Chapter 10

Image understanding – HOGs, RFs

10.5.3 Histograms of oriented gradients—HOG

- feature extraction chain that describes local object appearance and shape by constructing histograms of local intensity gradients
- takes relatively coarse spatial context into consideration and employs a classifier to detect objects of interest
- locally normalized **histograms of oriented gradients** or **HOGs** build on earlier concepts like edge-oriented histograms [Freeman and Roth, 1995] and SIFT descriptors
- HOGs utilize a classifier for object localization and recognition
- linear SVMs were used in the original application (upright human detector [Dalal and Triggs, 2005])

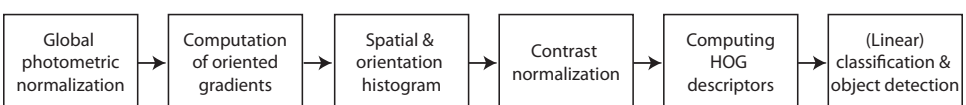


Figure 10.1: HOG-description chain for object detection and localization. The image window that is subjected to HOG description is covered by overlapping blocks, in which HOG features are computed and subsequently sent to a classifier.

- image region (**window**) is divided into smaller subregions (**cells**)
- local 1D histogram of gradient directions or edge orientations is constructed over all pixels of the cell
- configurations of several cells form **blocks** (Figure 10.2).

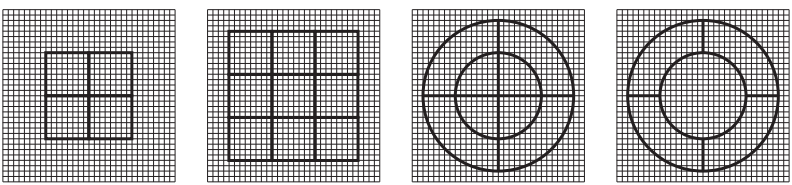
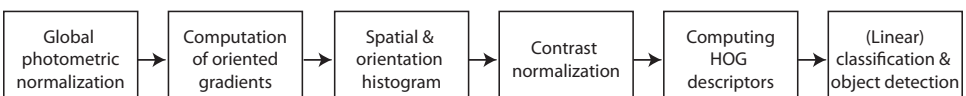


Figure 10.2: Examples of rectangular and circular blocks of cells that may be used in HOG descriptors. (a-d) The small squares corresponds to image pixels, each outlined area of pixels depicts a cell, and the respective configurations of cells give examples of block definitions.

- specific configuration of adjacent image pixels forms a *cell*, a specific configuration of cells forms a *block*, and a number of blocks (possibly overlapping) can be used to cover the image *window* using a specific block/overlap grid
- $\rightarrow$ defining the pixel/cell/block configurations and the block/window grid is part of the implementation

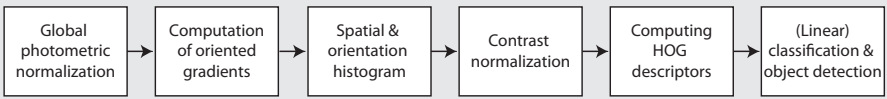
- contrast normalization can be employed to gain insensitivity to illumination, shadowing, and other photometric transformations (Figure 10.1)
- combined histogram entries from an overlapping dense grid of local HOG descriptors form the final HOG description feature vector associated with the window
- this feature vector is used for classification.
- HOG representation captures edge/gradient structure corresponding to the underlying local shape with controllable insensitivity to rotation and translation



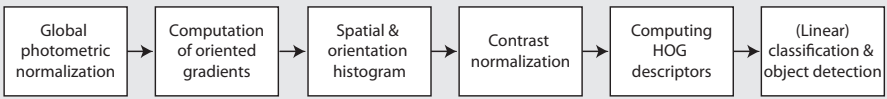
Algorithm 10.1: HOG object detection and localization (Figure 10.1)

1. **Determination of window, cell, block sizes/shapes and overlap.** Based on the image detection task at hand, the size and shape of the image window must be determined (64×128 windows were used for pedestrian detection as shown in Figure 10.3; a sufficient margin around the object of interest should be included in the window—a 16-pixel margin was appropriate in the pedestrian detection case).

