PAL 0.2 user documentation
How to integrate PAL in a GIS

Maxence Laurent

September 17, 2009

1 Introduction

This document shortly explains how to use the PAL library.

It links common GIS actions – such adding a layer and so on – with PAL classes and methods.

2 Common GIS actions and events

2.1 A new GIS project is created

- Create a new pal::Pal object and set properties :
  - Set the resolution (dpi) with pal::Pal::setDpi(val) (default is 72 dot-per-inch)

2.2 A vector layer is added

- Add the layer in your pal object with pal::Pal::addLayer()
  - layer name (unique within a pal instance)
  - scale range in which the label should be labelled (min scale, max scale)
  - Arrangement mode for label (see figure 1)
  - Unit of label size (pixel or meters, see pal::Units)
  - defaultPriority : a double between 0 and 1, 0 is the most prioritar (typically a slider in the GUI)
  - obstacle: do pal treats feature of the layers as obstacle ?
  - active: is the layer displayed on the map ?
  - toLabel: do you want to label this layer ?
- register every feature in the pal layer with pal::Layer::registerFeature():
  - an unique identifier for the feature within the layer ($geom\_id$)
  - the size ($label_x, label_y$) for the label (linked with Layer::set/getUnits()), defaults values $(-1, -1)$ prevents feature to be labelled
  - a pointer to the user geometry, which implements the pal::PalGeometry interface

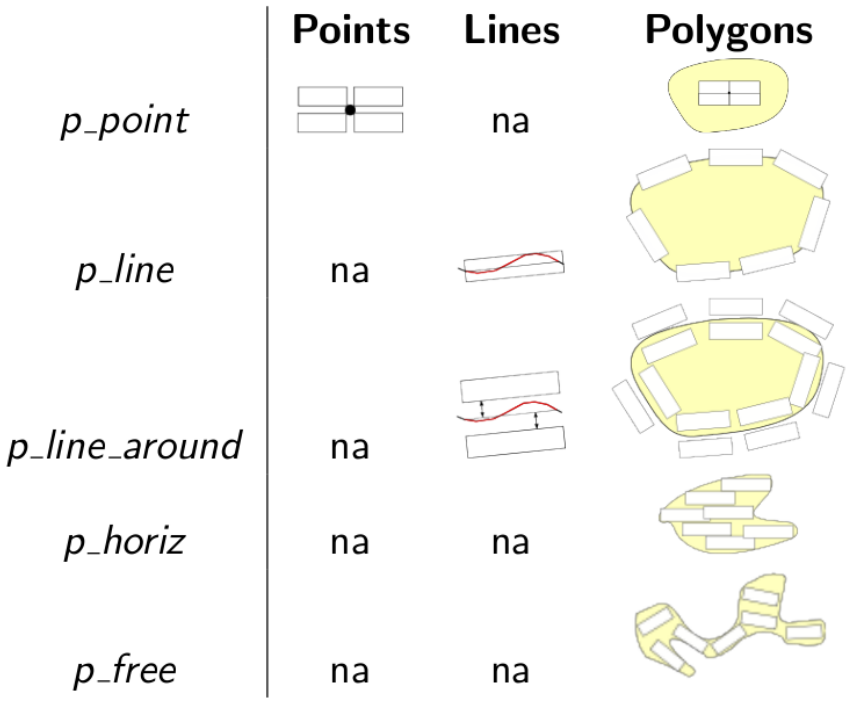

Figure 1: Arrangement mode for label, depending on geometry type

2.3 A vector layer is removed

- remove the layer with pal::Pal::removeLayer(the_layer)

2.4 Zooming in or zooming out, panning

- remove all displayed labels
- run the labelling process (pal::Pal::labeller())
  - scale
  - mapExtent
  - PalStat pointer to an unalloced PalStat pointer (can be NULL)
  - if *displayAll* is set to *true*, pal will label every feature even if overlaps occurs. Nevertheless, pal tries to minimize the amount of overlaps
- draw labels computed by pal::Pal::labeller(). The result is a linked list or Label ($std :: list < Label* >$, see Table 1)

2.5 Show or hide a layer, (scale depend rendering, final user toggle hides/shows the layer, etc.)

- *the_layer* →setActive(true/false)

method	description
double getOrigX()	return the down-left x coordinate of the label
double getOrigY()	return the down-left y coordinate of the label
double getX(size_t i)	get a specific x coordinate (0 is down-left, 1 down-right, 2 up-right and 3 is up-left)
double getY(size_t i)	get a specific y coordinate (0 is down-left, 1 down-right, 2 up-right and 3 is up-left)
double getRotation()	return the rotation of the label [rad]
const char * getLayerName()	return layer's name
const char * getFeatureId()	return feature's id
PalGeometry * getGeometry()	return a pointer to the userGeom registred in the layer

Table 1: Description of a Label object, which describe where and how dispay the label on the map

- run the labelling process (see 2.4)
- When the layer is not active, pal will completly ignore it

2.6 Toggle labelling of a layer

- *the_layer* →setToLabel(true/false)
- run the labelling process (see 2.4)

2.7 Label's style change for a layer (font, font size, text, etc.)

- for each feature of the layer, set the new label size with *the_layer* →setFeatureLabelSize()

2.8 Symbology change for a layer of point

- give the radius (in pixel) of the circle that contains the symbol for each features of the layer with *the_layer* →setFeatureDistlabel()

2.9 Symbology change for a layer of line

- give the line width (in pixel) for each features of the layer with *the_layer* →setFeatureDistlabel()

2.10 Arrangement mode change for a layer

- set the new mode with *the_layer* →setArrangement()

2.11 Unit of label size change for a layer

- set the new unit with *the_layer* →setLabelUnit()
- set the new label size for every feature of the layer with *the_layer* →setFeatureLabelSize()

2.12 A priority of a layer change

- set the new priority with *the_layer* →setPriority()

2.13 The scale depend labelling range change for a layer

- set the new range with *the_layer* →set{Min,Max}Scale()
- the value -1 will deactivate min or max threshold