

The rulercompass package: code

Andrew Stacey
stacey@math.ntnu.no

December 11, 2013

1 Implementation

Load in useful tikzlibraries.

```
1 \usetikzlibrary{intersections,calc}
```

Are we in draft mode (so display point labels)?

```
2 \newif\ifrc@draft
```

Is beamer loaded?

```
3 \newif\ifrc@beamer
```

When drawing arc segments, do we flip the segment?

```
4 \newif\ifrc@fliparc
```

A picture id that stays the same on beamer frames to make it easier to compare bounding boxes and other information of the “same” tikzpicture.

```
5 \newcount\rc@picture@id
```

Counter for our point labels.

```
6 \newcounter{pointlabels}
```

Set the beamer boolean.

```
7 \@ifclassloaded{beamer}{%
```

```
8   \rc@beamertrue
```

```
9 }{}
```

Internal separator for the path-naming scheme

```
10 \def\rc@pathsep{0}
```

Are we running under beamer?

```
11 \ifrc@beamer
```

Make our counters reset on frames

```
12 \resetcounteronoverlays{pointlabels}
```

```
13 \resetcountonoverlays{rc@picture@id}
```

Define an overlay-aware style

```

14 \tikzset{
15   alt if exist/.code args={#1#2#3}{%
16     \ifundefined{path@the\rc@picture@id @#1}{%
17       \pgfkeysalso{#2}%
18     }{
19       \alt<.-\csname path@the\rc@picture@id @#1\endcsname>{%
20         \pgfkeysalso{#2}%
21       }{
22         \pgfkeysalso{#3}%
23       }%
24     }%
25   },

```

Save a path when it is used to compute a point

```

26   intersection/save/.code={%
27     \only<.>{%
28       \begingroup
29       \tikz@intersect@path@names@parse#1\tikz@stop
30       \protected@write\pgfutil@auxout{}{%
31         \string\global\string\@namedef{path@the\rc@picture@id @\tikz@intersect@path@a}{\the\bb
32         \string\global\string\@namedef{path@the\rc@picture@id @\tikz@intersect@path@b}{\the\bb
33       }%
34     \endgroup
35   }%
36 }
37 }

```

Define overlay-aware versions of the main macros.

```

38 \newcommand<>\compass [3] [] {%
39   \draw#4[#1,ruler compass/compass={#2}{#3}];
40 }
41
42 \newcommand<>\ruler [3] [] {%
43   \draw#4[#1,ruler compass/ruler={#2}{#3}];
44 }

```

The `\point` macro is doubly overlay aware. If the intersection has already been computed (say, on another slide of the same picture), reuse it.

```

45 \newcommand<>\point [4] [] {%
46   {%
47     \advance\c@pointlabels by 1\relax
48     \xdef\rc@temp{\thepointlabels}%
49   }%
50   \edef\rc@tempa{\the\rc@picture@id}%
51   \expandafter\ifx\csname rc@id@\rc@temp\endcsname\rc@tempa
52   \path#5 (\rc@temp) node[ruler compass/point,#1] {};
53   \else
54   \path#5 [name intersections={use=#2 and #3}] (intersection-#4) node[ruler compass/point,#1]
55   \fi
56 }

```

Now for the non-beamer versions

```
57 \else
```

The overlay-aware style defaults to the first option.

```
58 \tikzset{
59   alt if exist/.code args={#1#2#3}{%
60     \pgfkeysalso{#2}%
61   },
```

Saving this means we don't have to have two versions of a more complicated bit of code.

```
62   intersection/save/.code={%
63     \begingroup
64     \tikz@intersect@path@names@parse#1\tikz@stop
65     \protected@write\pgfutil@auxout{}%
66       \string\global\string\@namedef{path@the\rc@picture@id @\tikz@intersect@path@a}{\thepa
67       \string\global\string\@namedef{path@the\rc@picture@id @\tikz@intersect@path@b}{\thepa
68     }%
69     \endgroup
70   }
71 }
```

