



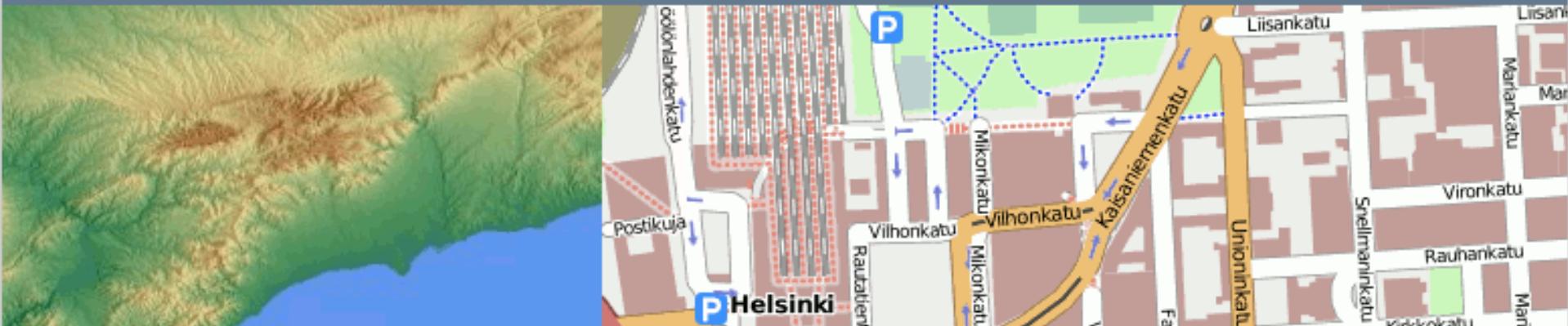
we'd  
~~What I'd~~ like to do with mapnik



Steve Chilton



© OpenStreetMap contributors, CC-BY-SA



With particular thanks to: **Artem, Dane, Lennard, Colin, Jon**



# Problems in non-manual cartography

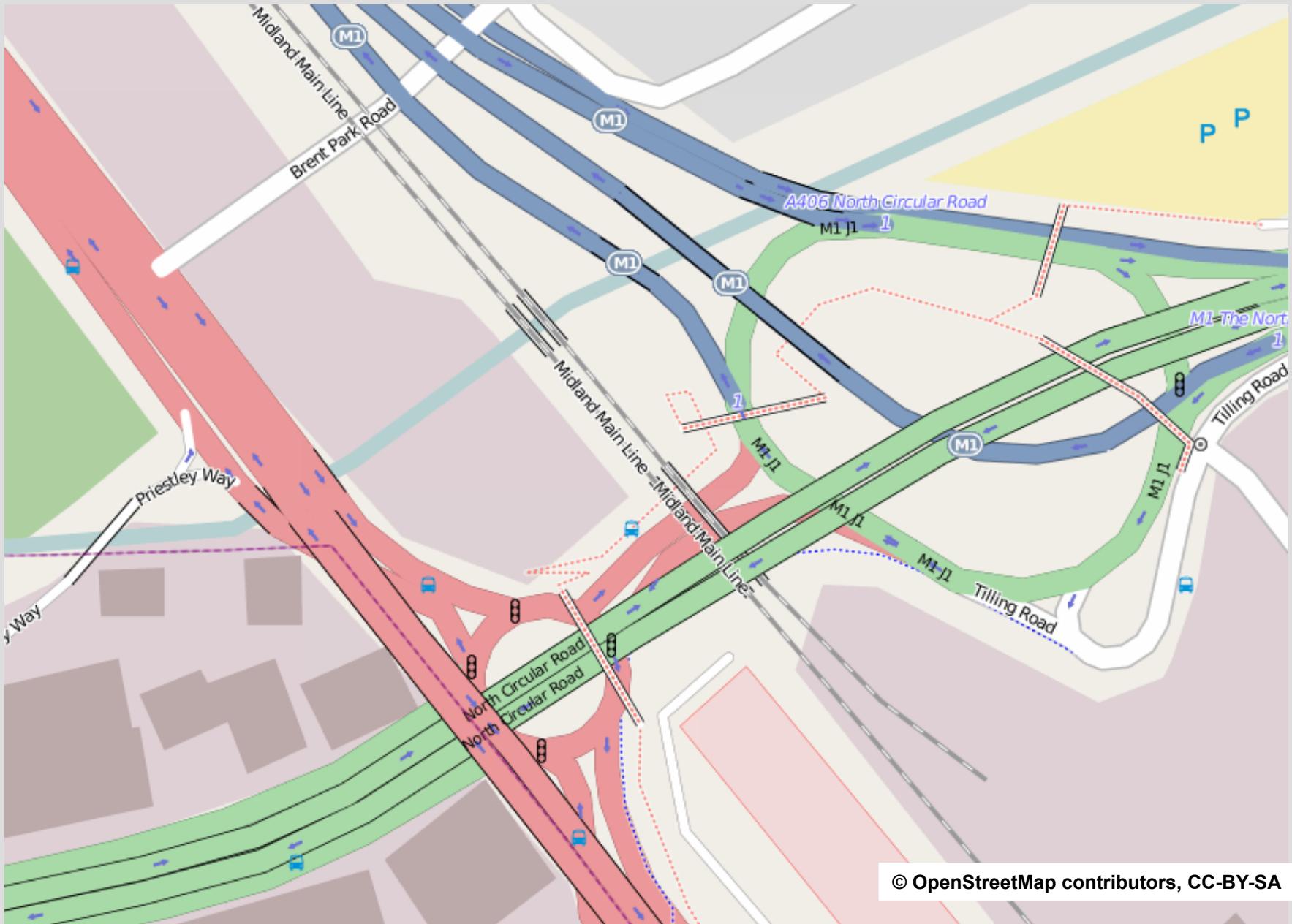
.... three classic problems that computer-generated map output has to deal with. These are **generalization**, **type placement**, and **overlaid detail** on other detail.

*It will be our task as a project to incorporate these principles and techniques as the project develops, solving problems along the way. I am sure it will be an interesting ride.*

