sunpath – Draw Sun Path*

Reference

Hồng-Phúc Bùi†

Released 2024/10/20

Contents

1 Documentation

1.1 Context and Terms

The position of the sun from perspective of an observer is defined by two parameters:

- the azimuth Φ, which tells the observer, how far (in degree) he must turn around from the North direction,
- the altitude θ, which tells the observer, how height (in degree) about the horizon he must look to see the sun.

*This file describes v0.5, last revised 2024/10/20.

†E-mail: hong-phuc.bui (at) htwsaar dot de

The azimuth can take a value in the interval $[0, 360)$. The altitude can take a value in the interval $[0, 90]$, whereas 0 is the horizon, 90 is the zenith. We do not care so much about how far is the sun, so we normalize this distance to 1.
The figure 1 shows these parameter. The coordinate system, which takes the position of the observer as the centre, and the observer's local horizon as the fundamental plane, is called horizontal coordinate system.[1]

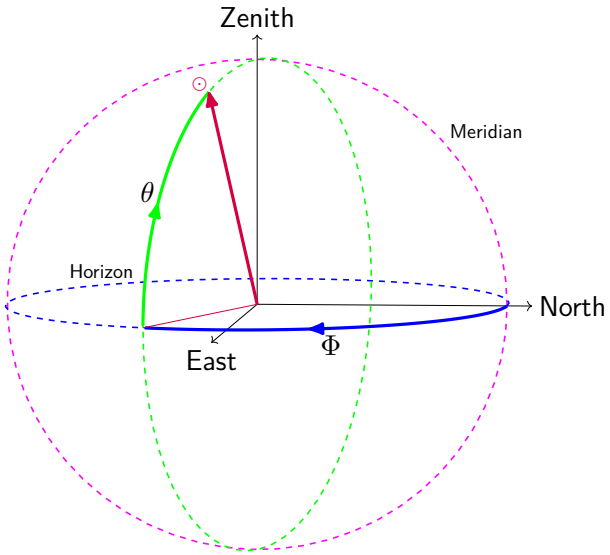


Figure 1: Horizontal coordinate system

In this package, the cardinal points have specifics values of azimuth as following:

North	East	South	West
0°	90°	180°	270°

The projection of the sun on the horizon plane is a point, which can be defined by two parameters:

- the angle Φ,
- the distance $r = \cos(\theta)$ from the centre to the sun.

Figure 2 shows the projection of the sun on the horizontal plane. If we track the position of sun on the horizontal plane changes from time to time, we will get a curve. This curve is called the sun path. A chart which shows position of the sun from time to time is called a sun path chart. Of course there are many type of sun path chart. This package provides tools to plot sun path on the horizontal plane.

[1] dt.: topozentrisches Koordinatensystem

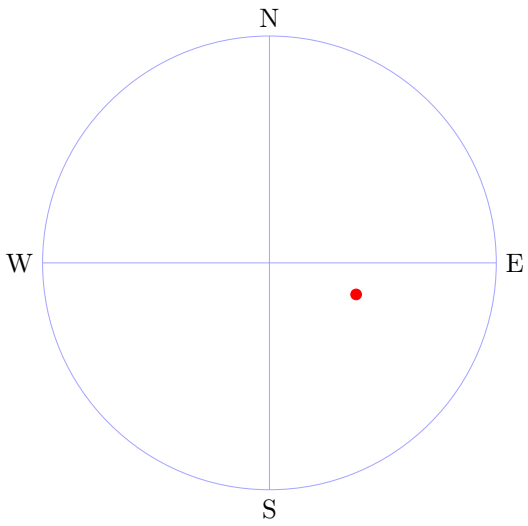


Figure 2: Projection of the sun on the horizon plane

1.2 Draw a Sun path chart

Figure 2 is a very rudimentary sun path chart. There is neither scalar, nor time on the chart. A more usable Sun path chart may look like one in the figure 3. In this section we will create this chart.

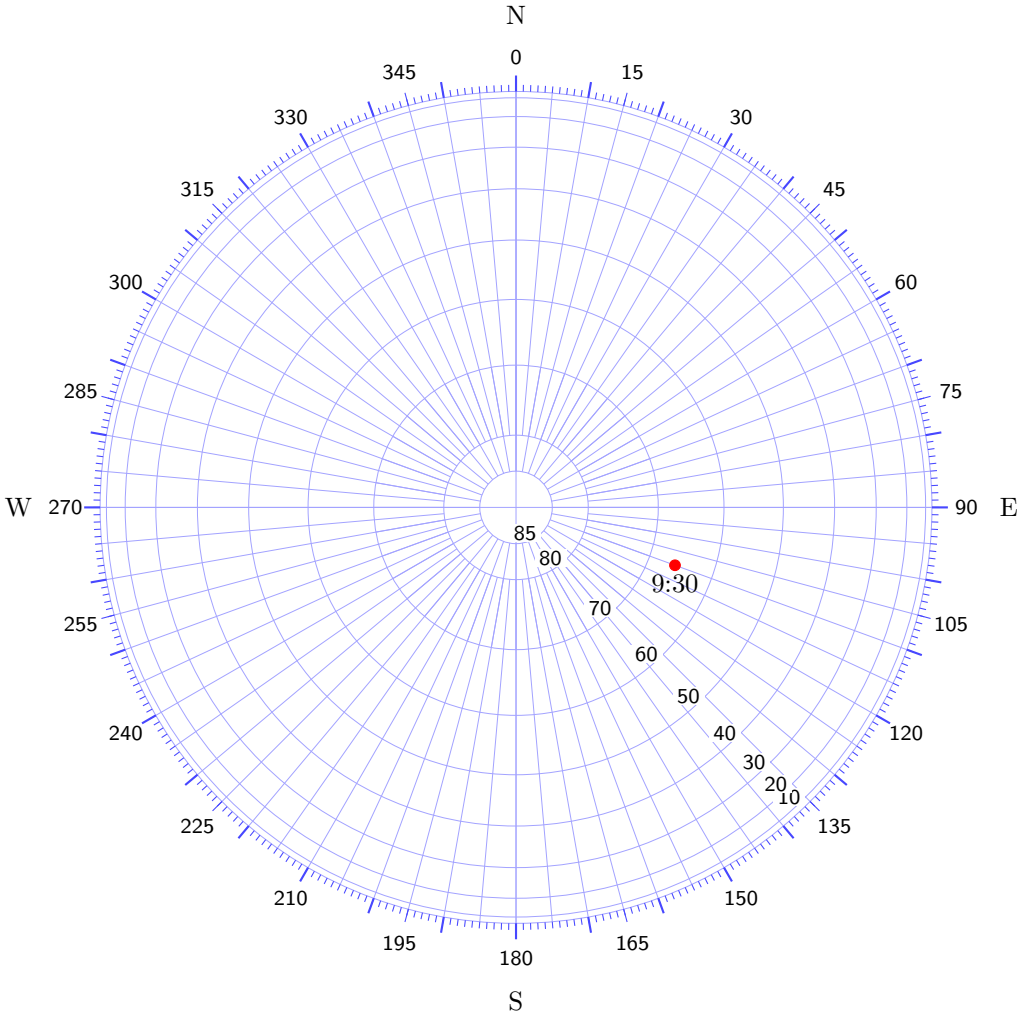


Figure 3: A Sun path chart

1.2.1 Outlines

User has to place `\usepackage{sunpath}` in the preamble part of the document. The chart is a Ti*k*Z-picture, so we need a `tikzpicture`-environment. We can also customize the distance from the centre of the chart to the horizon line by setup the option `spradius`. By default it is 5.5 in PGF xy coordinate. In this example we make it a little bigger:

