. A shorter processing time therefore corresponds to a higher computational efficiency. It is assessed to ensure that improvements in accuracy do not lead to insurmountable computational costs that prohibit real-time operation. We consider the RC features superior to the PC features when the detector incorporating the RC features outperforms the detector featuring the PC features on both evaluation criteria.
1.5. Related Work

Our approach for improved detection accuracy in depth data deals effectively with background noise, without requiring additional computational power. It relates to several contributions in the fields of image refinement, computer vision and image understanding. In what follows, the related work is discussed.

First, several approaches aiming to counteract background noise in depth data include advanced depth image filter and refinement techniques \[35, 36, 37\]. Although image refinement is likely to improve the quality of the input depth data, it comes at the cost of computational power. This may influence the detection time negatively. An interesting approach was presented by Fanello et al. \[38\] in the form of their ‘filter forests’. Using location-dependent adaptive filters, their approach can be used to refine the quality of depth images. Such filters are computationally demanding and therefore not suitable for our goals. Inspired by their approach, our RC features incorporate a more straightforward - and computationally less demanding - way to filter noisy depth images.

Second, Nanni et al. \[42\] aim to detect human faces by applying the well-known Viola-Jones detector \[8\] to visual (RGB - Red Green Blue) images. Aligned depth images are then used to validate the detection results. Although their approach does not deploy depth data as its main data source, using the Viola-Jones detector in this context provides an interesting element. Inspired by Nanni et al. and Viola and Jones, our approach incorporates Haar-like features \([7, 8]\) to detect objects in depth images.

Third, the face-detection method proposed by Fanelli et al. \[40, 41\] operates on large, randomly selected patches in depth images (typically the size of a face), rather than on individual pixels (as seen in \[30\]). Their method includes a decision forest for the automatic labelling of the patches. The use of patches instead of individual pixels makes the method less prone to noise. Fanelli et al. suggest that using the integral image representation \[39\] of a depth image (rather than the depth image itself) may facilitate an efficient evaluation of the patches in the decision forest. Inspired by their suggestion, the RC features aim to describe individual pixel locations by computing depth comparison features.
over patches of various dimensions. The RC features can therefore be seen as a generalization of the patch-based method by Fanelli et al., which provide an indication of the direction and the magnitude of depth transitions in a depth image. The RC features include small and large patches of depth images through a decomposition of the integral depth image. This ensures an efficient feature computation process, which may therefore result in short prediction times.

Fourth, Buys et al. [43] incorporate the Pixel Comparison (PC) features that were proposed by Shotton et al. [30] in their sophisticated method to detect human bodies and to estimate their pose in single depth images. They label pixels using a random decision forest classifier (see Subsection 2.2). To deal with the noisy labels generated by their decision forest (which are partly due to the noisy nature of the individual pixels), Buys et al. perform a smoothing procedure on the pixel labels by means of a mode blur filter. In agreement with Buys et al., we acknowledge the importance of smoothening depth data to counteract the noise contained in depth images. In contrast to Buys et al.’s method for pixel comparison, the RC features do not require explicit smoothing. Instead, the RC features perform an implicit smoothing procedure by integrating over depth image regions of varying dimensions, rather than relying on individual pixels. Lacking the need for a post-hoc smoothing procedure is likely to contribute to the efficiency of our approach.

1.6. Outline

Section 2 describes our region comparison detector and its constituent components in more detail. Section 3 describes the experimental setup and the criteria used to evaluate and compare the performances of the detectors, while Section 4 describes the experiment and its results. Finally, the results and their implications are discussed in Section 5. The effectiveness of the RC features is concluded upon in Section 6.
2. The Region Comparison Detector

The region comparison detector detects human body parts in depth data. The detector classifies individual pixel locations in a point cloud (i.e., a subset of pixel locations) as either belonging to an object (e.g., a face) or to the background. The detector consists of two stages: (1) a preprocessing stage to compute our RC features for the individual pixel locations in the point cloud, and (2) a classification stage that uses a random decision forest classifier to predict the corresponding labels of the pixel locations. The labelled point cloud forms the final output of the classifier. In what follows, the RC features are reviewed in Subsection 2.1. Subsection 2.2 describes the random decision forest classifier that is used in the classification stage to classify the pixel locations. Figure 1 shows a diagram of the region comparison detector. The Figure shows (1) the preprocessing stage, and (2) the classification stage. These consecutive stages are represented by dark grey, rectangular areas. Their constituent sub-stages (A to D) are represented by white boxes. Below, the preprocessing stage and the classification stage are discussed briefly.
Preprocessing. In the preprocessing stage, the integral image representation of the input depth image is computed (sub-stage A). Then, a point cloud of pixel locations is selected at random from the input depth image (sub-stage B). The advantage of selecting a subset of random pixel locations from the input image (as proposed in [28]) is twofold: (1) it allows for the detection of partially oc-
cluded objects, and (2) it reduces the time required to process an entire depth image. After selecting the pixel locations, the region comparison detector computes multiple RC features for each pixel location in the subset by calculating the depth transitions over rectangular regions around the selected pixel location (sub-stage C). The features are then combined into a single RC feature vector, which provides a mathematical description for that particular pixel location.

The set of feature vectors (i.e., a single feature vector per pixel location in the point cloud) forms the input for the detector’s classification stage. Figure 2 shows an example of a visual image (Figure 2a) and the corresponding depth image (Figure 2b) of a person. Figures 2c and 2d show three examples of RC feature types yielding a response, i.e., a depth transition over regions. The spatial positions of the green/blue rectangles (in these Figures also indicated with the capital letters \(A\) and \(B\), respectively) represent the regions and direction over which a depth transition is measured, for straightforward (2c), and more complex depth transitions (2d). The RC features are discussed in-depth in Subsection 2.1.

Classification. In the classification stage, a random decision forest (RDF) [44] is used to classify the RC feature vectors that are computed for the pixel locations in the point cloud (sub-stage D in Figure 1). After classifying a feature vector, the RDF maps a class label (OBJECT, NO OBJECT) onto the corresponding pixel location in the point cloud. The labelled point cloud forms the final output of the classifier. Given the output of the classifier, groups of pixel locations with similar labels provide an indication of the presence and location of a person’s body parts in the image. As a case in point, Figures 4, 5, and 6 show six examples from our test set, in which the RDF of the region comparison detector classified individual pixel positions as belonging to a head (Figures 4 and 5) or person (Figure 6). In these examples, a green dot represents a pixel location that is correctly classified as belonging to a head or a person, while a red dot represents a pixel location that is (correctly) dismissed by the detector. The orange and blue dots represent the detector’s misclassifications. The examples
show that groups of pixel locations with the same labels reveal the location of a person’s head or body. In what follows, Subsection 2.2 provides a more detailed description of the random decision forest classifier.