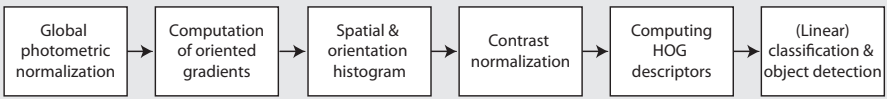
   Local information is binned in relatively small cells consisting of adjacent pixels, and the size and shape of the cells must be determined. Cells consisting of 6×6 to 8×8 pixels (6–8 pixel width corresponds to the width of a human limb) and organized in rectangular 2×2 or 3×3 blocks of cells were used in the pedestrian detection case. Alternatively, rectangular or circular blocks (of cells) may be defined. Figure 10.2 shows an example of rectangular and circular block options, among the many possible block designs. Rectangular cells are primarily used to construct the blocks due to their computational efficiency. Additionally, the block features are computed for overlapping blocks and therefore, a grid must be designed to determine parameters of the overlap.



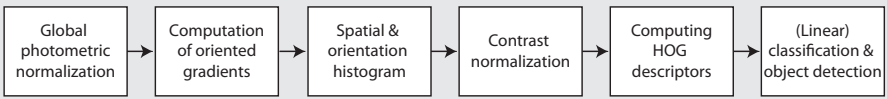
2. **Photometric normalization.** Global image data normalization and gamma correction is performed over the entire image. Use of color (multi-band) image data is recommended when applicable and independent channel-specific gamma correction is recommended in that case.
3. **Computation of oriented gradients.** 1D or 2D directional gradient detectors with different levels of image smoothing can be used (higher-D gradient detectors are foreseeable in volumetric or higher-dimensional images). Most (if not all) implementations employ a centered local 1D gradient detector [-1, 0, 1] with no smoothing ($\sigma = 0$, it was also reported to work best in the pedestrian detection case). The 1D gradient detectors are applied vertically and horizontally, In color images, separate channel–specific image gradients can be computed and the largest-norm channel-specific gradient used.



4. **Spatial and orientation binning—constructing the histogram.** For each pixel of the analyzed cell, its gradient orientation is used to increment the respective histogram bin in proportion to the gradient magnitude. To gain invariance to minor orientation differences, these histogram bin contributions are linearly or bilinearly interpolated between the neighboring bin centers—each gradient direction thus contributes to several neighboring bins with interpolated weights. Histogram bins are evenly spaced over the $[0^o, 180^o)$ interval when working with unsigned gradients or over $[0^o, 360^o)$ when gradient orientation is used in addition to direction. Dense directional binning is important. 20^o increments, leading to 9 bins, were shown to give good results in the pedestrian detection case when unsigned gradients were used—20^o is quite small when dealing with edge direction differences. Signed gradients were shown appropriate in other applications.



5. **Contrast normalization:** To deal with positionally-varying gradient strengths due to illumination changes and foreground–background contrast differences, locally-sensed contrast must be normalized. Normalization is performed in blocks and each block is contrast-normalized separately (see equations 10.1–10.3 below, Figure 10.4, and [Dalal and Triggs, 2005; Dalal, 2006] for details). Even if individual blocks overlap, each (overlapping) block is normalized independently.
6. **Forming the final HOG descriptor.** A vector of components of normalized responses from each cell forming the block, and combined for all (overlapping) blocks in the detection window forms the final descriptor. The HOG descriptor is therefore associated with the entire window. The overlap of blocks allows local image information from individual cells to contribute to several block-based feature vectors, each of them subjected to a block-specific normalization, Figure 10.4e.



7. **Classification:** The HOG description vector is used for training and recognition employing any of the available feature-based classifier, working well with efficient linear classifiers—linear support vector machines performed very well for the pedestrian detection/localization task [Dalal and Triggs, 2005], Figure 10.4f,g.
8. **Object detection.** The detection window is moved across the image and the HOG description vector is obtained for all positions and scales. Non-maximum suppression is used for object detection and localization in the multi-scale image pyramid. PASCAL overlap non-maximal suppression is widely used and is (virtually) parameter-free [Everingham et al., 2010].

Figure 10.3: Example images used for HOG-based pedestrian detection/localization. *A color version of this figure may be seen in the color inset—Plate 1.*

This approach is the most popular and successful person-detection approach in existence today, including humans in difficult or unusual poses.

Other applications include

- human face detection
- deer detection in thermal camera images to reduce animal–vehicle collisions
- 3D extension to detect regions of interest in medical images
- database image retrieval using hand-drawn shape sketches
- ...

