ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΕ ΧΡΗΣΗ ΕΙΚΟΝΩΝ ΒΑΘΟΥΣ ΣΕ ΠΑΡΑΜΟΡΦΩΣΙΜΑ ΜΟΝΤΕΛΑ

Γρηγόριος Γ. Χρυσός

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέποντες:

Πέτρος Μαραγκός, Καθηγητής Ε.Μ.Π.

Ιάσονας Κόκκινος, Αναπληρωτής Καθηγητής Ε.C.P.

Αθήνα, Σεπτέμβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΕ ΧΡΗΣΗ ΕΙΚΟΝΩΝ ΒΑΘΟΥΣ ΣΕ ΠΑΡΑΜΟΡΦΩΣΙΜΑ ΜΟΝΤΕΛΑ

Γρηγόριος Γ. Χρυσός

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέποντες:

Πέτρος Μαραγκός, Καθηγητής Ε.Μ.Π.

Ιάσονας Κόκκινος, Αναπληρωτής Καθηγητής Ε.C.P.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Σεπτεμβρίου 2014.

.....	.....	.....
Πέτρος Μαραγκός Καθηγητής Ε.Μ.Π.	Ιάσονας Κόκκινος Αναπληρωτής Καθηγητής Ε.C.P.	Κωνσταντίνος Τζαφέστας Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2014

.....
Γρηγόριος Γ. Χρυσός
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Acknowledgments

I consider it my duty to mention the people that contributed to my thesis study and their overall support throughout my studies.

My first experience in the field of Computer Vision was during the undergraduate course of Computer Vision, taught by Professor Petros Maragos, in the National Technical University of Athens. What originally attracted me to Computer Vision was that it requires knowledge from several fields including Mathematics, Physics, Computer Science and Algorithms. Additionally, I consider it very intriguing extracting information out of images and understanding the story behind each scene. Therefore, I would like to thank Professor Maragos, both for stimulating my interest in the field, and for providing me the liberty/flexibility to select the research topic as well as the direction of my research. He accepted my request to develop and implement significant part of the current thesis in Ecole Centrale Paris (ECP) under the supervision of Professor Iasonas Kokkinos.

Professor Kokkinos welcomed me to the CVC team and contributed significantly in my thesis. Prof. Kokkinos provided fresh ideas regarding the direction of my research and triggered my interest for the field of Computer Vision while sharing his experience with me. His guidance and advice were always welcome and aided my interest in the field. Considering those along with the endless hours he spent monitoring my progress, his contribution to the present work is deeply appreciated.

Except for my supervisors, I owe my deepest appreciation to all my close friends. Sharing ideas and experiences with all those people was a significant source of inspiration to continue with my work. They were always present, listening to my problems, to my ideas and urging me to follow my dreams. Special thanks to Ina, who was always motivating me and believing in my capabilities even more than I was.

Along with my friends, I would like to thank three special teams: The first one is Professor Koziris and the all members of okeanos team that provided the software ([56, 57]) which supported plenty of my experiments. The second team is the CVC team in ECP, where I spent several months conducting research for this dissertation. The EESTEC (Electrical Engineering STudent's European assoCiation) 'family' is the third one with all the friends I have gained. Last but not least, I would like to thank my biological family, whose love and support throughout my life has been fundamental in accomplishing my goals.

Abstract

Στη παρούσα διπλωματική εργασία εξερευνήσαμε την εργασία της ανίχνευσης αντικειμένων με χρήση $RGB + Depth$ ($RGBD$) εικόνων, ενώ προτείνουμε πέντε βελτιώσεις στο σύστημα $Deformable\ Part\ Models$ (DPM) του [29]. Η συμβολή μας διαχωρίζεται σε δυο κατηγορίες: σε επίπεδο γεωμετρίας και σε επίπεδο χαρακτηριστικών. Σε επίπεδο γεωμετρίας: (i) πετύχαμε αύξηση των δεδομένων εκπαίδευσης με μια γεωμετρική τεχνική $rendering$, (ii) επαυξήσαμε το $pairwise\ term$ ([29]) ώστε να περιλαμβάνει πληροφορίες βάθους, (iii) πετύχαμε καλύτερη αρχικοποίηση των διαφορετικών ομάδων δεδομένων εκπαίδευσης. Σε επίπεδο χαρακτηριστικών πετύχαμε: (i) την σχεδίαση νέων χαρακτηριστικών, τα οποία αποκαλούμε $displacement\ features$ (χαρακτηριστικά μετατόπισης), (ii) επαύξηση των $sparse\ codes$ για ανίχνευση αντικειμένων σε $RGBD$ εικόνες. Διεξάγαμε εκτενή πειράματα, όπου παρουσιάζουμε ότι οι μέθοδοι που προτείνουμε υπερέχουν από τα τρέχοντα συστήματα για ανίχνευση αντικειμένων σε $RGBD$ εικόνες.

Keywords

όραση υπολογιστών, αναγνώριση αντικειμένων, εικόνες $RGBD$, βάθος, χαρακτηριστικά, $rendering$

Abstract

In this thesis we explored the task of object detection for RGB+Depth (RGBD) images and propose five improvements to the Deformable Part Models (DPM) system of [29]. Our contributions are divided in two categories: the geometry based and the feature based. In the geometric extensions: (i) we augmented the training data using a geometric rendering technique, (ii) we modified the pairwise term of [29] to account for the depth information and (iii) we accomplished a better initialization of the training groups. Our feature based extensions consist in: (i) introducing our new features which we call displacement features, (ii) augmenting sparse coding for object detection for RGBD images. We have conducted extensive experimentation, where we demonstrate that our proposed system outperforms the current state-of-the-art DPM system in object detection with RGBD images.

Keywords

computer vision, object detection, RGBD images, depth, features, rendering

Contents

List of Figures

Chapter 1

Introduction

1.1 Objectives of the study

The current diploma thesis belongs to the research field of computer vision and more specifically in object detection. Computer vision is a field that includes methods for acquiring, processing, analyzing, understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions ([67]).

Object detection is one of the core research topics of computer vision having numerous applications that require the extraction of high-level information from images. Object detection refers to recognizing objects in a scene, like in Fig. 1.1. Object detection is distinguished into two major types: instance recognition and category level recognition.

Instance recognition is the task of identifying whether an object is physically the same object that has previously been seen. Instance recognition depends highly on the RGB and instance-specific information ([58]). Capturing such instance specific information has been extensively studied and has presented successful recognition results in most cases ([12, 47, 58, 95]). Therefore, we will not refer to instance recognition in the current thesis.

Category level recognition is the task of detecting previously unseen objects as belonging in the same category as objects that have previously been seen.

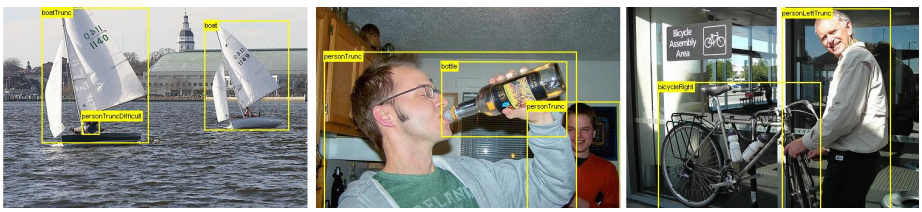


Figure 1.1: Example images, with overlaid ground-truth bounding boxes.The images are from PASCAL VOC Challenge 2007 ([26]).

Category level recognition depends on generalization of the properties or functionalities of the object to capture unseen instances of the same category. The representation of the object can undergo changes such as scaling, translation, occlusion, or other deformations, which make category level recognition a challenging topic, therefore in this thesis we develop methods to improve category level recognition.

The overall goal of the thesis is to increase the robustness of object detectors, while also moving towards the 3D recognition. We explore the use of depth images alongside with RGB images for object detection. We present five contributions that prove that the depth images are a useful tool to increase the performance of the object detectors.

1.2 Historical overview of detection

Since our work belongs in the field of object detection, we present a brief historical overview to explain the current progress in the task. Object recognition by a computer has been an active area of research for several decades. A thorough historical analysis of object recognition can be found in [23, 68]. In Fig. 1.2 there is a summary of the historical evolution of object recognition since the 1970's. Using Fig. 1.2, we describe below selected lines of work and significant trends that dominated each decade.

The preamble of the recognition problem was in 1950's and 1960's. A dominant method was the blocks world idea ([93]), best described in [77]. The blocks world idea restricted objects to be polyhedral shapes on a uniform background.

In the 1970's, the idea of modeling objects as 3D volumetric parts was applied ([4, 8, 9]). Some significant principles emerged during the decade including the significance of shape, viewpoint invariance, hierarchical and 3D representations. The following decade the mainstream of work was focused on 3D models that were capable of recognizing real objects ([18, 63]), which was a breakthrough for the period. 3D models declined significantly during the following decade (1990's) with the appearance-based models gaining the attention of the community ([10, 14, 66, 70]).

In the previous decade (2000's), there was a vast increase in the number of works. A method that dominated the first years of the decade was the use of Adaboost ([34]). A representative work of the use of Adaboost is the Viola and Jones detector ([90]). In 2004 and 2005 the introduction of Scale Invariant Feature Transform (SIFT) ([62])) and Histograms of Oriented Gradient (HOG) ([20]) respectively allowed the shift from global to local features. Both SIFT and HOG are among the most popular feature schemes. An additional line of research was the addition of pairwise spatial constraints ([28, 31, 32]) that captured the deformation of different object instances. The combination of the efficient feature representation and the pairwise spatial constraints were combined in the Deformable Part Models (DPM) in [29].

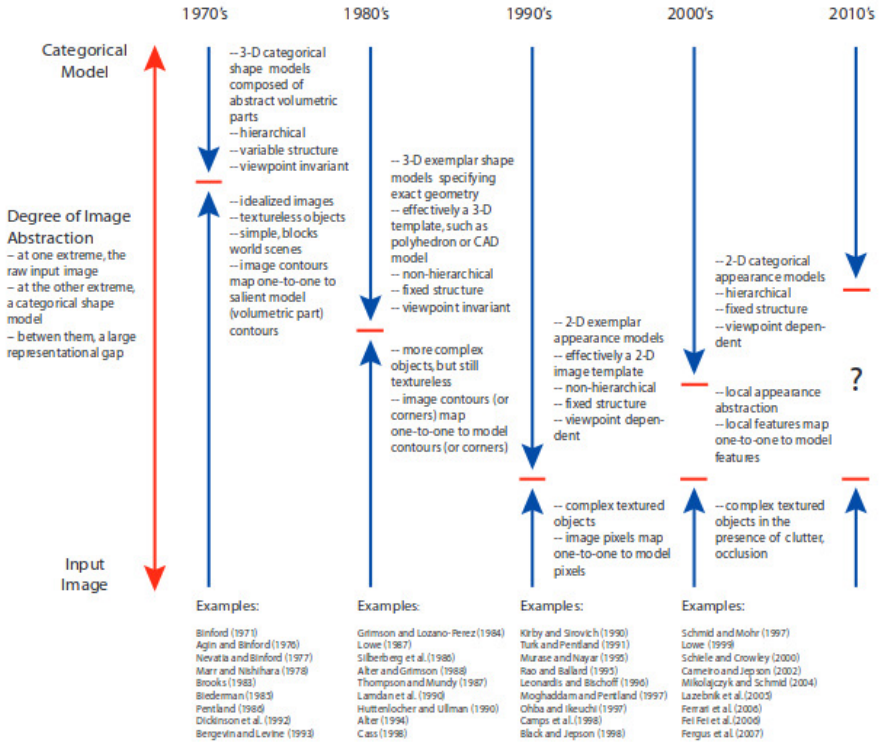
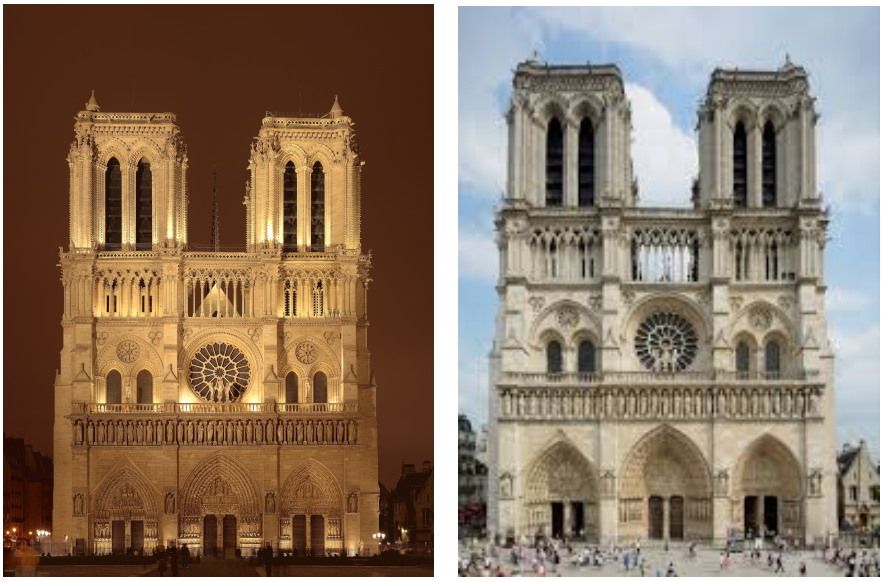


Figure 1.2: The historical evolution of object recognition as summarized in [23].