`spradius`

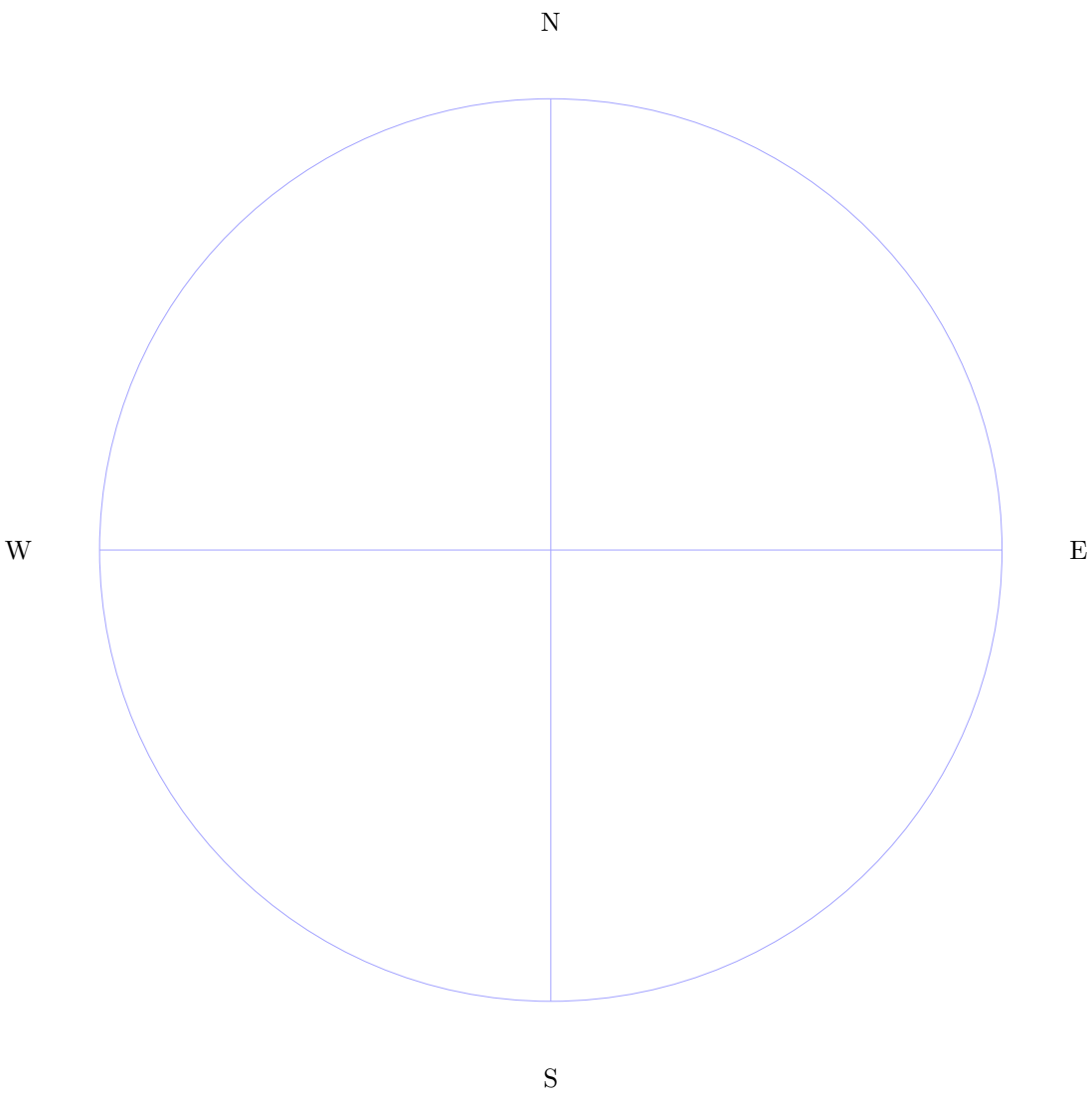
```
\begin{tikzpicture}[spradius=6]
\end{tikzpicture}
```

We also need the crosshair, the horizon line –in this type of sun path chart it is a circle–, the fours geographic direction. This can be done by adding more commands into the `tikzpicture`

```
\begin{tikzpicture}[spradius=6]
```

```
\spcrosshair
\spaltitudecircle{{0}}
\spgeodirection
\end{tikzpicture}
```

drawcrosshair
drawgeodirection
drawaltitudecircle



Man has to pay attention to the double curry brackets in the command `drawaltitudecircle`. The outer brackets delimit the argument of the command. The argument of the command is a valid Ti*k*Z-range, which is used in a `\foreach` command, so it has be placed in between a pair of curry bracket. That is the inner brackets.