2.1. Region Comparison Features

Region Comparison (RC) features are two dimensional filters (or masks) that translate transitions (e.g., depth contours or edges) in depth images into a numerical value, i.e., the feature value. The features are based on the well-known Haar wavelets [45]. The RC features provide an indication of the direction and magnitude of depth transitions in an area of a depth image by comparing the depth differences over regions, i.e., large groups of pixels, instead of pixel pairs (as seen in [30]). Varying the dimensions of the regions over which the RC features are computed, allows for the description of smaller or larger depth transitions. The advantage of comparing regions rather than individual pixel values is that it allows to average over larger areas. As a result, RC features are less prone to local pixel noise. Averaging over larger regions, however, results in a loss of spatial precision. By virtue of the Viola-Jones approach [8], which combines (1) Haar wavelets, and (2) integral images, the RC features combine the best of both worlds. On the one hand, the RC features include the averaging (summing) over large regions, which makes the features insensitive to local pixel noise. On the other hand, the features take individual pixel pairs, i.e., small regions, into account. To extract the RC features for a pixel location, the sum of the pixel values enclosed in a region around that pixel location is computed. The additional computational cost to calculate the surfaces of the regions, i.e., the sum of the pixel values, is negligible when integral images are employed [40, 41]. Thus, the RC features are computed using the integral depth image rather than the depth image itself. In what follows, the RC features are described in detail.

Definition. RC features are defined in terms of symmetrically located square regions in an image. The feature type depends on (1) the parameter $r$ defining...
the size of the individual square areas, and (2) the spatial configuration \(i\) defining the orientation of the constituent square regions. An RC feature value for pixel location \(P(x, y)\) in a depth image can be computed by calculating the sums of the pixels enclosed by two or more square areas and subtracting these sums from each other [8]. This results in a single feature value that provides an indication of the direction and magnitude of the depth transition over an area. Formally, the RC feature value of type \(i\) at location \(P\) in depth image \(I\), \(f_i(P, I)\), is defined as follows:

\[
f_i(P, I) = \sum_{n=1}^{d(i)} S(A_n(i), r) - \sum_{n=1}^{d(i)} S(B_n(i), r),
\]

where \(r\) is the linear size of the square region, \(A_n(i)\) and \(B_n(i)\), are the square regions of feature type \(i\), \(d(i)\) is the number of square regions for feature type \(i\), and \(S(X, r)\) represents the sum of the pixels enclosed by square region \(X\) of size \(r\). The feature value is computed by subtracting these sums for the regions \(A\) and \(B\). The square image regions, containing \(r^2\) pixels each, define the regions over which the depth difference is calculated. The value of \(r\) determines the spatial scale of analysis. For a small value of \(r\), the associated feature encodes depth transitions at a small scale, while large values of \(r\) allow the associated feature to encode for depth transitions at a large scale.

**Feature types.** The number of square regions and their relative spatial positions in relation to each other are predefined in terms of 11 different feature types. The feature types are based on the well-know square Haar features as proposed by Papageorgiou and Poggio [46]. Following up on their work, Viola and Jones [8] extended the Haar features by including arbitrary rectangular shapes. In addition, they distinguished two, three and four rectangle configurations. We adopt (1) the square regions of [46], and (2) two and four rectangle configurations [8]. In line with the Haar features we imposed the constraint that the squares are adjacent in the horizontal, vertical, diagonal, or anti-diagonal direction. For two-square configurations, this gives rise to four different feature types (Figure 3, a - d). For four square configurations, we obtain seven feature types [3] e -
Figure 3: An enumeration of the 11 RC feature types that are defined for our region comparison detector. The 4 two-rectangle feature types (a - d) allow for the computation of (a) horizontal, (b) vertical, (c) diagonal, and (d) anti-diagonal depth transitions. Combining several two-rectangle feature types results in 7 four-rectangle feature types (e - k), which are able to encode more complex depth transitions.

k). Please note that the individual feature types incorporate both variants of the A (green) and B (blue) regions: the original configuration and the one in which the A and B regions are swapped.

Given a feature type $i$, the spatial dimensions of the area over which the feature value is computed are defined by (1) the number of rectangular regions $d(i)$, and (2) the dimensions $r$ of the individual regions. If a feature type consists of a limited number of small rectangles, it typically encodes for very local depth transitions in a depth image. Similarly, feature types that are created using (a larger number of) large regions allow for the computation of features over larger areas, i.e., global depth transitions.

Figure 3 shows an overview of the 11 feature types that are used in our experiments, i.e., Figure 3 (a-d) shows the 4 two-rectangle feature types, while
Figure 3 (e-k) shows the 7 four-rectangle feature types of the detector. In this Figure, the green rectangles represent the rectangular areas $A_n(i)$ and the blue rectangles represent the rectangular areas $B_n(i)$ as defined in eq. 1. Both are used for the computation of the RC features. In the Figure, the red dot represents pixel location $P(x, y)$. Moreover, Figure 3a shows an example of a two-rectangle feature type (represented as green rectangle and a blue rectangle; $d(i) = 1$), while Figure 3c shows an example of a four-rectangle feature type (represented by two green rectangles $A$ and two blue rectangles $B$; $d(i) = 2$).

Feature vector. Given a pixel location $P(x, y)$ in a depth image, the features for the point are calculated in multiple iterations. In each iteration, the features are computed using the feature types as defined in Figure 3. This results in a series of feature values: one feature value for each feature type employed in the iteration. The feature types incorporate square regions that enclose $r^2$ pixels per rectangle. With each new iteration, the dimensions of the rectangles are increased: $r = \{1, 2, ..., r_{max}\}$. The feature vectors created after each iteration are then concatenated in the final RC feature vector. Calculating the sum of the rectangular areas for all possible square sizes up to $r_{max}$ can be done efficiently using the integral image representation. Because features are calculated over a relatively large spatial region (containing $r^2$ pixels), the feature vector extracted from location $P$ provides (1) an indication of the orientation, and (2) the extent of depth differences in the near vicinity, as well as at a larger spatial distance from location $P$.

2.2. The Random Decision Forest

To perform the actual classification, the standard random decision forest (RDF) classifier [44] is trained and used to classify individual pixel locations in depth images. The inputs of the RDF classifier are high-dimensional RC feature vectors, which are mapped onto pixel specific class labels. In what follows, the classification algorithm is described briefly.

RDF classifiers are fast and effective multi-class classifiers [47] that typically use an ensemble ("forest") of slightly different decision trees. RDF classifiers are
suitable for various supervised machine learning tasks, such as object classification \cite{48}. Each individual tree in an RDF classifier consists of multiple binary split nodes and leaf nodes. Individual split nodes compare single features from the feature vector with a threshold, branching left or right depending on the outcome of the comparison. The leaf nodes of the trees contain the prediction results. In a forest, the predictions of all constituent decision trees are averaged to obtain the final classification.

To grow the trees of an RDF classifier, each individual split node in a tree selects a random subset of features from the collection of candidate features from the training set. The number of features to select at random is (by default) the square root of the number of candidate features per pixel location. The best splitting candidate, i.e., the feature that best separates the subset of training examples, is selected as the split node’s threshold. A tree can be grown until each leaf node contains a limited number of observations, hence pruning the trees is not necessary.

3. Evaluation Procedure

This Section describes the experiments performed to evaluate the performance of the Region Comparison (RC) features. In a comparative evaluation, we compare the performances of our region comparison detector with a variant based on Shotton et al.’s Pixel Comparison (PC) features \cite{30}. The aim of our experiments is to investigate to what extent our RC features enable fast and effective face or person detection in noisy depth images, as compared to PC features. To perform our evaluation, the region comparison detector and the same detector incorporating PC features, the so-called \textit{pixel comparison detector}, are trained and evaluated on three challenging datasets. We emphasize that the training and evaluation procedure is performed using subsets of random pixel location that are extracted from the input depth image. Thus, we train and evaluate both detectors using feature vectors that are computed for individual pixel locations, rather than an entire depth image. While the number of depth images is rather
limited, this still allows us to generate sufficient training and test samples. As stated in Subsection 1.4, the evaluation investigates (1) the classification performance, and (2) the computational efficiency (i.e., the prediction speed) of the detectors. In what follows, Subsection 3.1 describes the datasets that are used in the experiments. Subsequently, we give the implementation details of both detectors in Subsection 3.2. Finally, we describe the experiments performed and the criteria employed to evaluate their performances in Subsection 3.3 and Subsection 3.4 respectively.

