<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basic Usage</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Advanced Usage</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Features</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Filtering</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>I/O</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Keypoints</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>KdTree</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>Octree</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>Range Images</td>
<td>25</td>
</tr>
<tr>
<td>10</td>
<td>Recognition</td>
<td>27</td>
</tr>
<tr>
<td>11</td>
<td>Registration</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Sample Consensus</td>
<td>31</td>
</tr>
<tr>
<td>13</td>
<td>Segmentation</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>Surface</td>
<td>37</td>
</tr>
<tr>
<td>15</td>
<td>Visualization</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>Applications</td>
<td>43</td>
</tr>
<tr>
<td>17</td>
<td>GPU</td>
<td>45</td>
</tr>
</tbody>
</table>
The following links describe a set of basic PCL tutorials. Please note that their source codes may already be provided as part of the PCL regular releases, so check there before you start copy & pasting the code. The list of tutorials below is automatically generated from reST files located in our git repository.

**Note:** Before you start reading, please make sure that you go through the higher-level overview documentation at [http://www.pointclouds.org/documentation/](http://www.pointclouds.org/documentation/), under **Getting Started**. Thank you.

As always, we would be happy to hear your comments and receive your contributions on any tutorial.
CHAPTER 1

Basic Usage

- walkthrough

---

Title: PCL Functionality Walkthrough  
Author: Razvan G. Mihalyi  
Compatibility: > PCL 1.6  
Takes the reader through all of the PCL modules and offers basic explanations on their functionalities.

- basic_structures

---

Title: Getting Started / Basic Structures  
Author: Radu B. Rusu  
Compatibility: > PCL 1.0  
Presents the basic data structures in PCL and discusses their usage with a simple code example.

- using_pcl_pcl_config
In this tutorial, we will learn how to link your own project to PCL using cmake.

**compiling_pcl_posix**

In this tutorial, we will explain how to compile PCL from sources on POSIX/Unix systems.

**building_pcl**

In this tutorial, we will explain the basic PCL cmake options, and ways to tweak them to fit your project.

**compiling_pcl_dependencies_windows**

In this tutorial, we will explain how to compile PCL’s 3rd party dependencies from source on Microsoft Windows.
In this tutorial, we will explain how to compile PCL on Microsoft Windows.

This tutorial explains how to build the Point Cloud Library from MacPorts and source on Mac OS X platforms.

This tutorial explains how to install the Point Cloud Library on Mac OS X using Homebrew. Both direct installation and compiling PCL from source are explained.

This tutorial shows you how to get your PCL as a project in Eclipse.
This tutorial shows you how to generate and use a local documentation for PCL.

- matrix_transform

This tutorial shows you how to transform a point cloud using a matrix.
Advanced Usage

- **adding_custom_ptype**

  Title: **Adding your own custom** PointT point type  
  Author: *Radu B. Rusu*  
  Compatibility: > PCL 0.9, < PCL 2.0  
  This document explains what templated point types are in PCL, why do they exist, and how to create and use your own PointT point type.

- **writing_new_classes**

  Title: **Writing a new PCL class**  
  Author: *Radu B. Rusu, Luca Penasa*  
  Compatibility: > PCL 0.9, < PCL 2.0  
  This short guide is to serve as both a HowTo and a FAQ for writing new PCL classes, either from scratch, or by adapting old code.
CHAPTER 3

Features

• how_3d_features_work

Title: How 3D features work
Author: Radu B. Rusu
Compatibility: > PCL 1.0
This document presents a basic introduction to the 3D feature estimation methodologies in PCL.

• normal_estimation

Title: Estimating Surface Normals in a PointCloud
Author: Radu B. Rusu
Compatibility: > PCL 1.0
This tutorial discusses the theoretical and implementation details of the surface normal estimation module in PCL.

• normal_estimation_using_integral_images
In this tutorial we will learn how to compute normals for an organized point cloud using integral images.

pfh_estimation

This tutorial introduces a family of 3D feature descriptors called PFH (Point Feature Histograms) and discusses their implementation details from PCL’s perspective.

fpfh_estimation

This tutorial introduces the FPFH (Fast Point Feature Histograms) 3D descriptor and discusses their implementation details from PCL’s perspective.

vfh_estimation
This document describes the Viewpoint Feature Histogram (VFH) descriptor, a novel representation for point clusters for the problem of Cluster (e.g., Object) Recognition and 6DOF Pose Estimation.

