User manual

Ground Plane Classification ("*GPC*")

Introduction

GPC was created as an assignment for the 2007 Game & Media Technology Master course *Image Processing* at Utrecht University, The Netherlands.

For panoramic images of CycloMedia (*cycloramas*) this application can detect the ground plane using a combination of edge based image segmentation and neural networks. The ground plane is considered to consist of everything directly visible in the image where one can typically walk upon (for example: street, sidewalk, grass, etc).

GPC uses image segmentation to split the image in a number of segments, after which the resulting segmentation is enhanced and 'cleaned' using several methods.

Finally, a neural network is used to classify each segment as belonging to the ground plane or not.

System requirements

- 1 GHz processor
- 512 MB RAM
- Java 1.6 or newer[1]

Executing the program

Prior to running GPC, it is highly recommended to copy all files from the disc to your local hard disk, as configuration files, training sets and output images can otherwise not be saved.

Start the application simply by executing the batch file (`GPC.bat`) located in the `\bin` directory. After the program window appears, please wait a moment to allow the program to initialize.

[1] Java version 1.6 was used to create and test the program; older Java versions might not be compatible and may yield unexpected results.

User interface

The user interface consists of three sections:

1. A toolbar
2. Two image panes
3. A status and progress bar (including a hidden log screen with additional buttons)

Upper toolbar

- **Load** Load a selected cyclorama image[2] into the upper image pane.
- **Batch** Process all the JPEG images inside a selected directory at once.
- **Save** Allows to save the image in the lower plane to a file, or to save the result of classification as a binary image.
- **Options** Shows the options window (See *Options* section).
- **Performance** Calculates the error rate for a classified image.
- **About** Shows the about window with program information.
- **Copy down** Copies the image in the upper image pane to the lower image pane.
- **Copy up** Copies the image in the lower image pane to the upper image pane.
- **Other image operations**
  - **Scale down** Scales the upper pane image down with scale ½.
  - **Trim** Trims any gray strips from the top or bottom of the upper image.
  - **Invert** Inverts the colors in the image.
  - **Grayscale** Converts the upper image to grayscale.
  - **Histogram eq.** Performs histogram equalization on the upper image.
  - **Morphology** Performs a morphology operation on the upper image.
  - **Edge detection** Displays the results of an edge detection algorithm.
- **Start** Automatically performs the segmentation and classification steps.
- **Cancel** Stops the current operation.

[2] The image must be either in JPEG or PNG format.

Image panes

The upper image pane shows the original image or reference images. All the image operations are performed on the upper image and subsequently outputted to the lower image pane.

If you desire to perform multiple image operations, please do so as follows:

1. Perform an operation via the “Other image operations” toolbar button;
2. After each operation press the “Copy up” toolbar button.

Status and progress bar

The bottom of the main window contains the status and progress bar. The status bar shows the current operation the program is performing and the progress bar shows its progress.

Should you wish to see the status log, you can expand the debug panel by pressing the button in the far bottom left corner. This debug panel also contains several extra buttons to perform some frequently used operations:

- **Segment** Segments the image using the chosen edge detection method en pre- and postprocessing steps set in the options window (see *Options* section). The edges of the resulting segments are drawn in black above the semi-transparent original image.
  After segmentation the segments can be clicked upon for manual classification. To classify the segment as the ground plane (green), left-click a segment. Likewise, right-click a segment to classify it as non-ground plane (red).
- **NN: Save** Adds the manually classified segments to the training set file (“*training_set.txt*”) for the neural network.
  Generally, you will not need to use this button unless you’re making a training set. Please note that choosing which segments to add to the training set must be done with care, more details about this process can be found in the paper entitled *Ground Plane Detection using edge based segmentation and artificial neural networks*.
- **NN: Load** Adds the manually classified segments to the training set file.
  By default, this operation is performed automatically when the application is started, but can be disabled

in the options window.

- **NN: Classify** Classifies all the segments of the current image as either ground plane (green) or non-ground plane (red) using the neural network. The image must be segmented first and the neural network must be trained before this operation can be performed.

Options window

The options window can be opened by pressing the options button on the upper toolbar and consists of five tabs.

- **General:** The general segmentation settings:
    - **Edge detection method:** Choose one of the edge detection methods.
        - **Squared:** The edge squared edge detection method.
        - **Canny:** The canny edge detection method.
        - **Squared + Canny:** Both edge detection methods combined. (default)
- **Preprocessing steps:** The steps performed before doing edge detection during segmentation.
    - **Histogram equalization:** Performs histogram equalization before edge detection.
    - **Median filter:** Performs a median filter on the image before edge detection, which smooths the image details and reduces noise.
    - **Opening:** Performs opening on the image before edge detection.
- **Postprocessing steps:** The steps performed after doing edge detection during segmentation, these deal with removing small segments.
    - **Separation:** Number of times a seperation filer is used after segmentation, used to separate not fully closed segments.
    - **Modus filter:** Number of times a modus filter is used after segmentation, to remove small segments.
    - **Segment filter:** Number of times a small segment filter is used after segmentation. This merges small segments into larger ones.
- **Orientation:** The trimmed top area for the car orientation mask (don't change unless you made a new clipped car mask).
- **Squared:** Options for the squared edge detection algorithm
    - **Threshold:** the threshold value to use for the algorithm (lower means more edges, higher means less edges).
- **Canny:** Options for the canny edge detection algorithm

    - **Threshold:** the threshold value to use for the algorithm (lower means more edges, higher means less edges).
    - **Canny parameters:** Canny algorithm parameters to be tweaked to possibly get better edge detection on certain images.
    - **Canny color conversion:** The canny color conversion fix, which actually gives worse results in most cases. If turned off it will use the red RGB value only.
    - **Use closing on edge image:** Performs closing on the edge image when doing edge detection. When turned off you must turn on the separation filter to close edges for the segmentation.
- **Customize:**
    - **Load neural network at startup:** Automatically train the neural network with the training set file after starting the program.
    - **Automatically segment loaded image:** Automatically segments the image after it's loaded.
    - Automatically classify loaded image: Automatically classify the image segments after it's loaded. “*Automatically segment loaded image*” option must be on.
    - **Detect car:** Detects the car, which captured the image, in the image and makes one big segment from it during segmentation. Requires a program restart.
    - **Save GP after classification:** Automatically saves a binary ground plane image after classification. The black areas in the image represent the extracted ground plane.
    - **Beep when done:** The program gives an audio beep when it's done calculating.

Credits

GPC was a collaboration of:

- Wesley van Beelen
- Mathijs Lagerberg
- Paul Lammertsma
- Jeff Ouwerkerk
- Erik van de Pol

Canny implementation:
    Timothy Sharman (http://homepages.inf.ed.ac.uk/rbf/HIPR2/)
Neural network engine:
    JOONE (www.jooneworld.com)
Icon set:
    Mark James (http://www.famfamfam.com)