Advanced TileMill Techniques

Dane Springmeyer
@springmeyer

Sotm-US 2013

TileMill

Ensure fast i/o

- PostGIS, Shapefile, SQLite (not geojson/csv/kml)
- Reproject to epsg:3857 before TileMill
- Understand indexes:
  - PosGIS: *gist* (built-in support, postgis builds)
  - Shapefile: *quadtree*, create with `shapeindex` tool
  - SQLite: *rtree* (built-in support, TileMill builds)

TileMill

Ensure fast i/o

ogr2ogr -t_srs EPSG:3857 data_merc.shp data.shp
shapeindex data_merc.shp

ogr2ogr -F SQlite data.db data_merc.shp

shp2pgsql -s 4326 data.shp | psql data

gochas:

- Shapefiles are cached / app restart to refresh data
- SQLite indexes (.index) must be user-cleared
- PostGIS extent calculations can be slow so TileMill caches

TileMill

Ensure fast i/o

Limit attributes to the required columns

PostGIS:

```
(select geometry, name from big_table) as faster_table
```

Other datasources:

```
ogr2ogr faster.shp world.shp -sql “select NAME from world”
```

TileMill

Ensure fast rendering

CartoCSS “attachments” can draw features in batches (needed for road casings) but use them sparingly

```
#layer::outline { ... } // queries data
#layer::inline { ... } // queries data again

// query once, draw twice, with “instances”
#layer {
    outline/line-width: 5;
    inline/line-width:2;
}
```

TileMill

Ensure fast rendering

If you want points to overlap, then ensure collision detection is fully turned off:

```
#layer {
    marker-allow-overlap:true;
    marker-ignore-placement:true;
}
```

TileMill

Ensure fast rendering

Filter at the layer level instead of the style level

Fast:

```
(select * from table where building not null)
    as urban_footprint
```

Works, but not as fast:

```
#urban_footprint[building!=null] { ... }
```

TileMill

Ensure fast rendering

Reduce label clutter at the layer level instead of relying on collision avoidance.

Fast:

```
(select * from table where !scale_denominator! > zoom_scale ) as road_shield_labels
```

Not as fast:

```
#labels {  text-min-distance:10;  }
```

TileMill

Ensure fast rendering

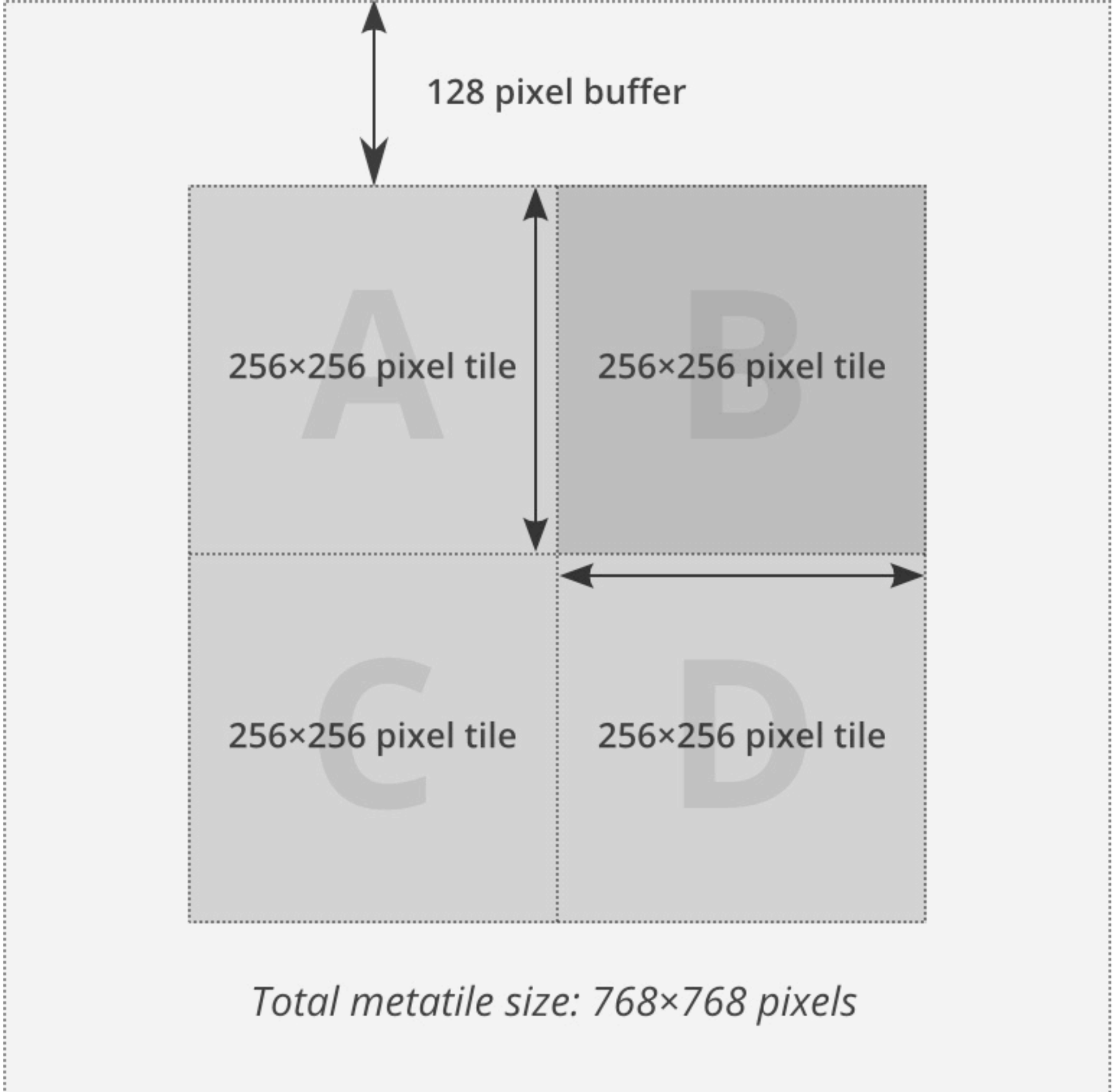
- line-rasterizer: fast;
- text-halo-rasterizer: fast;
- marker-file: use png/jpeg not svg
- only use ‘round’ line-cap or line-joins when necessary
- line-clip and polygon-clip:true - useful for reducing overhead of rendering large geometries that cross tile boundaries
- throw labels out early: e.g. text-max-char-angle-delta above 50-70 will avoid placement before glyph measurements