- **narf_feature_extraction**

- **moment_of_inertia**

- **rops_feature**
This document describes the Globally Aligned Spatial Distribution (GASD) global descriptor to be used for efficient object recognition and pose estimation.
CHAPTER 4

Filtering

• passthrough

Title: Filtering a PointCloud using a PassThrough filter
Author: Radu B. Rusu
Compatibility: > PCL 1.0
In this tutorial, we will learn how to remove points whose values fall inside/outside a user given interval along a specified dimension.

• voxelgrid

Title: Downsampling a PointCloud using a VoxelGrid filter
Author: Radu B. Rusu
Compatibility: > PCL 1.0
In this tutorial, we will learn how to downsample (i.e., reduce the number of points) a Point Cloud.

• statistical_outlier_removal
In this tutorial, we will learn how to remove sparse outliers from noisy data, using StatisticalRemoval.

• project_inliers

In this tutorial, we will learn how to project points to a parametric model (i.e., plane).

• extract_indices

In this tutorial, we will learn how to extract a set of indices given by a segmentation algorithm.

• remove_outliers

In this tutorial, we will learn how to remove outliers from noisy data, using ConditionalRemoval, RadiusOutlierRemoval.
I/O

- pcd_file_format

Title: The PCD (Point Cloud Data) file format  
Author: Radu B. Rusu  
Compatibility: > PCL 0.9  
This document describes the PCD file format, and the way it is used inside PCL.

- reading_pcd

Title: Reading Point Cloud data from PCD files  
Author: Radu B. Rusu  
Compatibility: > PCL 1.0  
In this tutorial, we will learn how to read a Point Cloud from a PCD file.

- writing_pcd
In this tutorial, we will learn how to write a Point Cloud to a PCD file.

- **concatenate_clouds**

In this tutorial, we will learn how to concatenate both the fields and the point data of two Point Clouds. When concatenating fields, one PointClouds contains only \( \text{XYZ} \) data, and the other contains \( \text{Surface Normal} \) information.

- **openni_grabber**

In this tutorial, we will learn how to acquire point cloud data from an OpenNI camera.

- **hdl_grabber**

In this tutorial, we will learn how to acquire point cloud data from a Velodyne HDL.
• dinast_grabber

Title: Grabbing Point Clouds from Dinast Cameras  
Author: Marco A. Gutierrez  
Compatibility: >= PCL 1.7  
In this tutorial, we will learn how to acquire point cloud data from a Dinast camera.

• ensenso_cameras

Title: Grabbing point clouds from Ensenso cameras  
Author: Victor Lamoine  
Compatibility: >= PCL 1.8.0  
In this tutorial, we will learn how to acquire point cloud data from an IDS-Imaging Ensenso camera.

• david_sdk

Title: Grabbing point clouds / meshes from davidSDK scanners  
Author: Victor Lamoine  
Compatibility: >= PCL 1.8.0  
In this tutorial, we will learn how to acquire point cloud or mesh data from a davidSDK scanner.

• depth_sense_grabber
Title: Grabbing point clouds from DepthSense cameras
Author: Sergey Alexandrov
Compatibility: &ge; PCL 1.8.0
In this tutorial we will learn how to setup and use DepthSense cameras within PCL on both Linux and Windows platforms.
Keypoints

- narf_keypoint_extraction

Title: How to extract NARF keypoints from a range image
Author: Bastian Steder
Compatibility: > 1.3
In this tutorial, we will learn how to extract NARF keypoints from a range image.
In this tutorial, we will learn how to search using the nearest neighbor method for k-d trees.

Title: KdTree Search
Author: Gabe O'Leary
Compatibility: > PCL 1.0

In this tutorial, we will learn how to search using the nearest neighbor method for k-d trees.
In this tutorial, we will learn how to compress a single point cloud and streams of point clouds.

- **octree_compression**

  **Title:** Point cloud compression  
  **Author:** Julius Kammerl  
  **Compatibility:** > PCL 1.0  
  In this tutorial, we will learn how to compress a single point cloud and streams of point clouds.

- **octree_search**

  **Title:** Octrees for spatial partitioning and neighbor search  
  **Author:** Julius Kammerl  
  **Compatibility:** > PCL 1.0  
  In this tutorial, we will learn how to use octrees for spatial partitioning and nearest neighbor search.