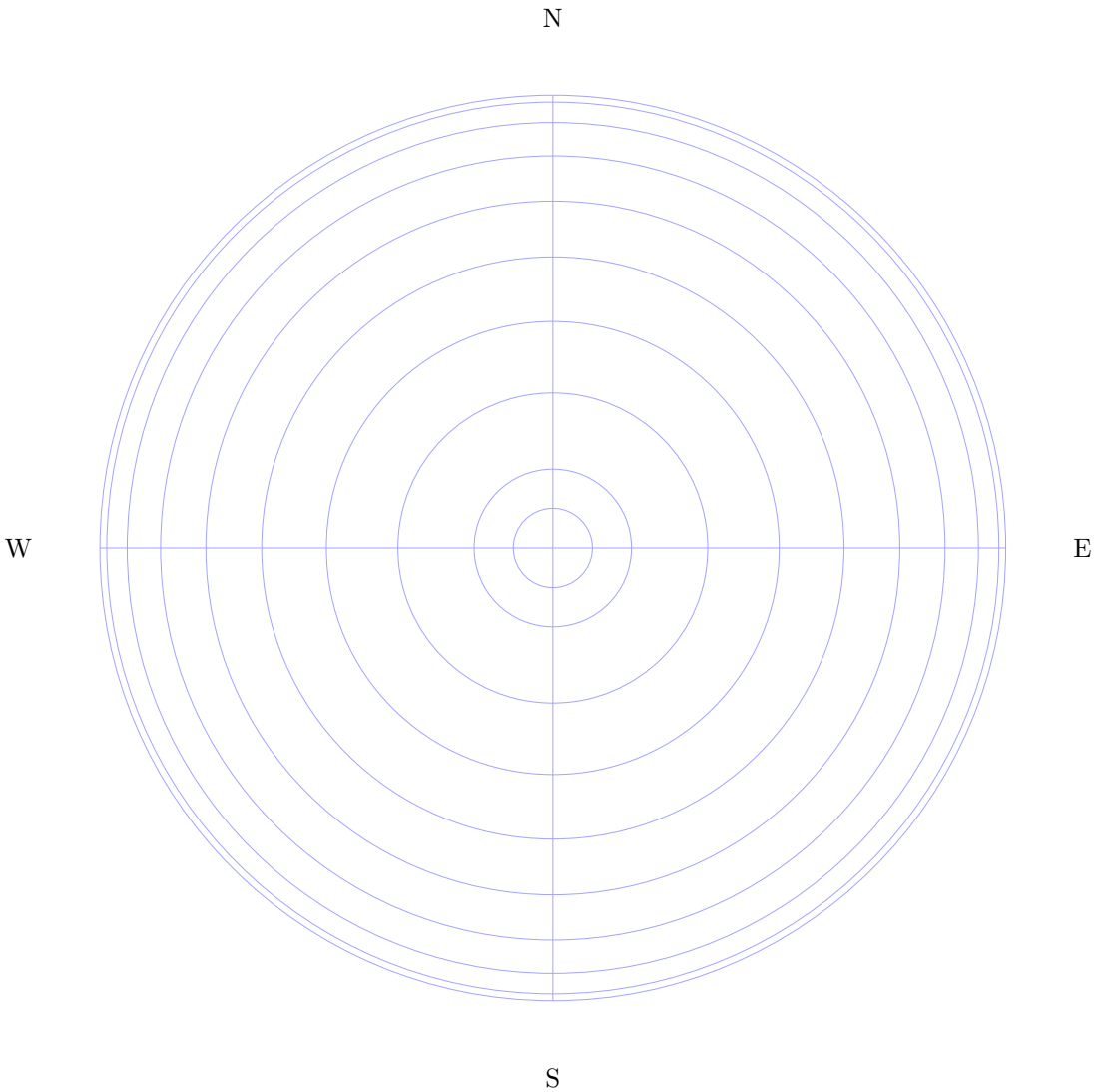
1.2.2 Scalar and labels

As the name of the commando says, we can also draw more than the horizon line by adding some values of altitude in the range of the argument of the command

drawaltitudecircle

`\spaltitudecircle`. For example `\spaltitudecircle{{0,10,...,80,85}}` draws 10 circles of altitude.

```
\begin{tikzpicture}[spradius=6]
\spcrosshair
\spaltitudecircle{{0,10,...,80,85}}
\spgeodirection
\end{tikzpicture}
```



We can use the command `\drawazumuthline{r}{h}{l}` to draw azimuth lines in range `r`, from the higher altitude `h` to the lower altitude `l`.
For example

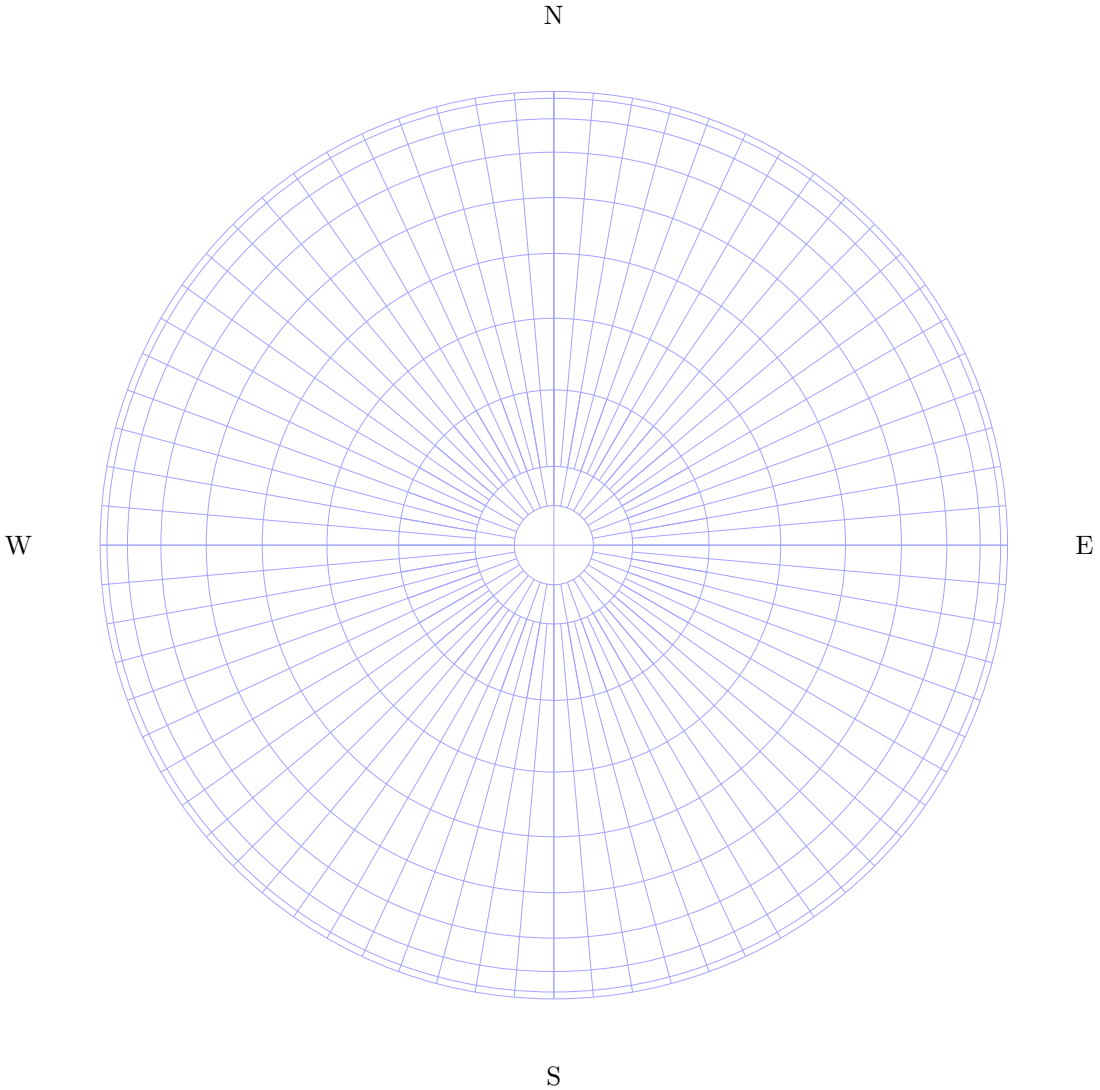
- `\spazimuthline{{0,10,...,360}}{85}{70}` draws every 10° azimuth from the 85° altitude to to 70° altitude.