1.3 Challenges of object detection

Despite the rapid progress in the task of object detection, recognizing objects in a systematic way is challenging from several aspects. We present the following four major challenges:

- **Illumination**: The lighting conditions in an image, e.g. in Fig. 1.3. Humans trivially recognize that the building in Fig. 1.3 is the same, but for an algorithm the two images differ significantly. [33, 59] are two methods to estimate the likely condition of illumination of the scene, however the challenge is far from solved.



(a) night-light (b) day-light

Figure 1.3: Representation of an image under different lighting conditions. The feature representation of the images will likely differ significantly.

- **Camera Viewpoint**: The specific location/angle at which a camera is placed to take the shot, e.g. in Fig. 1.4. Without a 3D model of the scene, as is the case with the datasets we work on, it remains challenging to achieve complete camera invariance in object detection.

(a) front-view (b) side-view

Figure 1.4: A landmark from two different viewpoints. Without prior knowledge of the landmark it remains questionable whether even a human would recognize that the two images refer to the same building.

- **Texture**: The spatial arrangement of color or intensities in a region of an image, e.g. in Fig. 1.5.

(a) (b)

Figure 1.5: Two chairs with different texture. Not only the color of the two chairs differs, but also their material.

- **Occlusion**: Part of the object is hidden (occluded) either by other part of the object or by another object, e.g. in Fig. 1.6. In [30, 88] there are techniques developed to detect occluded objects, however the outcome depends on the occluded part and the occlusion level.

(a)

(b)

Figure 1.6: In (a) human legs are occluded, while in (b) the body of a dog is occluded.

1.4 3D object representation and depth maps

Obtaining the exact 3D representation of a scene can facilitate object detection ([21, 48]) by reducing intra-class variation. Intra-class variation in object detection is due to both intrinsic (texture and shape), and extrinsic (viewpoint and illumination) factors.

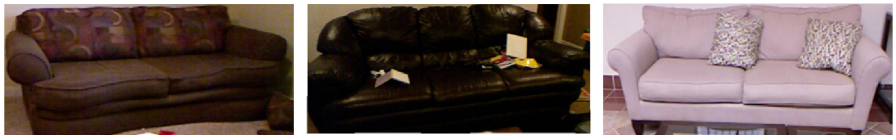
However, obtaining a 3D representation of a scene was computationally expensive. Therefore the focus was in recovering information about the shape of the objects from 2D images. The recovered shape can be expressed with the form of a depth map.

A depth map is an image or image channel that contains information relating to the distance of the surfaces of objects from the camera that captured the shot. In Fig. 1.7 an RGB image is presented along with its depth counterpart. The default indication is that the lighter the shade of grey is, the further the item is from the camera.

(a) RGB

(b) Depth

Figure 1.7: Representation of an RGB image and its respective depth map.



(a) RGB

(b) depth

Figure 1.8: Illustration of the invariance of depth images to illumination and texture. In the first row, there are 3 different objects of class sofa, while in the second the corresponding depth maps. It is notable that in the RGB images the sofas differ significantly in illumination and texture, while in the depth they are similar. Note that the different shades of grey in different depth images depend on the relative depth that each sofa appears in the image.

As illustrated in Fig. 1.8 depth maps are invariant to texture and illumination changes, thus they reduce intra-class variation. For several decades, techniques to recover the shape had been studied ([13, 43, 46, 51, 65, 80]), but these techniques did not work well in realistic scene images. The introduction of the Kinect sensor ([1, 2]) provided an inexpensive and accessible way of acquiring RGB and aligned depth images in real time. This accessibility led to a proliferation of works around holistic scene analysis through shape, appearance and even physics-based cues ([61, 100, 101]). A more thorough review of the Kinect-based algorithms exists in [41].

In the current thesis, we explore the use of Kinect RGBD images to increase the performance of the Deformable Part Models system.

1.5 Thesis outline

In Chapter 2 we present the popular framework of Deformable Part Models (DPM) ([29]). We also present representative works previously published in object detection with RGBD images.

In Chapter 3 we introduce our proposals to extend the existing DPM system to account for depth images. We divide our contributions in geometric based and feature based and analyze their theoretical side and the motivation for their development.

The experimental setup and the results that we have obtained are studied in Chapter 4. We evaluate our contributions and conclude that they improve the state-of-the-art DPM system in object detection with RGBD images.

The last Chapter of our thesis concerns the summation of all the contributions

and some further proposals for future work.

Chapter 2

Previous work in object detection

The goal of this Chapter is twofold: (i) to demonstrate the most significant attributes of the Deformable Part Models, in which we integrate our contributions, and (ii) to present representative works on object detection with RGBD images to motivate our proposed improvements.

2.1 Deformable Part Models (DPM)

Deformable Part Models (DPM) constitute an object detection framework, which have been really effective for learning object detectors for thousands of classes [22] or in tasks like face detection [102], articulated pose estimation [97], video detection [89]. The ability of DPM to recognize objects that appear in highly varying size, shape and scale is the key feature that established DPM as a successful framework for detection.

The DPM are a learning-based system, they learn to detect and localize an object based on the training data provided to the system, while they only require weakly-labeled data. Weakly-labeled data constitute of images in which only the bounding boxes around each object of interest are given. The goal of DPM for object detection is to assign a label $w \in \{0, 1\}$ indicating whether a small image patch x contains a specific object ($w = 1$) or not ($w = 0$). To achieve this goal, DPM learn a model for each class and then they can ideally detect all instances of this class in a new test image. In the following paragraphs we will present in more details the model and the training procedure followed by DPM.

2.1.1 Model

DPM represent objects as a star-shaped graphical model with the central node or 'root' and several leaf nodes, named 'parts', connected in a star shape with the root. In Fig. 2.1 we demonstrate a star-shaped model for human faces.

The root variable of the DPM accounts for the appearance of the entire object, while the leaf nodes correspond to the appearance of the parts of the object.

The edges of the star-shaped model represent the spatial relationship between the root and the parts.

The root consists of a linear filter with a fixed position, while the parts are linear filters which are allowed to change their relative positions. The features of the parts are computed at twice the resolution of the features in the root level.

(a) face (b) star shape model in real face

Figure 2.1: Illustration of a star-shaped model for the human face. In (a) an original face and in (b) a representative star-shaped model with the root to be located in the nose and 4 additional points as parts.

2.1.1.1 Deformations and parts

Based on the Dalal-Triggs detector ([20]), DPM use a number of improvements. The deformation of the parts and the feature pyramid are two improvements that reduce the impact of shape and scale variance of the objects.

- **Deformations**: A significant hypothesis of DPM is that the objects are composed of a large number of rigid parts that are linked together through nonrigid connections. This is the motivation for the star-shaped model with the root and the parts, a formation which allows the objects to vary in shape from other objects of the same category. An object 'deforms' when its parts change their relative positions or orientations.

- **Feature pyramid**: Apart from the shape differences, objects appear at a wide range of scales. DPM enforce the extraction of features in different scales and with different size of patches. In practice, each feature pyramid is computed through a standard image pyramid via repeated smoothing and sub-sampling.

- **Score**: Detecting objects requires a high-scoring object hypothesis. An object hypothesis specifies the location of each filter in the model in a feature pyramid. The score of an object hypothesis is composed by three terms, the unary potential, the pairwise term and the bias factor b.

The unary potential $U(p_i)$, with p_i indicating the part i and p_0 the root, is the score of each filter in its respective position. Both root and part filter unary scores are defined by the dot product between a filter and a sub-window of a feature pyramid. The pairwise term captures the deformation cost that depends on the relative position of each part with respect to the root. In a mathematical formulation:

$$S(p_0, p_1, \ldots, p_P) = \sum_{i=0}^{P} U(p_i) - \sum_{i=1}^{P} d_i \cdot \varphi_d(p_i) + b \tag{2.1}$$

where node 0 is the root and the remaining P parts connect to the root with a deformation cost of

$$\varphi_d(p_i) = (dx_i, dy_i, {dx_i}^2, {dy_i}^2) \tag{2.2}$$

To detect objects in an image an overall score for each root location is computed according to the best possible placement of the parts:

$$score(p_0) = \max_{p_1,\ldots,p_P} score(p_0, p_1, \ldots, p_P) \tag{2.3}$$

The locations of the parts that yield a high-scoring root location define an object hypothesis and a high-scoring root location defines a detection. Fig. 2.2 illustrates the afore-mentioned matching process, while Fig. 2.3 presents high-scoring detections of the model for several classes.

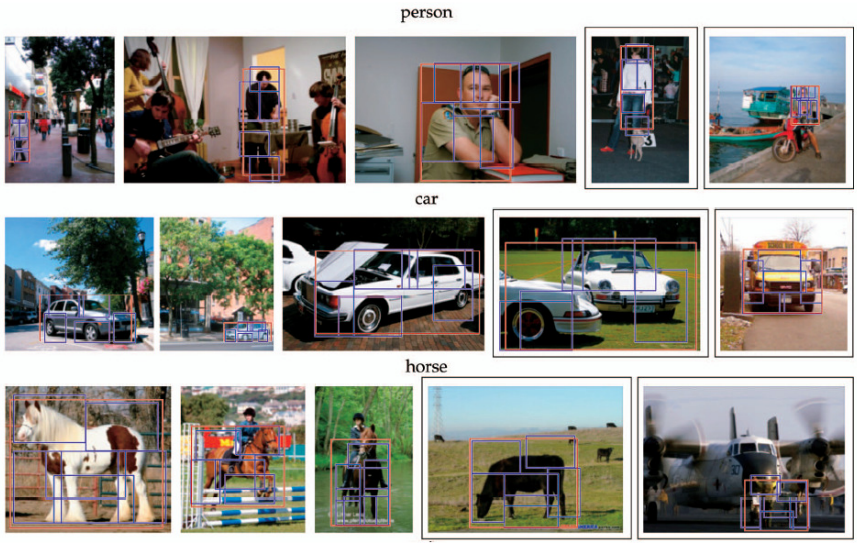


Figure 2.3: Examples of high-scoring detections in the PASCAL VOC 2007 ([26]). The framed images (last two in each row) illustrate false positives for each category.

2.1.1.2 Mixture of models

Objects can appear in several views (frontal, side, back views) and this is increasing the complexity of detections. DPM capture the different views with

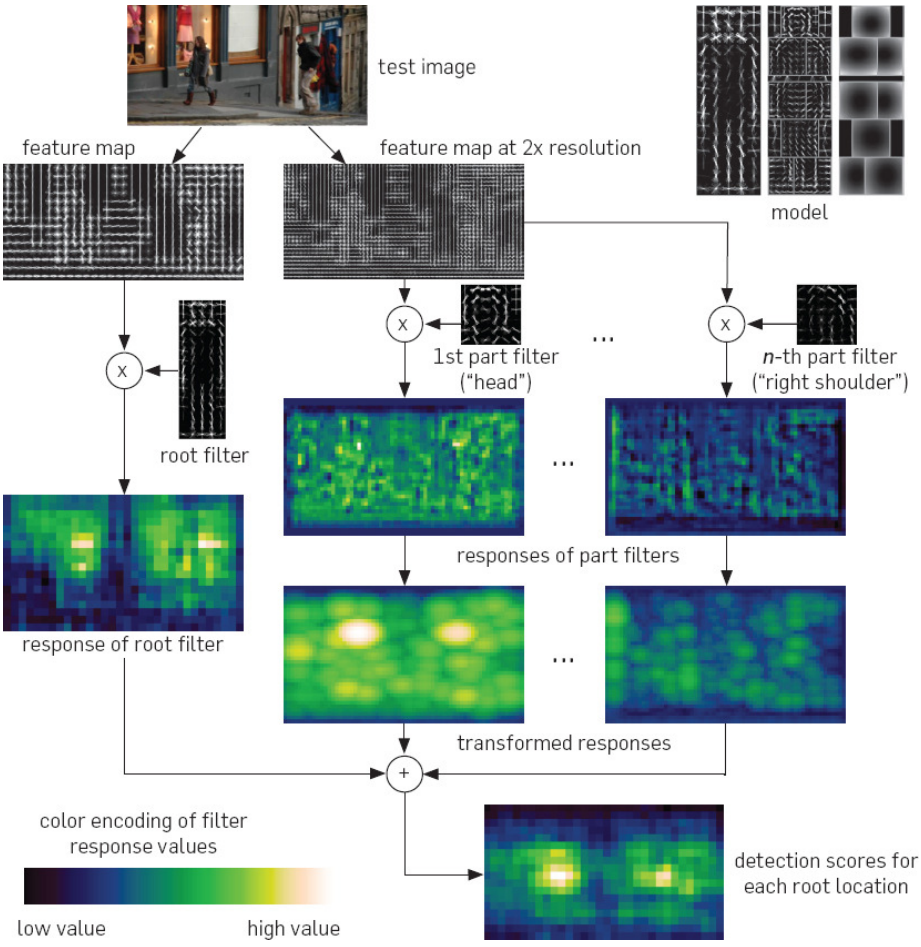


Figure 2.2: The matching process at one scale for the class person. Due to space limitations, responses and transformed responses for the 'head' and 'right shoulder' parts are shown. Note how the 'head' filter is more discriminative. The combined scores indicate two high scoring hypotheses for the object.

the use of m components. Each component is described with a model as analyzed in the Paragraph 2.1.1.1. Each component captures a different view of the object. The equation 2.1 of detecting objects by a mixture model is used for every component independently. Fig. 2.4 presents a mixture model for the class bicycle.