- **octree_change_detection**
In this tutorial, we will learn how to use octrees for detecting spatial changes within point clouds.
CHAPTER 9

Range Images

• range_image_creation

Title: Creating Range Images from Point Clouds
Author: Bastian Steder
Compatibility: > PCL 1.0
This tutorial demonstrates how to create a range image from a point cloud and a given sensor position.

• range_image_border_extraction

Title: Extracting borders from Range Images
Author: Bastian Steder
Compatibility: > PCL 1.3
This tutorial demonstrates how to extract borders (traversals from foreground to background) from a range image.
CHAPTER 10

Recognition

- correspondence_grouping

Title: The PCL Recognition API
Author: Tommaso Cavallari, Federico Tombari
Compatibility: > PCL 1.6
This tutorial aims at explaining how to perform 3D Object Recognition based on the pcl_recognition module.

- implicit_shape_model

Title: Implicit Shape Model
Author: Sergey Ushakov
Compatibility: > PCL 1.7
In this tutorial we will learn how the Implicit Shape Model algorithm works and how to use it for finding objects centers.

- global_hypothesis_verification
### Verification for 3D Object Recognition

**Author:** Daniele De Gregorio, Federico Tombari  
**Compatibility:** > PCL 1.7  

This tutorial aims at explaining how to do 3D object recognition in clutter by verifying model hypotheses in cluttered and heavily occluded 3D scenes.
### Registration

- **registration_api**

  ![Image of registration_api](image)

  **Title:** The PCL Registration API  
  **Author:** Dirk Holz, Radu B. Rusu, Jochen Sprickerhof  
  **Compatibility:** > PCL 1.5  
  In this document, we describe the point cloud registration API and its modules: the estimation and rejection of point correspondences, and the estimation of rigid transformations.

- **iterative_closest_point**

  **Title:** How to use iterative closest point algorithm  
  **Author:** Gabe O’Leary  
  **Compatibility:** > PCL 1.0  
  This tutorial gives an example of how to use the iterative closest point algorithm to see if one PointCloud is just a rigid transformation of another PointCloud.

- **pairwise_incremental_registration**

  **Title:** How to incrementally register pairs of clouds  
  **Author:** Raphael Favier  
  **Compatibility:** > PCL 1.4  
  This document demonstrates using the Iterative Closest Point algorithm in order to incrementally register a series of point clouds two by two.
• interactive_icp

**Title:** Interactive ICP  
**Author:** Victor Lamoine  
**Compatibility:** > PCL 1.5  
This tutorial will teach you how to build an interactive ICP program.

• normal_distributions_transform

**Title:** How to use the Normal Distributions Transform algorithm  
**Author:** Brian Okorn  
**Compatibility:** > PCL 1.6  
This document demonstrates using the Normal Distributions Transform algorithm to register two large point clouds.

• in_hand_scanner

**Title:** How to use the In-hand scanner for small objects  
**Author:** Martin Saelzle  
**Compatibility:** >= PCL 1.7  
This document shows how to use the In-hand scanner applications to obtain colored models of small objects with RGB-D cameras.

• alignment_prerejective

**Title:** Robust pose estimation of rigid objects  
**Author:** Anders Glent Buch  
**Compatibility:** >= PCL 1.7  
In this tutorial, we show how to find the alignment pose of a rigid object in a scene with clutter and occlusions.
Sample Consensus

- random_sample_consensus

In this tutorial we learn how to use a RandomSampleConsensus with a plane model to obtain the cloud fitting to this model.
SEGMENTATION

• planar_segmentation

Title: Plane model segmentation
Author: Radu B. Rusu
Compatibility: > PCL 1.3
In this tutorial, we will learn how to segment arbitrary plane models from a given point cloud dataset.

• cylinder_segmentation

Title: Cylinder model segmentation
Author: Radu B. Rusu
Compatibility: > PCL 1.3
In this tutorial, we will learn how to segment arbitrary cylindrical models from a given point cloud dataset.

• cluster_extraction
In this tutorial we will learn how to extract Euclidean clusters with the `pcl::EuclideanClusterExtraction` class.

• region_growing_segmentation

In this tutorial we will learn how to use region growing segmentation algorithm.

• region_growing_rgb_segmentation

In this tutorial we will learn how to use color-based region growing segmentation algorithm.