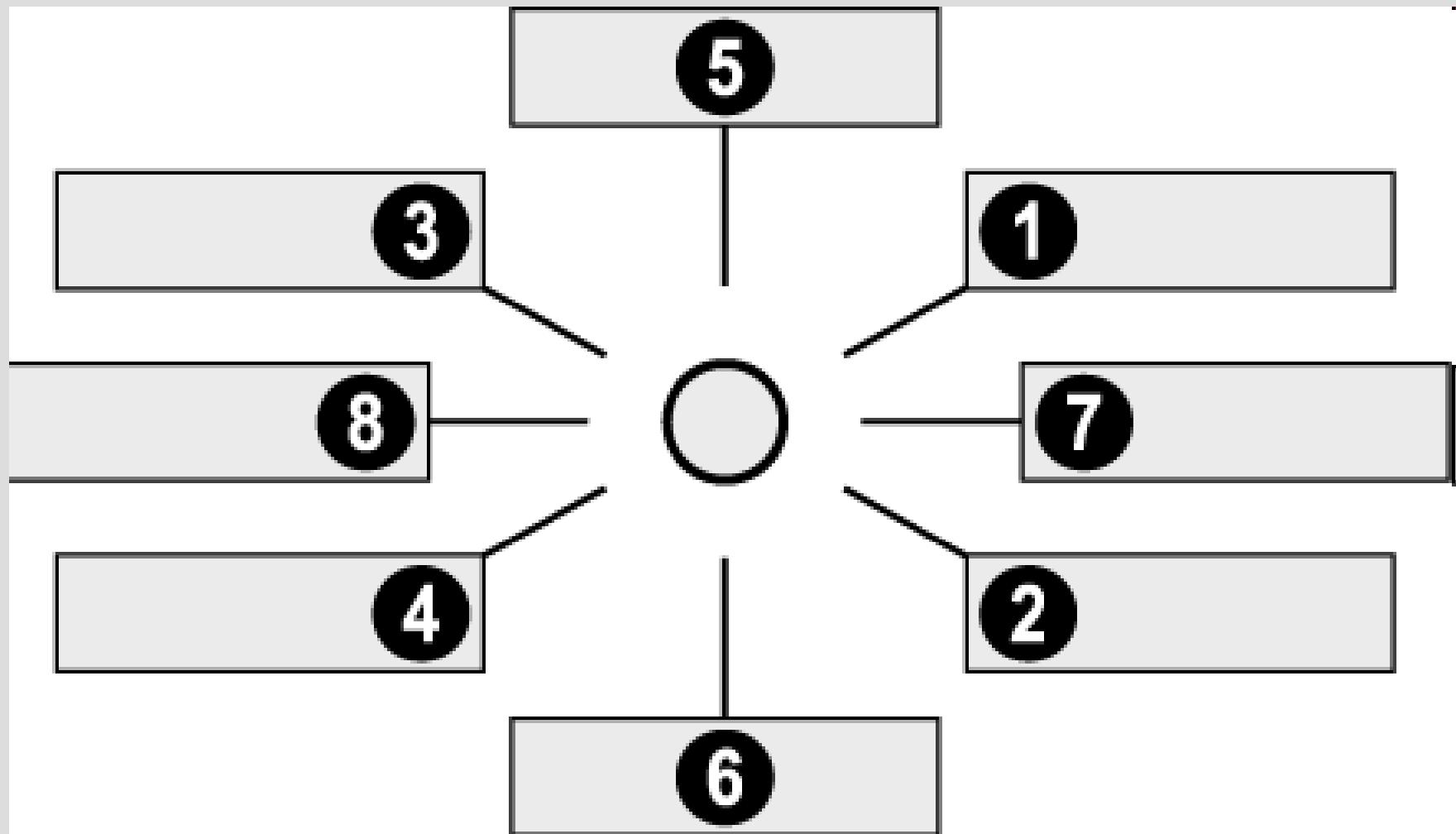
# Respect the layer tag implicitly



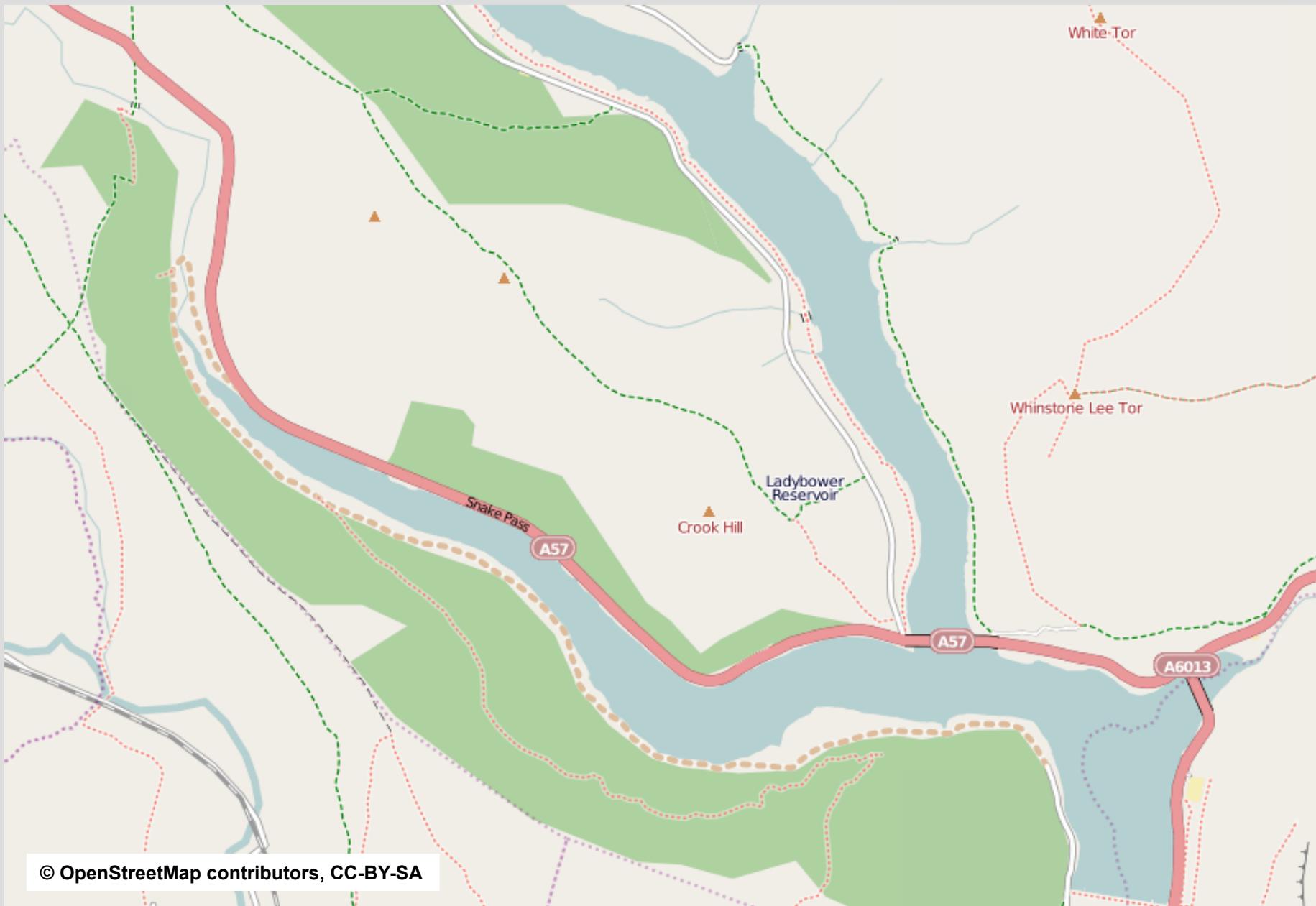
# Not place labels of underlying roads on bridges