3.1. Datasets

To assess to what extent the RC features are able to deal effectively with background noise in depth images, the region comparison detector and the pixel comparison detector are trained and evaluated on the following three (publicly available) databases with depth images.

1. Biwi Kinect Head Pose Database by Fanelli et al. [41]
2. RGB-D Face Database by Høg et al. [49]
3. RGB-D People Dataset by Spinello & Arras [33]

These databases vary in (1) the amount of background noise, and (2) the objects captured in the depth data, i.e., human faces or entire humans. Below, these datasets are reviewed briefly.

**Biwi Kinect Head Pose Database** is developed by Fanelli et al. [41]. The dataset contains over 15,000 visual (RGB) and depth (D) images of people with various head poses sitting in front of a Kinect device. It provides annotations in the form of masks that indicate the location of a person’s face in a depth image. The masks use logical flags to indicate whether a pixel location belongs to a face or not. All depth images in this database have an image resolution of 640 × 480 pixels. The background of the depth data is removed using a threshold on the distance. The depth values are rescaled to an interval with values ranging from 0 to 4,095 (both inclusive). Removing the background is likely to result in a
Figure 4: Two examples of the classification results that are achieved by the region comparison detector on test images from the first head detection task, using an RDF classifier of 10 trees. In these examples, a green dot indicates a true positive prediction for pixel location \( P(x, y) \), while red indicates a true negative prediction. Orange represents false negative predictions, while blue represents the false positive ones.

RGB-D FACE DATABASE is developed by Høg et al. [49]. The dataset contains 1,581 visual (RGB) and depth (D) images of the heads and shoulders of human participants in various poses and with different facial expressions. As no annotations were provided for this database, each depth image in the subset was manually annotated\(^1\) by selecting a rectangular area that encloses the person’s face in the depth image. The boundaries of the annotation area were aligned with the left, top and right side of the face, and the lowest point of the person’s lower jaw. Similar to the Biwi Kinect Head Pose Database, the image resolution of the depth images is 640 × 480 pixels. The depth values of the depth images range from 0 to 4,095 (both inclusive). The background of the depth data is left intact. Figure 5 shows two examples of depth images from this database.

\(^1\)The annotations for this database are available upon request from the author.
Figure 5: Two examples of the classification results that are achieved by the region comparison detector on test images from the second head detection task, using an RDF classifier of 10 trees. In these examples, a green dot indicates a true positive prediction for pixel location \( P(x, y) \), while red indicates a true negative prediction. Orange represents false negative predictions, while blue represents the false positive ones.

**RGB-D PEOPLE DATASET** is developed by Spinello & Arras [33]. The dataset contains over 3,000 visual (RGB) and depth (D) images of mostly upright walking and standing people in a populated indoor environment, seen from different orientations and with different degrees of occlusions. This dataset is acquired in a university hall using three vertically mounted Kinect devices. In total, the dataset contains 1,133 annotated depth images. As the Kinect devices used in this experiment were mounted vertically, the depth images in the dataset are rotated 90 angular degrees. Hence, the image resolution of the depth images is 480 × 640 pixels. The maximal between the Kinect device and the hand of the subject is 1.0 meter. The annotations consist of rectangular bounding boxes enclosing a person’s body. Similar to the **RGB-D Face Database**, the background of the depth data is left intact. The depth values, however, are rescaled to an interval with values ranging from 0 to 4,095 (both inclusive). Figure 6 shows two examples of depth images from the **RGB-D People Dataset**.
Figure 6: Two examples of the classification results that are achieved by the region comparison
detector on test images from the person detection task, using an RDF classifier of 10 trees.
In these examples, a green dot indicates a true positive prediction for pixel location \( P(x, y) \),
while red indicates a true negative prediction. Orange represents false negative predictions,
while blue represents the false positive ones.

3.2. Implementation Details

The experiments are described in Subsection 3.3. Unless specified otherwise,
four types of parameters are used, viz. for (1) the selection of the random pixel
locations and spatial search area, (2) the RC and PC features, (3) the RDF
classifier, and (4) the implementation of the detectors.

The selection of the random pixel locations and spatial search area. For each
depth image, a subset of 2,000 random pixel locations is selected, for which the
PC and RC features are computed. To ensure a fair comparison, both types of
features operate on exactly the same pixel locations.

The maximal dimensions of the spatial search area over which the PC de-
tector computes its features are 150 \( \times \) 150 pixels. The PC detector normalises
the dimensions of the spatial search area based on the distance (depth value)
at point \( P(x, y) \). As a result, the search area is small for objects far from the Kinect device (high depth value at point \( P(x, y) \)) but large for objects close to the Kinect device (low depth value at point \( P(x, y) \)).

The spatial search area over which the RC features are computed is the same as the maximal (i.e., not normalised) search area used by the PC features. As such, the maximal dimensions of the rectangles incorporated by the RC features are \( 38 \times 38 \) pixels. As the feature types with the largest spatial dimensions deploy 4 rectangles (positioned horizontally, vertically, or (anti)-diagonally next to each other), the resulting search area is \((4 \times 38) \times (4 \times 38) \approx 150 \times 150\) pixels. Contrary to the PC features, the search area used for the RC features is not normalised for the distance.

**The RC and PC Features.** The rectangle size parameter \( r \) for the feature types that are used to compute the RC features, is defined as an integer value that increases with each iteration. In the first iteration, the value of \( r \) is initiated at 1. After each iteration, the value of \( r \) increases with step size 1, up to its maximum value of 38. Hence, the value of \( r \) over the iterations is defined as: \( r = \{1, 2, 3, \ldots, 38\} \). The resulting RC feature vectors may at most contain \( 11 \times 38 = 418 \) unique elements for each pixel location.

The parameters employed to compute the PC features are as specified in [30]. The resulting PC feature vectors contain 2,000 unique elements for each pixel location.

**RDF classifier.** For the experiments, the MATLAB implementation of the random decision forest (the so-called “TreeBagger”\(^2\)) is used. For the RC features, each split node of the forest selects a random subset of \( \sqrt{418} \approx 20 \) candidate features. For the PC features, each split node of the forest tests \( \sqrt{2,000} \approx 44 \) candidate features to find the best splitting threshold. Each tree of the random decision forest is trained until a minimum number of one observation per tree leaf is reached. The trees are not pruned.

\(^2\)http://nl.mathworks.com/help/stats/treebagger.html
The implementation of the detectors. Both detectors are implemented in MATLAB scripts. The implementations of the detectors are available upon request from the author. The entire training and evaluation procedure takes several days on a 50-core Linux calculation server.