• min_cut_segmentation

In this tutorial we will learn how to use min-cut based segmentation algorithm.
• conditional_euclidean_clustering

**Conditional Euclidean Clustering**

Author: Frits Florentinus  
Compatibility: >= PCL 1.7  
This tutorial describes how to use the Conditional Euclidean Clustering class in PCL: A segmentation algorithm that clusters points based on Euclidean distance and a user-customizable condition that needs to hold.

• don_segmentation

**Difference of Normals Based Segmentation**

Author: Yani Ioannou  
Compatibility: >= PCL 1.7  
In this tutorial we will learn how to use the difference of normals feature for segmentation.

• supervoxel_clustering

**Supervoxel Clustering**

Author: Jeremie Papon  
Compatibility: >= PCL 1.8  
In this tutorial, we show to break a pointcloud into the mid-level supervoxel representation.

• progressive_morphological_filtering
• model_outlier_removal

This tutorial describes how to extract points from a point cloud using SAC models.
In this tutorial, we will learn how to construct and run a Moving Least Squares (MLS) algorithm to obtain smoothed XYZ coordinates and normals.

In this tutorial we will learn how to calculate a simple 2D concave or convex hull polygon for a set of points supported by a plane.

In this tutorial we will learn how to construct a concave or convex hull polygon for a plane model.
In this tutorial we will learn how to run a greedy triangulation algorithm on a Point-Cloud with normals to obtain a triangle mesh based on projections of the local neighborhood.

- bspline_fitting

In this tutorial we will learn how to reconstruct a smooth surface from an unordered point-cloud by fitting trimmed B-splines.
CHAPTER 15

Visualization

• cloud_viewer

Title: Visualizing Point Clouds
Author: Ethan Rublee
Compatibility: > PCL 1.0
This tutorial demonstrates how to use the pcl visualization tools.

• range_image_visualization

Title: Visualizing Range Images
Author: Bastian Steder
Compatibility: > PCL 1.3
This tutorial demonstrates how to use the pcl visualization tools for range images.

• pcl_visualizer
This tutorial demonstrates how to use the PCLVisualizer class for powerful visualisation of point clouds and related data.

- pcl_plotter

This tutorial demonstrates how to use the PCLPlotter class for powerful visualisation of plots, charts and histograms of raw data and explicit functions.

- visualization

This tutorial will give an overview on the usage of the PCL visualization tools.

- qt_visualizer
<table>
<thead>
<tr>
<th>Title: Create a PCL visualizer in Qt with cmake</th>
<th>Author: Victor Lamoine</th>
<th>Compatibility: &gt; PCL 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>This tutorial shows you how to create a PCL visualizer within a Qt application.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- qt_colorize_cloud

<table>
<thead>
<tr>
<th>Title: Create a PCL visualizer in Qt to colorize clouds</th>
<th>Author: Victor Lamoine</th>
<th>Compatibility: &gt; PCL 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>This tutorial shows you how to color point clouds within a Qt application.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Applications

- template_alignment

  **Aligning object templates to a point cloud**
  
  Author: Michael Dixon
  Compatibility: > PCL 1.3
  This tutorial gives an example of how some of the tools covered in the previous tutorials can be combined to solve a higher level problem — aligning a previously captured model of an object to some newly captured data.

- vfh_recognition

  **Cluster Recognition and 6DOF Pose Estimation using VFH descriptors**
  
  Author: Radu B. Rusu
  Compatibility: > PCL 0.8
  In this tutorial we show how the Viewpoint Feature Histogram (VFH) descriptor can be used to recognize similar clusters in terms of their geometry.

- mobile_streaming
This tutorial describes how to send point cloud data over the network from a desktop server to a client running on a mobile device.

**ground_based_rgbd_people_detection**

**Detecting people on a ground plane with RGB-D data**
Author: Matteo Munaro
Compatibility: >= PCL 1.7
This tutorial presents a method for detecting people on a ground plane with RGB-D data.
CHAPTER 17

GPU

• gpu_install

Title: GPU Installation
Author: Koen Buys
Compatibility: PCL git master
This tutorial explains how to configure PCL to use with a Nvidia GPU

• using_kinfu_large_scale

Title: Using KinFu Large Scale to generate a textured mesh
Author: Francisco Heredia and Raphael Favier
Compatibility: PCL git master
This tutorial demonstrates how to use KinFu Large Scale to produce a mesh from a room, and apply texture information in post-processing for a more appealing visual result.

• gpu_people
Title: People Detection
Author: Koen Buys
Compatibility: PCL git master
This tutorial presents a method for people and pose detection.