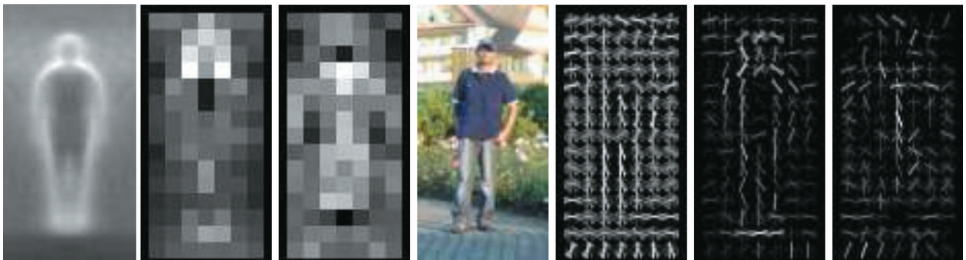
Contrast normalization:

- contrast normalization schemes can be employed to construct HOG description features
- $\boldsymbol{\xi}$ — non-normalized vector of histogram features
- $||\boldsymbol{\xi}||_k$ — its k-norm, $k = 1, 2$
- ϵ — a small positive constant

The following normalization schemes were proposed and tested in [Dalal and Triggs, 2005; Dalal, 2006]:

- **L2 norm**:
$$\boldsymbol{\xi} \rightarrow \frac{\boldsymbol{\xi}}{\sqrt{||\boldsymbol{\xi}||_2^2 + \epsilon^2}} . \tag{10.1}$$
- **L2-Hys norm**—L2 norm subjected to clipping, thus limiting the maximum to a pre-specified value—0.2 was shown to be appropriate—followed by an additional step of re-normalizing as in [Lowe, 2004].
- **L1 norm**:
$$\boldsymbol{\xi} \rightarrow \frac{\boldsymbol{\xi}}{||\boldsymbol{\xi}||_1 + \epsilon} . \tag{10.2}$$
- **L1-sqrt norm**—L1 norm followed by square root, thus effectively treating the descriptor vector as probability distributions:
$$\boldsymbol{\xi} \rightarrow \sqrt{\frac{\boldsymbol{\xi}}{||\boldsymbol{\xi}||_1 + \epsilon}} . \tag{10.3}$$

All four presented contrast normalization schemes markedly improve overall performance when compared to no normalization, but the simple L1 norm was least successful. Low sensitivity to the value of ϵ was observed.



(a) (b) (c) (d) (e) (f) (g)

Figure 10.4: *A color version of this figure may be seen in the color inset—Plate 2.* Example of HOG features when used for pedestrian detection. The detector is mostly responding to the pedestrian body contours (e.g., head, shoulders, feet). (a) Average gradient magnitude image constructed from all training samples. (b) Maximum positive weights of the SVM classifier, shown associated with individual blocks (larger and overlapping blocks, positioned at their center pixel). (c) Maximum negative weights of the SVM classifier. (d) Example test image window. (e) Rectangular HOG block descriptors from window (d). (f,g) Rectangular HOG block descriptors from window (d) weighted by positive (f) and negative (g) SVM weights. The linear SVM classifier correctly identifies window (d) as depicting a pedestrian.

Notable properties:

- highly local gradients and their orientations, derived from non-smoothed images at fine scales, thus representing local abrupt edges, outperform features derived from smoothed image information
- the gradient orientation should be sampled quite finely
- spatial smoothing—applied after the local edge detection and performed at small blocks—can be relatively coarse
- local contrast normalization is essential for good performance and multiple independent local contrast normalizations can be combined in the overall descriptor—offering information redundancy that improves performance

10.8 Image understanding using random forests

- RFs for image analysis and understanding for multi-class object detection
- simultaneous use of classification and regression capabilities
- RFs require large training datasets to be available and their generalization capabilities are not suited for small training datasets
- size of the training dataset is the single most important parameter influencing random forest performance
- highly successful commercial use of RFs — Microsoft Kinect for XBox
  - trained on 900,000 examples of depth image data
  - recognizing 31 separate human body parts of a human being
  - in virtually any position and orientation
- massive training requires substantial time to complete
- training a random forest consisting of merely three trees to a depth of 20 required one full day using a 1000 node cluster [Shotton et al., 2011]
- $\rightarrow$ body part detection runs at a frame rate of 200 frames per second on commercial 2013 CPU/GPU hardware taking advantage of the natural parallelization of the recognition process

Specifics of RFs for image analysis:

- RF classification capability used for object recognition (RFs well suited for many-class recognition)
- RF regression capability predicts object location
- image is divided into patches of pre-determined size
  - training set: each object of class ω_i is outlined by its bounding box
  - image patches falling within the bounding box are associated with a respective *class label*
  - for a patch to 'fall within' the box, either the entire patch or its center must be inside of the bounding box
  - not requiring the entire patch to be located inside the object box permits better sampling of the object boundary information and is especially useful for tight bounding boxes
  - remaining non-object patches of the image form background and are associated with a *background* label
    —of course, no bounding boxes are used to outline background
  - patches may but do not have to be densely sampled
  - for each patch, a set of features is calculated
    —low-level features such as color, gradients, Gabor filter indices, and similar are frequently used since they can be computed efficiently
    —alternatively, SIFT or SURF sparse features can be employed