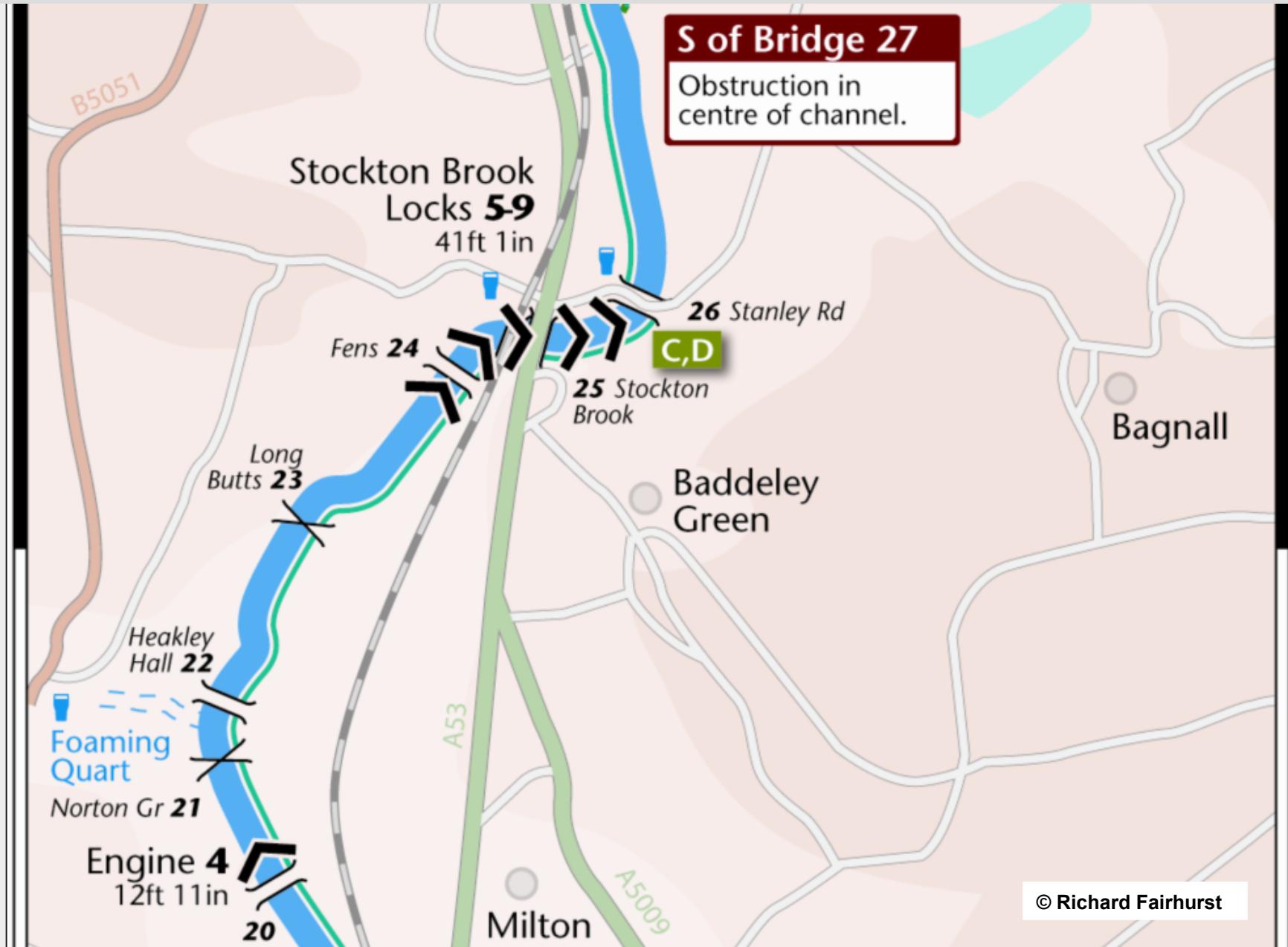
# Iterate alternative label placements



# Apply point-in-polygon to text placement



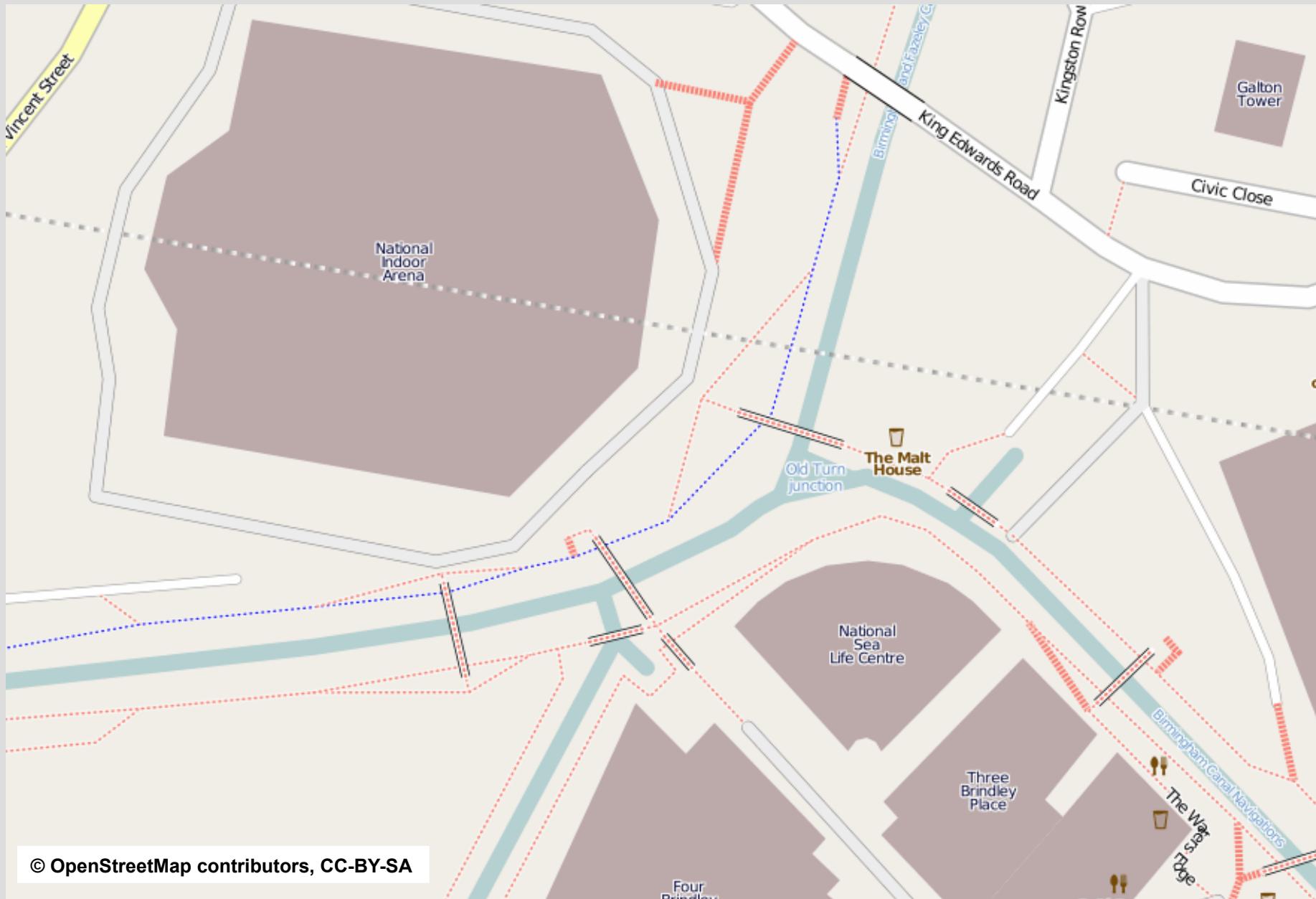
# Allow rotation of icons



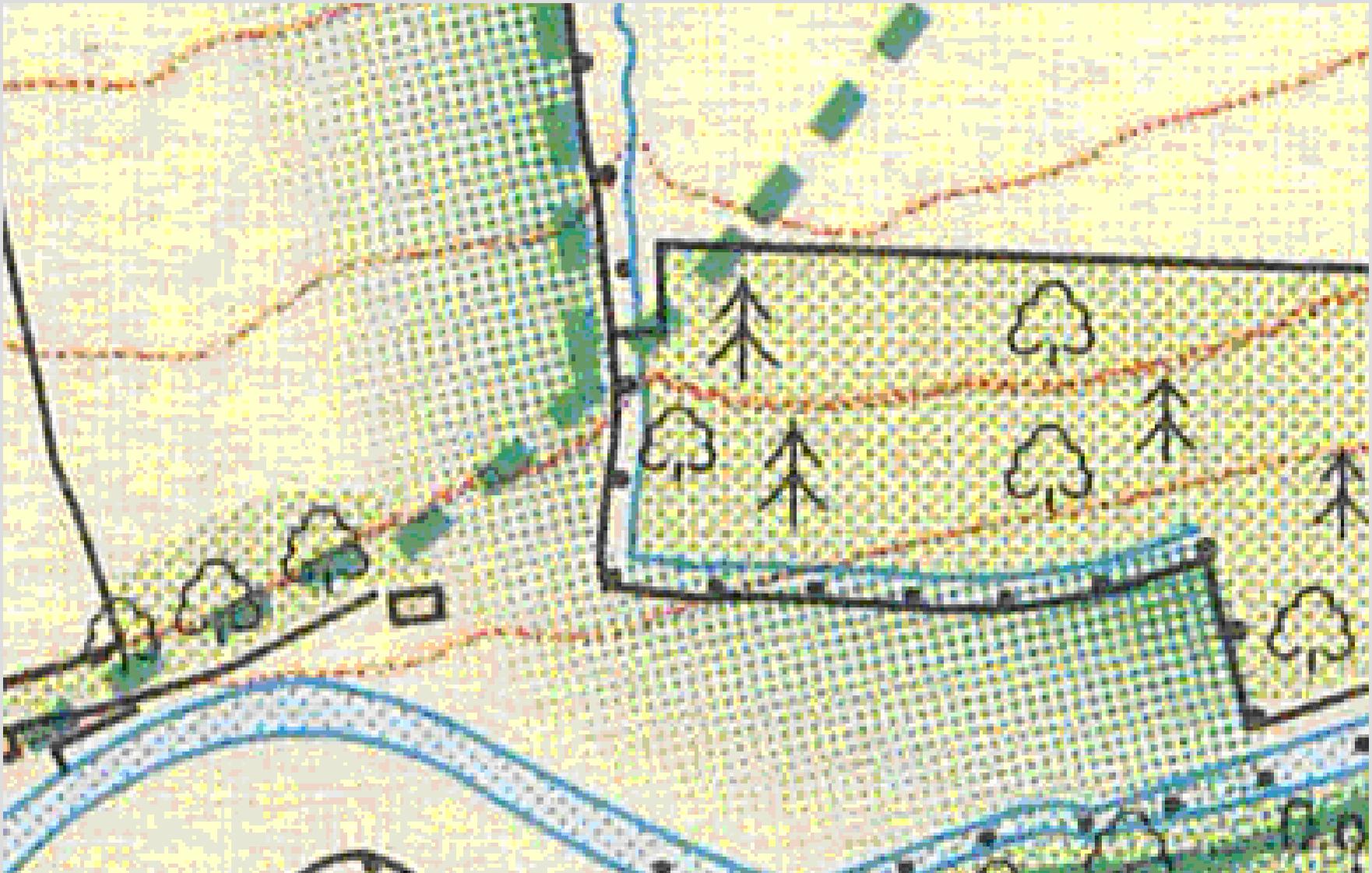
# Accept SVG icons



# Apply variable width to canals/rivers

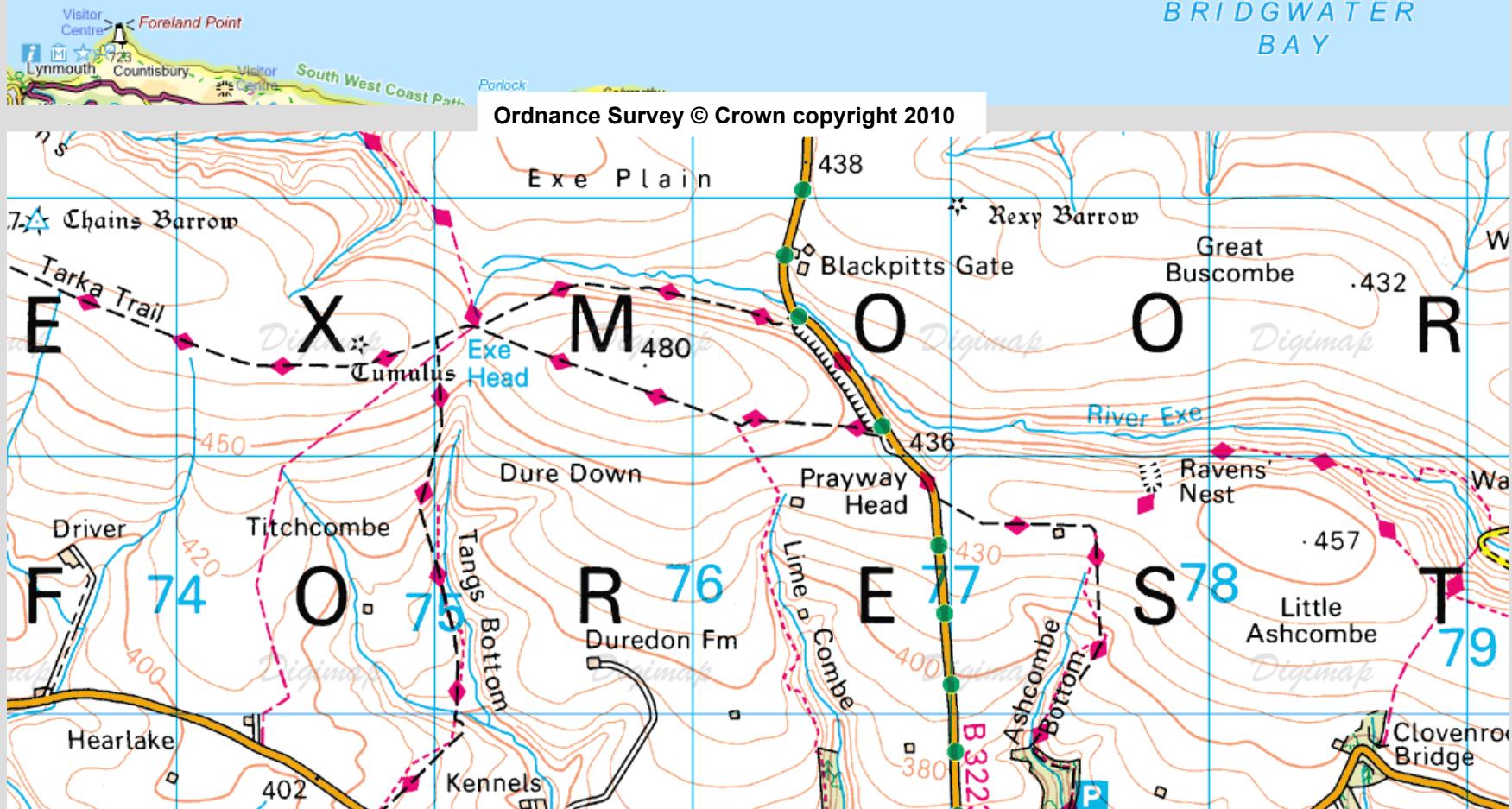


# Apply vignettes inside polygons

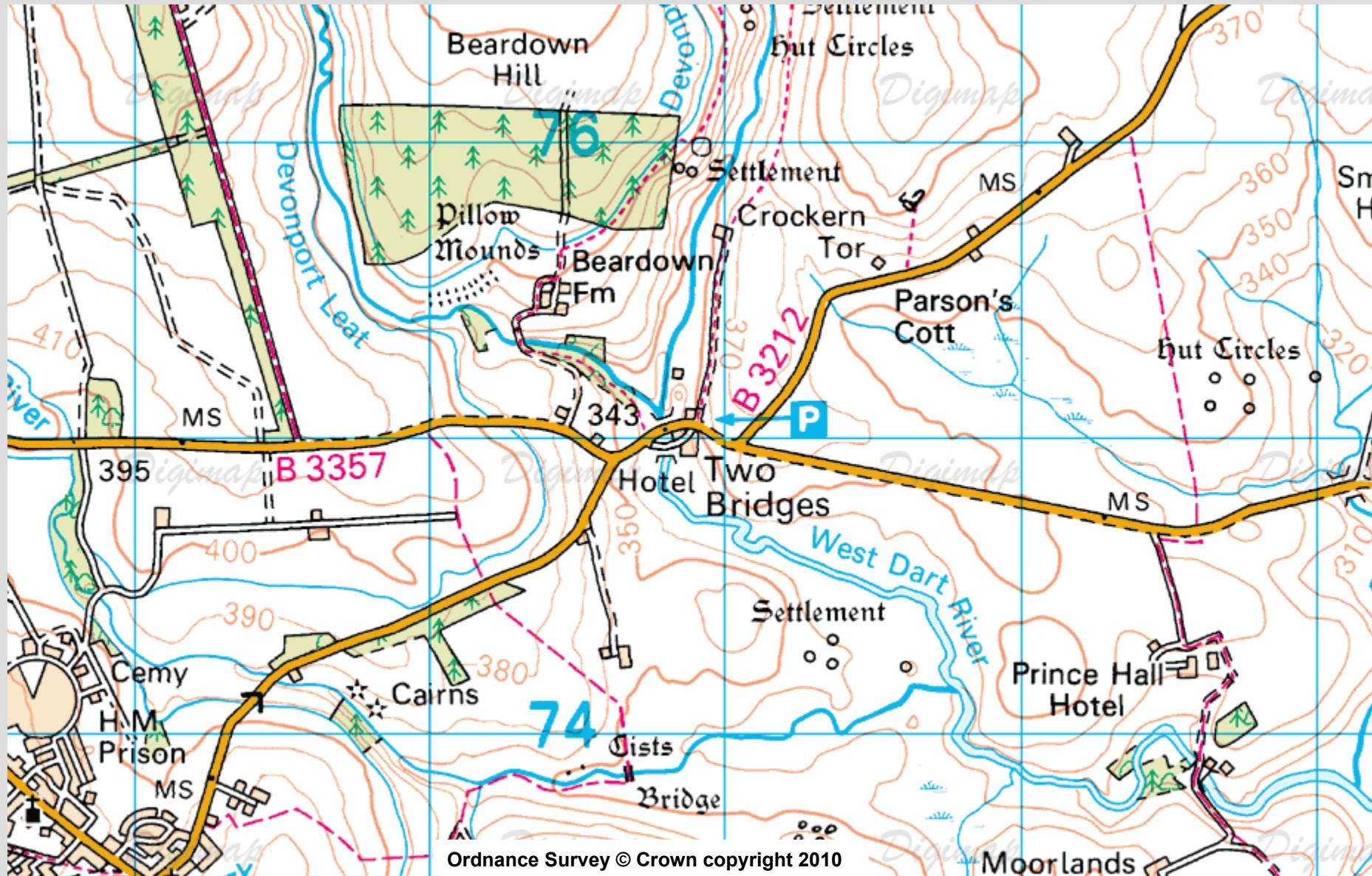