Non-overlay aware versions of the primary commands.

```
72 \newcommand\compass[3] [] {%
73   \draw [#1,ruler compass/compass={#2}{#3}];
74 }
75
76 \newcommand\ruler[3] [] {%
77   \draw [#1,ruler compass/ruler={#2}{#3}];
78 }
79
80 \newcommand\point[4] [] {%
81   \path [name intersections={use=#2 and #3}] (intersection-#4) node[ruler compass/point,#1] {}
82 }
```

End of non-beamer specific section

```
83 \fi
```

The following macros process the path/point specifications. The first looks to see if the first character is a period.

```
84 \def\rc@processpt#1{%
85   \pgfutil@ifnextchar.{\rc@processpt@@#1}{\rc@processpt@#1}}
```

Nope, so now look for a plus or a minus.

```
86 \def\rc@processpt@#1#2\pgf@stop{%
87   \pgfutil@in@+{#2}%
88   \ifpgfutil@in@
89     \let\@next=\rc@processpt@plus
90   \else
91     \pgfutil@in@-{#2}%
92     \ifpgfutil@in@
93       \let\@next=\rc@processpt@minus
94     \else
```

```

95     \let\@next=\rc@processpt@bare
96     \fi
97     \fi
98     \@next#1#2\pgf@stop
99 }

```

Okay, we got a period. That means the current point, possibly with an offset.

```

100 \def\rc@processpt@#1.#2\pgf@stop{%
101     \def\rc@temp{#2}%
102     \ifx\rc@temp\pgfutil@empty
103     \edef\rc@temp{\thepointlabels}%
104     \else
105     \pgfmathparse{int(\the\c@pointlabels + #2)}%
106     \begingroup
107     \c@pointlabels=\pgfmathresult\relax
108     \xdef\rc@temp{\thepointlabels}%
109     \endgroup
110     \fi
111     \let#1\rc@temp
112 }

```

We need to add an offset to the given label, so need to compute the index of the label.

```

113 \def\rc@processpt@plus#1#2+#3\pgf@stop{%
114     \pgfmathsetmacro\rc@temp{0}%
115     \rc@reverse#2\pgf@stop%
116     \pgfmathparse{int(\rc@temp + #3)}%
117     \begingroup
118     \c@pointlabels=\pgfmathresult\relax
119     \xdef\rc@temp{\thepointlabels}%
120     \endgroup
121     \let#1\rc@temp
122 }

```

Same, but with a minus.

```

123 \def\rc@processpt@minus#1#2-#3\pgf@stop{%
124     \pgfmathsetmacro\rc@temp{0}%
125     \rc@reverse#2\pgf@stop%
126     \pgfmathparse{int(\rc@temp - #3)}%
127     \begingroup
128     \c@pointlabels=\pgfmathresult\relax
129     \xdef\rc@temp{\thepointlabels}%
130     \endgroup
131     \let#1\rc@temp
132 }

```

Simplest case.

```

133 \def\rc@processpt@bare#1#2\pgf@stop{%
134     \def#1{#2}}

```

Reverse lookup the counter value from its displayed format.

```

135 \def\rc@reverse#1{%
136     \ifx#1\pgf@stop

```

```

137 \let\@next=\pgfutil@empty
138 \else
139 \let\@next=\rc@reverse
140 \pgfutil@tempcnta=#1\relax
141 \pgfmathsetmacro\rc@temp{\rc@temp * \rc@factor + \the\pgfutil@tempcnta - \rc@initial}%
142 \fi
143 \@next}

```

Compute the actual path name from the three token (lists).

```

144 \def\rc@parsepath#1#2#3#4\pgf@stop{%
145 \rc@processpt\rc@tempa#3\pgf@stop
146 \rc@processpt\rc@tempb#4\pgf@stop
147 \xdef#1{#2\rc@pathsep\rc@tempa\rc@pathsep\rc@tempb}%
148 }

```

Now we install all our styles

```

149 \tikzset{

```

Code which initialises our counters at the start of a picture.

```

150 every picture/.append style={
151 ruler compass/at begin picture
152 },