3.3. Experiments

To evaluate the performance obtained by the RC features (as compared to the PC features), we perform three classification experiments. In the experiments, we train and evaluate the performance of both feature types (RC and PC) using publicly available databases. In the first experiment, the Biwi Kinect Head Pose Database \[41\] is used to detect human faces in smoothened depth images. Subsequently, the RGB-D Face Database \[49\] is used to detect human faces in non-smoothened depth images. In the third and last experiment, the RGB-D People Dataset \[33\] is used to detect entire people in non-smoothened depth images. In what follows, the experimental setup (A) and the individual experiments (B, C, and D) are described briefly.

A: Experimental setup. While the Biwi Kinect Head Pose Database and the RGB-D Face Database both contain depth images of annotated human faces, the depth images in the first database are likely to contain less background noise than the depth images in the latter database. This is due to the removal of the background of the depth images in the Biwi Kinect Head Pose Database. Thus, training and evaluating the detectors on these databases in the first two experiments provides an indication of the extent to which the RC features are able to deal effectively with background noise. Compared to the face detection tasks, detecting an entire human is likely to be a more challenging task. To investigate whether our results also extend to more complex detection tasks, we therefore compare the performance of both detectors in the third experiment (the person detection task).

As the feature extraction procedure is computationally demanding (especially during the training procedure of the pixel comparison detector, due to
the large feature vectors required for this detector), our experiments are performed using subsets of randomly selected depth images. The resulting subsets are considerably smaller than the original datasets. For all experiments, the generalisation performance is estimated using a 10-fold cross-validation procedure. The datasets used in our experiments are partitioned into separate training sets and test sets. In our experiments, the complexity of the RDF classifiers is not optimised prior to the experiment. Hence, we did not create a validation set. Moreover, the PC features are not optimised beyond the parameters provided by Shotton et al. [28, 29, 30]. The RC features are optimised to match the parameters of the PC features, which ensures a fair comparison between both feature computation approaches.

**B: Experiment 1.** Experiment 1 deals with face detection in smoothened depth images. For this experiment, a random subset of 100 depth images is selected from the *Biwi Kinect Head Pose Database*. Using the 10-fold cross-validation procedure, individual folds are created that each consist of 90 training images and 10 test images. Inspired through the work by Shotton et al. [28], a subset of 2,000 random pixel locations is selected from each depth image, and labelled in a binary fashion, i.e., *face* when a pixel is located within the region annotated as belonging to the face, or *other* in all other cases. For each fold, the resulting dataset of this experiment consists of 180,000 training examples (pixels) and 20,000 test examples. For each individual training example, the RC and PC vectors are computed. Combined with the labels, the training examples are used to train RDF classifiers, with forests ranging from 1 tree up to 10 trees (both inclusive). The test examples and the corresponding labels are used to assess the generalisation performance of the detectors.

**C: Experiment 2.** Experiment 2 deals with face detection in non-smoothened depth images. The underlying idea of this experiment is as follows. We expect that RC features are suitable to deal with noise in depth data. Removing the background in depth images, however, also reduces the amount of noise in the depth data, which may influence the performance of the RC and PC features.
We therefore repeat in some sense experiment 1 while using the *RGB-D Face Database*. Contrary to the previous experiment, this database contains depth images from which the background is not removed.

For experiment 2, a random subset of 93 depth images is selected. After performing 10-fold cross-validation, the resulting folds consist of 84 training images and 9 test images. Similar to the previous experiment, a subset of 2,000 random pixel locations is selected from each depth image, and labelled accordingly. The resulting dataset contains 168,000 training examples and 18,000 test examples per fold. The training and evaluation procedure for this experiment is the same as in experiment 1.

**D: Experiment 3.** Experiment 3 deals with person detection in non-smoothened depth images. The underlying idea of this experiment is as follows. Whereas detecting human faces in depth images in general might be relatively easy due to the fairly consistent shape of the human face, detecting entire humans is likely to be a more challenging task.

For experiment 3, a subset of 100 random depth images is selected from the *RGB-D People Dataset*. The background of the depth images in this dataset is left intact. Similar to the previous experiments, the resulting subset is divided into separate training and test sets using the 10-fold cross-validation procedure. We again select 2,000 random pixel locations from each depth image. Each individual fold therefore consists of 180,000 training examples and 20,000 test examples. Each example is labelled as either PERSON or OTHER. The training and evaluation procedure of this experiment is the same as in the previous experiments.

### 3.4. Performance Metrics

The performance of the detectors was quantified using two performance metrics: (1) classification performance metrics to report on the average per-class segmentation accuracy, and (2) computational efficiency metrics to measure the time required by a classifier to process an entire image. In our evaluation, RC features
are considered to outperform PC features when they achieve a higher average classification performance, without incurring an additional cost in terms of detection speed as compared to the PC features. In what follows, the performance metrics are defined.

**Classification Performance Metrics.** The detectors that are trained and evaluated in our experiments classify individual pixel locations as either belonging to an object (e.g., a face), or to the background. Given the binary nature of the experiments, we use the (1) precision, (2) recall, (3) balanced accuracy, (4) F1-score, and (5) receiver operating characteristic curve (AUC) as classification performance metrics to measure the classification performance of a detector. Moreover, we measure (6) the average number of levels, and (7) the average number of leaf nodes of the trees to estimate the complexity of the random forest classifier (see Subsection 2.2). The precision indicates the percentage of recognized instances that are relevant, while recall provides an indication of the percentage of relevant instances that are identified by the detector. As the class distributions of the datasets are highly skewed, i.e., a low percentage of FACE- or PERSON-samples versus a high percentage of OTHER-samples, the performances may be biased towards the most frequent class in a dataset. To deal with this bias, the balanced accuracy (as proposed in [50]) is adopted. The balanced accuracy is defined as the arithmetic mean of class-specific accuracies, which considers the recall of the positive and negative class [51]. In addition, as additional performance metrics that are able to handle unbalanced datasets, the F1-score and the area under the receiver operating characteristic curve (AUC) are computed. The F1-score can be interpreted as the weighted average of the precision and recall values of the positive class [52]. The AUC score is taken from, for example, [53]. Moreover, the average number of levels per tree provides an indication of the (average) number of tests that are performed on each feature vector. Moreover, the combination of (a) the average number of levels, and (b) the average number of leaf nodes provides an indication of the efficiency of the tree. An efficient tree is typically characterised by a low number of levels,
yet a relatively high number of leaf nodes.

Computational Efficiency Metrics. The computational efficiency metrics measure the time required by a detector to process individual depth images. A shorter prediction time corresponds to a higher computational efficiency. Thus, the prediction times provide indication of the computational efficiency of a detector.

4. Experimental Results

In this section, we describe the results of the experiments performed to evaluate the RC features. Subsection 4.1 describes the results of the first face detection task, which uses a dataset with smoothened depth images, i.e., depth data from which the background is removed using a threshold. Subsequently, Subsection 4.2 describes the results of the second face detection task, which incorporates a dataset with non-smoothened depth images. Subsection 4.3 describes the results of the person detection task using non-smoothened depth images.

4.1. Results Experiment 1 (Face Detection, Smoothened Background)

The results of the first face detection task are shown in Figures 7 and 8, and Tables 1, 2 and 3. The dataset used in the experiment contains smoothened depth images, i.e., images from which the background is removed. In what follows, the results will be discussed in more detail.