- `\spazimuthline{{0,5,...,360}}{80}{0}` draws every 5° azimuth from the 80° altitude to to 0° altitude.

```
\begin{tikzpicture}[spradius=6]
\spcrosshair
\spaltitudecircle{{0,10,...,80,85}}
\spazimuthline{{0,10,...,360}}{85}{70}
\spazimuthline{{0,5,...,360}}{80}{0}

\spgeodirection
\end{tikzpicture}
```

drawazimuthline



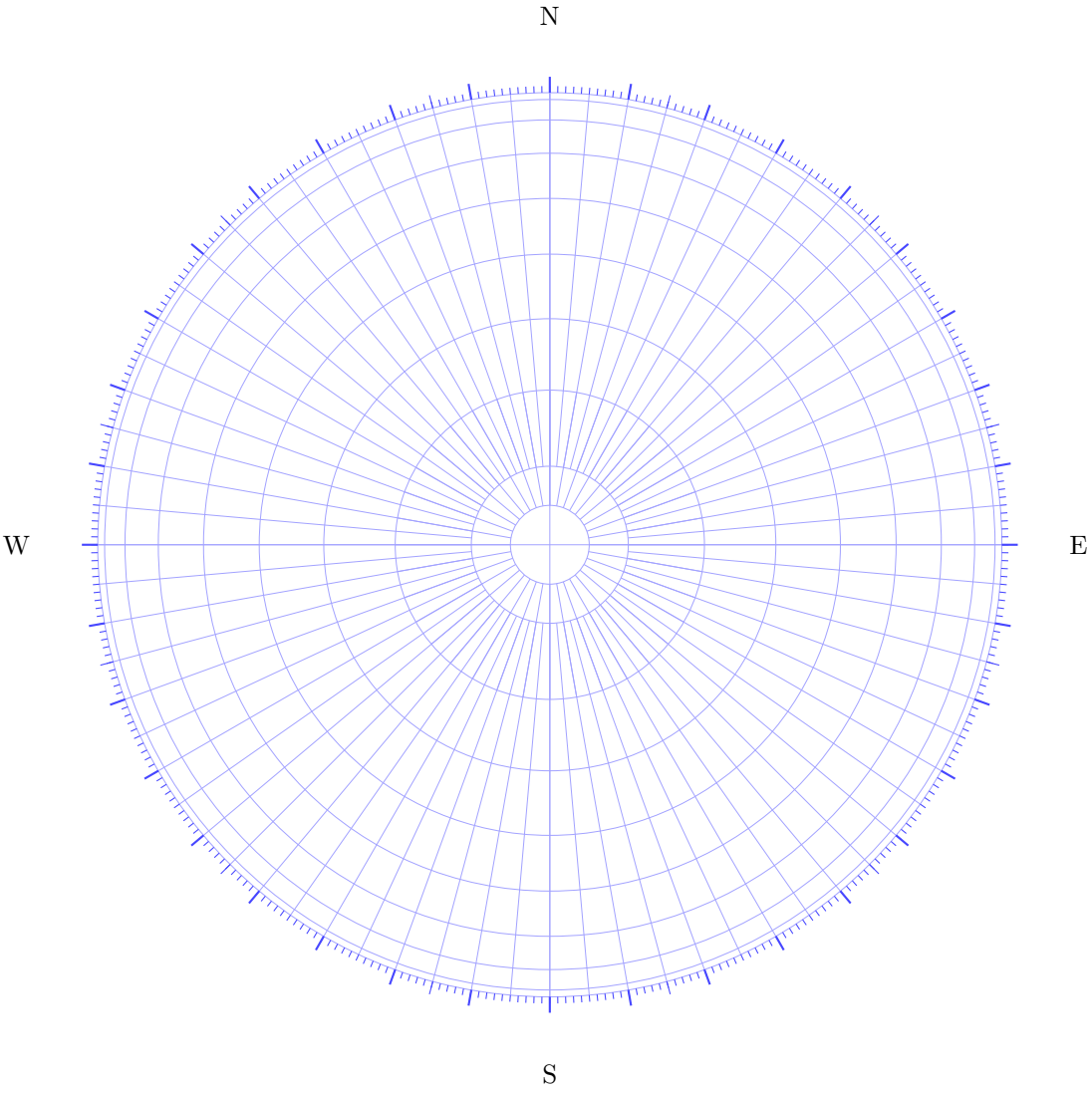
To draw azimuth ticks outside the horizon line, we can use `\spazimuthtick`. This command expects for now no argument.

```
\begin{tikzpicture}[spradius=6]
\spcrosshair
```

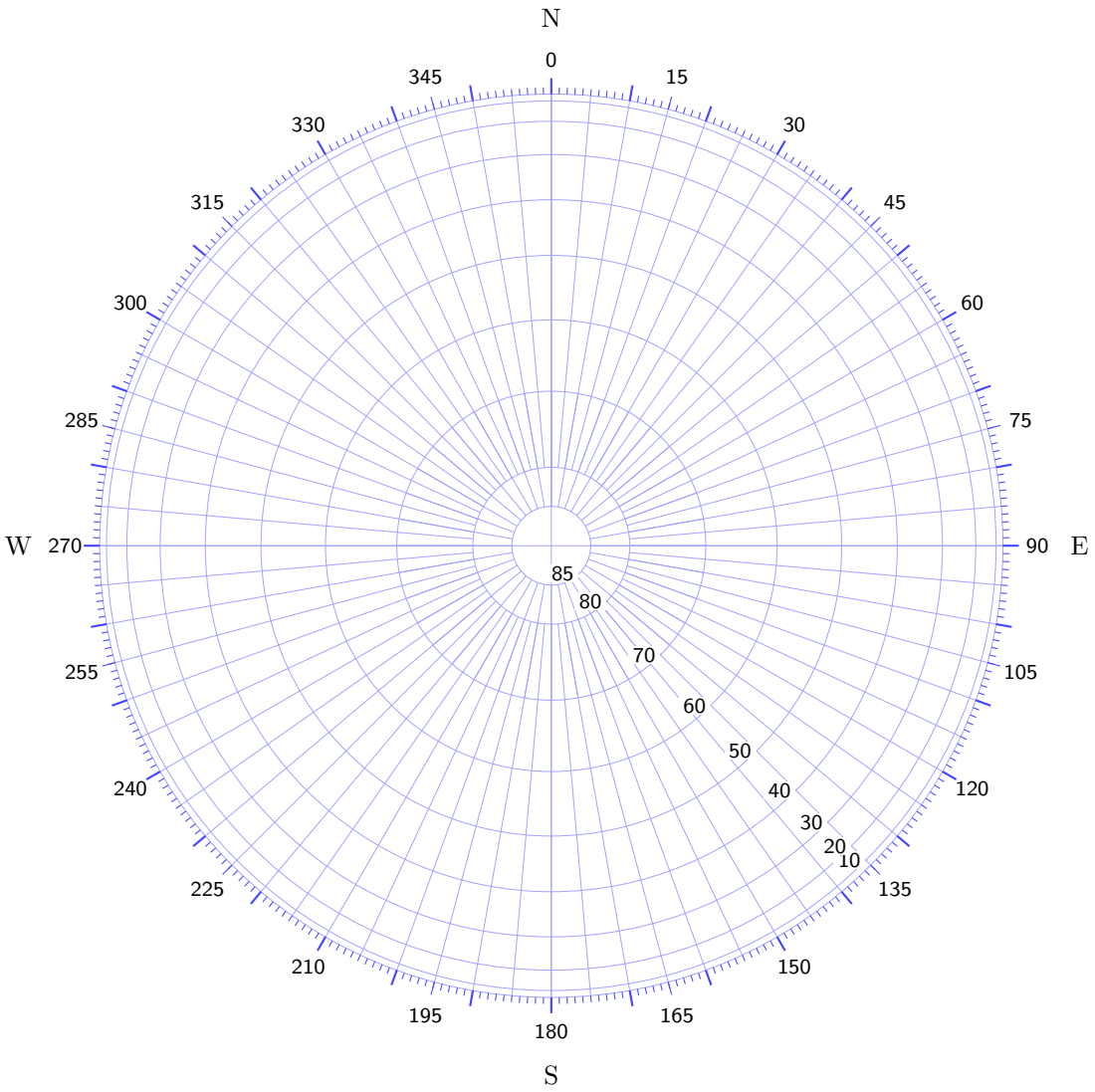
```
\spaltitudecircle{{0,10,...,80,85}}
\spazimuthline{{0,10,...,360}}{85}{70}
\spazimuthline{{0,5,...,360}}{80}{0}
\spazimuthtick

\spgeodirection
\end{tikzpicture}
```

drawazimuthtick



To draw labels of azimuth lines and altitude circles in the chart, we can use the commands
`\spaltitudelabel{r}` and
`\spazimuthlabel{r}`.