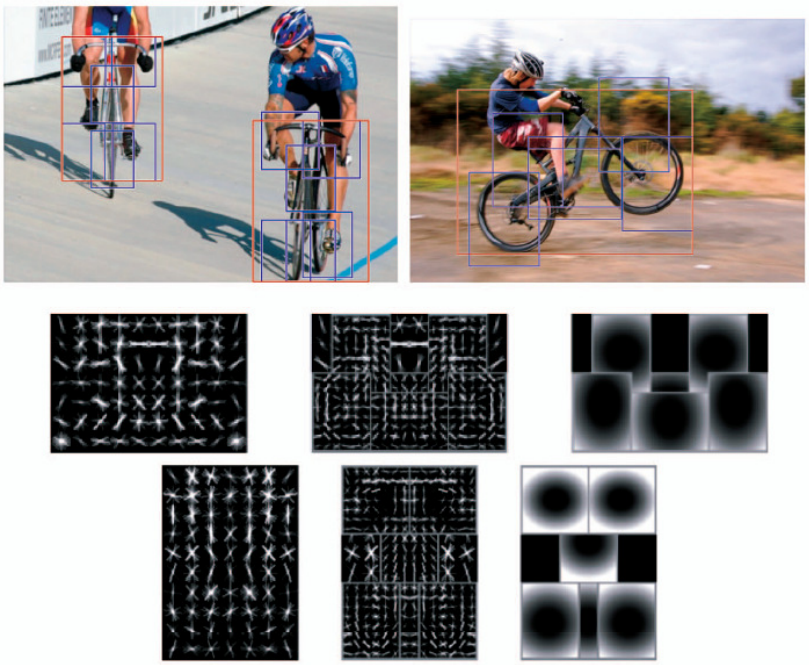


Figure 2.4: Detections obtained in class bicycle with 2 components. In this model, the first component captures sideways views of bicycles while the second component captures frontal and near frontal views.

2.1.2 Learning

DPM use discriminative training with an optimization method called latent SVM ([29]). A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. A latent SVM scores an example x with

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z) \tag{2.4}$$

where β is the concatenation of the root filter, the part filters and the deformation cost weights, z are latent values that specify a component label and a configuration of the parts for this component and $Z(x)$ is a set of the possible components and their configurations. β is trained by minimizing the loss function

$$L_D(\beta) = \frac{1}{2} \mid\mid \beta \mid\mid^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)) \tag{2.5}$$

where (x_i, y_i) are the labeled examples with $y_i \in \{-1, 1\}$. If x_i contains the object, then $y_i = 1$, otherwise $y_i = -1$. The classifier described by equation 2.4 is not linear and the problem is non-convex.

For $y_i = -1$ (negative examples) the hinge loss, $\max(0, 1 - y_i f_\beta(x_i))$, is convex in β because the maximum of a set of convex functions is convex. For the negative examples, both 0 and $(1 + f_\beta(x_i))$ are convex, since $(1 + f_\beta(x_i))$ is the sum of two convex functions. For a positive example though the hinge loss is the maximum of a convex (0) and a concave $(1 - f_\beta(x_i))$ function. However, if there is a single possible latent value for each positive example, the loss due to each positive is convex and therefore (2.5) becomes convex for both the positive and the negative examples. In a mathematical formulation, let Z_p specify a latent value for each positive example in D. The auxiliary objective function $L_D(\beta, Z_p)$ is defined by restricting the latent values for the positive examples. It is proven that

$$L_D(\beta) = L_D(\beta, Z_p) \tag{2.6}$$

In DPM, $L_D(\beta, Z_p)$ is minimized with a coordinate descent method based on two iterative steps:

1. Relabel positive examples: Optimize $L_D(\beta, Z_p)$ over Z_p by selecting the highest scoring latent value for each positive example.

2. Optimize β: Optimize $L_D(\beta, Z_p)$ over β by solving the convex optimization problem.

It should be noted that both steps are guaranteed to either improve or maintain the value of $L_D(\beta, Z_p)$. Step 1 enforces the examination of an exponentially large space of latent values for positive examples while step 2 searches over all possible models.

2.1.2.1 Hard negative mining

When training a detector, it is often important to use large training sets to achieve high performance. However, the training problem is then highly unbalanced because there exist many more background structures than objects. This motivates a process of eliminating the easily detected background examples and keeping a relatively small number of potential false positives, or hard negative examples.

A similar methodology was adopted in [20], but in DPM there are several rounds of data-mining. The idea is to initialize a model with a subset of negative examples, and then collect negative examples that are incorrectly classified by this initial model to form a set of hard negatives. A new model is trained with the hard negative examples.

2.1.2.2 Sensitivity to initialization

Let P denote the set of positive examples for the class and N the background examples. The optimization algorithm used is susceptible to local minima and thus there are 4 phases of initialization before the final training step. The training phases in detail are:

- **Phase 1: Root initialization**: For each of the m components one detector is trained with a different root filter and different training data. The different splits $P_1, P_2, ..., P_m$ of training data are determined based on the aspect ratio of the bounding boxes of the positive examples. The training data are sorted based on the aspect ratio and then split into m groups of equal size.

The dimension of each root filter is selected automatically by extracting statistics of the bounding boxes in the training data. Concretely, it selects the mean aspect ratio of the boxes in P_i and the largest area not larger than 80% of the boxes.

Each model is trained with a standard SVM with no latent information. The positive examples are anisotropically scaled to the size and aspect ratio of the filter, while the negative examples are extracted from random sub-windows from N.

- **Phase 2: Mixture of roots**: For each aspect ratio group P_i a mixture of two root filters is trained. Each pair of root filters are horizontally mirror images of each other. The positives examples are not warped and hard negative examples are used in this step.

- **Phase 3: Merging components**: All the initial models from the previous step are combined into a mixture model which is retrained in the full (unsplit) training data. In this step the component label and root location are considered latent variables.

- **Phase 4: Part Initialization**: A fixed number of parts is initialized using a heuristic. The parts are greedily placed in the high-energy regions of the root filter with energy of a region defined by the norm of the positive weights in a sub-window. Once a part is placed, the energy of the region covered is set to zero, and the next region is searched. The part filters are initialized from the root filter values in the sub-window selected for the part, but filled in to handle the higher spatial resolution of the part. The resulting model serves as the initial model for the last round of parameter learning.

2.2 Object recognition with RGBD images

The depth maps provide rich surface and shape information of the scene, but it has not yet been studied extensively how to optimally exploit this information for statistical model learning. Recent approaches have exploited 3D object models ([72]) for object detection from RGB images and only partially used surface information ([82, 87, 98]). Below we describe the aforementioned works in more details.

2.2.1 Histograms of Oriented Normal Vectors

In [87], the authors introduce new features (Histograms of Oriented Normal Vectors or HONV) computed only by the depth images. Their contribution consists in computing the normal vector and forming the histograms as a concatenation

of local histograms of azimuthal and zenith angle.

Concretely, they firstly consider a plane L that contains the point $p(x, y, d(x, y))$ with $d(x, y)$ indicating the depth at p. They restrict L to be parallel to xy-plane. They compute the normal vector, N, of L and prove that

$$N = \begin{pmatrix} -\frac{\partial d(x,y)}{\partial x} \\ -\frac{\partial d(x,y)}{\partial y} \\ 1 \end{pmatrix}$$

and since the third dimension is 1, in the spherical coordinates, N can be computed based only on the zenith angle θ and the azimuth angle ϕ. Following the idea of [20], they divide the detection window into $m \times n$ non overlapping cells. In each cell, the orientation of the normal vector at each pixel is quantized and voted into a 2D histogram of ϕ and θ.

2.2.2 Histogram of vector quantized surface descriptors

In [98], the authors introduce new feature descriptors that they integrate in the DPM framework.

Their algorithm is based on computing surface normals and then aggregating them in a histogram. To be more precise, they first compute the surface normals and then cluster them using a standard K-means ([42]) on cosine distance metric. They segment each image patch into a square cell and then compute the histogram by vector quantizing surface normals to the nearest centroid produced during the K-means step.

2.2.3 Geometry DPM

In [82], the authors extend the training phases of [29] to capture geometric information of the objects. Their fundamental assumption is that every object has constituent parts with consistent 3D geometric properties, for instance a table includes a horizontal part and some legs supporting it. In their work, they require RGBD data during training time and RGB data during test time. They use the depth data as weak supervision to impose geometric constraints.

Their algorithm includes two phases: initializing parts and training a model. In the first phase, they compute surface normals and train a dictionary with 3D information. Then they use the dictionary to initialize the parts for each class. In the second phase they train geometry DPM (gDPM). gDPM defers from [29] in the computation of the score of an object hypothesis. In gDPM they augment the score function with a term that restricts the spatial movement of the parts.

Chapter 3

Extending DPM for RGBD images

In this Chapter we describe our work in extending DPM from the existing framework of RGB images to take into account the depth images as well. We divide our contributions in two categories: the geometry based and the feature based. The geometry based exploit the geometric information of the scene introduced by the depth images. The feature based augment the existing feature representation by extracting information from the additional cue.

3.1 Geometric extensions

A depth image provides rich geometric information about the 3D formation of a scene. We exploit the geometric information to achieve three improvements: (i) augment the training data, (ii) modify the pairwise term to include depth displacements, (iii) accomplish better initialization of training groups.

3.1.1 Dataset Augmentation

3.1.1.1 Geometric Jittering

A major challenge in object recognition is variation due to camera viewpoint; camera rotations can modify the appearance of an object so radically that mixtures of viewpoint-tuned classifiers are imperative for multi-view detection. This challenge remains with RGBD sensors, as they only record the side of the object's surface that is facing the camera. However, we can exploit depth information to improve the robustness of our detectors by accommodating variability due to moderate camera rotations.

We propose a dataset augmentation scheme that uses geometric information to take 'a different look' at the published images of the dataset. In particular, we simulate the effects of small camera rotations around the initial viewpoint that each image is captured. Thus, we acquire new images that reveal how each scene would seem from a novel viewpoint. Our technique can be understood as a generalization of the 'jittering' technique that is known to drastically

improve detection accuracy by using translated/scaled/rotated samples of an object during training. All these transformations assume that the azimuth and elevation of the camera stays fixed; here instead we let azimuth and elevation vary moderately ($\pm 10^o$) and use the resulting images to enhance the variability of our training set.

3.1.1.2 Image-based rendering

Viewing a scene from different angles is an attractive feature for applications such as 3D reconstruction ([44, 60]), medical imaging ([78]) and multimedia services ([99]). Since the number of cameras is practically restricted and consequently the number of viewing angles, research has been devoted to interpolate views between the cameras. The creation of such artificial views in 3D is called rendering.