Figure 7 shows the performance for both feature computation methods (RC and PC) for ten sizes of the RDF classifier: (a) the balanced accuracy, (b) precision and recall, (c) F1-scores, and (d) the AUC scores. Figure 8 shows (a) the average tree depth, and (b) the average number of leaf nodes per tree in the classifiers, which provides an indication of the complexity of the classifiers. Table 1 shows the minimum and maximum performances for the region comparison detector, while Table 2 shows this information for the pixel comparison detector. Table 3 shows the AUC values and the associated prediction times (i.e., the
Figure 7: (Experiment 1: face detection) The classification performance for the first face detection task for ten sizes of the random decision forests: (a) balanced accuracy, (b) precision and recall, (c) F1-scores, and (d) the AUC scores (higher is better). The continuous line represents the performance obtained with the RC features, while the dotted line represents the performance obtained with the PC features. The x-axes of the graphs represent the number of trees in the RDF. The y-axes represent the classification performance. More details and interpretations are provided in the text.

computation times required to process an entire depth image), for ten sizes of the forest.

Figure 4 shows two depth images from our test set, in which the region comparison detector recognized the location of a persons’ head by classifying the pixel locations in the point cloud. In these examples, a green dot indicates a true positive prediction for a given pixel \( p \), while red indicates a true
Table 1: [Experiment 1 - RC features] The minimum and maximum scores for the balanced accuracy, recall, precision, and F1-scores obtained by the RC features in experiment 1.

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Min. score (SD)</th>
<th>Max. score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced accuracy (%)</td>
<td>79.4 (1.7)</td>
<td>88.8 (1.5)</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>59.7 (3.5)</td>
<td>79.1 (3.1)</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>52.6 (7.3)</td>
<td>74.0 (7.4)</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.59 (0.04)</td>
<td>0.70 (0.03)</td>
</tr>
</tbody>
</table>

Table 2: [Experiment 1 - PC features] The minimum and maximum scores for the balanced accuracy, recall, precision, and F1-scores obtained by the PC features in experiment 1.

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Min. score (SD)</th>
<th>Max. score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced accuracy (%)</td>
<td>51.2 (0.4)</td>
<td>63.5 (1.1)</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>2.63 (0.9)</td>
<td>31.8 (2.1)</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>12.9 (1.8)</td>
<td>19.7 (7.3)</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.05 (0.01)</td>
<td>0.18 (0.02)</td>
</tr>
</tbody>
</table>

negative prediction. Orange represents false negative predictions, while blue represents the false positive ones. The black background of the images is the (visualized) result of the background removal (smoothening). Figure 13a shows the AUC curve for the optimal detection parameters, i.e., an RDF classifier of 610 trees.

The results of this experiment show that the region comparison detector achieves a significantly higher classification performance than the pixel comparison detector. The results also reveal that both detectors approach their optimal classification performance using a random decision forest of rather small dimensions, i.e., a forest consisting of only a limited number of trees (say, three to five). Training additional trees does not affect the balanced accuracy (see Figure 7a) of the region comparison detector significantly, although it decreases slightly
Figure 8: [Experiment 1: face detection] The bar plot of (a) the average tree depth, and (b) the average number of leaf nodes per tree for the pixel comparison detector (represented in gray) and the region comparison detector (represented in blue), and the corresponding error bars. The results are averaged over all trees and all folds.

for the pixel comparison detector. When increasing the number of trees in the forest, recall and precision (Figure 7b), and the F1-score (Figure 7c) increase slightly for the region comparison detector, while the precision of the pixel comparison shows a slight increase, and even a decrease in recall and F1-score. For both detectors, the Area Under the Curve (AUC) score increases with the size of the forest, approaching its optimal score using a forest of three trees and five trees for the region and pixel comparison detector, respectively (Figure 7d).

Our results show that the trees that are grown for the region comparison detector are significantly smaller than the ones that are grown for the pixel comparison detector. Averaged over all folds and dimensions in the experiment, the trees of the region comparison detector are 13.0 levels deep (SD = 0.21). The trees that are grown for the pixel comparison detector, however, reach an average depth of 17.6 levels (SD = 0.41). Moreover, our results show that the trees of the region comparison detector contain an average of 1,001 leaf nodes (SD = 23), while the trees of the pixel comparison detector contain an average of 2,015 leaf nodes (SD = 23). These results imply that random forests that are trained by using the RC feature vectors, require, on average, a lower number of
Table 3: [Experiment 1: face detection] The AUC scores and prediction times per image for the both detectors while using RDF classifiers of ten sizes in experiment 1.

<table>
<thead>
<tr>
<th>Forest size</th>
<th>RC features</th>
<th></th>
<th>PC features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Pred. time (s) (SD)</td>
<td>AUC</td>
<td>Pred. time (s) (SD)</td>
</tr>
<tr>
<td>1</td>
<td>0.794</td>
<td>1.08 (0.16)</td>
<td>0.578</td>
<td>0.01 (0.00)</td>
</tr>
<tr>
<td>2</td>
<td>0.889</td>
<td>1.38 (0.01)</td>
<td>0.635</td>
<td>0.01 (0.00)</td>
</tr>
<tr>
<td>3</td>
<td>0.924</td>
<td>1.74 (0.01)</td>
<td>0.674</td>
<td>0.01 (0.00)</td>
</tr>
<tr>
<td>4</td>
<td>0.947</td>
<td>2.09 (0.02)</td>
<td>0.722</td>
<td>0.01 (0.00)</td>
</tr>
<tr>
<td>5</td>
<td>0.959</td>
<td>2.45 (0.01)</td>
<td>0.761</td>
<td>0.02 (0.00)</td>
</tr>
<tr>
<td>6</td>
<td>0.969</td>
<td>2.83 (0.02)</td>
<td>0.784</td>
<td>0.02 (0.00)</td>
</tr>
<tr>
<td>7</td>
<td>0.975</td>
<td>3.17 (0.03)</td>
<td>0.801</td>
<td>0.02 (0.00)</td>
</tr>
<tr>
<td>8</td>
<td>0.980</td>
<td>3.52 (0.02)</td>
<td>0.824</td>
<td>0.02 (0.00)</td>
</tr>
<tr>
<td>9</td>
<td>0.982</td>
<td>3.86 (0.02)</td>
<td>0.842</td>
<td>0.02 (0.00)</td>
</tr>
<tr>
<td>10</td>
<td>0.980</td>
<td>4.24 (0.01)</td>
<td>0.846</td>
<td>0.02 (0.00)</td>
</tr>
</tbody>
</table>

Although the RC features seem to enable a significantly higher classification performance than the PC features, the average prediction times indicate that the region comparison detector requires more time to process a depth image than the pixel comparison detector. This may (partially) be due to the time required to (1) compute the integral image representation, and (2) extract the RC features. While the results of this experiment show that RC features enable a higher classification performance in smoothened depth data, their classification...
performance comes at the cost of computational efficiency.

4.2. Results Experiment 2 (Face Detection, Non-Smoothened Background)

We now investigate to what extent the pattern of results of experiment 1 are related to non-smoothened depth data. As noisy depth data is likely to result in erroneous depth measurements in the feature vectors, accurately separating the feature vectors may become a challenge. Classifiers that aim to separate feature vectors with erroneous feature values require additional tests to achieve an optimal separation of the data. This may lead to an increase in the number of split nodes and leaf nodes, i.e., an increase in the complexity of the classifiers. Increasing the number of tests that are performed on the input feature vectors may influence a detector’s prediction time negatively. This is assessed in our second face detection experiment, in which both detectors are trained and evaluated on a dataset with non-smoothened depth images.