- to inject randomness to the process, each tree $\mathcal{T}_t$ of the forest uses a randomly selected subset A_t of image patches for training
  - if an image patch is associated with an object label, an additional piece of information may be associated with each patch
    —for example, the distance from and orientation to a reference point of the training object may be associated with each training patch
    this reference point may be class-specific or a center of the bounding box may be used for simplicity
- ... intuitively clear that recognizing an object patch as belonging to a specific object class and determining distance from and direction to its reference point may help recognize (classification) as well as locate (regression) the object in the recognition stage

... Notice a similarity with the Hough transform in which individual image features (usually edges) contribute to identification of an object instance in the accumulator space

... the exact definition of the reference point is of secondary importance as long as it is defined consistently across all training samples.

Figure 10.5: Detection and localization of cars in outdoor scenes. Image patches associated with the 'car' class are shown in red, and those denoting background are shown in blue. Green vectors connect centers of individual non-background patches with a car-object reference point. With kind permission from Springer Science+Business Media: Outdoor and Large-Scale Real-World Scene Analysis, "An introduction to random forests for multi-class object detection," 2012, pp. 243-263, J. Gall, N. Razavi, and L. Gool *A color version of this figure may be seen in the color inset—Plate 3.*

- scale considerations are addressed during the recognition stage — therefore all image patches are all of the same size
- patch size of 16×16 was shown appropriate for images that have been previously scaled so that the length of the bounding box outlining the object is about 100 pixels [Gall and Lempitsky, 2009]
- Figure 10.5 shows examples of object and background image patches that are sized according to these recommendations

- tree-specific training set A_t consists of a set of patches P_i, which hold image, patch class information, and its relative location:

$$P_i = [\mathbf{I}_i, \omega_i, \mathbf{d}_i] \,, \tag{10.4}$$

  where $\mathbf{I}_i$ holds the patch image information (e.g., as a set of calculated features), ω_i is the patch class label, and $\mathbf{d}_i$ is an offset vector from the patch center to the reference point

- background patches are not associated with any reference point, a pseudo-offset of $\mathbf{d}_i = 0$ is used

- each tree is trained in parallel
- class probability and class-specific distribution of training patches need to be learned from the training set and associated with each leaf L from the set of all leaves, forming leaf-specific prediction models
- leaf-specific class probability $p(\omega_r|L)$ can be derived from A^L_{t,ω_r}—the number of patches of class ω_r that arrive at leaf L of tree $\mathcal{T}_t$ after training, normalized to account for uneven distribution of classes in the training set of patches:

$$p(\omega_r|L) = \frac{|A^L_{t,\omega_r}| \cdot b_{t,\omega_r}}{\sum_{r=1,\ldots,R}(|A^L_{t,\omega_r}| \cdot b_{t,\omega_r})}\,, \tag{10.5}$$

$$b_{t,\omega_r} = \frac{|A_t|}{|A_{t,\omega_r}|}\,, \tag{10.6}$$

  where A_t is the entire set used to train tree $\mathcal{T}_t$ and A_{t,ω_r} is a set of all patches in A_t belonging to class ω_r and R is the number of classes.
- class-specific spatial distribution of patches $p(\mathbf{d}|\omega_r, L)$ is derived from the offsets $\mathbf{d} \in D^L_{\omega_r}$ of all patches A^L_{t,ω_r}, where $D^L_{\omega_r}$ is the set of offsets associated with patches of class ω_r reaching node L

- Figure 10.6 shows examples of leaf-specific statistics of trees for detection of cars in images from Figure 10.5

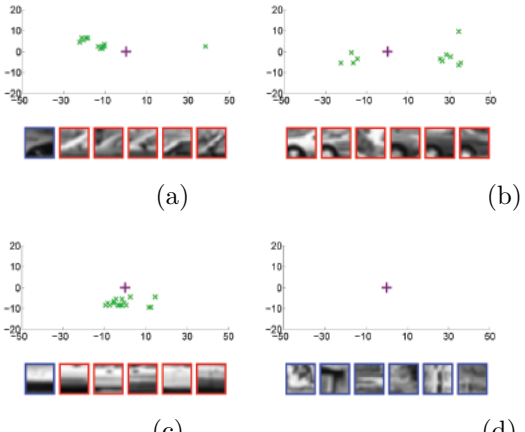


Figure 10.6: Information contained in several samples of tree leaves in car detection with a random forest (see Figure 10.5). Probabilities $p(\omega_r|L)$ that patches reach a specific tree leaf L are stored for each leaf and result from the relative numbers of positive (red) and negative (blue) examples that reach the leaf during training. The end-points of all offset vectors $\mathbf{d}$ are shown as green crosses for all positive examples (all negative examples have $\mathbf{d} = 0$). (a,c) The distribution of vectors $\mathbf{d}$ is frequently multimodal, showing correspondence of the positive patches with multiple object parts. (b) The wheel patches may be associated with either the front or the rear wheels. (d) The tree leaf associated with this panel only contains negative patches. *A color version of this figure may be seen in the color inset—Plate 4.*