In our work we use Depth-Image-Based Rendering (DIBR) which is a method for synthesizing novel 'virtual' views of an image, based on its intensity and depth values. Conceptually, this process can be understood by first warping the points on the original image plane to the 3D world coordinates and then back-projecting the real 3D points onto the virtual image plane which is located at the required viewing position. Following the conventions as described in [27] the creation of a novel view representing an horizontal movement to the left can be simplified to computing the horizontal displacement of each pixel, which is:

$$u' - u = \frac{f(a, \theta)}{Z} \tag{3.1}$$

where $f(a, \theta)$ depends on the camera parameters, and the rendering angle θ, Z is the depth value at pixel (x, y).

In this work, we use the free viewpoint rendering method described in [25] to render both depth and RGB images. Even though the geometric transformation involved in DIBR is straightforward, a host of image processing problems emerge ([25, 27, 54, 69, 86]), involving ghost contours, cracks, and most importantly, disocclusions, namely areas that cannot be viewed from the original viewpoint. More recently, [75] uses structural information from 3D models to synthesize novel-views of cars from images, which are used to amplify training data for DPM. However, the method proposed in [75] requires alignment of real images with 3D models, which was achieved manually.

3.1.1.3 Inpainting-based post-processing

The method in [25], applies two methods to tackle both the ghost contour and the cracks. The ghost contour is solved with preprocessing, where a Gaussian smoothing is applied, while the cracks are filled with inverse warping of the hole to the original view. Therefore, the rendered views suffer only from the presence of holes/disocclusions. The common way to fill such holes is to apply some inpainting technique.

Inpainting is a method to modify an image in a way that the filled areas are non-detectable for an observer who does not know the original image. The

mathematical description of inpainting is the following: Let Ω denote a complete 2D image domain and D denote the missing values which are a subset of Ω. The goal of inpainting is to recover the original ideal image u on the entire domain Ω, based only on partial observation $u_{0|\Omega/D}$.

A variety of techniques, which can be divided into diffusion based and exemplar based, have been applied for image inpainting ([19, 37, 71]). The diffusion based techniques ([7, 73, 74, 81, 92]) focus on filling the missing regions with local information. However, a drawback is that sometimes the local information do not characterize the missing region. More recently, the exemplar based techniques ([15, 19, 94]) utilize non-local information by finding a matching sample from the whole image region.

We consider that the authors of the related papers for DIBR techniques have already explored several inpainting methods to remedy the holes, therefore we do not conduct an extensive research on inpainting algorithms. We post-process the RGB image renderings by performing an inverse warping of the missing pixels to retrieve the textures as described in [25], and the depth renderings using the bilateral filtering approach as proposed in [83]. We experimented with different algorithms for depth inpainting, including the works of [36, 45], and our extension of [19] to consider the depth patches. Since the databases for depth images involve over 1000 images, we required the method to be both qualitative efficient and fast and thus we concluded that the bilateral filtering provides the best result/time ratio with an average time of 0.08 seconds per image for depth inpainting. In Fig. 3.1 the discrete steps of rendering, crack elimination and inpainting are presented in an example.

3.1.2 Pairwise term with depth deformation

As described in the Paragraph 2.1.1, the position of the parts is allowed to change to capture the varying shape of different objects. The availability of depth data can lead to a deeper insight into the deformation of each object.

The default deformation in the DPM is expressed with the equation 2.2:

$$\varphi_d(p_i) = (dx_i, dy_i, {dx_i}^2, {dy_i}^2) \tag{3.2}$$

Recall that this equation expresses the horizontal and the vertical displacement of the parts from their nominal position. We augment this equation to consider the deformation of the parts in the third dimension, depth. The motivation is to restrict the movement of the parts according to the shape of the object. For example, if there is an accountable difference of depth around the object, the deformation cost should increase to penalize the existence of a part there.

We experimented with adding: (i) values of quantized difference of depth, (ii) a quadratic function, (iii) only a quadratic term. The best outcome is provided by adding only a quadratic term, dz^2. Therefore, the equation of the deformation was modified to

$$\varphi_d(p_i) = (dx_i, dy_i, {dx_i}^2, {dy_i}^2, {dz_i}^2) \tag{3.3}$$

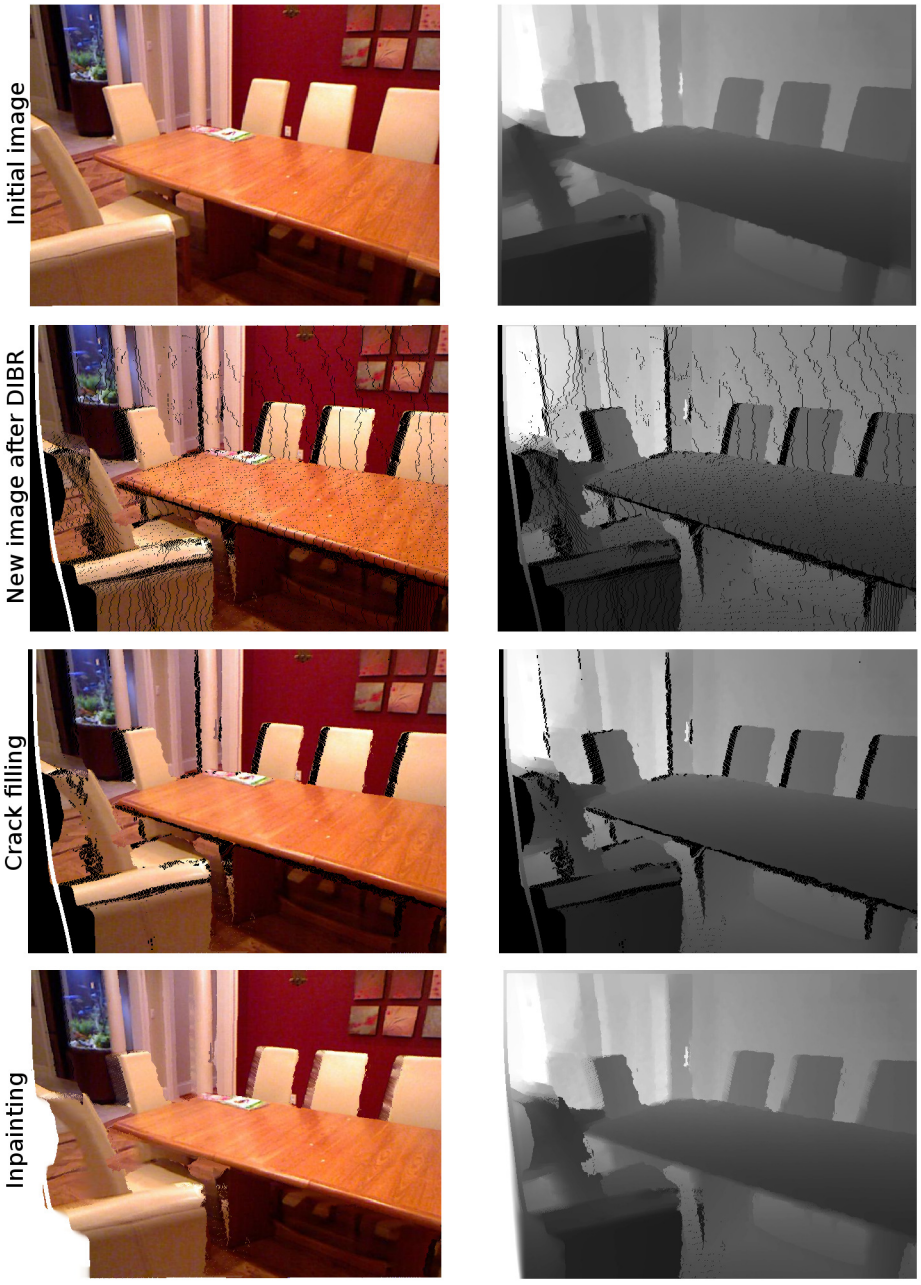


Figure 3.1: Representation of the 3 steps to create a new sample. The first row presents the initial samples, the second the outcome of DIBR, the third how the new images look after crack filling and the last the final outcome. In the first column are the RGB images, while in the second the respective depth. The distortions in the second and third row are apparent.

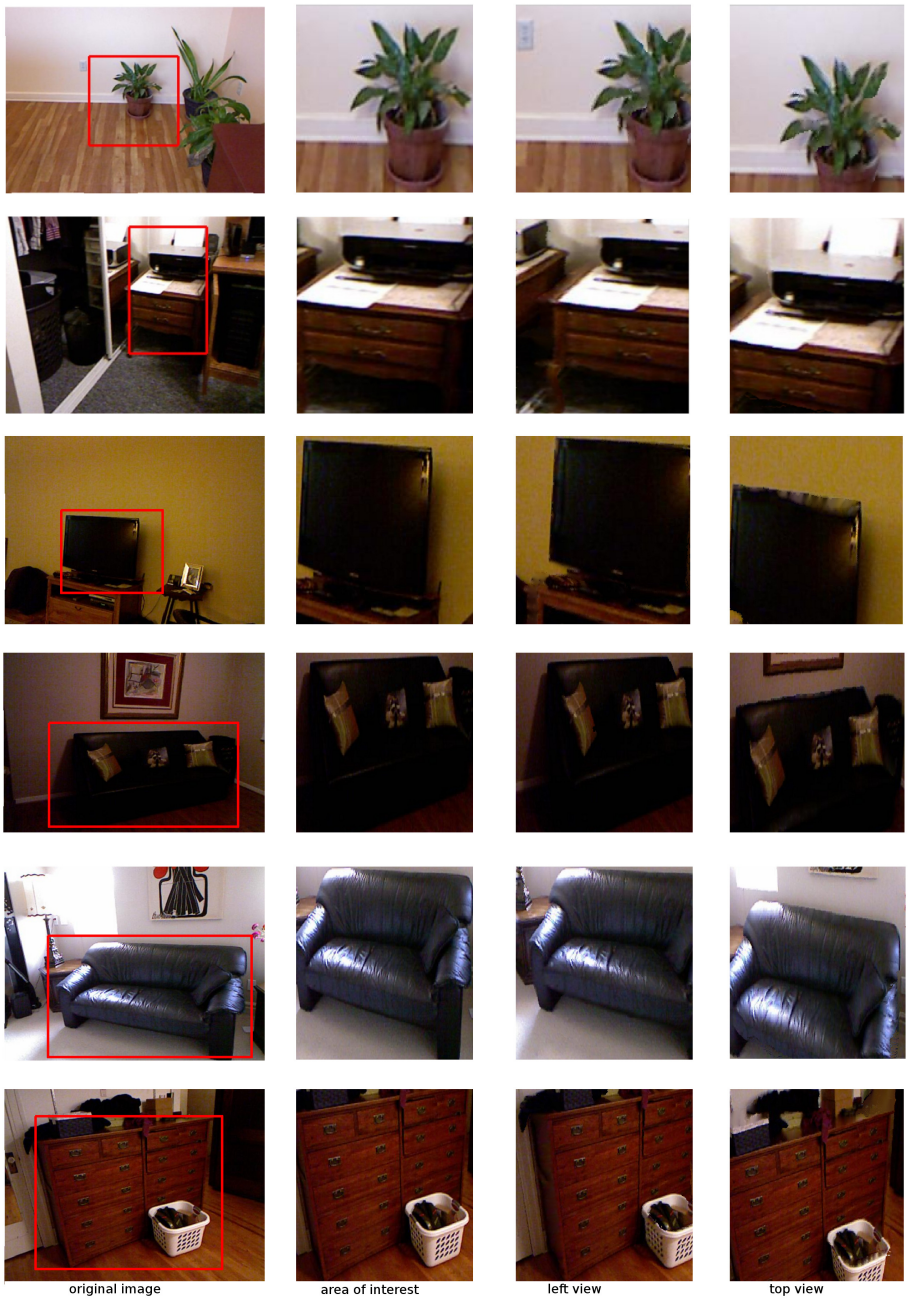


Figure 3.2: Rendering novel views using Depth Image-Based Rendering: in the first column the original images, in the second the zoomed in area of interest of the original image, and the last two include the two views rendered by moving the camera to the left and top respectively. We observe that there are noticeable differences among the different views and the original image. The rendered views capture all the essential information of the original image and reproduce it from the new viewpoint. There are some small distortions in the points where there is considerable depth discontinuity or highly cluttered object, e.g. the leaves in the first image. However, we demonstrate that performing this geometric dataset augmentation during training is beneficial.

3.1.3 Component initialization with 3D geometric split

As described in the Paragraph 2.1.1, a mixture model is used to capture different views of an object, like in Fig. 2.3 for bicycle. Felzenszwalb *et al.* ([29]) clustered the object samples into m groups according to the aspect ratios of their corresponding bounding boxes. However, due to the occlusion or truncation of objects in the bounding boxes, aspect ratio often leads to suboptimal clustering ([24]). We improve the aspect ratio clustering by incorporating depth information.