That it's, now we have a nice chart, on which we can draw positions of the sun from time to time.

1.2.3 Position of the sun

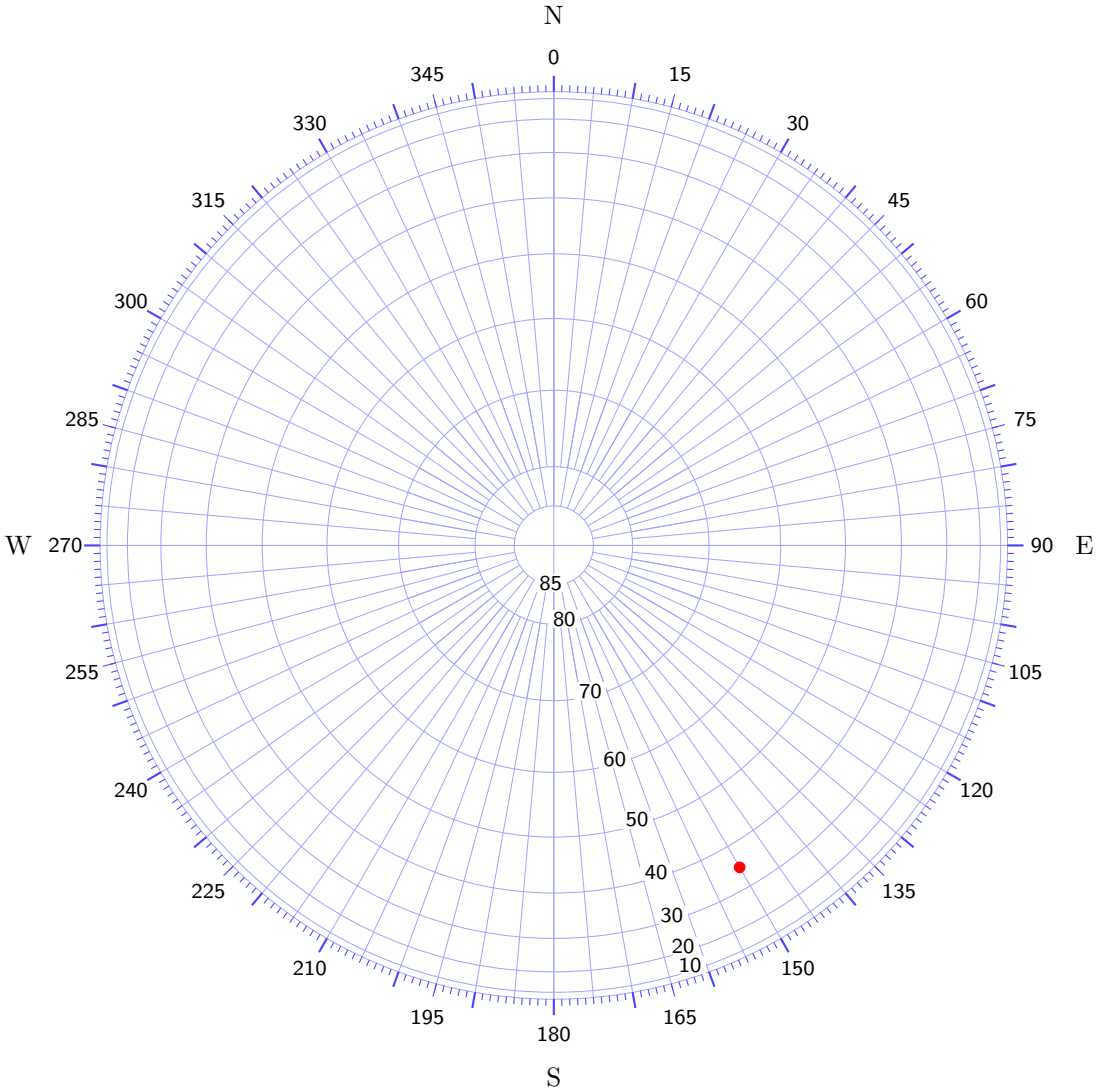
We can easily plot the position of the sun in the chart with the coordinate `sunpath`, if the azimuth and the altitude are given. For example, to plot the position of the sun with 150° Azimuth and 22° Altitude, we just use the `path` command as following:

```
\path[fill=red,draw=red] (sunpath cs:azi=150,alt=22);
```

The result would be

```
 ...
\path[fill=red,draw=red] (sunpath cs:azi=150,alt=35) circle[radius=2pt];
\spaltitudelabel{{10,20,...,80,85}}[160]
 ...
```