Detection stage:

- patches are densely sampled on previously unseen images
- and evaluated in each node of each tree forming the forest
  —starting at their roots to be sent to one of the two child nodes until they reach a tree leaf
- each patch $P(\mathbf{y})$,
  where $\mathbf{y}$ denotes patch image location,
  eventually ends up in one of the tree leaves $L_t(\mathbf{y})$ per tree $\mathcal{T}_t$

- to detect and localize an object in the image, contributions from multiple patches are considered and the patch configuration needs to point to a sufficiently consistent reference point $\mathbf{x}$, which then represents the location of the identified object
- probabilities of object-class-and-location hypotheses $h_t(\omega_r, \mathbf{x}, s)$ need to be computed for
  - each tree $\mathcal{T}_t$
  - each class ω_r
  - each reference point location $\mathbf{x}$
  - each object scaling factor s

- for any patch located at $\mathbf{y}$
  —single-tree probability that a patch $P(\mathbf{y})$ is associated with an object labeled ω_r at reference point $\mathbf{x}$ is calculated as [Gall et al., 2011]:

$$p(h_t(\omega_r, \mathbf{x}, s)|L_t(\mathbf{y})) = p(\mathbf{d}(\mathbf{x}, \mathbf{y}, s)|\omega_r, L_t(\mathbf{y}))p(\omega_r|L_t(\mathbf{y})) \,, \tag{10.7}$$

  where

$$\mathbf{d}(\mathbf{x}, \mathbf{y}, s) = \frac{s_u(\mathbf{y} - \mathbf{x})}{s} \,. \tag{10.8}$$

  s_u represents the unit size of the training-object bounding box that is known from training

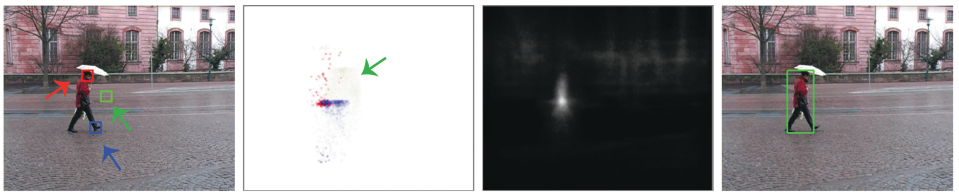
- probabilities $p(\mathbf{d}(\mathbf{x}, \mathbf{y}, s)|\omega_r, L_t(\mathbf{y}))$ and $p(\omega_r|L_t(\mathbf{y}))$ are known from training as explained above (see also equation 10.5)
- distribution $p(h_t(\omega_r, \mathbf{x}, s)|L_t(\mathbf{y}))$ combines both the classification and regression aspects of the object detection and localization task

- as suggested earlier, a voting approach can be employed to approximate the distributions $p(\mathbf{d}(\mathbf{x}, \mathbf{y}, s)|\omega_r, L_t(\mathbf{y}))$
  - let the distance vectors $\mathbf{d}$ associated with class ω_r and patch locations $\mathbf{y}$ that reached leaf L_t in tree $\mathcal{T}_t$ form a set $D_{\omega_r}^{L_t(\mathbf{y})}$
  - equation (10.7) can be rewritten as

$$p(h_t(\omega_r, \mathbf{x}, s)|L_t(\mathbf{y})) = \frac{1}{|D_{\omega_r}^{L_t(\mathbf{y})}|} \left(\sum_{\mathbf{d} \in D_{\omega_r}^{L_t(\mathbf{y})}} \delta_{\mathbf{d}} \cdot \left(\frac{s_u(\mathbf{y} - \mathbf{x})}{s} \right) \right) p(\omega_r|L_t(\mathbf{y})) \,, \tag{10.9}$$

    where δ is a Dirac delta function

- Equations (10.7–10.9) give the probabilities for a single tree
- Figure 10.7 further demonstrates the approach
- alternatively, distributions can be approximated using Gaussian mixture models