# Produce spread text labels

# Digimap Bristol Channel



# Different casing either side of roads



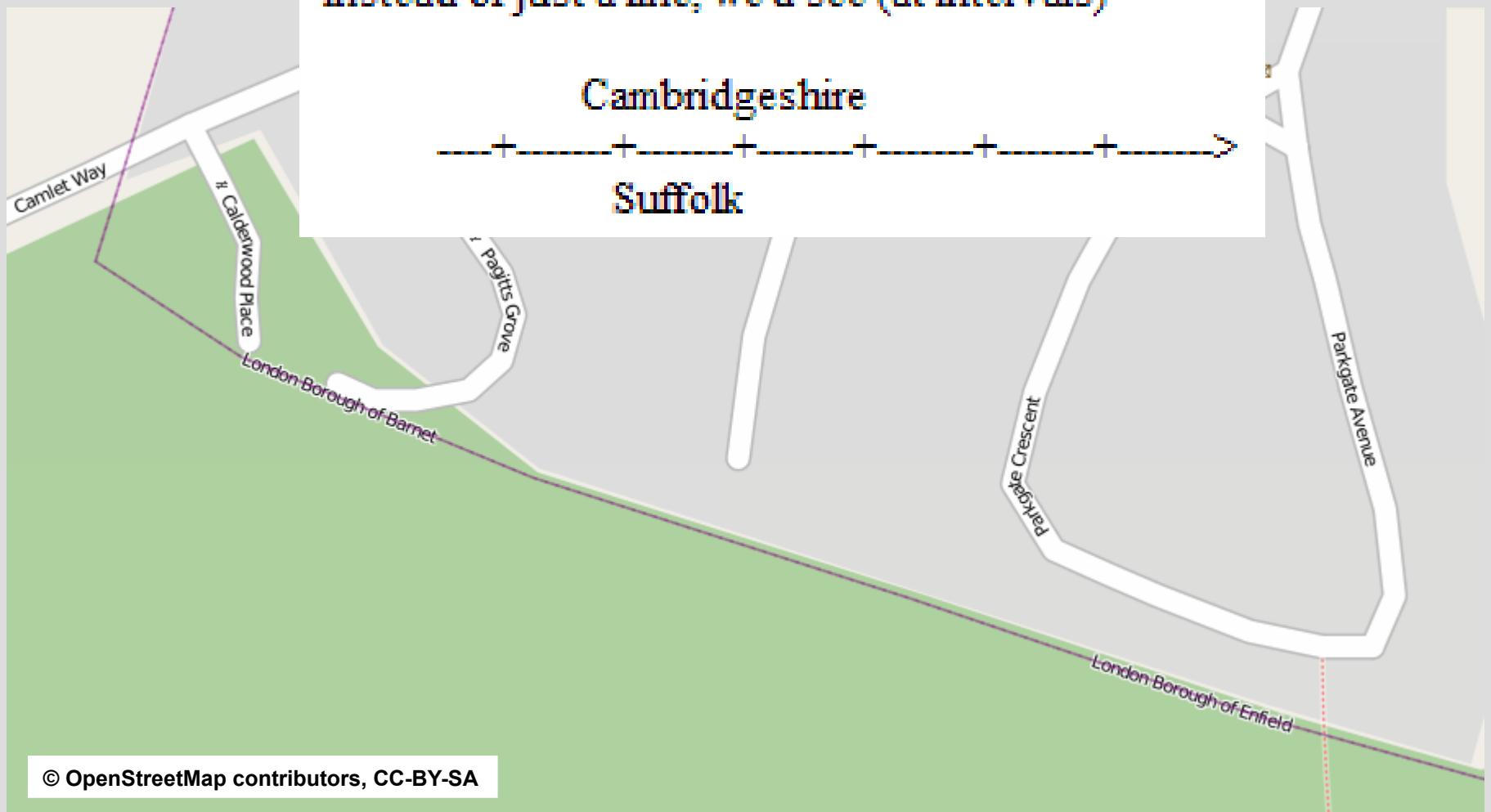
# Text labels either side of a line

`boundary=administrative`

`left:county=Cambridgeshire`

`right:county=Suffolk`

instead of just a line, we'd see (at intervals)



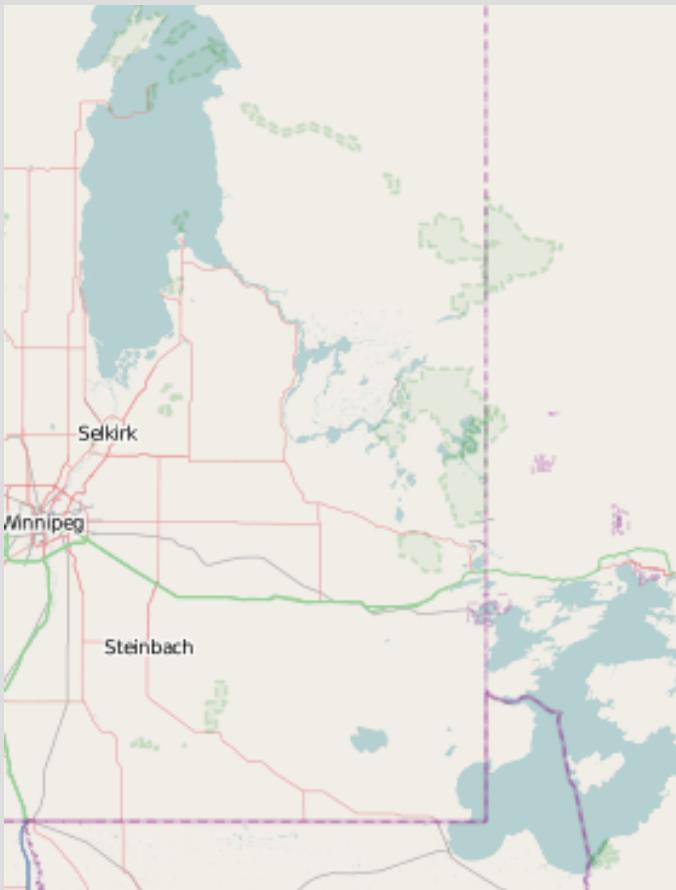
# Nudge an icon a bit



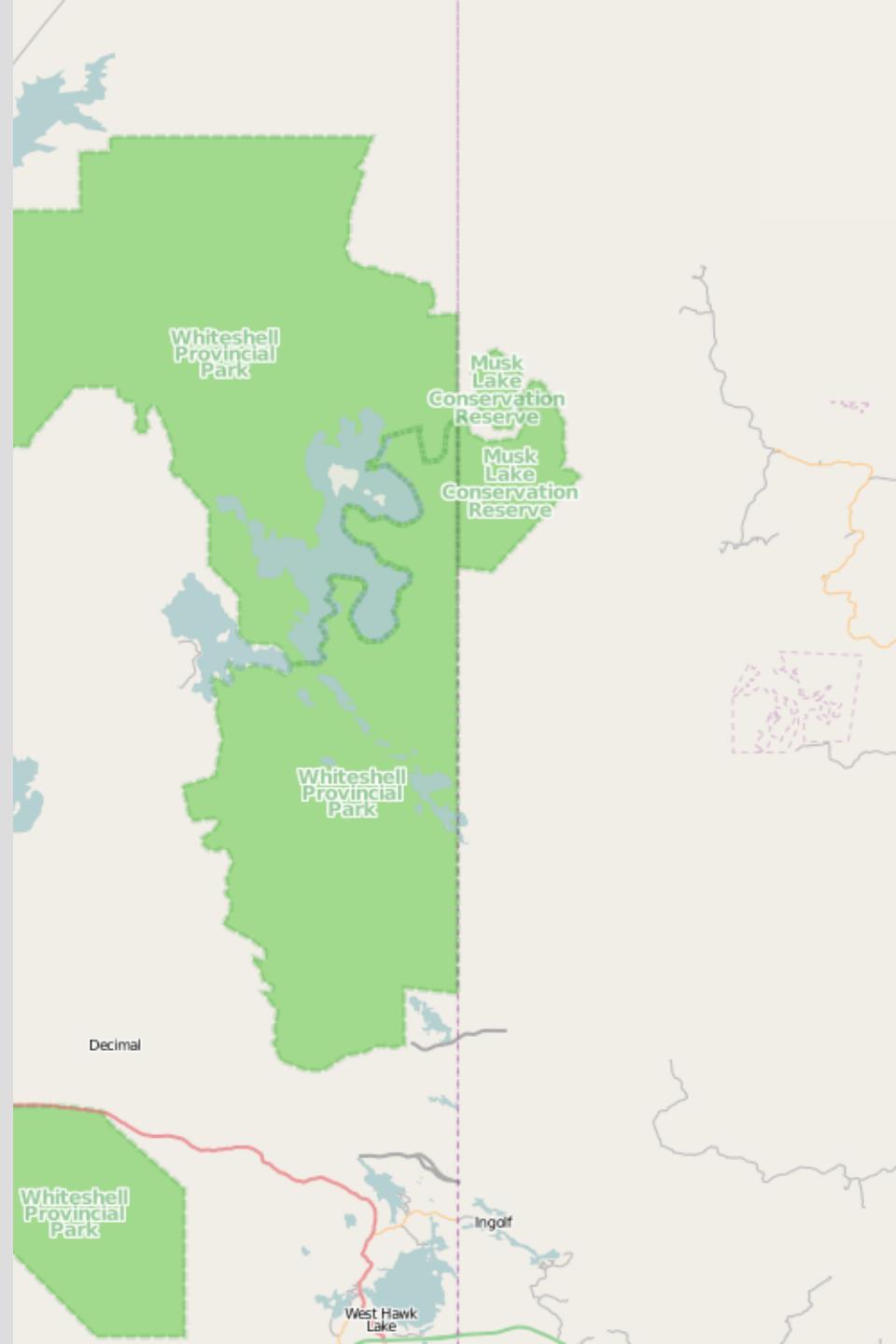
# Mountain range names following a hint line