```

Passes the paths to the intersection but also takes note of their use.

```

153 intersection/use/.code args={#1 and #2}{%
154 \rc@parsepath\rc@pta#1\pgf@stop
155 \rc@parsepath\rc@ptb#2\pgf@stop
156 \pgfkeysalso{
157 of={\rc@pta} and \rc@ptb,
158 save={\rc@pta} and \rc@ptb,
159 }
160 },

```

to path for a circle through a given point.

```

161 circle through/.style={
162 to path={
163 \pgfextra{
164 \tikz@scan@one@point\pgfutil@firstofone(\tikztostart)\relax
165 \pgf@xa=\pgf@x
166 \pgf@ya=\pgf@y
167 \tikz@scan@one@point\pgfutil@firstofone(\tikztotarget)\relax
168 \pgfmathsetmacro\rc@radius{\veclen(\pgf@x - \pgf@xa,\pgf@y - \pgf@ya)}%
169 }
170 circle[radius=\rc@radius pt] (\tikztotarget)
171 }
172 },

```

Flip an arc when drawing a segment.

```

173 arc flip/.is if=rc@fliparc,

```

Arc from one point to another centred on specified point.

```
174   centred arc to/.code 2 args={%
175     \tikz@scan@one@point\pgfutil@firstofone(#1)\relax
176     \pgfmathsetmacro\rc@radius{veclen(\tikz@lastx-\pgf@x,\tikz@lasty-\pgf@y)}%
177     \pgfmathsetmacro\rc@sangle{atan2(\tikz@lasty - \pgf@y,\tikz@lastx - \pgf@x)}%
178     \pgf@xa=\pgf@x
179     \pgf@ya=\pgf@y
180     \tikz@scan@one@point\pgfutil@firstofone(#2)\relax
181     \pgfmathsetmacro\rc@eangle{atan2(\pgf@y - \pgf@ya,\pgf@x - \pgf@xa)}%
182     \pgfmathsetmacro\rc@eangle{\rc@eangle < \rc@sangle ? \rc@eangle +      360 : \rc@eangle}%
183     \ifrc@fliparc
184     \pgfmathsetmacro\rc@eangle{\rc@eangle - 360}%
185     \fi
186     \pgfkeysalso{start angle=\rc@sangle, end angle=\rc@eangle, radius=\rc@radius pt}
187   },
```

Most of our keys are in this family.

```
188   ruler compass/.is family,
189   ruler compass/.cd,
```

Draft mode displays the labels.

```
190   draft mode/.is if=rc@draft,
191   % beamer mode/.is if=rc@beamer,
```

This is the actual code for resetting the counters.

```
192   at begin picture/.style={
193     execute at begin picture={
194       \global\advance\rc@picture@id by 1\relax
195       \setcounter{pointlabels}{0}%
196     },
197   },
```

Styling the point labels.

```
198   point labels/.is choice,
199   point labels/arabic/.code={%
200     \renewcommand\thepointlabels{\@arabic \c@pointlabels}%
201     \def\rc@initial{48}%
202     \def\rc@factor{10}%
203   },
204   point labels/alph/.code={%
205     \renewcommand\thepointlabels{\@alph \c@pointlabels}%
206     \def\rc@initial{96}%
207     \def\rc@factor{26}%
208   },
209   point labels/Alph/.code={%
210     \renewcommand\thepointlabels{\@Alph \c@pointlabels}%
211     \def\rc@initial{64}%
212     \def\rc@factor{26}%
213   },
```

Styling the point labels with the `alphalph` package.

```

214 point labels/alphalph/.code={%
215   \@ifundefined{alphalph}{%
216     \message{The "alphalph" option only works if the 'alphalph' package has been loaded (usi
217     \renewcommand\thepointlabels{\@alph \c@pointlabels}%
218     \def\rc@initial{96}%
219     \def\rc@factor{26}%
220   }{%
221     \renewcommand\thepointlabels{\alphalph \c@pointlabels}%
222     \def\rc@initial{96}%
223     \def\rc@factor{26}%
224   }%
225 },
226 point labels/AlphAlph/.code={%
227   \@ifundefined{alphalph}{%
228     \message{The "AlphAlph" option only works if the 'alphalph' package has been loaded (usi
229     \renewcommand\thepointlabels{\@Alph \c@pointlabels}%
230     \def\rc@initial{64}%
231     \def\rc@factor{26}%
232   }{%
233     \renewcommand\thepointlabels{\AlphAlph \c@pointlabels}%
234     \def\rc@initial{64}%
235     \def\rc@factor{26}%
236   }%
237 },