We consider a 3D Cartesian coordinate system in which the axes are height, width and depth values. Each training sample is a point in the coordinate system, since we have knowledge of its height, width and depth. Before inserting a sample, we normalize its dimensions with its volume value. Each training sample is described with a vector of dimensionality 3 in the coordinate system space. We insert all training samples in the coordinate system and we perform K-means clustering ([42]) to separate the data into different groups. In particular, we require K-means to separate the data into m groups, each group of which will initialize one of the m components.

We visualize the coordinate system in which each sample is represented by a point. As presented in Fig. 3.4 there is significant variance in the difference of depth among different positive examples, thus the variation in the data is improved. In Fig. 3.3 a visual indication why the 3D geometric split improves the component initialization is presented.

3.2 Feature-based extensions

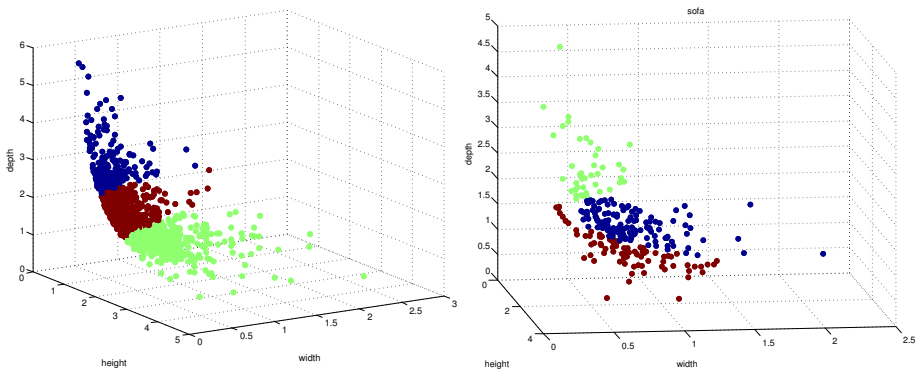
Extending DPM from RGB to RGBD images, allows us to extract additional information from the new cue, depth. Our first step is to extend the HOG RGB features to HOG RGBD. We, also, use the depth channel to accomplish two improvements: (i) our novel displacement features, (ii) sparse codes for RGBD.

3.2.1 Histograms of Depth Gradient

A straightforward extension of feature extraction in RGBD images is the implementation of [20] for depth images. Histograms of Depth Gradient (HOD) ([85]) consist this extension. In our experimentation HOD outperformed HOG RGB of [29] and especially in classes that have a distinctive shape from the background, e.g. sofa and bed.

Since RGB and depth images include complementary information, we concatenate the feature vectors of HOG RGB and HOD ([85]). The concatenated feature vector, HOG RGBD, outperforms the individual features in the performance of the respective DPM detectors. Therefore, we use this as the baseline feature technique that we compare our contributions.

Figure 3.3: Training samples that indicate the enrichment of the split with the incorporation of depth. Each image represents a training sample for the class of chair. All these 9 samples belong in the same aspect ratio group ([38]). However, in the proposed method of the 3D geometric split, each row of chairs belongs to a different group. We observe that there is a correlation between the depth of the visible part of the chair and the group, i.e. in the first row the narrow parts, while in the last one, the bigger examples. This intuition would be lost in the aspect ratio split, since all of them would be in the same group.



(a) Chair (b) Sofa

Figure 3.4: Representation of the training samples in the 3D Cartesian coordinate system for the classes of (a) chair and (b) sofa. The axes of the coordinate system are height, width and depth. Each point represents a training sample while the different colours represent the 3 different groups obtained after K-means. We observe that there is significant variance in the 3rd axis of depth, therefore it is meaningful to take into account this variance.

3.2.2 Displacement Features

Here we describe our novel displacement features, which extend our detector by using depth information. Displacement features are local, depth-based descriptors that are typically computed over regular interest points in a dense grid.

Most standard feature extraction schemes, for example [20, 62], employ a sequence of differentiation and L2-norm normalization; these two steps discard the effects of additive and multiplicative illumination changes respectively. This processing is combined with spatial pooling to render the descriptors robust with respect to small translations. Even though the aforementioned steps yield increased robustness, a substantial part of the signal information is lost during the processing steps, and is considered to be partially responsible for the saturation of HOG-based detection performance [91].

More dedicated point cloud descriptors have been proposed early on in the literature, starting from SPIN images [35, 53] and moving on to more recent and sophisticated variants, including Fast Point Feature Histograms [79], normal-based descriptors [79], as well as full-fledged learning-based descriptors, using sparse coding [11] or deep learning [84].

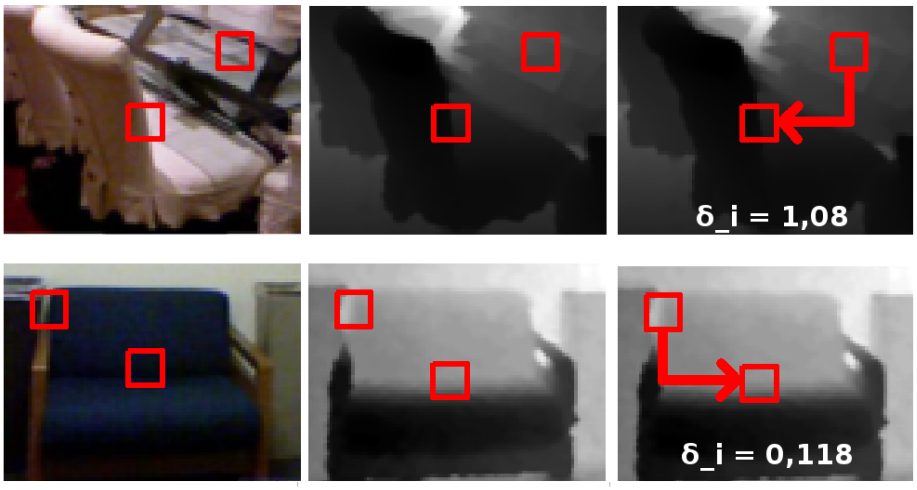
Still, the point cloud descriptors may come at substantial computational cost if extracted over points in a dense grid. They have been assessed either in the setting of image classification, where a global image descriptor is constructed, or in conjunction with interest point detectors, where a shortlist of positions is provided from the point detector.

We propose efficient, densely computable depth features which complement

Histogram-of-Depth Gradient features with surface-based information. Instead of relying on the few, and variable, positions where the depth signal changes, as in HOG/HOD, we describe our signal within a window in terms of the depth displacements with respect to its center.

Given a region/bounding box $\mathbf{x}$ in a depth image, and a cellsize s, we first compute a depth summary descriptor of the image region $\mathbf{x}$. This depth descriptor, $\widetilde{\mathbf{x}}$, is a depth field where each pixel describes a $s \times s$ cell in $\mathbf{x}$. We compute $\widetilde{\mathbf{x}}$ by scanning $\mathbf{x}$ with a sliding window of size $s \times s$, and summarizing each cell using the mean value of depth intensities in the cell. Given the descriptor $\widetilde{\mathbf{x}}$, we compute the displacement of each pixel p_i in $\widetilde{\mathbf{x}}$ from the center pixel p_0. This displacement can be expressed as, $\delta_i = depth^{p_i} - depth^{p_0}$.

Since the values of δ_i can fluctuate widely, we quantize its value into a set of N displacement bins, using a hard quantization function: $q(\delta) : R \rightarrow R^N$. Since δ_i can assume both positive and negative values, we have symmetric displacement bins corresponding to the positive and negative values. Thus δ_i is expressed as a sparse-indicator-feature vector of size N. This indicator feature vector has exactly one non-zero entry, corresponding to the displacement bin which consumed δ_i. The displacement feature of $\mathbf{x}$ is the concatenation of these sparse-indicator-feature vectors, capturing the depth variation in each cell with respect to the center of the image region. Fig. 3.5 illustrates an example of computation of δ_i.



(a) RGB (b) Depth (c) Absolute displacement

Figure 3.5: Representation of the computation of δ_i for two objects of the class chair. In the first column (a), there are two RGB images with indicated the center of the filter region and the sliding window, in the second column (b) the respective depth maps and in the third the computation of δ_i (distance in meters).

We observed that hard quantization yields better results than soft quantization

schemes. Another advantage of hard quantization is that it results in sparse features, which we exploit for achieving faster convolutions. The number of displacement bins, N, was set with cross-validation, while the interval endpoints consist of a geometric sequence in log space. The cellsize s is chosen so as to ensure that the number of displacement cells in $\mathbf{x}$ is equal to the number of HOG cells in $\mathbf{x}$. While HOG features measure the directional derivatives of the depth field, the displacement features capture the depth variations in a region with respect to the center.

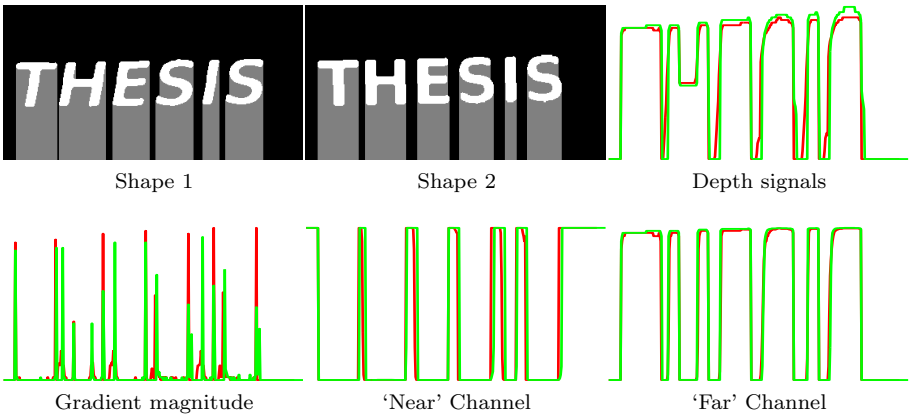
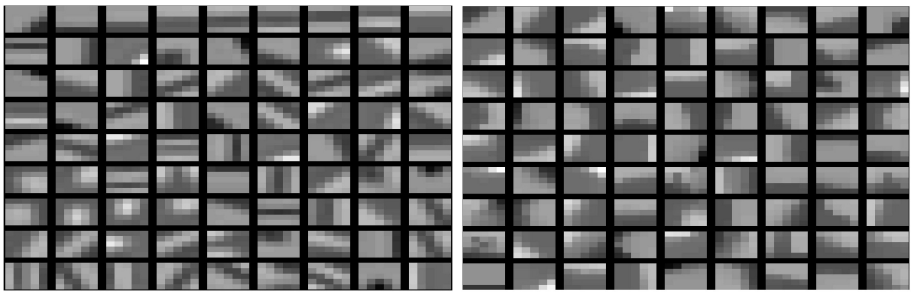


Figure 3.6: Illustration of Gradient-based versus our Displacement-based features for two synthetic 'flatland', two-dimensional, shapes: considering that the shapes are seen from above, their respective 'depth' signals would correspond to the functions shown on the top right. Gradient-based features (bottom-left) underlying HOG are sensitive to shape variation, while our displacement-based features (bottom-middle and right) encode relative depth, which is similar for both shapes on a larger extent of their domains.

This processing would not be meaningful for RGB images, as there the signal's value depends on a host of factors, including color, illumination and shading. However, for depth it provides us with valuable information whether the neighborhood of a point lies closer to the camera or not.

A feature extraction pipeline that relies on signal gradients (bottom left) will either consider their, non-overlapping, gradient signals distinct, or resort to smoothing to make them comparable. Instead, our displacement-based features (bottom middle and right) quantize the signal's domain into regions that are 'far' or 'near' from the center of the signal in depth, delivering features that exhibit a smaller amount of intra-class variation. In a certain sense both our displacement-based and the gradient-based representations contain the same information, but in different 'formats': intuitively, our displacement features are more appropriate when the boundaries are variable, but the depth variability is consistent, while the HOD features could be more appropriate for well localized boundaries but potentially a larger breadth of depth differences.



(a) (b)

Figure 3.7: Two dictionaries learned with 5×5 patches. Specifically, (a) is the default dictionary used in [76] that is trained and used in RGB images. (b) is a dictionary that we trained based on patches from depth images of [83] using [64].

3.2.3 Sparse coding

The introduction of Histograms of Sparse Codes for detection [76] proved that achieving a richer representations of objects than the one provided by HOG is possible.

Sparse Coding is increasingly used in several fields of Computer Vision including learning feature representation [5], tracking [50], classification [16, 17, 96] and detection [55].