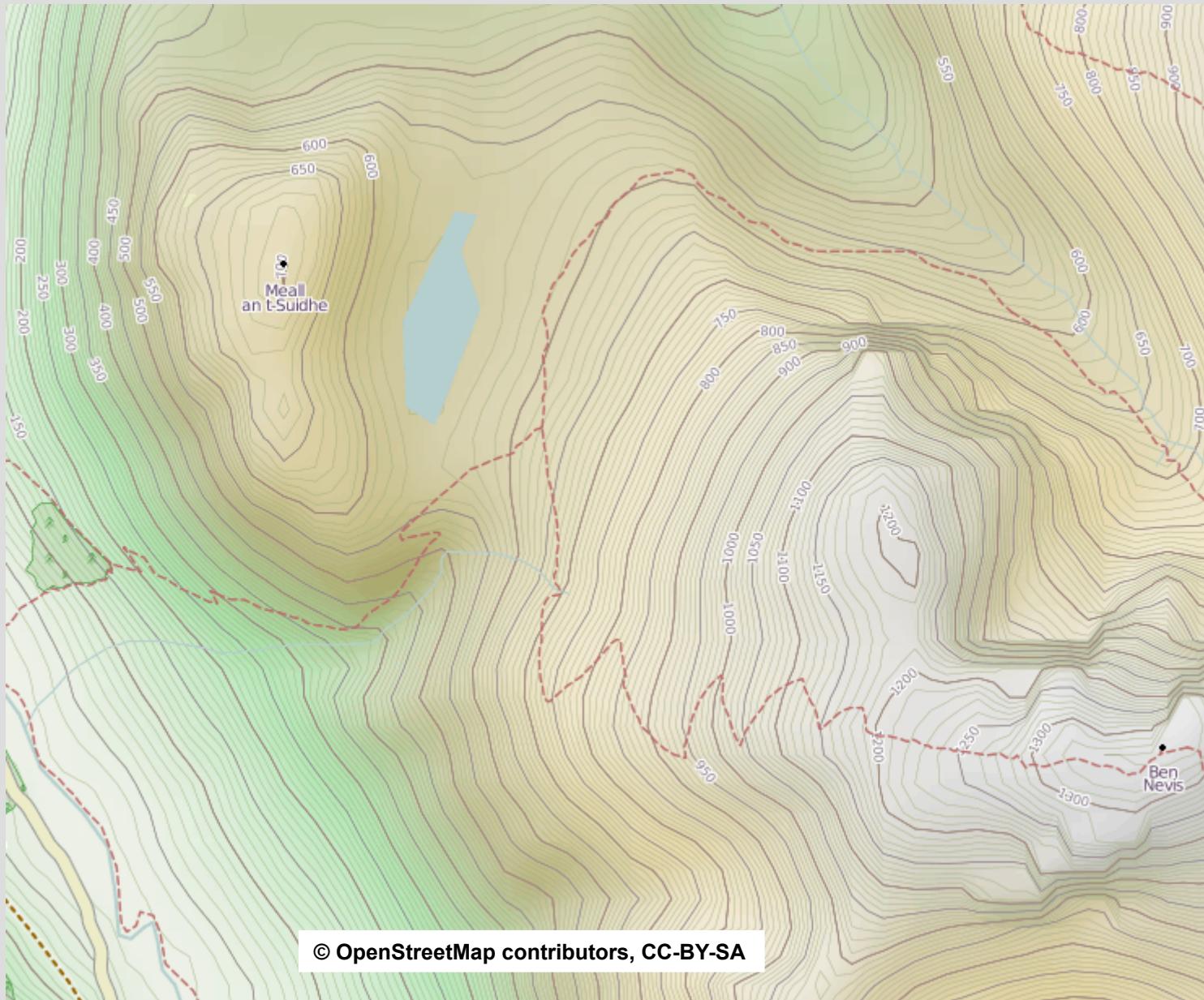
# Collapse small areas into larger one



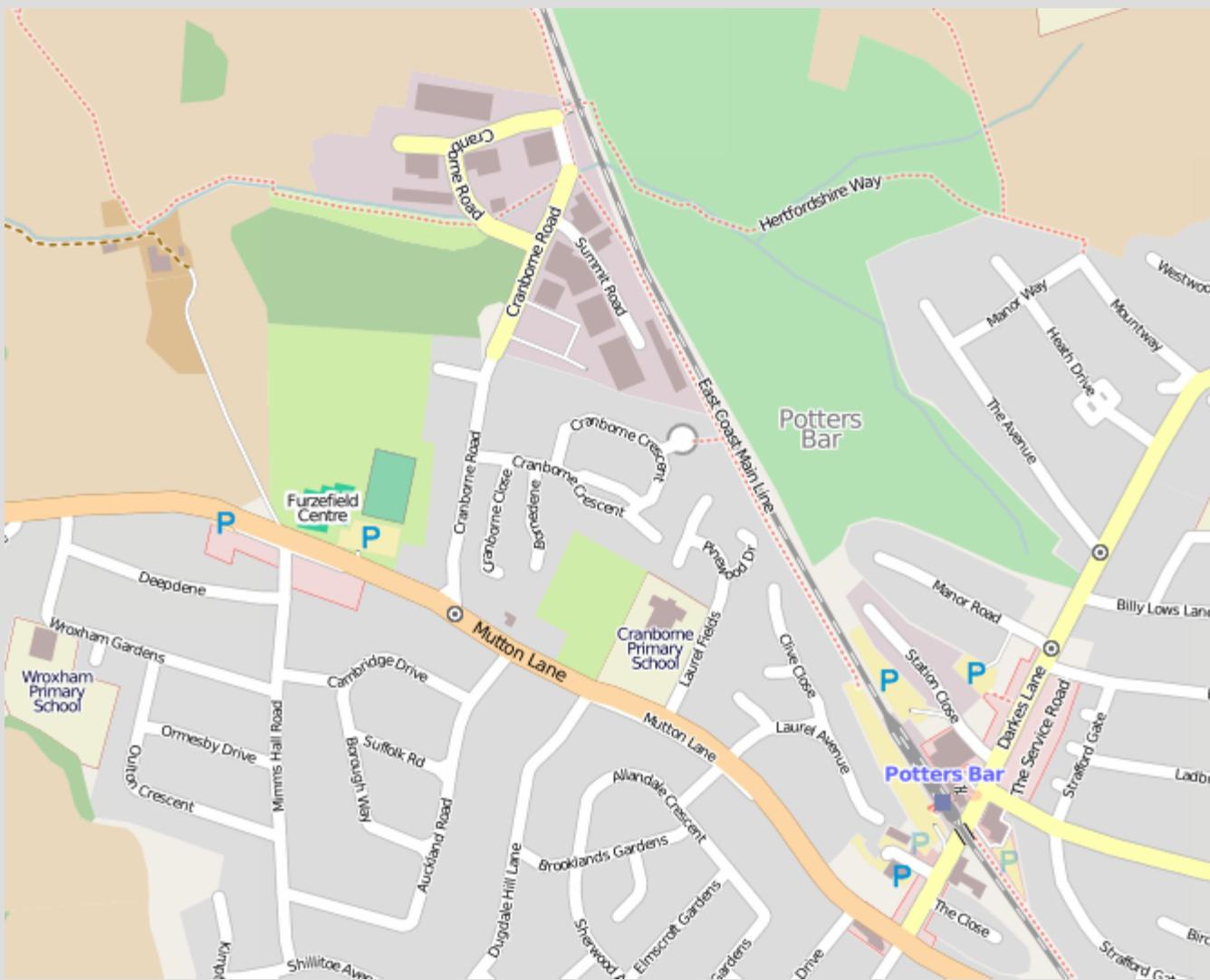
© OpenStreetMap contributors, CC-BY-SA



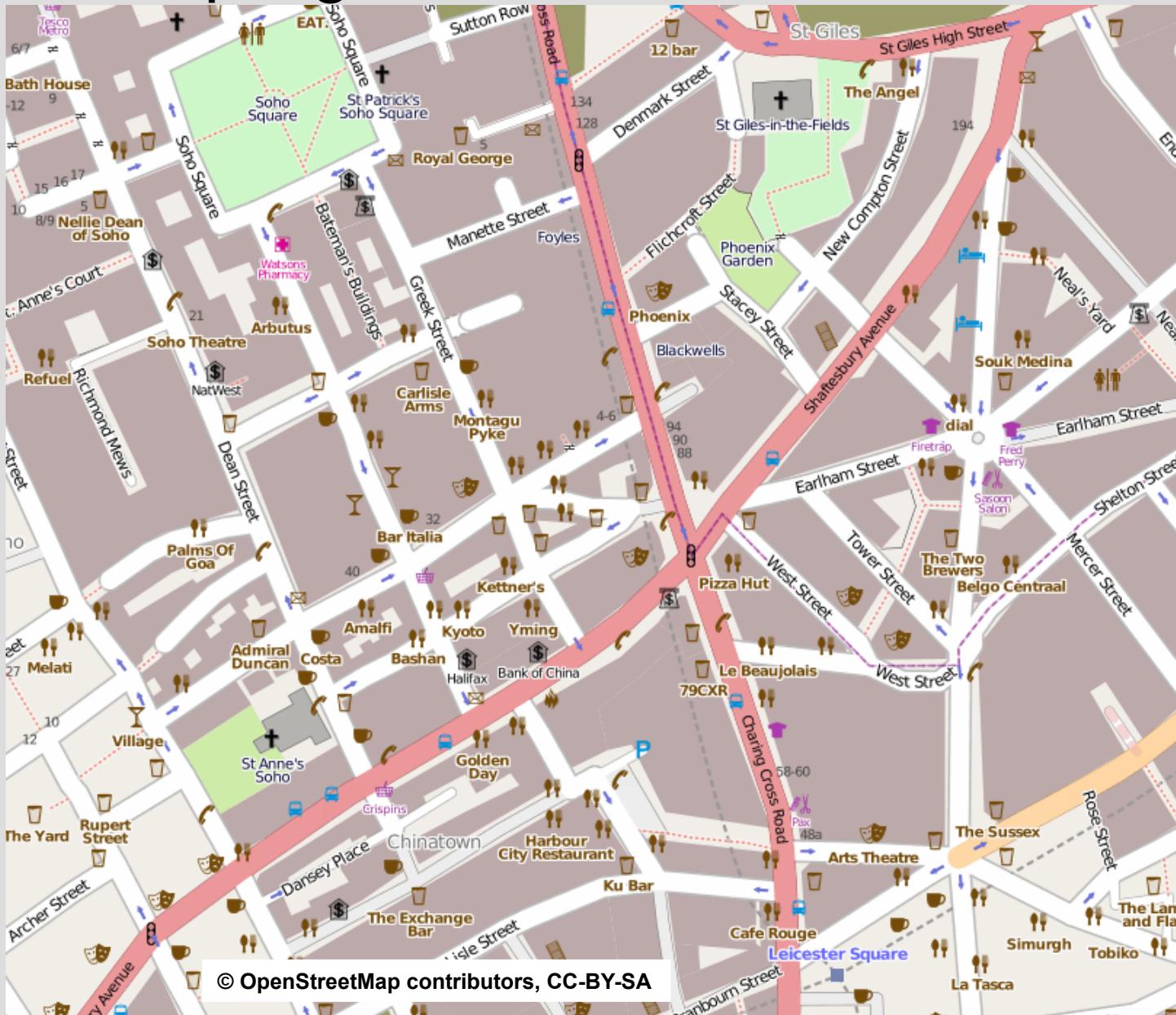
# More control over text rotation



# Combine point and line data in single style



# Coupling between icon and text.



# Separate two lines by minimum distance



# Add more Natural Earth data at low zooms



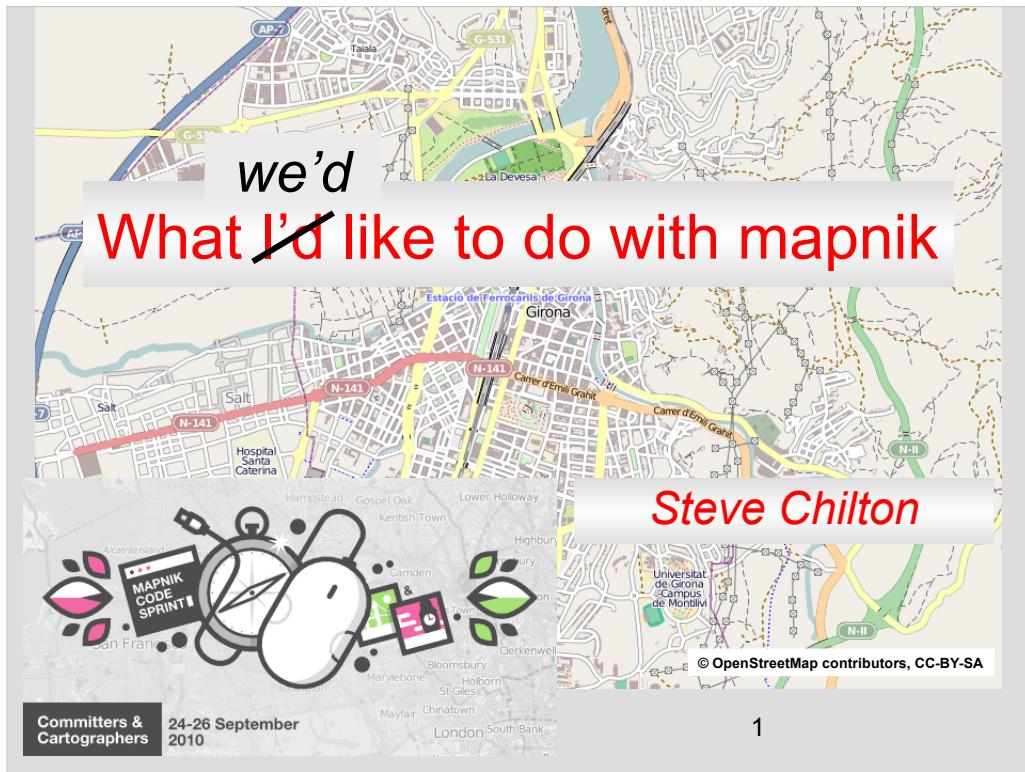


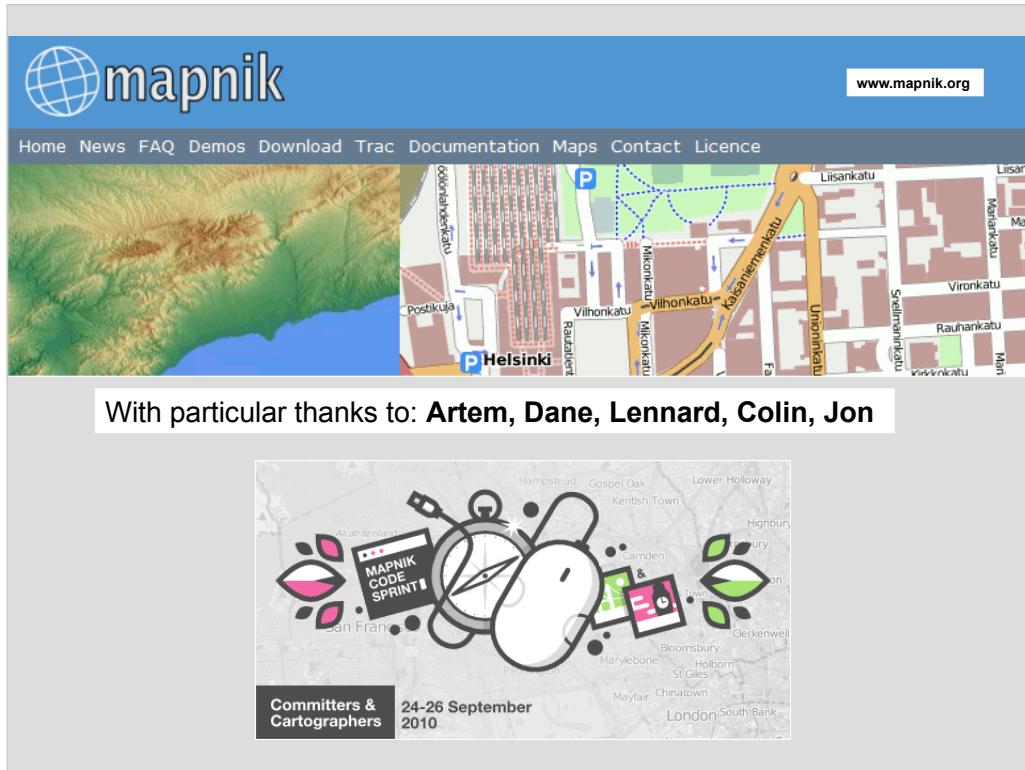
*I'm going*  
~~What I'd like~~ to do with mapnik

Steve Chilton

Email: [steve8@mdx.mdx.ac.uk](mailto:steve8@mdx.mdx.ac.uk)

Twitter: [steev8](#)





Artem

Dane

Lennard

Colin

Jon

# Problems in non-manual cartography

.... three classic problems that computer-generated map output has to deal with. These are **generalization**, **type placement**, and **overlays of detail** on other detail.

*It will be our task as a project to incorporate these principles and techniques as the project develops, solving problems along the way. I am sure it will be an interesting ride.*



Quotes from SOTM07?

..... at end of my talk?

Still holds true....

## Respect the layer tag implicitly



## Respect the layer tag implicitly

*ds: No way to do this quite yet. But, we are half way there. Mapnik2 now has the ability to use “expressions” to pull values out the data for each feature and put those to use in determining symbolizer behavior. So currently it is now possible in Mapnik2 to pull the ‘layer’ tag. Next step will be thinking through how/if that could be used to determine paint order. Will need more thought.*

*Ldp: it's not so easy as it sounds. The order of <Layer> appearances is what drives object rendering order. The OSM stylesheet is split over dozens of layers, and ordering can currently only work within a <Layer>. Osmarender takes the approach of pulling every object (from the .osm file) and sorting them by layer=\*, and then iterates a huge list of object styles. The advantage is that sorting by layer=\* is global. Quite the opposite of what we are doing in the mapnik OSM stylesheet. We could mimic their way, but we'd then have only a single <Layer> with a huge <Style> for every geographic object. I don't think that's the direction we want to go.*

?