TileMill

Mitigating Cutoff labels

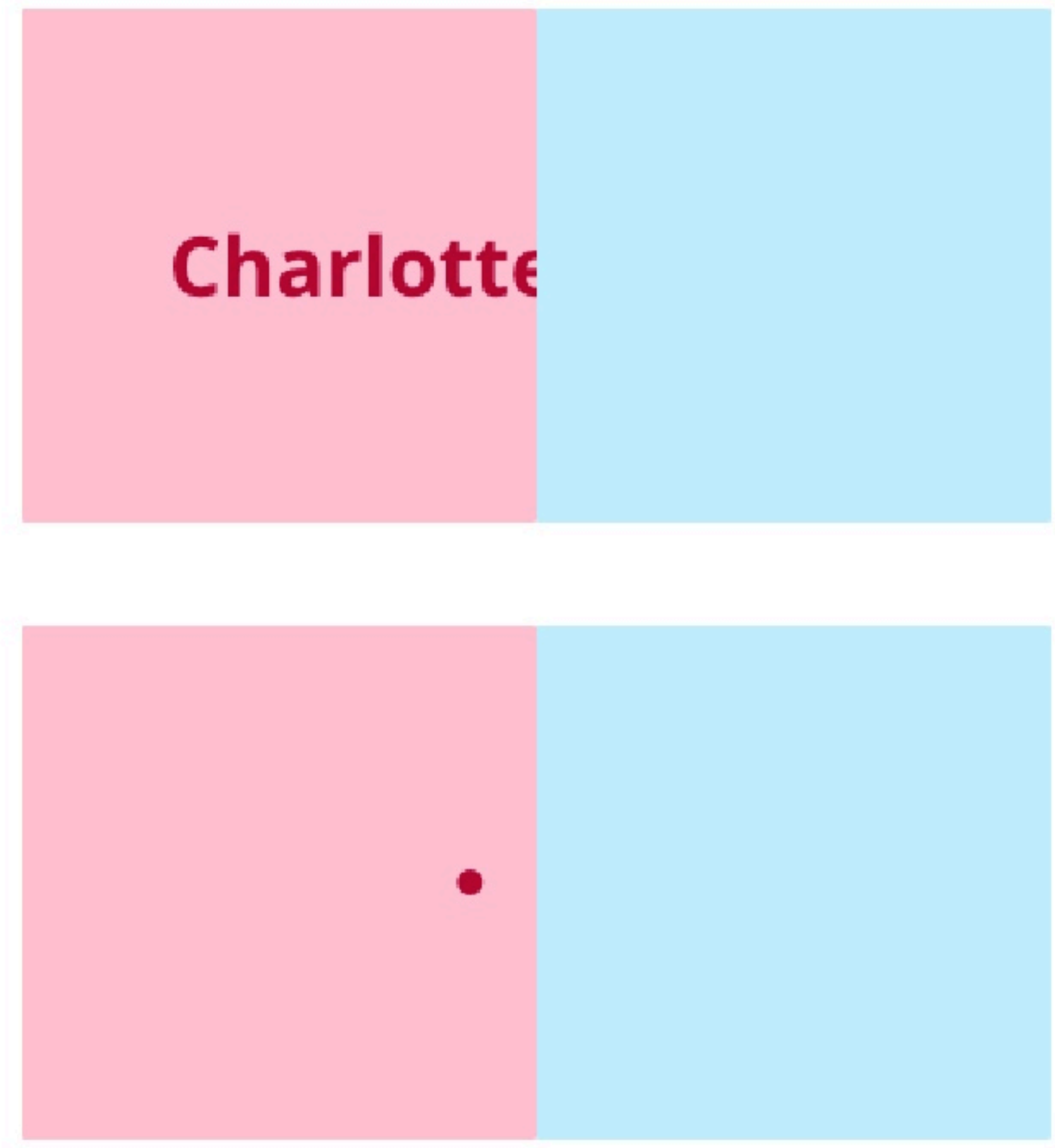
Factors for label volatility across tiles:

- #Map { buffer-size:256 }
- Metatile size
- Density/complexity of features
- Sort order of features by tile
- Width of labels/font size
- Labeling on lines / polygon centroids
- Alternative placements

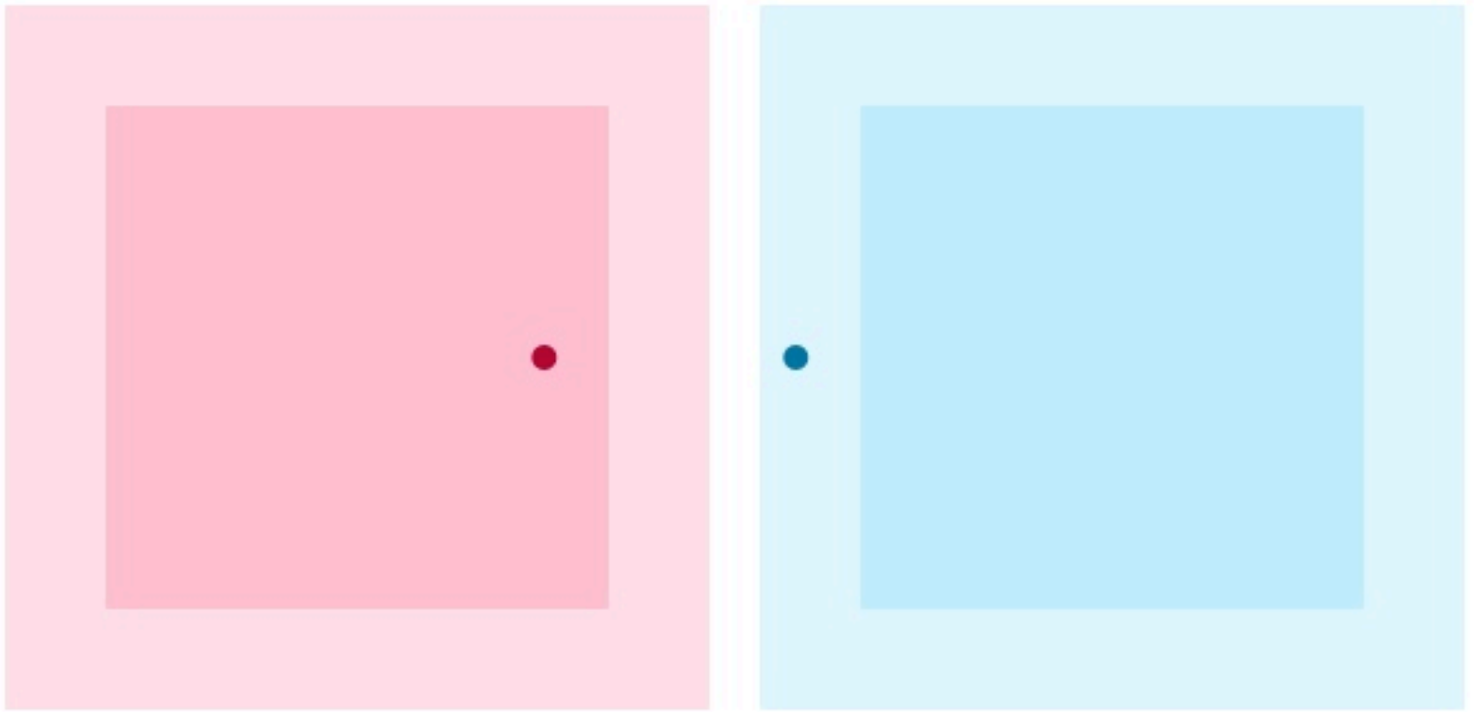


Total metatile size: 768×768 pixels