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

Figure 10.7: Pedestrian detection and localization using a random forest. (a) Three kinds of image patches are shown—head patch (red), foot patch (blue), background patch (green) (arrows). (b) Weighted votes of pedestrian's position, color coded with respect to which patch class contributes to a specific reference point location (equation 10.9). While the head patch class forms a single strong mode of possible reference point location (red), the foot patch class obtains similar responses from the left and right feet (blue), subsequently forming a two-mode response. A weak set of green responses (green arrow) with no clear mode(s) is associated with background patches. Here, the low probability of a background class to belong to the pedestrian object contributes to the low weights and overall weak response from background. (c) Accumulation of votes from all patches (equation 10.11 employed at one scale s)—a single strong mode emerges. (d) Pedestrian detection shown as a bounding box derived from the detected location of the reference point. *A color version of this figure may be seen in the color inset—Plate 5.*

- using an across-tree averaging approach, a forest-based probability can be obtained

$$p(h(\omega_r, \mathbf{x}, s)|P(\mathbf{y})) = \frac{1}{T}\sum_t p(h_t(\omega_r, \mathbf{x}, s)|L_t(\mathbf{y}))\,. \tag{10.10}$$

- using this forest-level probability, distribution over all patches and all trees results from accumulation

$$p(h(\omega_r, \mathbf{x}, s)|\mathbf{I}) = \frac{1}{|\mathcal{Y}|}\sum_{\mathbf{y}\in\mathcal{Y}} p(h_t(\omega_r, \mathbf{x}, s)|P(\mathbf{y}))\,, \tag{10.11}$$

  where $\mathbf{I}$ refers to the entire image and $\mathcal{Y}$ is the set of all patch locations $\mathbf{y}$.

- applied to the image of Figure 10.7a at one scale, this equation yields a single strong mode shown in Figure 10.7c

- processing an image at multiple scales is demonstrated in Figure 10.8
- to detect an object of scale s considering that the training objects were all scaled to a training size of s_u, each image needs to be scaled at s_u/s
- if all images are scaled at all feasible scaling levels prior to being analyzed, the training-introduced scaling is already accounted for (equation 10.8) and object detection via identification of strong modes of equation (10.11) can be efficiently accomplished by employing mean shift mode detection

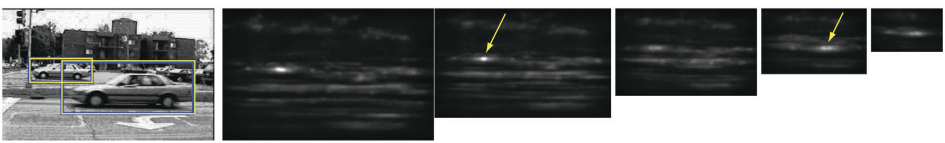


Figure 10.8: Objects are detected at multiple scales. (a) Two cars are marked in the original image, each located at a different distance from the observer. (b–f) Cars can be detected at multiple scales and locations by searching for modal maxima in the joint scale–location space. The larger car produces modal responses in panels (b–d) with a maximal modal peak associated with scale shown in panel (c). Similarly, the smaller car shows modal responses in panels (e–f) with a maximum shown in panel (e). With kind permission from Springer Science+Business Media: Outdoor and Large-Scale Real-World Scene Analysis, "An introduction to random forests for multi-class object detection," 2012, pp. 243-263, J. Gall, N. Razavi, and L. Gool *A color version of this figure may be seen in the color inset—Plate 6.*

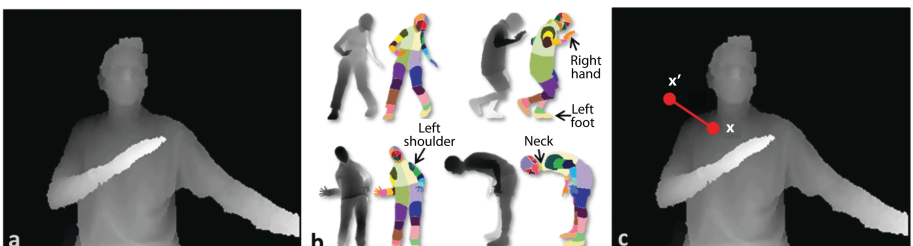


Figure 10.9: Employing random forests in Microsoft Kinect for XBox. (a) Original depth image (640 × 480 pixels), brightness corresponds to depth. (b) Color-coded ground truth for 31 body parts. (c) Reference point $\mathbf{x}'$ associated with a patch at location $\mathbf{x}$. A. Criminisi, J. Shotton, and E. Konukoglu, Decision Forests for Classfication, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft Research technical report TR-2011-114. © 2012 Microsoft Corporation. All rights reserved *A color version of this figure may be seen in the color inset—Plate 7.*