## Not place labels of underlying roads on bridges



### Understand bridges and not place incorrect labels on them

*Ldp: invisible dummy text on bridges, or any other geometric feature that we don't want text to render on, before we render the actual road names. This puts the extent of the bridge in the 'collision list' and other road names would then avoid it.*

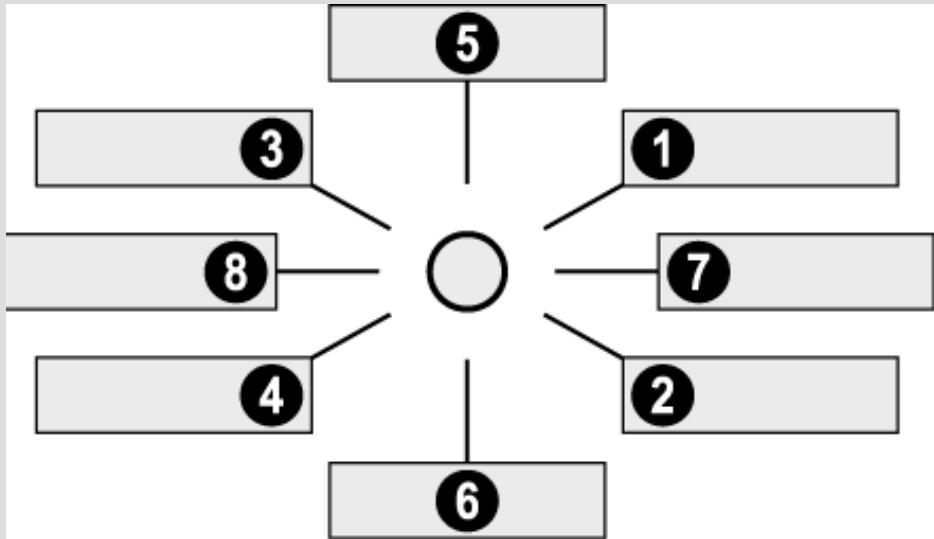
*Instead of an invisible dummy text, explicit support from mapnik to be able to place a line feature into the collision list would be an eventual goal.*

*This would also work for canal names. Currently, we render them very early on, but any bridges (or tunnels) now overpaint the canal name. If we could swap this to 1) render canal way 2) render bridge and any regular name that would apply to it 3) declare any remaining bridges that have no text yet as off-limits to further text placement 4) render canal names. These now avoid the bridges.*

*Basically, the idea is to be able to involve <LineSymbolizer> into the placement algorythms.*

?

## Iterate alternative label placements



6

## Apply a displacement when naming nodes/areas

For example if a collision occurs iterate through other possible placements.  
Would be complicated, but there are algorithms out there.

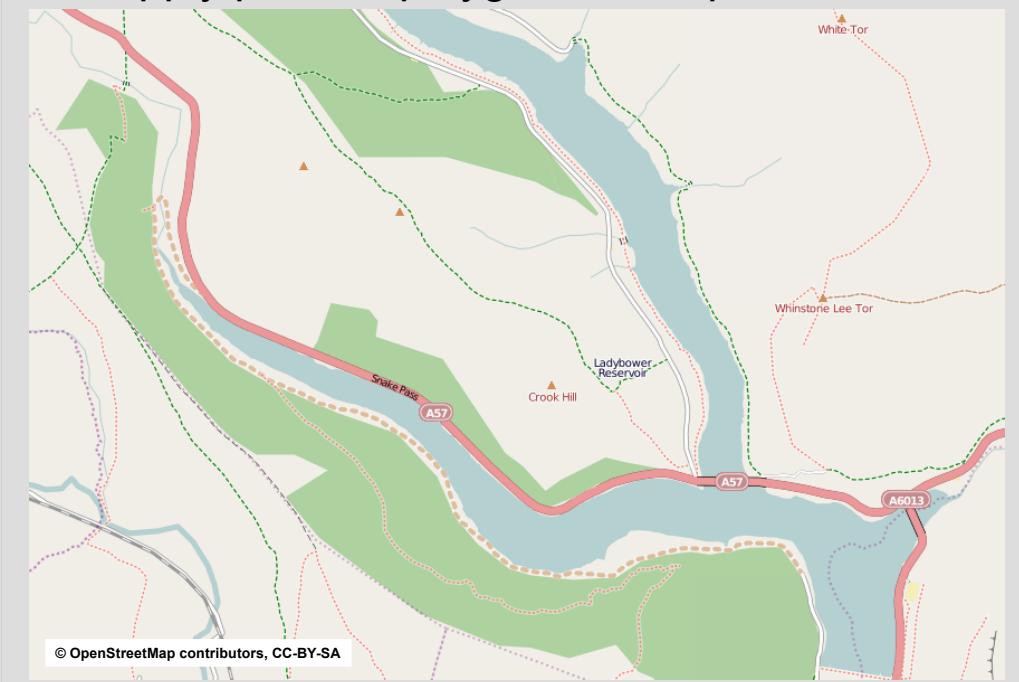
Ds: *not yet possible, but we have a patch for it from a russian cartography group: <http://trac.mapnik.org/ticket/463>.*

So, I could see this happening sometime **in the next year** (limit is just developer availability and funding, since this requires quite a bit of coding)

Ldp: *this is a feature that I would dearly like to see.*

N

## Apply point-in-polygon to text placement



### Place text labels for polygons more intelligently

For instance at the moment the name for a crescent shaped area fill will be placed at the centre of gravity of the shape, which can easily be outside the area itself.

*Ds: not yet possible, but we are currently discussing adding support for “**point\_on\_surface**” algorithm for **point in polygon** placement, that would ensure the label is within the polygon.*

*Ldp: exactly. We could already have a secondary geometry column in the polygon table, and have PostGIS calculate the ST\_PointOnSurface (either through triggers or explicitly from osm2pgsql), but explicit support from mapnik is more flexible, as it would also work for other input plugins..*

?