For the second face detection task, the average classification performances are shown in Figures 9 and 10, and Tables 4, 5 and 6. The dataset used in the experiment contains depth images from which the background is left intact. In what follows, the results will be discussed in more detail.

Figure 9 shows (a) the balanced accuracy, (b) precision and recall, (c) F1-scores, and (d) the AUC scores for both feature computation methods. Figure 10 provides an indication of the complexity of the classifiers by showing the average tree depth and the average number of leaf nodes per tree in the classifiers. Tables 4 and 5 show the minimum and maximum performances for the RC features and PC feature, respectively. Table 6 shows the AUC values and the associated prediction times for ten sizes of the RDF classifier.

Figure 5 shows two examples of depth images from our test, in which the region comparison detector (featuring an RDF classifier of 10 trees) classified individual pixel locations. We remark the presence of the “depth shadow”, i.e., empty parts in the depth image, on the right side of the person. It is a direct result of a part of a scene that is not illuminated by the laser of the Kinect device, and therefore not captured by its infrared sensor. Consequently, it results in
Figure 9: [Experiment 2: face detection] The classification performance for the second face detection task for ten sizes of the random decision forests: (a) balanced accuracy, (b) precision and recall, (c) F1-scores, and (d) the AUC scores (higher is better). The continuous line represents the performance obtained with the RC features, while the dotted line represents the performance obtained with the PC features. The x-axes of the graphs represent the number of trees in the RDF classifier. The y-axes represent the classification performance. More details and interpretations are provided in the text.

The results of the experiment (Figure 9 and Table 4) show that the RC features again achieve a significantly higher classification performance than the PC features (Table 5), even though the number of training samples is slightly...
Table 4: [Experiment 2 - RC features] The minimum and maximum scores for the balanced accuracy, recall, precision, and F1-scores obtained by the RC features in experiment 2.

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Min. score (SD)</th>
<th>Max. score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced accuracy (%)</td>
<td>91.3 (2.0)</td>
<td>95.5 (1.8)</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>83.4 (3.9)</td>
<td>92.4 (3.5)</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>76.9 (4.7)</td>
<td>92.0 (3.9)</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.83 (0.03)</td>
<td>0.91 (0.03)</td>
</tr>
</tbody>
</table>

Table 5: [Experiment 2 - PC features] The minimum and maximum scores for the balanced accuracy, recall, precision, and F1-scores obtained by the PC features in experiment 2.

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Min. score (SD)</th>
<th>Max. score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced accuracy (%)</td>
<td>51.7 (0.4)</td>
<td>63.1 (1.3)</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>3.81 (0.9)</td>
<td>33.9 (2.4)</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>18.8 (1.6)</td>
<td>34.6 (6.0)</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.07 (0.02)</td>
<td>0.24 (0.01)</td>
</tr>
</tbody>
</table>

smaller than in the first detection task. Similar to the results of the previous experiment, Figure [9] shows that both types of features approach their optimal classification performance using a random decision forest of rather small dimensions (again, say three to five). Training additional trees does not affect the accuracy (Figure 9a) of the RC features significantly, although it decreases slightly for the PC features. When increasing the size of the forest, recall, precision (Figure 9b), and the F1-score (Figure 9c) increase slightly for the RC features. The difference in performances of both feature types is also reflected in their AUC scores (Figure 9d).

Compared to the previous experiment (experiment 1), the depth data used in this experiment contains a higher amount of background noise. Following the increase in the amount of background noise, our results (see Figures 7d and 9d)
Figure 10: [Experiment 2: face detection] The bar plot of (a) the average tree depth, and (b) the average number of leaf nodes per tree for the pixel comparison detector (represented in gray) and the region comparison detector (represented in blue), and the corresponding error bars. The results are averaged over all trees and all folds.

show that the classification performance of the PC features decreases slightly. The performance of the RC features, however, shows a minor increase.

The average prediction times (as shown in Table 6) indicate that, for this experiment, the region comparison detector requires less time to process an entire depth image than the pixel comparison detector. Contrary to the results of experiment 1, the region comparison detector now achieves a detection speed that is up to 2.5 times faster than the detection speed of the pixel comparison detector.

The difference in detection speed may partially be due to an increase in the complexity of the RDF classifier of the pixel comparison detector. While the classifier reaches an average depth of 17.6 levels (SD = 0.41) in the previous experiment, the average tree depth of the classifier quadruples to 68.1 levels (SD = 2.36). This increase is also reflected in the number of leaf nodes of classifier, as they double from 2,015 (SD = 23) in the previous experiment to an average of 3880 leaf nodes (SD = 58). The trees of the region comparison detector that are grown in this experiment, however, reach an average depth of 13.1 levels (SD = 0.22) and 997 leaf nodes (SD = 31), which translates to a minimal increase.
Table 6: [Experiment 2: face detection] The AUC scores and prediction times per image for the both detectors while using RDF classifiers of ten sizes in experiment 2.

| Forest size | RC features | | | PC features | | |
|-------------|-------------|-------------|-------------|-------------|-------------|
|             | AUC | Pred. time (s) (SD) | | AUC | Pred. time (s) (SD) | | |
| 1           | 0.912 | 1.36 (0.23) | | 0.415 | 7.55 (0.22) | | |
| 2           | 0.958 | 1.72 (0.02) | | 0.507 | 8.17 (0.03) | | |
| 3           | 0.973 | 2.18 (0.03) | | 0.583 | 8.82 (0.07) | | |
| 4           | 0.979 | 2.63 (0.02) | | 0.641 | 9.46 (0.06) | | |
| 5           | 0.984 | 3.09 (0.02) | | 0.680 | 10.1 (0.08) | | |
| 6           | 0.987 | 3.60 (0.05) | | 0.719 | 10.7 (0.07) | | |
| 7           | 0.987 | 3.98 (0.04) | | 0.745 | 11.4 (0.08) | | |
| 8           | 0.989 | 4.51 (0.04) | | 0.761 | 12.1 (0.15) | | |
| 9           | 0.991 | 4.89 (0.09) | | 0.782 | 12.7 (0.12) | | |
| 10          | 0.991 | 5.44 (0.08) | | 0.784 | 13.3 (0.20) | | |

...in their average depth, or even a small decrease in number of leaf nodes. These results imply that increased levels of background noise in depth data result in a significant increase in the number of tests that are performed by the RDF classifier of the pixel comparison detector. However, the results also imply that the RDF of the region comparison detector does not require additional tests to perform its classification task. Figure 10 shows (1) the average tree depth, and (2) the average number of leaf nodes per tree for both detectors experiment 2.

The results of this experiment show a significant increase in the prediction time of the pixel comparison detector, especially when compared to a relatively small increase in the prediction times of the region comparison detector. The results show that the complexity of the RDF classifier employed by the pixel comparison detector increases significantly with the level of background noise in the depth data. It indicates that the pixel comparison detector is more sensitive...
4.3. Results Experiment 3 (Person Detection)

We now turn to the third experiment, to assess if this pattern of results generalizes to the more complex task of person detection in depth images. While detecting human faces in depth images might be relatively easy, detecting entire humans is likely to be a more challenging task. The experiment explores to what extent RC features may outperform PC features in more complex detection tasks. The average classification performances obtained with RC features and PC features on the person detection task are shown in Figure 11 and 12, and Tables 7, 8, and 9. The dataset used in the experiment contains depth images with high levels of background noise. In what follows, the results will be discussed in more detail.