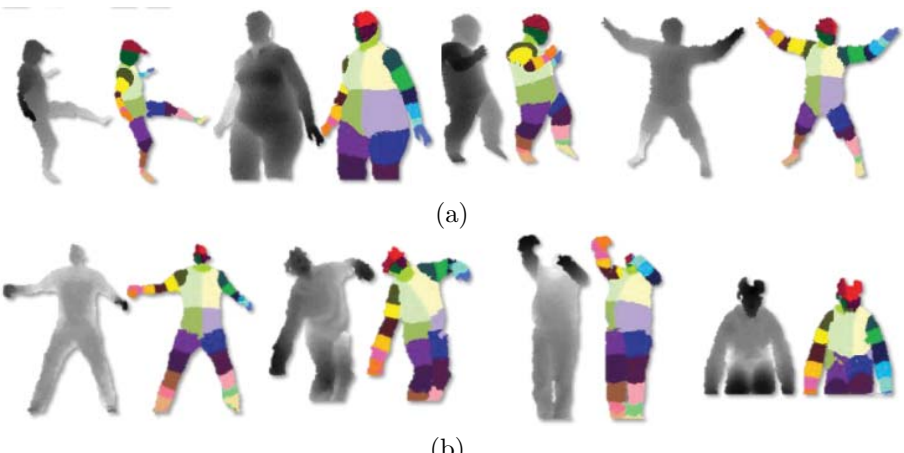


Figure 10.10: Examples of training and testing images with associated ground truth. (a) Training data consisted of a combination of real and synthetic depth datasets. (b) Real examples were used for testing.
A color version of this figure may be seen in the color inset—Plate 8.

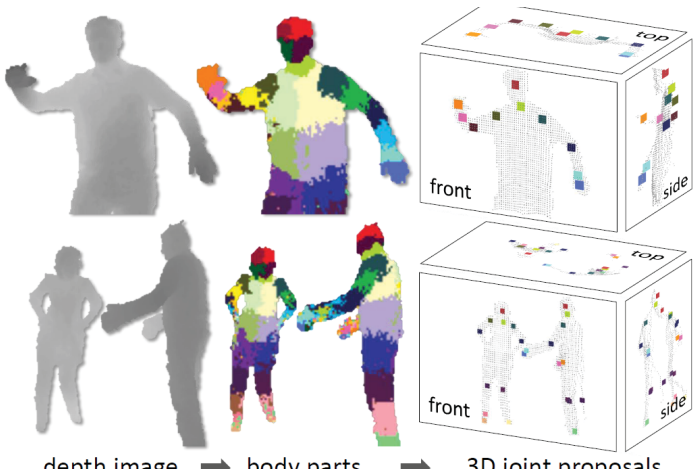


Figure 10.11: Body parts are detected and localized in 3D. *A color version of this figure may be seen in the color inset—Plate 9.*

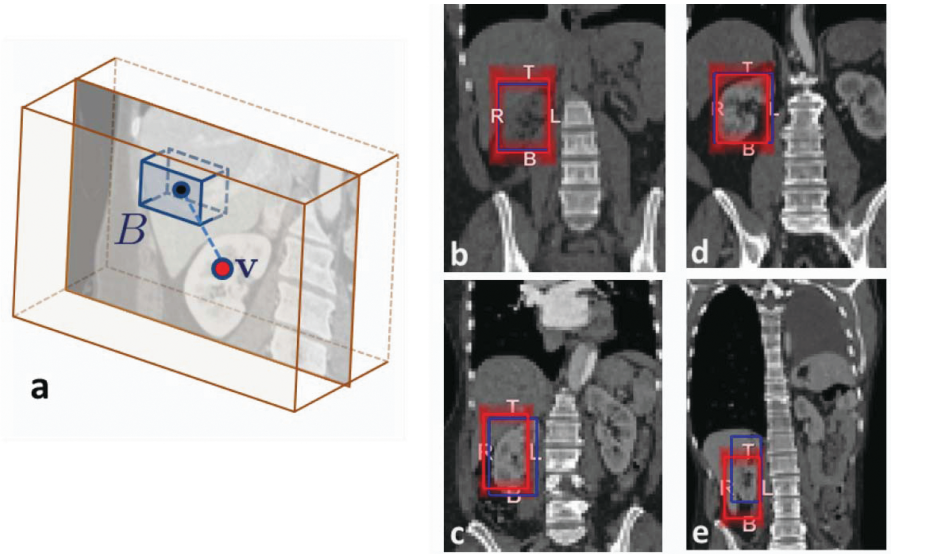


Figure 10.12: Application of random forests to 3D kidney detection and localization from abdominal X-ray CT images. (a) Distance vector between the center of a 3D patch and the reference point associated with a 3D kidney location is shown. (b–e) Example detections of kidneys show robustness of performance across subject-specific anatomical differences. Random-forest detections are shown in red while independent standard is in blue. A. Criminisi, J. Shotton, and E. Konukoglu, Decision Forests for Classfication, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft Research technical report TR-2011-114. *A color version of this figure may be seen in the color inset—Plate 9.*