## Allow rotation of icons



### Apply a rotation to an icon

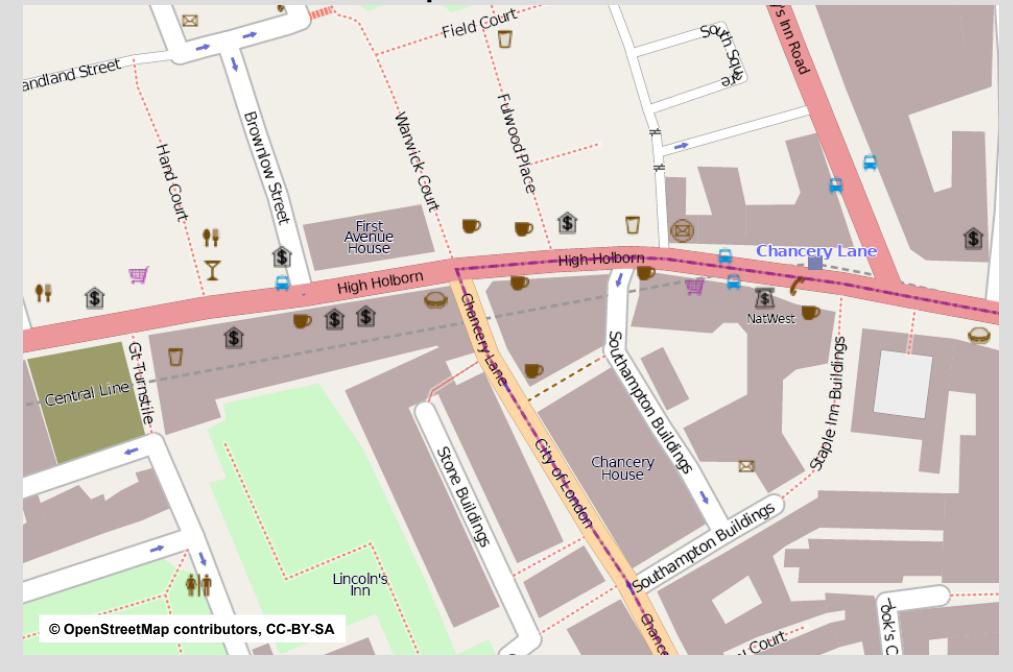
Based on the orientation of the node/way in the db or could be used for viewpoint when tagged with viewing angle, or to align a mountain pass symbol

*ds:we are planning on adding this soon. Basically, this requires the “expressions” work as noted above to be able to pull rendering directives directly from the database, so we are most of the way there. Rotation for the PointSymbolizer or LineSymbolizer, or both? I’d assume both!*

*Ldp: PointSymbolizer. Two things I’d like to see. 1) Basic rotation by defining a rotation angle in the symbolize. Either explicitly or using the new expressions engine to make it dynamic based on the data. The example shown is for locks, but it applies equally to mountain passes, where you’d want to rotate the symbol to align with an underlying way. Currently, the point and the way are completely separate concepts, and when rendering a lock\_gate node, it doesn’t know about the direction of the canal, nor that it’s even \*in\* the canal way. Something would have to be found to take the node, take a way, find out if they are related, and then translate that into a rotation angle. Quite tricky.*

?

## Accept SVG icons



## Accept SVG icons

At the moment PNG resolution is OK on screen but is let down when you use, for instance, PDF export - whereby lines and text are vectors but icons still are bitmaps

*ds: Artem has actually been working on this in the **last two weeks**. So, ability to READ svg icons is near. Having the ability to **keep icons as SVG/Vectors** in PDF output is not yet possible (not sure when it will be implemented), but at least the ability to read SVG (and render to PNG) will help ensure they are **not pixellated in PDF output** because they can be scaled up or down (depending of course on output size/resolution)*

?

## Apply variable width to canals/rivers



### Apply variable widths to canals and rivers

Based on the `width=tag`. At highest zooms, rivers and canals should be able to be displayed proportional to their actual width (if recorded).

*ds: great idea. This is not possible yet, but I know how to implement, so we can add in the future.*

*Ldp: again, we could probably translate the `width=*` tag into a **LineSymbolizer width**, depending on scale, but I fear the result would not work as nicely as you'd want. For example, what would you do at the spot where `width=*` changes? **Abruptly change the visual width**, or have a short span where it gradually changes width? The latter would require mapnik support.*

?

## Apply vignettes inside polygons



11

## Produce vignettes for areas

An example might be a National Park which encompasses a large area and many other landuses within it.

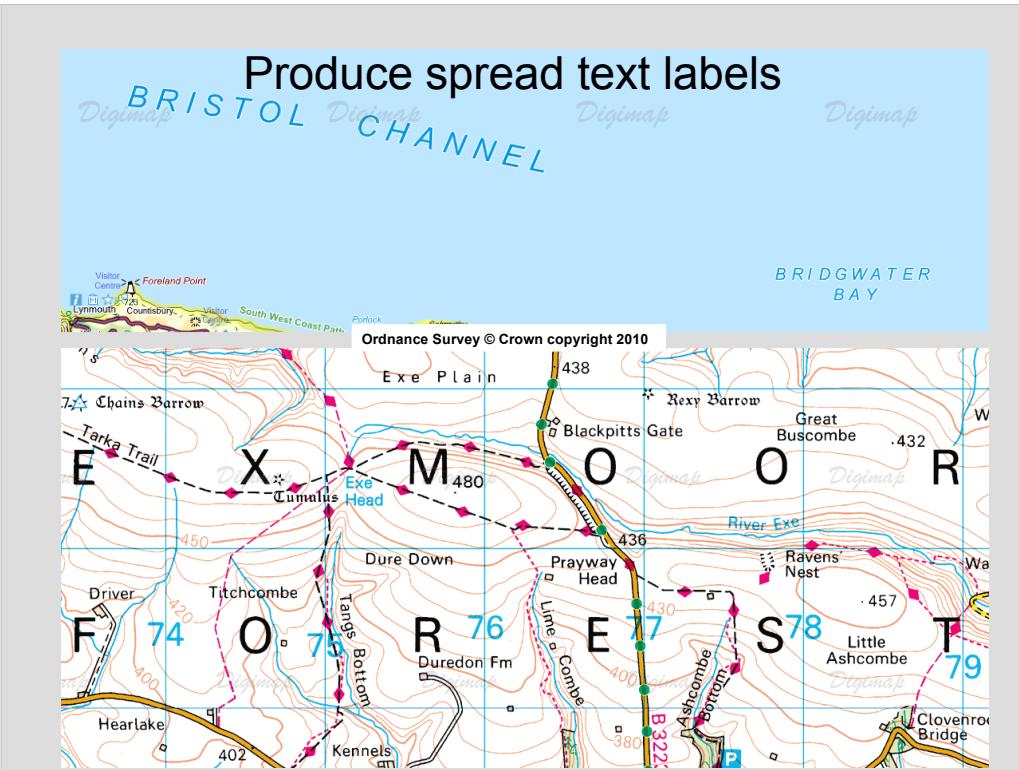
ds: **cannot do this yet.**

Ldp: *The vignette is the shaded border inside the outer extent of a large area. This should taper off the further from the outside you get.*

*There is one way we could already build this, using Postgis to **build a specific geometry for where we want the vignette**, but the tapering off would not be possible then. Also, building a geometry is much more cumbersome and slower than doing it in mapnik's pixel space.*

*Having **random symbol placement within polygons** is something else, but also good to have.*

N



## Produce spread text

An example might be naming a country or a mountain range, which extends over a wide area

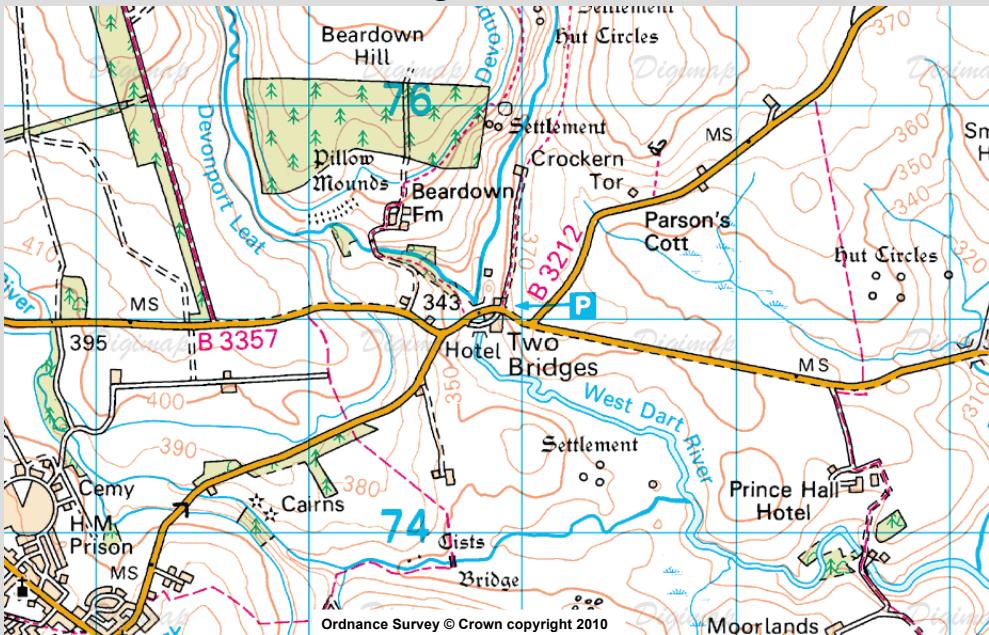
*Ds: This can be done in Mapnik 0.7.1, but not yet for lines. See the parameter 'character\_spacing'  
here: <http://trac.mapnik.org/wiki/TextSymbolizer>*

*Ldp: The current support is static. What would be needed in future is a way to make the font size and character/line spacing **dependent on the extent of the feature**. Could again be done with a bit of SQL (we already have the way\_area column from osm2pgsql), but a more generic way might be better.*

*Using only way\_area would also not work for long and narrow features.*

?

## Different casing either side of roads



Put a casing on one side of a way, or different on either side

OS: Unfenced roads or tracks

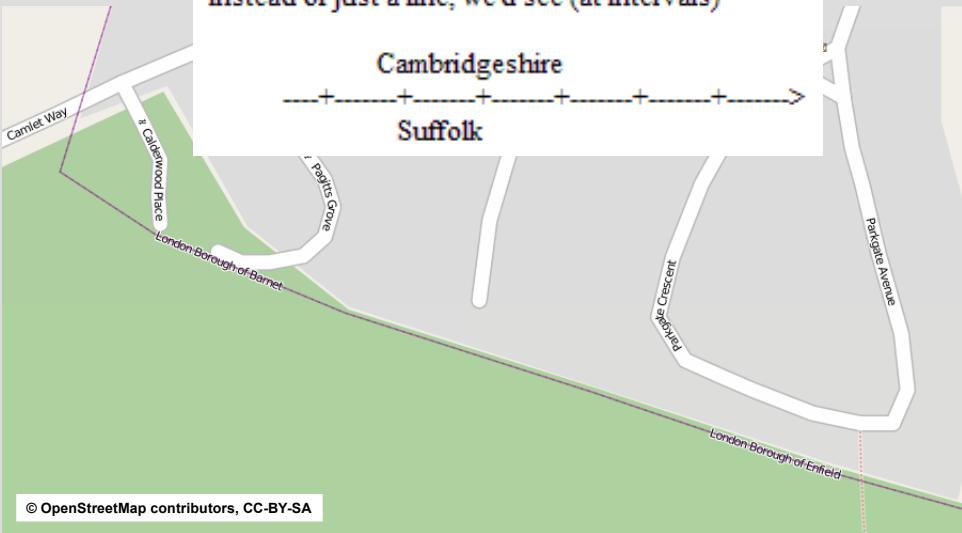
This might be what was required in the case of cycleway which appears on only one side of a highway, or on a road with verge on one side only

ds: *Not yet possible, but I have been working on adding support for Parallel lines: <http://trac.mapnik.org/ticket/180>. I hope to find time to work on this in the future, but not sure when. It will still not work properly for roads that need to be cased, but we can give that more thought.*

N

## Text labels either side of a line

```
boundary=administrative  
left:county=Cambridgeshire  
right:county=Suffolk  
instead of just a line, we'd see (at intervals)
```



### Put two text labels on either side of a line

An example might be the names of adjacent counties/parishes being shown either side of the administrative boundary

*ds: great idea. This **could be tricky**, I will think more about it!*

*Ldp: I really don't think this is hard to do. Although **keeping the labels on either side aligned to each other is the actual hard part**. We also have the issue that a **boundary could hold several labels**, one each for the various administrative levels involved. Country vs region vs municipality vs place, for instance.*

?