Sparse coding is a representation of objects that captures higher-level features in the data, for instance in Fig. 3.7. Sparse coding learns basis function from unlabeled data to achieve the higher-level features. Unlike some other unsupervised learning techniques such as PCA, sparse coding can be applied to learning overcomplete basis sets, in which the number of bases is greater than the input dimension. Sparse coding also models inhibition between the bases by sparsifying their activations.

Ren et al. in [76] compute sparse codes from data driven dictionaries, and then form local histograms. They use DPM supervision for initializing their optimization and they present an improvement over DPM results.

Our contribution lies in augmenting the initial framework to include depth maps. We train a dictionary using [64] and modify the framework to extract sparse codes in both RGB and depth images. We, also, explore whether a better initial localization of the parts provided by our novel displacement features affects significantly the outcome of the sparse-codes.

Chapter 4

Experimental Results

In this Chapter, we describe our experimental setup, alongside with the quantitative and qualitative results of our proposed techniques. Firstly, we introduce the evaluation method we use, though.

4.1 Framework for quantitative evaluation of the detector

In this Section, we describe the quantitative evaluation method that we use in the following paragraphs. Concretely, we refer to the precision-recall curves and the average precision metric.

4.1.1 Precision-recall curves

Precision-recall (PR) curves are a common evaluation technique in information retrieval and classification problems. In order to define those curves, we define the following terms first:

- **True positive**: An object is successfully detected in the correct location.

- **False positive**: An object is declared by the system in a location where no such object actually exists.

- **False negative**: An object is unsuccessfully declared as background.

Precision is the fraction of the detections that are true positives, while **recall** is the fraction of true positives that are detected. In a mathematical formulation:

$$Precision = \frac{tp}{tp + fp} \tag{4.1}$$

$$Recall = \frac{tp}{tp + fn} \tag{4.2}$$

where tp = True positive, fp = False positive and fn = False negative.

Ideally a detector has neither false positives nor false negatives, in other words it has a high precision and a high recall. However, these two measures are competitive, in the sense that there is a trade-off between high precision and high recall. To explain this intuitively, high recall means that the system detects all positives, but it may also classify as positives a lot of other regions, resulting in low precision. High precision on the other hand, means that the system is more selective in indicating something as positive, therefore some positives may be ignored, resulting in low recall.

4.1.2 Average precision with VOC

The development kit ([26]) used to compute the average precision score for the PASCAL VOC challenges has become the standard benchmark for detection results. The procedure followed is:

The PR curve is first replaced by its tightest monotonically decreasing upper bound. This transformation removes the characteristic 'sawtooth' shape found is many PR curves. The average precision (AP) is computed by sampling precision values from the PR curve at 11 points ($i/10$, $i \in \{0, 10\}$), and averaging those values. Starting in 2010, this sampling method was replaced by the area under the (upper-bounded) PR curve or put into a mathematical formulation

$$AP = \int_0^\infty p(r)\,\mathrm{d}r \tag{4.3}$$

where p(r) is the precision value as a function of recall value. This computation of average precision is more accurate and meaningful. We used both methods to ensure that our results are consistent and we confirm that with both methods the outcomes that we present below remain the same.

A detection is considered correct (true positive) if $\frac{area(BB \bigcup GT)}{area(BB \bigcap GT)} > 0.5$ where BB is the bounding box of the detection and GT is the ground truth bounding box of the class. Only one detection can be evaluated correct for a given ground truth box, with the rest considered false positives.

4.2 Experimental setup

4.2.1 Software implementation

We adapted the DPM of [38] using the default number of components (3), and the default parameters. For our contributions, which are divided in geometric based and feature based, we note the following implementation details:

In our geometric extensions:

Our augmented data was rendered by simulating the effect of the camera moving to the left and to the top from the original viewpoint. DPM pipeline augments the training data with laterally flipped images, so we do not render views where the camera moves to the right. We do not render views with the camera moving down, for reasons of efficiency. The rendering of novel views is done offline,

and takes approximately 2 seconds for each pair of images (RGB+Depth) on a single-core machine. The adaptation of the bounding boxes in each class is completed at the beginning of the DPM pipeline. For each training example, we use 2 new images per direction, therefore for each original positive example, we get a sum of 5 examples from different viewpoints.

Due to the distortions mentioned in [25], the augmented data are not suitable for the increased resolution of the parts' training. Therefore, the augmented training set is used only in the few initial iterations of the latent variable updates, which means that this method improves detection performance without increasing the training time significantly.

In our feature based extensions:

The features that we extracted were HOG RGBD and displacement features. The number of different bins in the displacement features was 15, i.e. displacement features are expressed as vector of size 15. The computation of the displacement features is implemented efficiently during the convolution with filters, exploiting the sparsity of the displacement features. The computation of the features and the convolution is approximately 33% faster for the displacement features than the fastest implementation of HOG.

4.2.2 Datasets

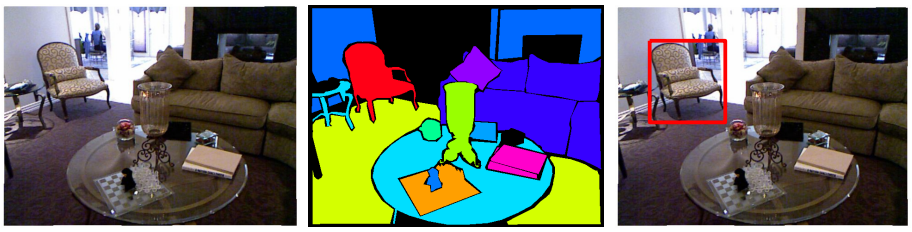
We use two different datasets with RGBD images: the NYU Depth v2 dataset ([83]) and the Berkeley 3-D Object Dataset ([52]). Both datasets constitute challenging benchmarks for the task of object detection ([6, 39, 40, 52, 82]).

The NYU Depth dataset contains 1449 RGBD images consisting of 464 different indoor scenes across 26 scene classes. The scenes contain multiple instances of objects. In total, there are 35064 distinct objects spanning 894 different classes. The NYU dataset comes with train-test splits, and pixel-wise object labels, like in Fig. 4.1. We use these pixel-wise labels to generate tight bounding box ground truth annotations for 5 object class categories, namely bed, chair, monitor + television (M.+TV), sofa, and table, as in [82].

The Berkeley Dataset includes 849 RGBD images. The database includes PASCAL like annotations and bounding boxes for all images, as well as 6 different train/test splits. We selected 5 classes, namely bowl, chair, mouse, pillow and phone, and used all 6 different splits for every class.

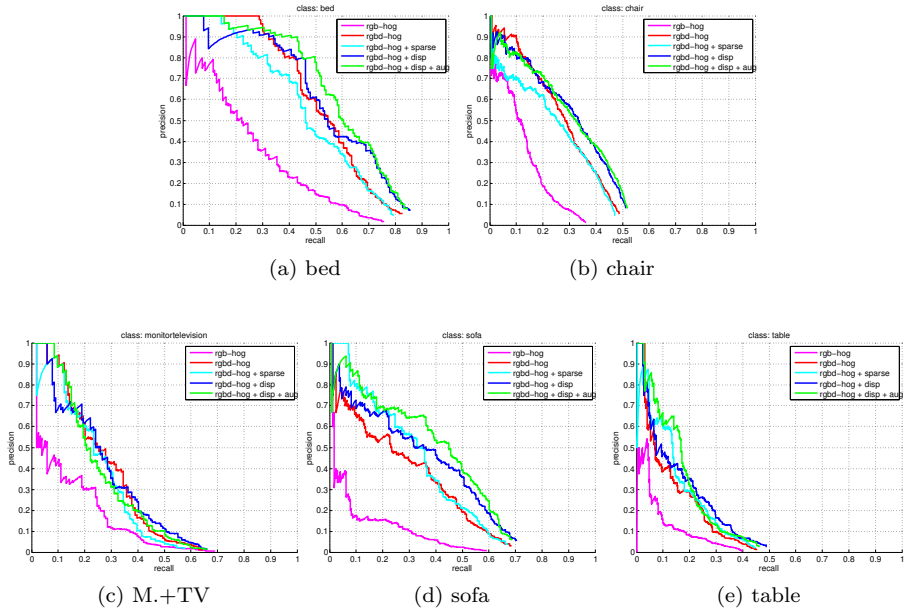
4.3 Experiment on NYU Dataset

The experiment was conducted on the NYU Dataset with the configuration mentioned in the Paragraph 4.2.1.



(a) Initial image (b) Pixel level labels (c) Bounding box of chair

Figure 4.1: Extraction of ground truth annotations. In (a) is the initial RGB image, in (b) the pixel level labels for all the pixels and in (c) the extracted tight bounding box of the chair from the image labels.



(a) bed (b) chair

(c) M.+TV (d) sofa (e) table

Figure 4.2: Precision recall curves for the detection task for the classes (a)bed, (b)chair, (c)M.+TV, (d)sofa and (e)table.

Method	Bed	Chair	M.+TV	Sofa	Table	Avg.
gdpm [82]	0.3339	0.1372	0.0928	0.1104	0.0405	0.1430
rgb-hog	0.2929	0.1635	0.1813	0.1380	0.07324	0.1698
rgbd-hog	0.5446	0.2760	0.2728	0.2734	0.1299	0.2993
rgbd-hog + sparse [76]	0.4907	0.2920	0.2765	0.3255	0.1524	0.3074
rgbd-hog + disp	0.5665	0.3003	0.2715	0.3422	0.1561	0.3273
rgbd-hog + disp + aug	0.6069	0.3025	0.2596	0.3911	0.1720	0.3464
improvement over rgbd-hog	0.0623	0.0265	-0.0132	0.1177	0.0421	0.0471

Table 4.1: Average Precision for the Object Detection Task on the NYU Dataset. The first row is dedicated to a previously published work. rgb-hog relates to the use of only color-based HOG features as in[29], rgbd-hog refers to the use of both color and depth HOG together. rgbd-hog+sparse refers to our proposed sparse coding for RGBD images. disp indicates the introduction of our displacement features, aug indicates the use of augmented data. The last row indicates the overall improvement we achieved with displacement features and training data augmentation. We observe that there is no improvement only in the class of monitor+TV.

In Table 4.1, we present the qualitative evaluation of our system based on the average precision for which the formula with the area is used. We report the Average Precision values for different methods, alongside the previously published state of the art result of [82]. We also present the precision recall curves of the detectors in Fig. 4.2. There are a number of significant remarks that can be extracted from this qualitative assessment:

- The extension of the sparse coding technique for RGBD data improves the HOG RGBD results. However, we observed that if we apply the sparse coding after our displacement features, the result improves marginally from the one with HOG RGBD. Additionally, the computational time required to train the model is significant, therefore we do not include this in the reference table.
- gdpm does not seem to perform better than HOG RGBD, thus we cannot compare with our improved results. We cannot reproduce the results, since the code is not publicly available.
- Displacement features demonstrate an improvement to 4 out of 5 classes with the 5^{th} one (monitor+TV) remaining almost constant. The most compelling improvement is in the class of sofa with an improvement of 6,90% average AP with an overall improvement of 2,80%.
- The training data augmentation technique also provided a noticeable improvement in our system. In 4 out of classes, there is a boost in the average precision with the overall improvement of 1,91% average AP over the training with the displacement features.

Apart from the experiments in Table 4.1, we also have the outcomes of using the 3D geometric split instead of the default aspect ratio. The experimentation

results are presented in Table 4.2. We observe that the use of a geometric split provides an improvement of 0.42% AP, indicating that our proposal improves the baseline. However, the improvement is not statistically significant to be considered for the rest of the experiments.

We additionally present the outcome of our experimentation with the augmented

Method	Bed	Chair	M.+TV	Sofa	Table	Avg.
rgbd-hog + disp	0.5665	0.3003	0.2715	0.3422	0.1561	0.3273
rgbd-hog + disp + geom-split	0.5875	0.2767	0.2215	0.3742	0.1974	0.3315
improvement over displ	0.0210	-0.0236	-0.0500	0.0320	0.0413	0.0042

Table 4.2: Supplemental Average Precision for the Object Detection Task with 3D geometric split of the training data. The first row refers to the displacement features results as mentioned in Table 4.1, while the second introduces the results for the geometric split. It is noticeable that there is considerable improvement in 3 classes and deterioration in the remaining 2, indicating that the geometric split could provide superior results, but it is not statistically significant to consider for the rest of the experiments.

pairwise term with depth deformation. In Table 4.3, we refer to the experiments conducted on the pairwise term. We deduct that the deformation of depth is a significant extension that provides additional information to our detector, but the improvement is not statistically significant, therefore due to the computational overhead we do not consider it in the rest experiments.