Figure 11 shows the classification performance of both types of features for different sizes of the random decision forest: (a) the balanced accuracy, (b) precision and recall, (c) F1-scores, and (d) the AUC scores. Figure 12 shows (a) the average tree depth, and (b) the average number of leaf nodes per tree in the classifiers, which provides an indication of the complexity of the classifiers. Tables 7 and 8 show the minimum and maximum performances of the aforementioned performance metrics for the RC features and PC features, respectively. Table 9 shows the performances expressed as the AUC of the detectors, versus the time required to process an entire depth image, i.e., the prediction time.

Figure 6 shows two examples of depth images from our test set, and the corresponding prediction results of the region comparison detector (using an RDF classifier of 10 trees). We remark that the region comparison detector is capable of detecting people at close range and at larger distances from the Kinect device. Figure 13c shows the AUC curve for the optimal detection parameters, i.e., an RDF classifier of 10 trees.
Figure 11: [Experiment 3: person detection] The classification performance for the person detection task for various sizes of the random decision forests: (a) balanced accuracy, (b) precision and recall, (c) F1-scores, and (d) the AUC scores (higher is better). The continuous line represents the performance obtained with the RC features, while the dotted line represents the performance obtained with the PC features. The x-axes of the graphs represent the number of trees in the RDF classifier. The y-axes represent the classification performance.

The results (see Figure 11) of the experiment show that the RC features again achieve a significantly higher classification performance than the PC features (see Tables 7 and 8). Both (balanced) accuracy and recall obtained with the RC features are largely independent of the number of trees in the RDF classifiers. The precision of the detector, however, increases significantly with the dimensions of the forest. The region comparison detector achieves its optimal AUC using a forest of 4 trees.
Table 7: [Experiment 3 - RC features] The minimum and maximum scores for the balanced accuracy, recall, precision, and F1-scores obtained by the RC features in experiment 3.

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Min. score (SD)</th>
<th>Max. score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced accuracy (%)</td>
<td>73.6 (1.1)</td>
<td>83.0 (1.0)</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>48.7 (2.1)</td>
<td>68.0 (2.0)</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>63.3 (2.9)</td>
<td>88.3 (2.2)</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.62 (0.02)</td>
<td>0.76 (0.02)</td>
</tr>
</tbody>
</table>

Table 8: [Experiment 3 - PC features] The minimum and maximum scores for the balanced accuracy, recall, precision, and F1-scores obtained by the PC features in experiment 3.

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Min. score (SD)</th>
<th>Max. score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balanced accuracy (%)</td>
<td>52.5 (0.6)</td>
<td>56.0 (0.6)</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>7.21 (1.1)</td>
<td>27.0 (1.3)</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>23.0 (1.8)</td>
<td>38.9 (2.7)</td>
</tr>
<tr>
<td>F1-score</td>
<td>0.12 (0.02)</td>
<td>0.248 (0.01)</td>
</tr>
</tbody>
</table>

An analysis of the complexity of both detectors (see Figure 12) reveals that the RDF classifier of the region comparison detector reaches an average depth of 24.0 levels (SD = 0.39), versus an average depth of 40.4 levels (SD = 1.02) for the pixel comparison detector. The trees of the region comparison detector consist of, on average, 6,610 leaf nodes (SD = 94), and 11,600 leaf nodes (SD = 90) for the pixel comparison detector. Compared to the results of the previous experiment (experiment 2; see Figure 10), the results show an increase in the number of levels of the RDF of the region comparison detector, but a decrease in the number of levels of its pixel comparing opponent. However, the results also indicate that the average number of leaf nodes increases significantly for both detectors. This suggests an increase in the average number of tests per tree of both detector, which translates to an increase in complexity of both
Figure 12: [Experiment 3: person detection] The bar plot of (a) the average tree depth, and
(b) the average number of leaf nodes per tree for the pixel comparison detector (represented in
gray) and the region comparison detector (represented in blue), and the corresponding error
bars. The results are averaged over all trees and all folds.

detectors. Although the complexity increases for both detectors, the RDF of
the region comparison detector again achieves a lower complexity than the pixel
comparison detector.

The prediction times for the person detection task are listed in table 9. The
results are two-fold. On the one hand, the results show that the RC features
allow for a considerably shorter prediction time per image (which therefore re-
results in a higher prediction speed) than the PC features. Our results show that
the detection speed obtained with the region comparison detector is about two
to four times higher than the detection speed obtained with its pixel comparing
opponent. On the other hand, the results show that both detectors require more
time to classify the pixel locations than in the previous experiments.

The results of experiment 3 indicate that the person detection task is, indeed,
a harder task than the face detection tasks (experiments 1 and 2). For more
complex classification tasks, the RC features seem to benefit from an increase
in the number of trees. When detecting entire people in non-smoothened depth
images, RC features outperform PC features at a two to four-fold increase in
Table 9: [Experiment 3: person detection] The AUC scores and prediction times per image for the both detectors while using RDF classifiers of ten sizes in experiment 3.

<table>
<thead>
<tr>
<th>Forest size</th>
<th>region comparison detector</th>
<th>pixel comparison detector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Pred. time (s) (SD)</td>
</tr>
<tr>
<td>1</td>
<td>0.785</td>
<td>1.70 (0.22)</td>
</tr>
<tr>
<td>2</td>
<td>0.862</td>
<td>2.54 (0.03)</td>
</tr>
<tr>
<td>3</td>
<td>0.892</td>
<td>3.34 (0.04)</td>
</tr>
<tr>
<td>4</td>
<td>0.913</td>
<td>4.19 (0.05)</td>
</tr>
<tr>
<td>5</td>
<td>0.923</td>
<td>5.00 (0.09)</td>
</tr>
<tr>
<td>6</td>
<td>0.931</td>
<td>5.92 (0.06)</td>
</tr>
<tr>
<td>7</td>
<td>0.936</td>
<td>6.75 (0.17)</td>
</tr>
<tr>
<td>8</td>
<td>0.940</td>
<td>7.63 (0.07)</td>
</tr>
<tr>
<td>9</td>
<td>0.944</td>
<td>8.48 (0.12)</td>
</tr>
<tr>
<td>10</td>
<td>0.946</td>
<td>9.30 (0.17)</td>
</tr>
</tbody>
</table>

The results of the third experiment thus show that the superiority of RC features also holds for the task of person detection.

5. Discussion

In this paper, we proposed the Region Comparison (RC) features as an accurate and efficient alternative to Pixel Comparison (PC) features. Our approach achieves high detection accuracy without requiring additional computational budget. The results of the empirical evaluation show that RC features do indeed outperform PC features in both classification performance as well as computational efficiency. This holds for face detection as well as for person detection. Below, the implications of the results are discussed in more details. Subsection 5.1 discusses the relative superiority of the RC features over the PC features, while Subsection 5.2 addresses the number of samples that are used in our experiments. Subsequently, Subsection 5.3 discusses our future work and the steps...
Figure 13: The AUC (Area Under the Curve) graphs of the detectors when using their optimal detector parameters (i.e., the parameters resulting in the highest prediction performance; a forest of 10 trees). Figure 13a shows an example of the AUC from experiment 1 (face detection in smoothened depth images), while Figure 13b shows this for experiment 2 (face detection in non-smoothened depth images). Subsequently, Figure 13c shows an example of the AUC from experiment 3 (person detection in non-smoothened depth image).

to be taken before the region comparison detector can actually be employed for object detection tasks.