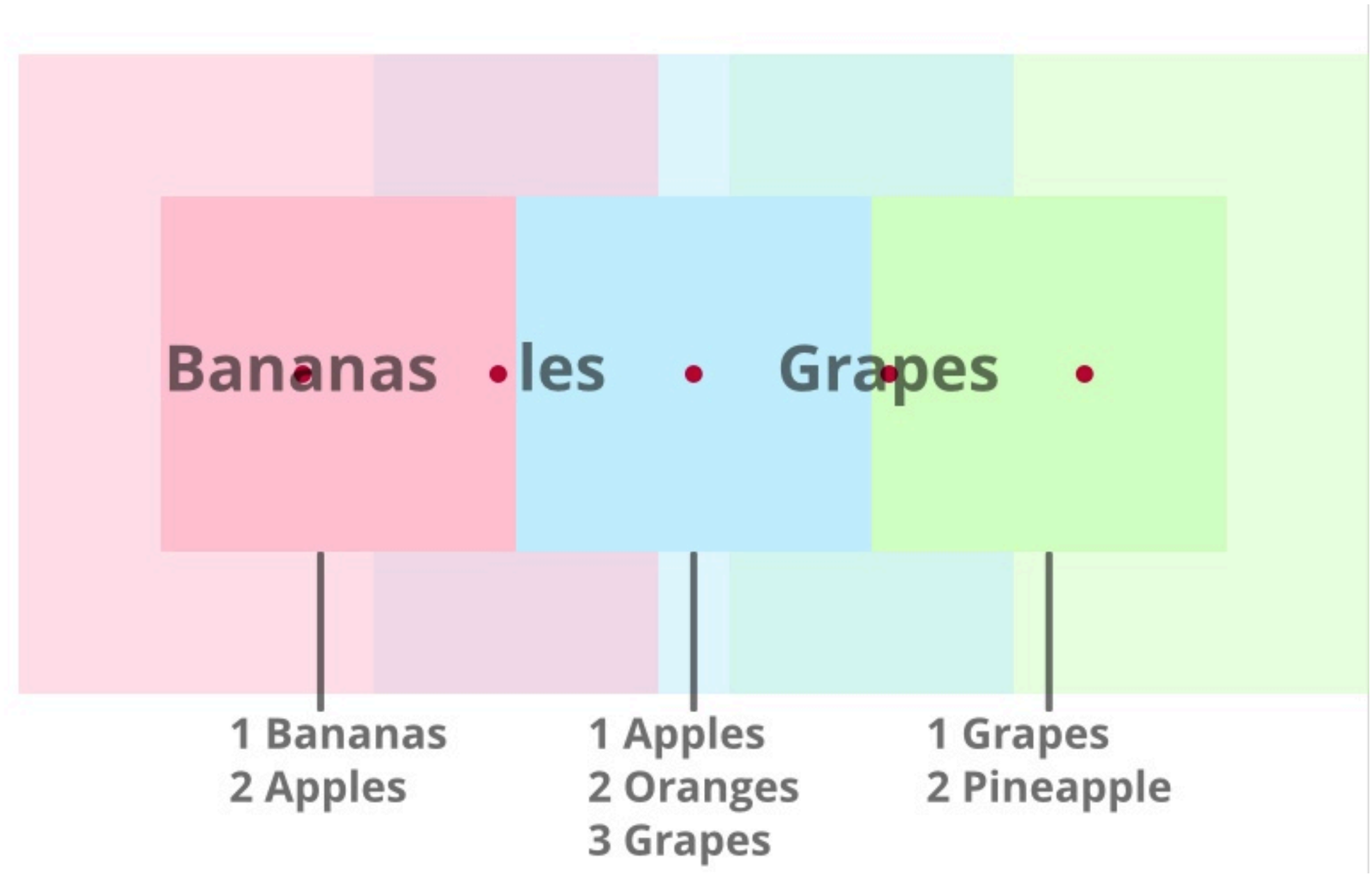
The Classic Case



Ravioli to the rescue



Chaos ensues when each tile contains different features in different order



TileMill

Mitigating Cutoff labels

The nuclear option:

text-avoid-edges:true;
text-min-padding:10;

(don’t let labels be near edges)