Extensions:

- Many applications have emerged that use the random forest approach.
- Figures 10.9–10.11 show body part detection/localization from depth image data used in Microsoft Kinect for XBox [Shotton et al., 2011]
  - depth images consisting of 640×480 pixels are acquired at 30 frames per second at a depth-resolution of a few centimeters (Figure 10.9)
  - images are used to identify $R = 31$ body parts to which each image pixel belongs:

$$\omega_r \in \{\text{left/right hand, left/right shoulder, left/right elbow, neck, etc.}\}\,. \tag{10.12}$$

  - Figure 10.10 shows examples of training and testing data
  - single-pixel image patches at location $\mathbf{x}$ are associated with depth-based features $f_{\mathbf{u},\mathbf{v}}(I, \mathbf{x})$

$$f_{\mathbf{u},\mathbf{v}}(I, \mathbf{x}) = d_I\left(\mathbf{x} \cdot \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} \cdot \frac{\mathbf{v}}{d_I(\mathbf{x})}\right)\,, \tag{10.13}$$

    where $d_I(\mathbf{x})$ is depth at pixel $\mathbf{x}$ in image I
    $\mathbf{u}$ and $\mathbf{v}$ are two vectors representing two positional offsets with respect to $\mathbf{x}$

- these offsets therefore allow depth at $\mathbf{x}$ to be simultaneously compared with depth at $\mathbf{x}+\mathbf{u}$ and depth at $\mathbf{x}+\mathbf{v}$, with $\mathbf{u}$ and $\mathbf{v}$ being parameters of this neighborhood depth comparison
- normalization factor $1/d_I(\mathbf{x})$ yields depth invariance of features and therefore 3D world coordinate invariance

- these features together with pixel-based class information are used for training a random forest, which in the image analysis stage assigns one of the 32 labels to each image pixel (31 body part classes and background).

- To obtain information about 3D positions of skeletal joints, the per-pixel information about body part labels must be pooled across pixels (for example) to find 3D centroids of all pixels with the same label.
- This approach however suffers from noise sensitivity and a mean-shift Gaussian-kernel weighted mode-finding approach was employed in the Kinect.
- Figure 10.11 demonstrates how three-dimensional information about the perceived body pose and location is provided by the described process.

- Random forests are also finding applications in medical imaging.
- For example, whole-body segmentation of anatomical structures from 3D CT or MR image data and automatic detection of presence/absence of individual structures has been reported in [Criminisi et al., 2010].
- Figure 10.12 demonstrates robustness of 3D kidney detection across subjects with natural anatomic variations.

10.9 References

Criminisi A. and Shotton J. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer Verlag, London, 2013.

Criminisi A., Shotton J., and Konukoglu E. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report MSR-TR-2011-114, Microsoft Research, Ltd., Cambridge, UK, 2011.

Criminisi A., Shotton J., Robertson D., and Konukoglu E. Regression forests for efficient anatomy detection and localization in CT studies. In *MICCAI 2010 Workshop MCV*, volume LNCS 6533, pages 106–117. Springer Verlag, 2010.

Dalal N. *Finding people in images and videos*. Ph.D. thesis, Institut National Polytechnique de Grenoble, July 2006. URL http://lear.inrialpes.fr/pubs/2006/Dal06.

Dalal N. and Triggs B. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, pages 886–893. IEEE, 2005.

Everingham M., Van Gool L., Williams C. K. I., Winn J., and Zisserman A. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88: 303–338, 2010.

Freeman W. T. and Roth M. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face- and Gesture-Recognition*, pages 296–301, Zurich, Switzerland, 1995. IEEE.

Gall J. and Lempitsky V. Class-specific hough forests for object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1022–1029, 2009.

Gall J., Yao A., Razavi N., Van Gool L., and Lempitsky V. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

Gall J., Razavi N., and Gool L. An introduction to random forests for multi-class object detection. In Dellaert F., Frahm J.-M., Pollefeys M., Leal-Taixe L., and Rosenhahn B., editors, *Outdoor and Large-Scale Real-World Scene Analysis*, volume 7474 of *Lecture Notes in Computer Science*, pages 243–263. Springer Berlin Heidelberg, 2012.

Geman S. and Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

Lowe D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A., and Blake A. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304, 2011.