Method	Bed	Chair	M.+TV	Sofa	Table	Avg.
rgbd-hog + disp	0.5665	0.3003	0.2715	0.3422	0.1561	0.3273
rgbd-hog + disp + pair-depth	0.5579	0.2985	0.2766	0.3554	0.17143	0.332
improvement over displ	-0.0086	-0.0018	0.0051	0.0132	0.0154	0.0047

Table 4.3: Supplemental Average Precision for the Object Detection Task with depth deformations. The first row refers to the displacement features results as mentioned in Table 4.1, while the second introduces the results for pairwise term with depth information. It is noticeable that there is considerable improvement in 3 classes and slight deterioration in the remaining 2, indicating that the pairwise term can provide additional information to our detector.

4.3.1 Qualitative results of the experiment

Apart from the PR curves and the mean AP that represent the quantitative outcome of our contributions, we also wanted to achieve a qualitative representation. We exhibit in Fig. 4.3 the top detections of the system with displacement features and training data augmentation.

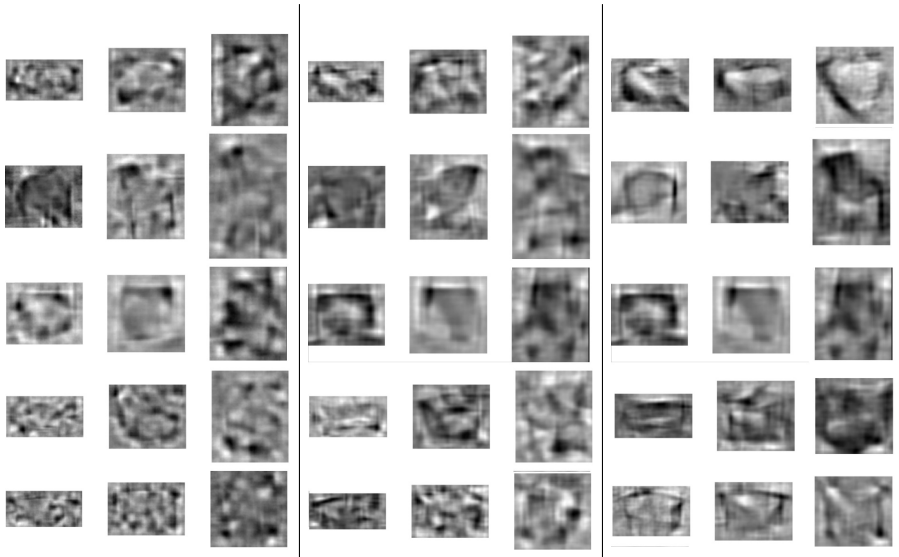
Figure 4.3: The top detections as scored by the detector with displacement features and training data augmentation. Each row represents the top detections in one class. From top to bottom: bed, chair, M.+TV, sofa, table. The yellow boxes are the ground-truth bounding boxes, while the green ones indicate the true positives. There is only one false positive in the class of sofa, which is a mistaken annotation. We note that the detections of our proposed system are accurate and most of them well localized.

Additionally, we use two popular analysis tools [49, 91] to acquire qualitative information about our contributions.

4.3.1.1 Visualization of what each detector expects to see

Using the HOGGLES visualization of [91], we visualize the weights learnt by each of the detectors. This algorithm visualizes the feature space of object detectors. It allows us to reveal the representation that the detector expects in order to make a detection, therefore it provides an insight into why our detectors fail in some regions.

We visualize in Fig. 4.4 the weights from the HOG (RGB) features (initial), the HOG RGBD and our final system with displacement features and the data augmentation. The difference between what the detectors expect to see is evident with the last columns of our contributions to be clearer. Particularly, we can recognize the objects that our proposed system visualizes in each case.



(a) HOG RGB (b) HOG RGBD (c) HOG RGBD + DISP + Augm

Figure 4.4: A hoggles [91] visualization of the model weights learnt by our 3 component detector. In each row, the first triple of images are from weights with HOG (RGB), the second triple are from HOG RGBD and the third with displacement features+data augmentation. The rows visualize respectively the class of bed, chair, monitor+TV, sofa, table. The difference between what the detectors expect to see is evident with the last columns of our contributions to be clearer. Especially, in the classes of chair and monitor+TV in the last three columns, the objects are clear.

4.3.1.2 Detector errors

Using large datasets like NYU and powerful detectors like DPM produces top performing results, however it makes it more complicated to compare qualitatively why one detector outperforms another detector. The authors of [49] provide an analysis framework to investigate the performance of the detector in several metrics.

Using [49], we analyzed our detector mistakes. We present the top false positives of the baseline HOG RGBD detector with our proposed detector that includes the displacement features and the training data augmentation. It can be observed that our detector makes fewer localization mistakes, while just the HOG RGBD detector makes several obvious mistakes, such as firing on random objects, and parts of objects.

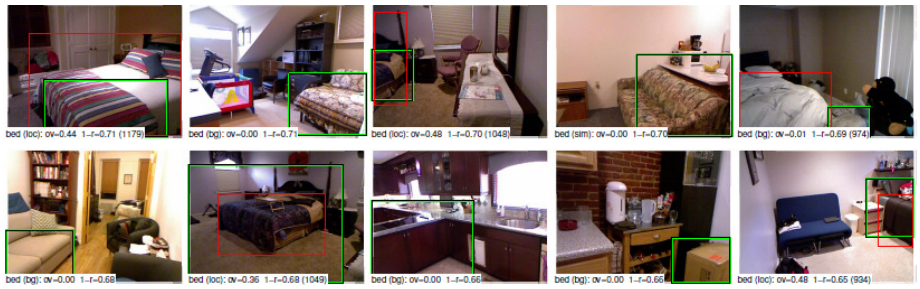
4.4 Experiment with data augmentation

This experiment was conducted on the NYU Dataset with the configuration of the Paragraph 4.2.1 with one difference. The only differentiating factor was that in this experiment, there were no parts in the last steps of training, but only root filters.

This allowed us to experiment with the training data augmentation. Since there was only root training, we applied the data augmentation in all training phases and it can be confirmed that augmenting the data in all phases provides additional improvement in the training. The quantitative results from this experimentation are summarized in Table 4.4. The precision-recall curves can be found in Fig. 4.8.

Method	Bed	Chair	M.+TV	Sofa	Table	Avg.
rgb-hog	0.2337	0.1335	0.1339	0.1094	0.0376	0.1296
rgbd-hog	0.4660	0.2773	0.2480	0.2295	0.1430	0.2628
rgbd-hog + sparse [76]	0.4835	0.2435	0.2769	0.2869	0.1726	0.2927
rgbd-hog + disp	0.5178	0.2771	0.2591	0.3440	0.1683	0.3133
rgbd-hog + disp + aug	0.5406	0.2919	0.2583	0.3470	0.1653	0.3206
rgbd-hog + disp + aug-all-phases	0.5639	0.2841	0.2732	0.3502	0.198	0.3339
improvement over rgbd-hog	0.0979	0.0068	0.0252	0.1207	0.055	0.0711

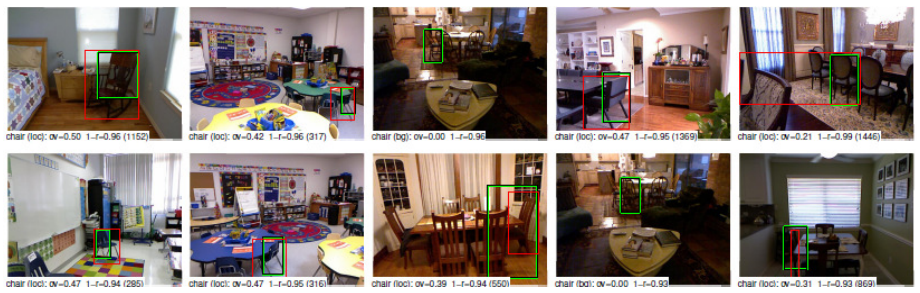
Table 4.4: Average Precision for the Object Detection Task on the NYU Dataset for root training. The results are presented in a similar way as in Table 4.1. The aug-all-phases declares that the training data augmentation technique was applied in all phases of the root training, while the aug only in the initial iterations of the latent variable updates. We observe that using the augmented data in all phases, provides a noticeable boost in the detector.



(a) (rgbd) bed

(b) (our) bed

(c) (rgbd) chair

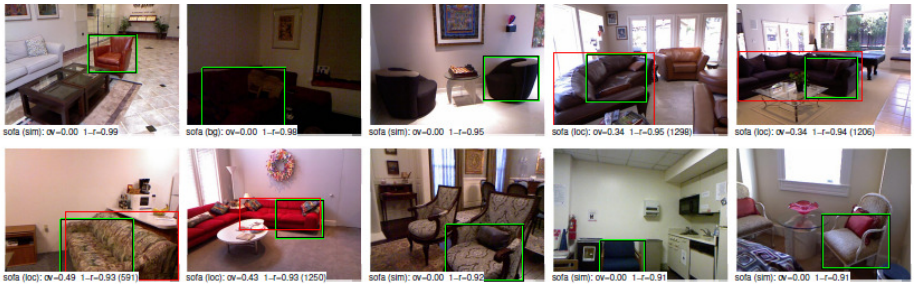


(d) (our) chair

Figure 4.5: Top false positives of the HOG RGBD detector and our proposed system with displacement features and training data augmentation. The parenthesis in front of each class indicates the system with those false positives. The green bounding box indicates the detector's 'best guess' while the red indicates the ground-truth bounding box.

(a) (rgbd) M.+TV

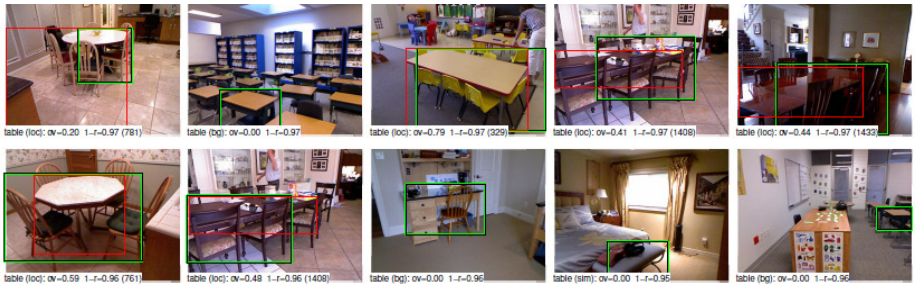
(b) (our) M.+TV



(c) (rgbd) sofa

(d) (our) sofa

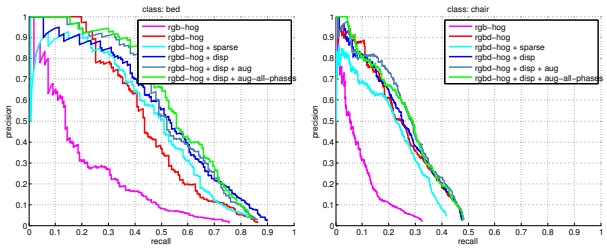
Figure 4.6: Continuation of Fig. 4.5.



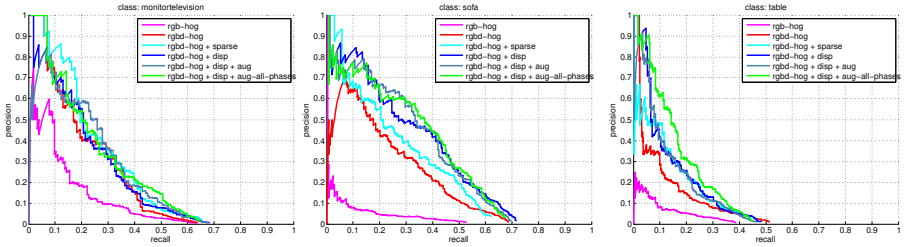
(a) (rgbd) table

(b) (our) table

Figure 4.7: Continuation of Fig. 4.5.



(a) bed (b) chair



(c) M.+TV (d) sofa (e) table

Figure 4.8: Precision recall curves for the detection task of root training for the classes (a)bed, (b)chair, (c)monitor+TV, (d)sofa and (e)table.

4.5 Experiment on Berkeley Dataset

We conducted the experiment on Berkeley Dataset with the default configuration. We present the quantitative results of the detector in Table 4.5. Note that the formula with the area is used for the average precision.

It can be deducted that the displacement features improve the detection results.

Method	Bowl	Chair	Mouse	Pillow	Phone	Avg.
rgb-hog	0.3703	0.1983	0.2721	0.0491	0.2403	0.226
rgbd-hog	0.4781	0.4107	0.3985	0.0621	0.2714	0.3242
rgbd-hog + disp	0.5041	0.4008	0.44	0.0675	0.2785	0.3382
improvement over rgbd-hog	0.026	-0.0099	0.0415	0.0054	0.0071	0.014

Table 4.5: Average Precision for the Object Detection Task on the Berkeley 3-D Object Dataset. The results are presented in a similar way as in the Table 4.1.