```

Styles for if the path is still in use or not.

```

238 construction in use/.style={
239   draw=blue
240 },
241 construction not in use/.style={
242   draw=gray!75,
243 },

```

Code for drawing a circle.

```

244 compass/.code 2 args={%
245   \rc@processpt\rc@pta#1\pgf@stop
246   \rc@processpt\rc@ptb#2\pgf@stop
247   \pgfkeysalso{ruler compass/compass aux={\rc@pta}{\rc@ptb}}%
248 },
249 compass aux/.style 2 args={
250   alt if exist={c\rc@pathsep#1\rc@pathsep#2}{
251     ruler compass/construction in use/.try,
252   }{
253     ruler compass/construction not in use/.try
254   },
255   name path=c\rc@pathsep#1\rc@pathsep#2,
256   ruler compass/every construction path/.try,
257   ruler compass/every compass/.try,
258   insert path={

```

```

259     (#1) to[circle through] (#2)
260     node[name=c\rc@pathsep#1\rc@pathsep#2\rc@pathsep centre,ruler compass/aux point={#1}] {}
261     node[name=c\rc@pathsep#1\rc@pathsep#2\rc@pathsep rim,ruler compass/aux point={#2}] {}
262   },
263 },

```

Code for drawing a straight line.

```

264 ruler/.code 2 args={%
265   \rc@processpt\rc@pta#1\pgf@stop
266   \rc@processpt\rc@ptb#2\pgf@stop
267   \pgfkeysalso{ruler compass/ruler aux={\rc@pta}{\rc@ptb}}%
268 },
269 ruler aux/.style 2 args={
270   overlay,
271   alt if exist={r\rc@pathsep#1\rc@pathsep#2}{
272     ruler compass/construction in use/.try
273   }{
274     ruler compass/construction not in use/.try
275   },
276   name path=r\rc@pathsep#1\rc@pathsep#2,
277   ruler compass/every construction path/.try,
278   ruler compass/every ruler/.try,
279   insert path={
280     ($(#1)!\pgfkeysvalueof{/tikz/ruler compass/ruler length}!(#2)$) -- ($(#2)!\pgfkeysvalueof{/tikz/ruler compass/ruler length}!(#1)$)
281     node[name=r\rc@pathsep#1\rc@pathsep#2\rc@pathsep start,ruler compass/aux point={#1}] {}
282     node[name=r\rc@pathsep#1\rc@pathsep#2\rc@pathsep end,ruler compass/aux point={#2}] {}
283   },
284 },

```

Sets the ruler length.

```

285 ruler length/.initial=20,

```

Auxiliary point style.

```

286 aux point/.style={
287   node contents/.try={},
288   at={(#1)},
289   ruler compass/every aux point/.try
290 },

```

Draws a point as a coordinate with another node for styling.

```

291 point/.style={
292   ruler compass/name it,
293   coordinate,
294   node contents/.try={},
295   insert path={
296     node[
297       fill=#1,
298       circle,
299       minimum width=1mm,
300       inner sep=0mm,
301       reset label anchor,

```



```

302     anchor=center,
303     node contents/.try={},
304     ruler compass/every point/.try,
305   ] {}
306 }
307 },

```

Forces a point to be named.

```

308 name it/.code={%
309   \ifx\tikz@fig@name\pgfutil@empty
310     \stepcounter{pointlabels}%
311     \pgfkeysalso{name=\thepointlabels}%
312     \expandafter\xdef\csname rc@id@\thepointlabels\endcsname{\the\rc@picture@id}%
313     \ifrc@draft
314     \pgfkeysalso{label={[ruler compass/draft label/.try]\thepointlabels}}%
315     \fi
316     \fi
317 },
318 point/.default=black,
319 }