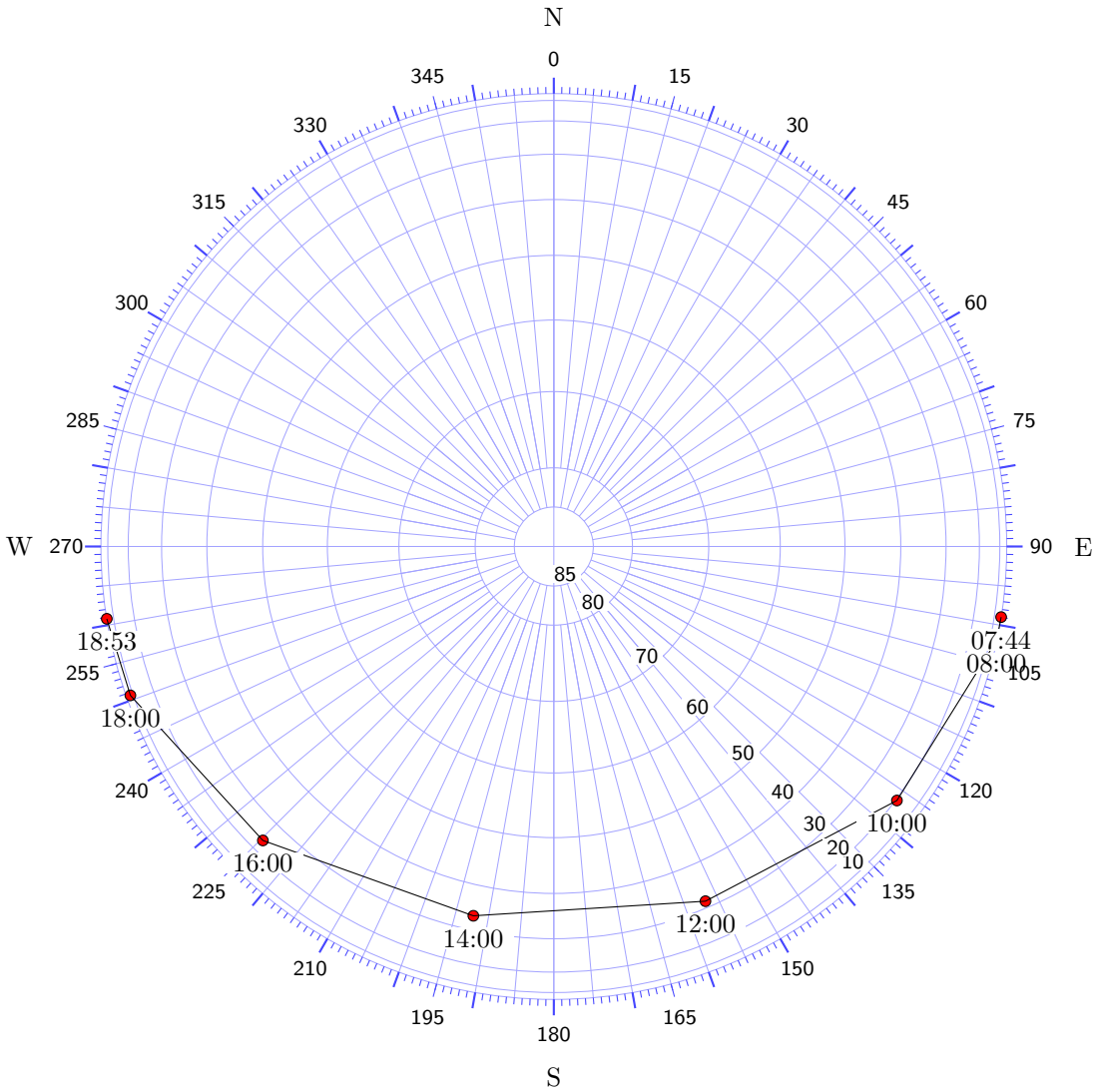
sunpath cs



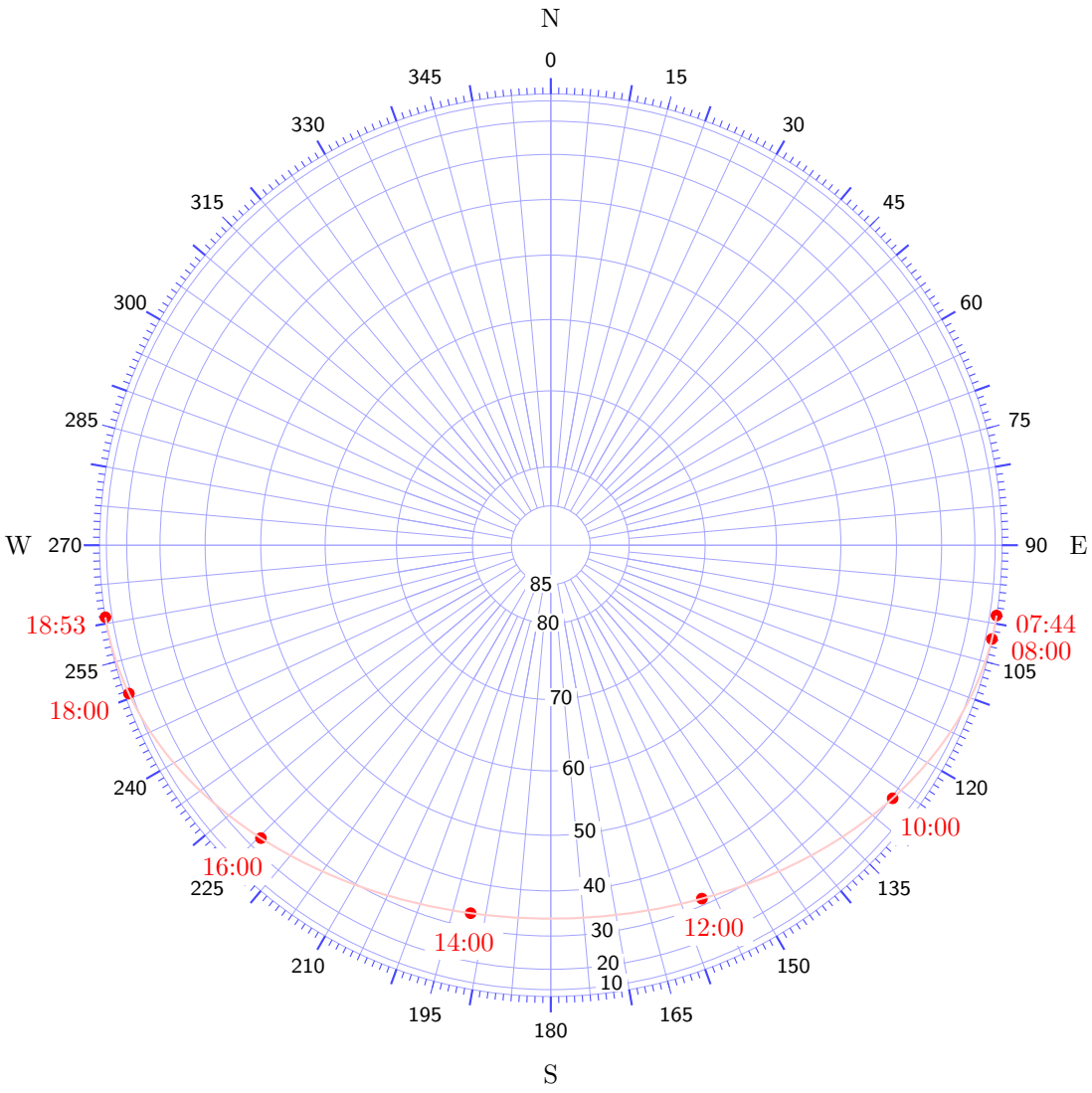
The command `\spaltitudelabel` can also take an optional argument to set altitude label on other azimuth. This can be useful if the labels cover distract important points on chart. In this chart it is set to be 160°. So one can easily read the azimuth of the sun on the chart.
We can also connect the position of the sun to a path, for example with the positions given in the following table

Time	Azimuth	Altitude
07:44	98.968673	-0.208672
08:00	102.009695	2.035492
10:00	126.513583	19.499874
12:00	156.854847	31.593335
14:00	192.292832	33.425294
16:00	224.708002	24.034984
18:00	250.626597	7.619801
18:53	260.810553	-0.244637

we can get a sun path like this:



But this chart is not nice. If the data is machine readable, we can generate all stuffs of the chart automatically. This chart below is generated from the table above. Just use your favourite programming language to process sun data.