## Nudge an icon a bit



Nudge an icon a bit if it would collide with others  
See restaurants in centre

*ds: can't do this yet.*

## Mountain range names following a hint line

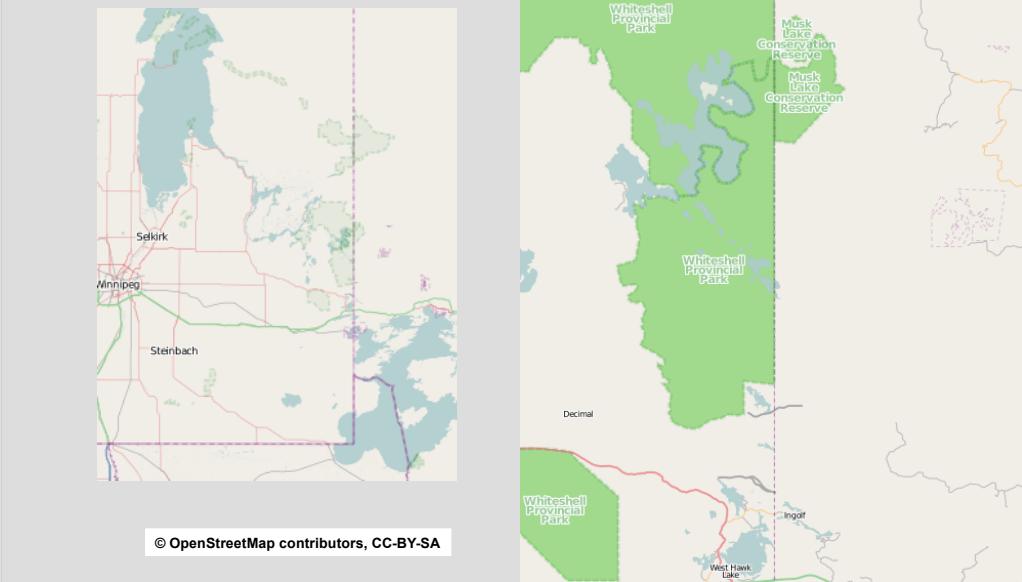


See Himalayas and Urals

*ds: can't do yet, not sure how to do this, but I will think more...*

*Ldp: again, one could build a dedicated geometry in geographic space to then place text on in mapnik. And again as well: support in pixel space from within mapnik sounds better..*

## Collapse small areas into larger one

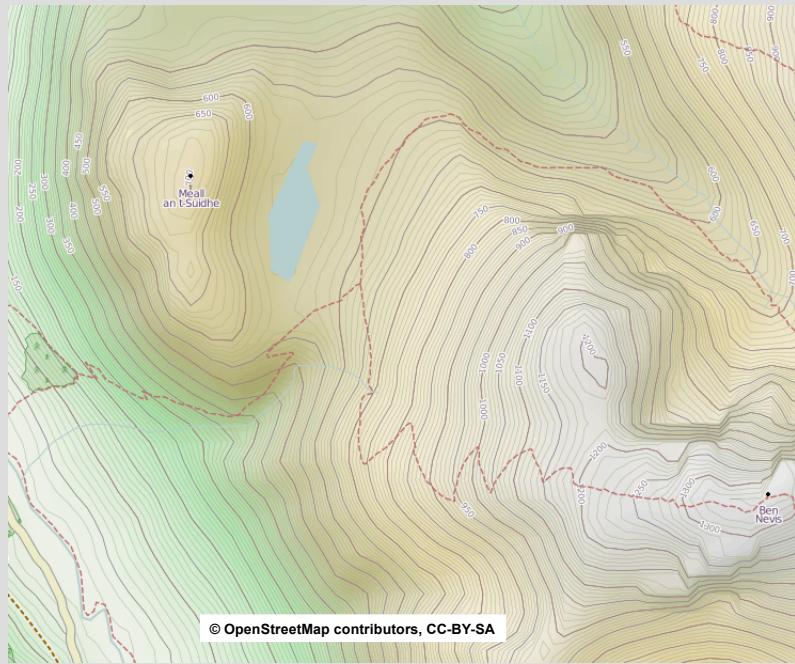


Collapsing of small areas with the same tags into a single larger one (for e.g. a big forest), so that e.g. the name is drawn only once, or that it can be simpler at different zooms

*ds: I can think of a way to do with in the PostGIS SQL query, using ST\_Collect and a group\_by statement. Lennard and I could likely get this working in Mapnik 0.7.1.*

*Ldp: that's one way, and could work where the forest is divided into a small number of distinct polygons. In NL however, due to a high res import, we now have forest areas built up of hundreds or more of tiny polygons. We were actually thinking of moving to a toponym way/relation tagging, where we then draw only the extent of the area, declare it to be a toponym and what kind of toponym (e.g. a forest toponym) and mapnik would then render the right label. The advantage is: no trickery in mapnik, no guessing about the style of label to render, no need to name hundreds or thousands of OSM polygons with the same name=\*, and direct control from the mapper on the extent of the toponym.*

## More control over text rotation



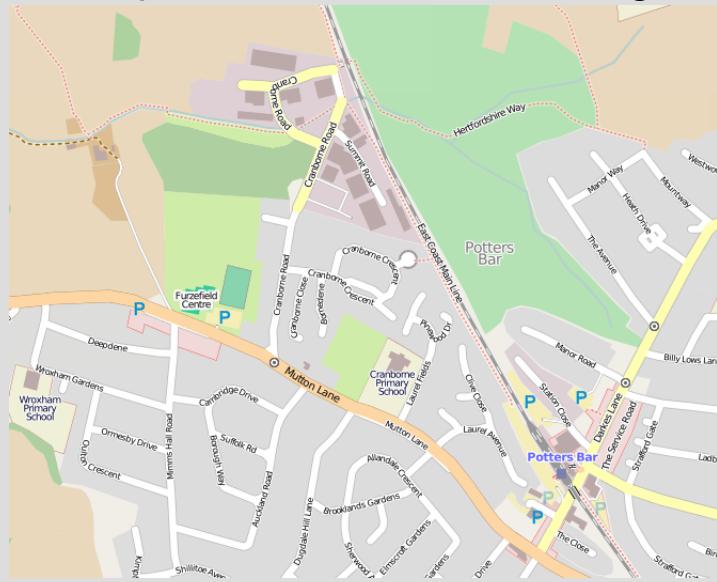
Be able to suppress automatic text rotation (per style or layer) - contour lines should always be displayed so that the base line of the text points to the lower ground

*Ds: interesting. We could add this I think. Could you show some visual examples?*

*Ldp: I don't need an example to know exactly what he wants to convey here.  
😊*

*What's needed is a modifier for the TextSymbolizer to disable the automatic 'upright' mode, so we can render text upside down as well. Or actually, always aligned to the line direction. Come to think of it, in addition to disabling the auto-upright, we need a way to flip the text upside down as well, when we know the line direction is reversed..*

## Combine point and line data in single style



© OpenStreetMap contributors, CC-BY-SA

19

### Have combined point and line data in same style

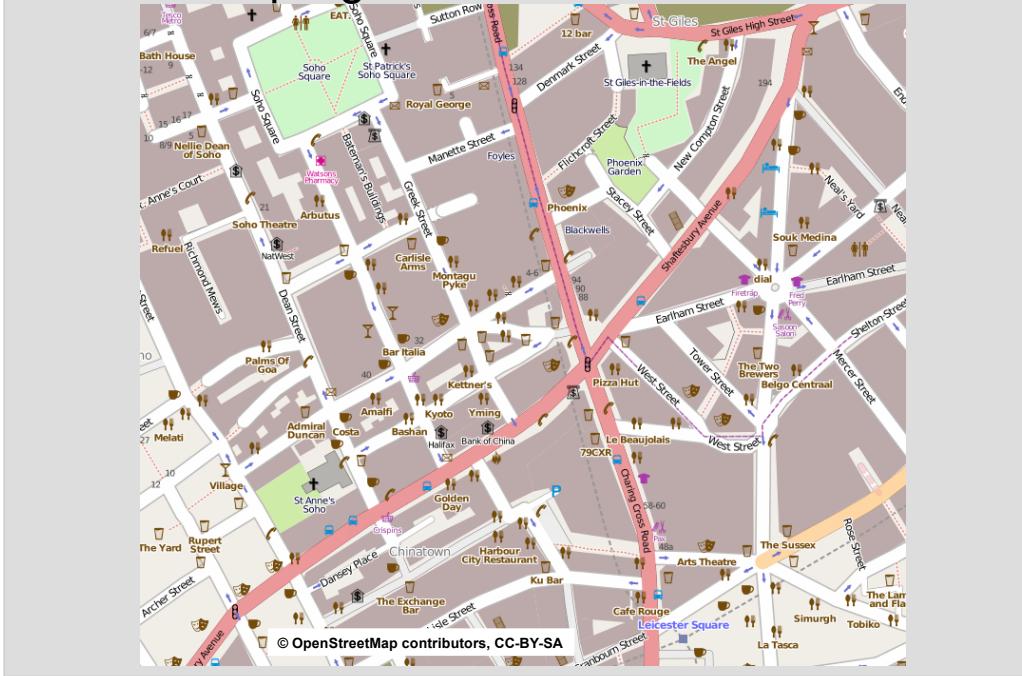
An example of this is highway = turning\_circle (node), which can't be drawn as an arc at present so that it can "merge" with its associated highway style.

Ds: great idea!

Ldp: Not a huge problem to solve, once you split turning circles into casing and fill and interlace accordingly with the road layer rendering.

The actual problem is that the highway=turning\_circle node has no idea \*at all\* about the road classification it should take. We'd have to spatially relate the node with any intersecting ways to find out which style the turning circle should take. It might be easier to see if osm2pgsql can do that for us.

## Coupling between icon and text.



Stronger coupling between icon and text (maybe similar to the ShieldSymbolizer?) so that manual text dy tweaking is not required anymore (want to be able to say: "place this icon in size 20x20 and put the text in size 12pt five pixels below it")

*ds: I think you can now do this in Mapnik 0.7.0, but I have not tested code closely enough to say. There are some new, not yet documented options added in: <http://trac.mapnik.org/changeset/1341>*

*Ldp: I think what Steve is alluding to is the ability to make text snugly below or above a symbol, no matter the size of the symbol, and without having to go through a few iterations of different dy values to make it come out nice.*

## Separate two lines by minimum distance



Separate lines so that there is a minimum distance between them (eg two rail lines, or dual carriageways)

*ds: okay, min distance in geographic units, or pixels?*

*Ldp: for me: pixels.*

## Add more Natural Earth data at low zooms



22

## Use more of the Natural Earth data for low zooms

JonB: There are some **obvious glitches** around the anti-meridian but I think it **looks nice**. I have not attempted to render anything other than the Natural Earth raster image.

The TIF file to render this is about 460MB and seems to render OK up to **around zoom level 6** before it starts becoming pixelated.

Y



Any questions – either now or in the rest of the session.

There are notes pages for each example, together with comments from Dane and Lennard, sometimes suggesting what might need to be done