```

Sets the initial scale and offset of the enclosing box.

```

320 \def\rc@scale{1}
321 \def\rc@offset{0pt}

```

Sets the bounding box from the auxiliary file.

```

322 \def\jump@setbb#1#2#3{%
323   \@ifundefined{jump@#1@maxbb}{-%
324     \expandafter\gdef\csname jump@#1@maxbb\endcsname{#3}%
325   }{%
326     \csname jump@#1@maxbb\endcsname
327     \pgf@xa=\pgf@x
328     \pgf@ya=\pgf@y
329     #3
330     \pgfmathsetlength\pgf@x{max(\pgf@x,\pgf@xa)}%
331     \pgfmathsetlength\pgf@y{max(\pgf@y,\pgf@ya)}%
332     \expandafter\xdef\csname jump@#1@maxbb\endcsname{\noexpand\pgfpoint{\the\pgf@x}{\the\pgf@y}}
333   }
334   \@ifundefined{jump@#1@minbb}{-%
335     \expandafter\gdef\csname jump@#1@minbb\endcsname{#2}%
336   }{%
337     \csname jump@#1@minbb\endcsname
338     \pgf@xa=\pgf@x
339     \pgf@ya=\pgf@y
340     #2
341     \pgfmathsetlength\pgf@x{min(\pgf@x,\pgf@xa)}%
342     \pgfmathsetlength\pgf@y{min(\pgf@y,\pgf@ya)}%
343     \expandafter\xdef\csname jump@#1@minbb\endcsname{\noexpand\pgfpoint{\the\pgf@x}{\the\pgf@y}}
344   }
345 }

```

Installs the code to save the bounding box.

```

346 \tikzset{
347   stop jumping/.style={
348     execute at end picture={%
349       \pgfmathsetlength\pgf@xa{\pgf@picminx/\rc@scale}%
350       \pgfmathsetlength\pgf@ya{\pgf@picminy/\rc@scale}%
351       \pgfmathsetlength\pgf@xb{\pgf@picmaxx/\rc@scale}%
352       \pgfmathsetlength\pgf@yb{\pgf@picmaxy/\rc@scale}%
353       \immediate\write\pgfutil@auxout{%
354         \noexpand\jump@setbb{\the\rc@picture@id}{\noexpand\pgfpoint{\the\pgf@xa}{\the\pgf@ya}}-
355       },
356       \pgf@x=\pgf@picminx
357       \pgf@y=\pgf@picminy
358       \csname jump@\the\rc@picture@id @minbb\endcsname
359       \pgf@xa=\pgf@x
360       \pgf@ya=\pgf@y
361       \pgf@x=\pgf@picmaxx
362       \pgf@y=\pgf@picmaxy
363       \csname jump@\the\rc@picture@id @maxbb\endcsname
364       \edef\rc@temp{\noexpand\path (\the\pgf@xa - \rc@offset,\the\pgf@ya - \rc@offset) -- (\the
365       \rc@temp
366     },
367   },

```

Scales the picture to fit inside a given rectangle.

```

368   max size/.code 2 args={%
369     \pgfutil@ifundefined{jump@\the\rc@picture@id @maxbb}{-}{%
370       \csname jump@\the\rc@picture@id @maxbb\endcsname
371       \pgf@xa=\pgf@x
372       \pgf@ya=\pgf@y
373       \csname jump@\the\rc@picture@id @minbb\endcsname
374       \advance\pgf@xa by -\pgf@x
375       \advance\pgf@ya by -\pgf@y
376       \advance\pgf@xa by \rc@offset
377       \advance\pgf@xa by \rc@offset
378       \advance\pgf@ya by \rc@offset
379       \advance\pgf@ya by \rc@offset
380       \pgfmathsetmacro\rc@xratio{\pgf@xa > #1 ? \pgf@xa/#1 : 1}%
381       \pgfmathsetmacro\rc@yratio{\pgf@ya > #2 ? \pgf@ya/#2 : 1}%
382       \pgfmathsetmacro\rc@scale{1/max(\rc@xratio,\rc@yratio)}%
383       \pgfkeysalso{scale=\rc@scale}%
384     }
385   },
386   enclosing box/offset/.store in=\rc@offset,
387   constrain/.style={
388     execute at begin picture=\constrain
389   }
390 }