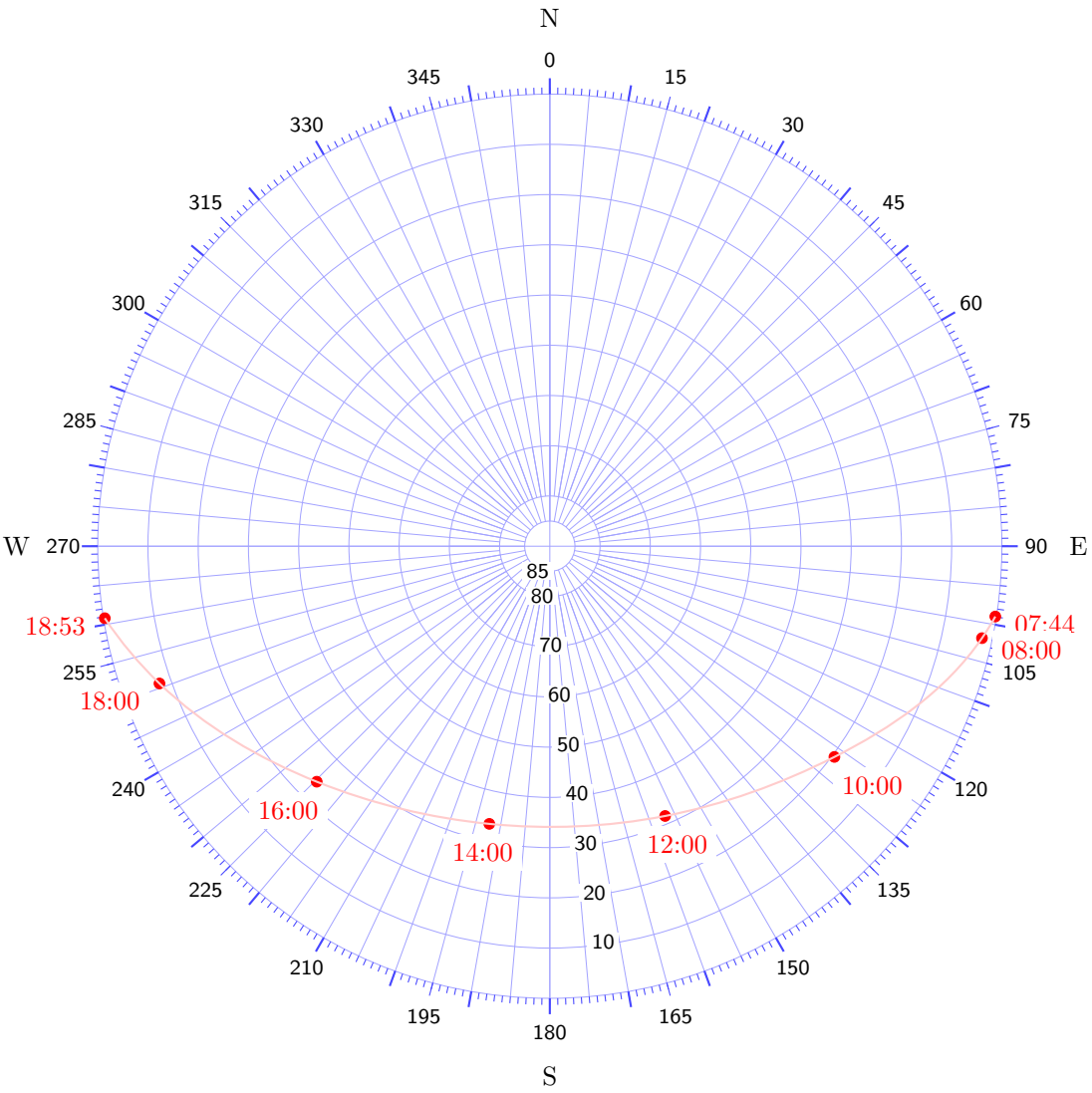

N
0
15
30
45
60
75
90
E
105
120
135
150
165
180
S
195
210
225
240
255
270
W
285
300
315
330
345
85
80
70
60
50
40
30
20
10
07:44
08:00
10:00
12:00
14:00
16:00
18:00
18:53

The part, which makes the chart nicer, is there:

```
...
\coordinate (P0) at (sunpath cs:azi=98.968673,alt=-0.208672);
\coordinate (P1) at (sunpath cs:azi=102.009695,alt=2.035492);
\coordinate (P2) at (sunpath cs:azi=126.513583,alt=19.499874);
\coordinate (P3) at (sunpath cs:azi=156.854847,alt=31.593335);
\coordinate (P4) at (sunpath cs:azi=192.292832,alt=33.425294);
\coordinate (P5) at (sunpath cs:azi=224.708002,alt=24.034984);
\coordinate (P6) at (sunpath cs:azi=250.626597,alt=7.619801);
\coordinate (P7) at (sunpath cs:azi=260.810553,alt=-0.244637);
\path[sun point] (P0) circle;
\path[sun point] (P1) circle;
\path[sun point] (P2) circle;
\path[sun point] (P3) circle;
\path[sun point] (P4) circle;
\path[sun point] (P5) circle;
\path[sun point] (P6) circle;
\path[sun point] (P7) circle;
\node[sun label,anchor=270-98.968673] at (P0) {07:44};
\node[sun label,anchor=270-102.009695] at (P1) {08:00};
\node[sun label,anchor=270-126.513583] at (P2) {10:00};
\node[sun label,anchor=270-156.854847] at (P3) {12:00};
\node[sun label,anchor=270-192.292832] at (P4) {14:00};
\node[sun label,anchor=270-224.708002] at (P5) {16:00};
\node[sun label,anchor=270-250.626597] at (P6) {18:00};
\node[sun label,anchor=270-260.810553] at (P7) {18:53};
\path[sun path curve] (P0) to [curve through={
    (P1) .. (P2) .. (P3) .. (P4) .. (P5) .. (P6)
    }]
  (P7) ;
...
```

To set the altitude scale (and also the altitude lines) equidistance, just use the option `altitude mapping=equidistance`. The chart above with same data looks like the chart below with this option.

`altitude mapping = equidistance`


N
0
15
30
45
60
75
90
E
105
120
135
150
165
180
S
195
210
225
240
255
W
270
285
300
315
330
345
85
80
70
60
50
40
30
20
10
07:44
08:00
10:00
12:00
14:00
16:00
18:00
18:53

2 Implementation

2.1 Package Dependenies

```
\RequirePackage{expl3}
\RequirePackage{tikz}
```

Load necsessary `tikz`-libraries.

```
\usetikzlibrary{calc,math,through}
```

2.2 `tikz`-Options for the new coordinate system

Setup options for `tikzpicture` environment.

`spradius` The radius of the 0° Altitude circle, default 5.5. This value can be accessed via macro `\spradius`.

`altitude mapping` How the altitude of the sun is mapped on the sunpath diagram. This mapping is a function $f(\theta) : [-90, 90] \to [0, r]$, where r is saved in `\spradius`.
Valid values are `spherical` and `equidistance`. Its default value is `spherical`.
This value can be accessed via macro `\altmapping`.
These options can be used like:

```
\begin{tikzpicture}[spradius=6,altitude projection=equidistance]
\coordinate (sunrise) at (sunpath cs:azi=105, alt=66.6);
\end{tikzpicture}
```

```
\pgfkeys{/tikz/.cd,
  spradius/.store in=\spradius,
  spradius=5.5,
  altitude mapping/.store in = \altmapping,
  altitude mapping=spherical
}
```

2.3 Define the new coordinate system `sunpath`

2.3.1 Azimuth and altitude

Define component `azi` (=Azimuth angle) and `alt` (=Altidude angle) for the coordinate system `sunpath`.

```
\tikzset{
  cs/azi/.store in=\tikz@cs@azi,
  cs/alt/.store in=\tikz@cs@alt,
}
```

2.3.2 Projection functions

Funtions to map the atitude of the sun to the altitude value on the sun path diagram.