A significant remark about this dataset is that there is a partial misalignment between the RGB and the depth images in the Berkeley Dataset, e.g. in Fig. 4.9. Therefore, we could not perform the training data augmentation since the misalignment resulted in significant distortions in the new images. This misalignment also forces the detector to fire in different location for the RGB and the depth image, which can explain the smaller improvement of displacement features in the average precision when compared to the previous experiments.

4.6 Summary

In this chapter we have demonstrated 3 diverse experiments with 2 different Datasets. We proved through both the quantitative and the qualitative results that our contributions improve the detection results and outperform all current algorithms for object detection with RGBD images.

(a) RGB (b) depth

Figure 4.9: Two images with apparent misalignment between RGB and depth image. The white lines are the edges of the depth image and where they are represented in the RGB image.

Chapter 5

Conclusion and future work

5.1 Conclusion

In this work, we presented a popular framework for object detection and we proposed strategies to improve object detection in RGBD images. The main intention was to increase the robustness of object detectors and ultimately to move towards 3D recognition.

Our contributions are divided into geometric based and feature based. The geometric based include: (i) training data augmentation, (iii) augmented pairwise term, (iii) 3D component initialization. The feature based include: (i) our novel displacement features, (ii) sparse coding for RGBD images.

In more details, training data augmentation relies on rendering novel views from 2D images. This augmentation can be beneficial in several applications, especially when there are not sufficient original data. Augmented pairwise term refers to the incorporation of depth displacement in the deformation cost of the parts. 3D component initialization provides an alternative geometric split for the training groups required for the initialization of the different mixture models.

Our displacement features are local, depth-based descriptors that provide statistical surface-based information in our detector. Sparse coding for RGBD images refers to our extension of the initial framework to include depth images and extract sparse codes in the depth field.

Displacement features and training data augmentation systematically improve detection performance on two popular benchmark RGBD datasets.

The code of our contributions will be publicly available under the open source license of 'Apache License, Version 2.0'. This will allow the researchers from the Computer Vision community to study and modify the code. The repository is in [3].

5.2 Future Work

Our contributions systematically improve the detector's performance, but further improvement can be achieved with the extension of the lines of work presented in the thesis.

One line of research could be the further exploitation of rendering techniques and data augmentation. The elimination or reduction of distortions introduced in the rendered images seems to be essential in order to achieve a further improvement. Comparing our inpainting techniques with the extensive work in the bibliography can provide a boost in this direction. Additionally, increasing the variation of the azimuth or the elevation is a potential line of research. The goal of achieving viewpoint invariant features is an alternative line of research that can provide significant improvement in the performance.

Another line of research is using some segmentation masks before the extraction of displacement features. This will allow a better statistical description of the objects' surface rather than include background objects.

Another direction is to explore further the pairwise term and enrich it further with the use of the depth information that is available. This can be combined with some extension of the DPM framework that will incorporate the convolution in 3D instead of 2D.

Bibliography

[1] Microsoft corp., http://www.xbox.com/en-us/kinect.

[2] Primesense corp., http://www.primesense.com/.

[3] https://github.com/grigorisg9gr/detection_for_rgbd.

[4] Gerald J Agin and Thomas O Binford. Computer description of curved objects. *Computers, IEEE Transactions on*, 100(4):439–449, 1976.

[5] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[6] Haider Ali, Faisal Shafait, Eirini Giannakidou, Athena Vakali, Nadia Figueroa, Theodoros Varvadoukas, and Nikolaos Mavridis. Contextual object category recognition for rgb-d scene labeling. *Robotics and Autonomous Systems*, 62(2):241–256, 2014.

[7] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer, 2006.

[8] Irving Biederman. Human image understanding: Recent research and a theory. *Computer vision, graphics, and image processing*, 32(1):29–73, 1985.

[9] Thomas O Binford. Visual perception by computer. In *IEEE conference on Systems and Control*, volume 261, page 262, 1971.

[10] Michael J Black and Allan D Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1998.

[11] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In *ISER*, 2012.

[12] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.

[13] J-Y Bouguet and Pietro Perona. 3d photography on your desk. In *Computer Vision, 1998. Sixth International Conference on*, pages 43–50. IEEE, 1998.

[14] Octavia I Camps, Chien-Yuan Huang, and Tapas Kanungo. Hierarchical organization of appearance-based parts and relations for object recognition. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 685–691. IEEE, 1998.

[15] Frédéric Cao, Yann Gousseau, Simon Masnou, Patrick Pérez, et al. Geometrically guided exemplar-based inpainting. 2009.

[16] João Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *Computer Vision–ECCV 2012*, pages 430–443. Springer, 2012.

[17] Yu-Tseh Chi, Mohsen Ali, Muhammad Rushdi, and Jeffrey Ho. Affine-constrained group sparse coding and its application to image-based.

[18] David T. Clemens and David W. Jacobs. Space and time bounds on indexing 3d models from 2d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1007–1017, 1991.

[19] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9):1200–1212, 2004.

[20] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 2, pages 886–893, 2005.

[21] Dima Damen, Pished Bunnun, Andrew Calway, and Walterio W Mayol-Cuevas. Real-time learning and detection of 3d texture-less objects: A scalable approach. In *BMVC*, pages 1–12, 2012.

[22] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1814–1821. IEEE, 2013.

[23] Sven Dickinson. The evolution of object categorization and the challenge of image abstraction. *Cambridge University Press*, pages 1–37, 2009.

[24] Santosh K Divvala, Alexei A Efros, and Martial Hebert. How important are "deformable parts" in the deformable parts model? In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 31–40. Springer, 2012.

[25] Luat Do, Sveta Zinger, et al. Quality improving techniques for free-viewpoint dibr. In *IS&T/SPIE Electronic Imaging*, pages 75240I–75240I. International Society for Optics and Photonics, 2010.

[26] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. The pascal visual object classes challenge 2007 (voc 2007) results (2007). In *URL http://www. pascal-network. org/challenges/VOC/voc2007/workshop/index.html*, 2008.

[27] Christoph Fehn. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In *Electronic Imaging 2004*, pages 93–104. International Society for Optics and Photonics, 2004.

[28] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4):594–611, 2006.

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

[30] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 2241–2248. IEEE, 2010.

[31] Robert Fergus, Pietro Perona, and Andrew Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303, 2007.

[32] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International Journal of Computer Vision*, 67(2):159–188, 2006.

[33] Graham Finlayson, Clément Fredembach, and Mark S Drew. Detecting illumination in images. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[34] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[35] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *ECCV*, 2004.

[36] Josselin Gautier, Olivier Le Meur, and Christine Guillemot. Depth-based image completion for view synthesis. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4. IEEE, 2011.

[37] Pascal Getreuer. Total variation inpainting using split Bregman. *Image Processing On Line*, 2012.

[38] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

[39] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 564–571. IEEE, 2013.

[40] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision–ECCV 2014*, pages 345–360. Springer, 2014.

[41] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. 2013.

[42] John A Hartigan. Clustering algorithms. 1975.

[43] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[44] Tal Hassner. Viewing real-world faces in 3d. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3607–3614. IEEE, 2013.

[45] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Depth map inpainting under a second-order smoothness prior. In *Image Analysis*, pages 555–566. Springer, 2013.

[46] Aaron Hertzmann and Steven M Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1254–1264, 2005.

[47] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):876–888, 2012.

[48] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 858–865. IEEE, 2011.

[49] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV*, pages 340–353. Springer, 2012.

[50] Zhibin Hong, Xue Mei, Danil Prokhorov, and Dacheng Tao. Tracking via robust multi-task multi-view joint sparse representation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 649–656. IEEE, 2013.

[51] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970.

[52] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013.

[53] Andrew Edie Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999.

[54] Müller Karsten, Smolic Aljoscha, Dix Kristina, Merkle Philipp, Kauff Peter, Wiegand Thomas, et al. View synthesis for advanced 3d video systems. *EURASIP Journal on Image and Video Processing*, 2009.

[55] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010.

[56] Vangelis Koukis, Constantinos Venetsanopoulos, and Nectarios Koziris. Synnefo: A complete cloud stack over ganeti. USENIX.

[57] Vangelis Koukis, Constantinos Venetsanopoulos, and Nectarios Koziris. okeanos: Building a cloud, cluster by cluster. *IEEE internet computing*, 17(3):67–71, 2013.

[58] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[59] J-F Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Estimating natural illumination from a single outdoor image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 183–190. IEEE, 2009.

[60] Carlos Leung and Brian C Lovell. 3d reconstruction through segmentation of multi-view image sequences. In *Workshop on Digital Image Computing*, volume 1, pages 87–92. Australian Pattern Recognition Society, 2003.

[61] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, 2013.

[62] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004.

[63] David G Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395, 1987.

[64] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[65] Jitendra Malik and Dror Maydan. Recovering three-dimensional shape from a single image of curved objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(6):555–566, 1989.

[66] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):696–710, 1997.

[67] Tim Morris. *Computer vision and image processing*. Palgrave Macmillan, 2004.

[68] Joseph L Mundy. Object recognition in the geometric era: A retrospective. In *Toward category-level object recognition*, pages 3–28. Springer, 2006.

[69] Ha T Nguyen and Minh N Do. Image-based rendering with depth information using the propagation algorithm. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc*, volume 2, pages 589–592, 2005.

[70] Kohtaro Ohba and Katsushi Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(9):1043–1047, 1997.

[71] George Papandreou, Petros Maragos, and Anil Kokaram. Image inpainting with a wavelet domain hidden markov tree model. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 773–776. IEEE, 2008.

[72] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.

[73] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[74] Shantanu D Rane, Guillermo Sapiro, and Marcelo Bertalmio. Structure and texture filling-in of missing image blocks in wireless transmission and compression applications. *Image Processing, IEEE Transactions on*, 12(3):296–303, 2003.

[75] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Image-based synthesis and re-synthesis of viewpoints guided by 3d models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. to appear.

[76] Xiaofeng Ren and Deva Ramanan. Histograms of sparse codes for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3246–3253. IEEE, 2013.

[77] Lawrence Gilman Roberts. *MACHINE PERCEPTION OF THREE-DIMENSIONAL solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[78] Daniel Ruijters and Svitlana Zinger. Iglance: transmission to medical high definition autostereoscopic displays. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009*, pages 1–4. IEEE, 2009.

[79] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, 2009.

[80] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–195. IEEE, 2003.

[81] Jianhong Shen and Tony F Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.

[82] Abhinav Shrivastava and Abhinav Gupta. Building part-based object detectors via 3d geometry. In *ICCV*, 2013.

[83] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012.

[84] Richard Socher, Brody Huval, Bharath Putta Bath, Christopher D. Manning, and Andrew Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *NIPS*, 2012.

[85] Luciano Spinello and Kai Oliver Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011.

[86] Wenxiu Sun, Lingfeng Xu, Oscar C Au, Sung Him Chui, and Chun Wing Kwok. An overview of free view-point depth-image-based rendering (dibr). In *APSIPA Annual Summit and Conference*, 2010.

[87] Shuai Tang, Xiaoyu Wang, Xutao Lv, Tony X Han, James Keller, Zhihai He, Marjorie Skubic, and Shihong Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Computer Vision–ACCV 2012*, pages 525–538. Springer, 2013.

[88] Siyu Tang, Mykhaylo Andriluka, and Bernt Schiele. Detection and tracking of occluded people. *International Journal of Computer Vision*, pages 1–12, 2012.

[89] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2642–2649. IEEE, 2013.

[90] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[91] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *ICCV*, 2013.

[92] Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

[93] Walter Whiteley. The machine interpretation of line drawings, 1987.

[94] Alexander Wong and Jeff Orchard. A nonlocal-means approach to exemplar-based inpainting. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2600–2603. IEEE, 2008.

[95] Jianixn Wu and James M Rehg. Where am i: Place instance and category recognition using spatial pact. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[96] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[97] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[98] Edmund Shanming Ye. Object detection in rgb-d indoor scenes. Master's thesis, EECS Department, University of California, Berkeley, Jan 2013.

[99] Cha Zhang. Multiview imaging and 3dtv. *IEEE Signal Processing Magazine*, 1053(5888/07), 2007.

[100] Jian Zhang, Chen Kan, Alexander G. Schwing, and Raquel Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *ICCV*, 2013.

[101] B. Zheng, Y. Zhao, Joey C. Yu, K. Ikeuchi, and S.-C. Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *CVPR*, 2013.

[102] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.