```

Defines a clip to the enclosing box.

```

391 \newcommand\constrain{%
392   \pgfutil@ifundefined{jump@the\rc@picture@id @minbb}{-%
393     \csname jump@the\rc@picture@id @minbb\endcsname
394     \pgfmathsetlength\pgf@xa{\pgf@x-\rc@offset}%
395     \pgfmathsetlength\pgf@ya{\pgf@y-\rc@offset}%
396     \csname jump@the\rc@picture@id @maxbb\endcsname
397     \pgfmathsetlength\pgf@xb{\pgf@x+\rc@offset}%
398     \pgfmathsetlength\pgf@yb{\pgf@y+\rc@offset}%
399     \edef\rc@temp{\noexpand\clip (\the\pgf@xa,\the\pgf@ya) rectangle (\the\pgf@xb,\the\pgf@yb)
400     \ifpgf@relevantforpicturesize
401       \pgf@relevantforpicturesizefalse
402       \rc@temp
403     \pgf@relevantforpicturesizetrue
404     \else
405       \rc@temp
406     \fi
407   }%
408 }
409

```

Defines the enclosing box node.

```

410 \expandafter\def\csname pgf@sh@ns@enclosing box\endcsname{rectangle}
411 \expandafter\def\csname pgf@sh@np@enclosing box\endcsname{%
412   \pgfutil@ifundefined{jump@the\rc@picture@id @minbb}{-%
413     \def\southwest{\pgfqpoint{\pgf@picminx}{\pgf@picminy}}%
414     \def\northeast{\pgfqpoint{\pgf@picmaxx}{\pgf@picmaxy}}%
415   }-%
416   \csname jump@the\rc@picture@id @minbb\endcsname
417   \pgfmathsetlength\pgf@xa{\pgf@x-\rc@offset}%
418   \pgfmathsetlength\pgf@ya{\pgf@y-\rc@offset}%
419   \edef\southwest{\noexpand\pgfqpoint{\the\pgf@xa}{\the\pgf@ya}}%
420   \csname jump@the\rc@picture@id @maxbb\endcsname
421   \pgfmathsetlength\pgf@xb{\pgf@x+\rc@offset}%
422   \pgfmathsetlength\pgf@yb{\pgf@y+\rc@offset}%
423   \edef\northeast{\noexpand\pgfqpoint{\the\pgf@xb}{\the\pgf@yb}}%
424   }%
425 }
426 \expandafter\def\csname pgf@sh@nt@enclosing box\endcsname{{1}{0}{1}{Opt}{Opt}}
427 \expandafter\def\csname pgf@sh@pi@enclosing box\endcsname{\pgfpictureid}
428
429

```

Defines the layer code for individual paths and nodes.

```

430 \tikzset{
431   on layer/.code={
432     \pgfonlayer{#1}\beginpgfgroup
433     \aftergroup\endpgfonlayer
434     \aftergroup\endgroup
435   },
436   node on layer/.code={
437     \gdef\node@@on@layer{%

```

```

438     \setbox\tikz@tempbox=\hbox\bgroup\pgfonlayer{#1}\unhbox\tikz@tempbox\endpgfonlayer\egroup
439     \aftergroup\node@on@layer
440   },
441   reset label anchor/.code={%
442     \let\tikz@auto@anchor=\pgfutil@empty
443     \def\tikz@anchor{#1}
444   },
445   reset label anchor/.default=center
446 }
447 \def\node@on@layer{\aftergroup\node@@@on@layer}

```

Sets the point label style.

```

448 \@ifpackageloaded{alphalph}%
449 {
450   \tikzset{ruler compass/point labels=alphalph}
451 }
452 {
453   \tikzset{ruler compass/point labels=alph}
454 }
455
456 \endinput

```