`spherical` maps an altitude angle θ to the altitude radius on the diagram with the function

$$s(\theta) = r\cos(\theta).$$

`equidistance` maps an altitude angle θ to the altitude radius on the diagram with the function

$$e(\theta) = r - r \cdot \frac{|\theta|}{90}.$$

`altradius` this function is used in the coordinate system `sunpath` to determinate the altitude radius of an azimuth angle on the sun path chart. It depends on the value of the option `altitude projection`.

`aziangle` maps the azimuth angle Φ to the azimuth angle on the diagram with the function

$$a(\Phi) = 90 - \Phi.$$

```
\tikzset{
  declare function = {
      spherical(\alt) = \spradius * cos(\alt);
      equidistance(\alt) = \spradius - \spradius*abs(\alt)/90;
      altradius(\alt) = \altmapping(\alt);
      aziangle(\x) = 90 - \x;
  }
}
```

2.3.3 Coordinate system `sunpath`

```
\tikzdeclarecoordinatesystem{sunpath}%
{
    \tikzset{cs/.cd,azi=0,alt=0,#1}
    \tikzmath{
      \r = altradius(\tikz@cs@alt);
      \angle = aziangle(\tikz@cs@azi);
    }
    \pgfpointadd{\pgfpointxy{0}{0}}{%
        \pgfpointpolarxy{\angle}{\r}
    }
}
```

2.4 Setup optical options for sunpath diagram

These are pre-defined Ti*k*Z style for components of the chart. They can be easily changed by using `\tikzset`.

`sunpath grid` style for azimuth lines and altitude circles

`sunpath tick` style for ticks around the horizon line

`sunpath minor tick` style for minor ticks around horizon line

`altitude label` 80

`azimuth label` 350 style for text label of altitude circle respective azimuth line

`direction label` N E S W style for text label of four directions

```
\tikzset{
  sunpath grid/.style={help lines,color=blue!45!white!80},
  sunpath tick/.style={draw,thick,color=blue!90!white!80},
  sunpath minor tick/.style={draw,thin,color=blue!90!white!80},
  altitude label/.style={
      font=\footnotesize\sffamily,
      fill=white,minimum width={width("90")+2pt},
      inner sep=0.5pt
  },
  azimuth label/.style={
      font=\footnotesize\sffamily,
      minimum width={width("360")+2pt},
      inner sep=0.5pt
```

```
  },
  direction label/.style={
      font=\normalsize\rmfamily
  }
}
```

2.5 Expose some commands for end-user

`\spcrosshair` `[`⟨*style*⟩`]`

Draws a thin line from North to South and a thin line from East to West. Default value of `[`⟨*style*⟩`]` is `sunpath grid`.

```
\NewDocumentCommand\spcrosshair{O{sunpath grid}}{
    \draw[#1] (-\spradius,0) -- (\spradius,0);
    \draw[#1] (0,-\spradius) -- (0,\spradius);
}
```

`\spgeodirection` `[`⟨*offset*⟩`]` `[`⟨*style*⟩`]`

Puts four geographic directions North, East, South, West around the horinzon line. `[`⟨*offset*⟩`]` is the distance from horizon to the Ti*k*Z node of the directions. Its default value is 22pt. Set it to zero causes that the directions are set very near to the horizon line.

```
\NewDocumentCommand\spgeodirection{O{22pt} O{direction label}}{
  \foreach \dname / \dgrad in {N/0, E/90, S/180, W/270}{
      \tikzmath{
        \polarangle = aziangle(\dgrad);
      }
      \coordinate (D) at (\polarangle:\spradius cm + #1);
      \node[#2,anchor=270-\dgrad] at (D) {\dname};
  }
}
```

`\spaltitudecircle` `{`⟨*range*⟩`}` `[`⟨*style*⟩`]`

Draws altitude circle given by `{`⟨*range*⟩`}`. The argument `{`⟨*range*⟩`}` must be a valid Ti*k*Z-range, which can be used in `\foreach`. For example `{{10,20,...,80,85}}`. The argument `[`⟨*style*⟩`]` define the style of altitude circles, default is `sunpath grid`.

```
\NewDocumentCommand\spaltitudecircle{m O{sunpath grid}}{
  \foreach \altitude in #1 {
    \coordinate (A) at (sunpath cs:azi=0,alt=\altitude) ;
    \path[draw,sunpath grid] (0,0) circle[radius=altradius(\altitude)];
  }
}
```

`\spaltitudelabel` `{`⟨*range*⟩`}` `[`⟨*azimuth*⟩`]` `[`⟨*style*⟩`]`

Draws the labels of altitude circles given by `{`⟨*range*⟩`}`. Range must be an in Ti*k*Z valid numeric range which can be used in `\foreach`. For example `{{10,20,...,80}}` The labels are placed along the azimuth `[`⟨*azimuth*⟩`]` (default 135) and typeset with style `[`⟨*style*⟩`]` (default `altitude label`).

```
\NewDocumentCommand\spaltitudelabel{m O{135} O{altitude label}}{
  \foreach \altitude in #1 {
      \coordinate (A) at (sunpath cs:azi=#2,alt=\altitude) ;
      \node [anchor=east,#3] at (A) {\altitude};
    }
}
```

`\spazimuthlabel` {⟨*range*⟩}[⟨*style*⟩]

```
\NewDocumentCommand\spazimuthlabel{m O{azimuth label}}{
  \foreach \azimuth in #1 {
      \tikzmath{
        \polarangle = aziangle(\azimuth);
      }
      \coordinate (D) at (\polarangle:\spradius cm + 13pt);
      \node[#2] at (D) {\azimuth};
  }
}
```

`\spazimuthline` {⟨*range*⟩}{⟨*start alt*⟩}{⟨*end alt*⟩}

```
\NewDocumentCommand\spazimuthline{m m m}{
  \foreach \azimuth in #1{
      \draw[sunpath grid]
        (sunpath cs:azi=\azimuth,alt={#2}) -- (sunpath cs:azi=\azimuth,alt={#3});
  }
}
```

`\spazimuthtick` [⟨*major*⟩][⟨*minor*⟩][⟨*mid*⟩]

Draws ticks along and outside the horizon circle. The optional arguments [⟨*major*⟩], [⟨*minor*⟩] and [⟨*mid*⟩] are the length of major ticks (every 10° from Zero), minor ticks (every 1°, from 1°) and the length of the middle ticks (every 30°, from 15°). Their default values are 6pt, 2.5pt and 5pt.

```
\NewDocumentCommand\spazimuthtick{O{6pt} O{2.5pt} O{5pt}}{
  \foreach \azimuth in {10,20,...,360}{
    \tikzmath{
        \pa = aziangle(\azimuth);
    }
    \path[sunpath tick] (\pa:\spradius) -- (\pa:{\spradius cm + #1});
  }

  \foreach \azimuth in {1,2,...,360}{
    \tikzmath{
        \pa = aziangle(\azimuth);
    }
    \path[sunpath minor tick] (\pa:\spradius) -- (\pa:{\spradius cm + #2});
  }

  \foreach \azimuth in {15,45,...,345}{
      \tikzmath{
          \pa = aziangle(\azimuth);
      }
      \path[sunpath minor tick] (\pa:\spradius) -- (\pa:{\spradius cm + #3});
    }
}
```

Change History

v0.1-Alpha

General: Initial implementation . . 15

v0.2-Alpha

General: Small fixes in

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.