5.1. RC Features Combine the Best of Both Worlds

What explains the superiority of the RC features over PC features? As indicated in Subsection 1.4, handling the noise in depth images is an important challenge for object detection methods. Individual depth pixels may have in-
correct values due to limited sensor resolution or false reflections. Comparing individual, incorrect pixel values may therefore lead to measurement errors. To counter the far-reaching effects of incorrect pixel values requires averaging over larger regions in the depth image. However, averaging over pixel values results in a loss of spatial precision. Analogously to the Viola-Jones approach, which combines integral images and Haar wavelets, the RC features combine the best of both worlds. On the one hand, RC features include the averaging (summing) over large regions, which makes the features insensitive to local pixel noise, while on the other hand the RC features take individual pixel pairs (as the PC features) into account. We believe that the combination of global averaging and local precision explains the relative superiority of the RC features over the PC features. The computational efficiency is a direct result of our use of the highly efficient integral image representation.

In this paper, we incorporated the combination of RC features and the integral image in our region comparison detector. The detector computes feature vectors with RC features for each randomly selected pixel locations in a depth image. The resulting RC feature vectors contain 418 elements. On the other hand, the PC feature vectors (that are created using the PC features of Shotton et al., as described in [30]) contain 2,000 elements per pixel location. This implies that calculating features over the same spatial area in a depth image results in RC feature vectors that are approximately 80% smaller than the feature vectors created for the PC features. As a result, the number of calculations required to create the individual feature vectors is likely to be in favour of the RC features. The additional computational cost required to compute the surface of the areas for the RC features is negligible when integral images are employed [40] [41]. Moreover, shorter feature vectors contain fewer features that need to be tested by the random decision forest. This may therefore add to the computational efficiency of the RC features. One may argue that calculating the integral image representation of the depth image itself also requires computational power. The results of our evaluation, however, reveal that processing an entire depth image using the RC features (by first calculating the integral image
of a depth image and subsequently computing the RC feature vectors) takes less
time than the time required to create and classify the PC feature vectors. This
is partly due to the fact that the integral image representation is computed only
once for each depth image. The time required to calculate the integral image is
therefore likely to be compensated by the efficient feature computation process.
We therefore argue that computing the integral image and the RC feature vec-
tors can be achieved more efficiently than the PC feature vectors, which works
in our favour.

Our results showed that the RC features achieve performances superior to
PC features. How do RC features compare to other state-of-the-art methods?
A direct comparison is difficult, because the RC and PC detectors assign labels
to individual pixels, rather than to entire objects. Still, some indication may
be given by relating our detection results to those obtained by Buys et al.
As discussed in Subsection 1.5 (Related Work), Buys et al. [43] developed a
sophisticated method for human body pose detection by building on the PC
features. Their person-classification performances range from 80 – 90%. Given
our findings, we expect that the detection accuracy of Buys et al.’s method
would improve beyond 90% when the PC features would be replaced by RC
features.

5.2. The Number of Samples Required
The PC features of Shotton et al. [30] rely on the comparison of depth pixel
pairs. They require many examples to encode objects uniquely against the
background. As a case in point, in their experiments, Shotton et al. [28, 29, 30]
use datasets of a size that are 150 – 9000-times as large as the 100-image subsets
that are used in our experiments.

As each of our experiments took several days to complete on our powerful
50-core calculation servers, our decision to use subsets of the original databases
was motivated by computational considerations. The experiments reported in
[28], for example, relied on 1000-core servers, which are not available to us.

Using subsets of the databases, however, may give rise to two challenges.
On the one hand, it may be the case that the performance obtained with PC features benefits from an increase in the number of training examples. On the other hand, using a small subset may result in overfitting on the data. To investigate these challenges, we performed an additional exploratory experiment using a larger subset of depth images of the Biwi Kinect Head Pose Database. In this experiment, we trained and evaluated both detectors on a subset of 1,250 depth images, using the same conditions as described in Section 3. The results of this experiment showed the same results as reported in Subsection 4.1.

While increasing the size of the datasets (and thereby the number of training examples) may increase the classification performance of PC features, we expect that this will increase the classification performance of the RC features as well. Due to the results of the up-scaled experiments (in combination with the established cross validation procedure), we feel confident that our results provide reliable estimates of the classification performance on large-scale datasets.

5.3. Future Work

The detection tasks performed in our evaluation procedure were limited to the automatic labelling of individual pixel locations as belonging to either an object (face/body) or to the background. Calculating features from pixel locations and classifying them accordingly are two important steps towards actual object detection. However, actual object detection requires an additional processing step which integrates the individual pixel labels into a higher-level detection of the object, i.e., the labelling of a larger region encompassing the face or body. We refrained from developing such a higher-level detection stage, because the focus of our study was on the evaluation of the RC features. Future work may therefore extend the region comparison detector with this stage. It is to be expected that the superiority of RC features (as compared to PC features) will be reflected in any higher-level detection method that takes the labels generated by the region comparison detector as input.

For our evaluation procedure, the detectors incorporating the RC and PC features were implemented as MATLAB scripts. While MATLAB allows for
rapid prototype development, it is not optimized for speed. Implementing the RC features in a dedicated programming language such as C++ may speed up their processing time. We expect that porting the RC features to a C++ implementation is feasible and that the RC features are therefore likely to run on reasonable hardware.

While this paper focuses on the comparison between PC and RC features, future studies may compare RC features with other depth-based features. A promising competing approach is the histogram-based ComboHOD method, that was developed by Spinello & Arras [33]. Future work may therefore include a comparison between our region comparison detector and other depth-based feature extraction methods, such as the ComboHOD method. After extending our detector with the necessary post-classification steps, we aim to compare our detector (and therefore the RC features) with the promising ComboHOD method.

6. Conclusions

This paper presents and evaluates the Region Comparison (RC) features. Combining the RC features with the integral image representation of depth images allows for fast and effective object detection in depth data. A comparative evaluation investigated in three different object detection tasks to what extent RC features contribute to fast and effective object detection in noisy depth images.

The results of the evaluation showed that RC features outperform the state-of-the-art PC features in classification performance and prediction speed, especially in noisy depth images. The RC features deal effectively with the noise in depth images. They maintain precision in the depth images by sampling depth transitions on scales varying from small to large image regions. Based on our results, we provisionally conclude (1) that RC features contribute significantly to fast and effective face and person detection in noisy depth images, (2) that RC features yield an improvement over PC features, and (3) that the RC features are able to operate with the same computational budget.
Employing RC features might be able to increase the classification performance of detectors based on the work by Shotton et al., such as the body part detector of Buys et al.

7. Acknowledgements

We gratefully thank dr. Jamie Shotton for his advice regarding our implementation of their PC features. The research described in this paper is carried out with a grant from Agentschap.nl under the EOS program for Long Term research, for which we would like to express our gratitude.

References


[20] C. Dal Mutto, P. Zanuttigh, G. M. Cortelazzo, Scene segmentation and video matting assisted by depth data, in: Time-of-Flight Cameras and Mi-


[34] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with Microsoft Kinect sensor: A review, Vol. 43, IEEE, 2013. doi:10.1109/TCYB.2013.2265378


[41] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests


