optics — a tikz library for optics drawings

Michel Fruchart
`michel (dot) fruchart [at] ens-lyon (dot) org`

2020-01-25 – version 0.2.5-alpha
`https://github.com/fruchart/tikz-optics`

1 Introduction

The aim of this library is to ease the creation of optical schematics including lenses, mirrors, and so on. The physically (in)accurate ray tracing is left to the user.

1.1 Legal matters

This library can be distributed and modified under the LaTeX Project Public License (LPPL), version 1.3c[1]. It can also be distributed and modified under the GNU General Public License (GNU GPL), either version 2[2], or any ulterior version published by the Free Software Foundation. Its documentation (that you are currently reading) can be distributed and modified under the LaTeX Project Public License (LPPL), version 1.3c[3]. It can also be distributed and modified under the GNU Free Documentation License (GNU FDL), either version 1.3[4], or any ulterior version published by the Free Software Foundation.

1.2 Installation of the library

If you have an up-to-date Texlive installation, the `tikz-optics` library might already be installed on your system. Try using the command `\usetikzlibrary{optics}` in your TEX file. If it does not work, you should either update your TEX installation or install the library manually (see below).

This library is a « tikz library ». It can be used it two ways :

— add the file `tikzlibraryoptics.code.tex` in a folder where TEX can find it, for instance in your `TEXMFHOME` folder[5], then use the command `\usetikzlibrary{optics}` in your TEX code ;

— directly include the file `tikzlibraryoptics.code.tex` with the `\input` command.

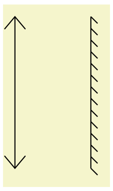
When the library is installed in a `TEXMF` folder, say `/home/agamemnon/texmf/`, the TDS structure must be respected for the library to be found[6]. As a consequence, the file `tikzlibraryoptics.code.tex` should be put in the folder `home/agamemnon/texmf/tex/latex/`, or in a subdirectory such as `home/agamemnon/texmf/tex/latex/tikzopt`

1.3 Basic usage

`/tikz/use optics` (no value)

Once the library is installed, it can be loaded with `\usetikzlibrary{optics}`. Then, it should be activated by applying the key `use optics` to a `tikzpicture`[7]

1. `http://latex-project.org/lppl/lppl-1-3c.txt`
2. `http://www.gnu.org/licenses/gpl-2.0.en.html`
3. `http://latex-project.org/lppl/lppl-1-3c.txt`
4. `https://www.gnu.org/licenses/fdl-1.3.en.html`
5. It can be found using `kpsewhich -var-value TEXMFHOME`.
6. See e.g. `https://www.ctan.org/TDS-guidelines`
7. The key `use optics` loads the relevant elements from `/tikz/optics/` to `/tikz/`, so they can be used directly. They are initially kept in a separate namespace to reduce the risk of name collision.



```
\begin{tikzpicture}[use optics]
  \node[lens] at (0,0) {};
  \node[mirror] at (1cm,0) {};
\end{tikzpicture}
```

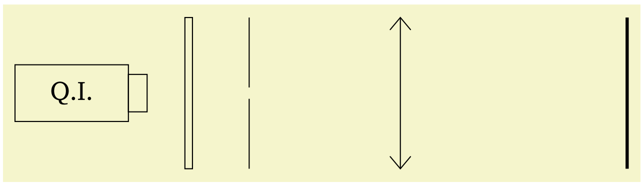
If `use optics` is not used, the code will fail miserably (in best case scenarios) or will display unexpected behaviors (in other scenarios).

2 Examples

2.1 Image of a slit on a screen

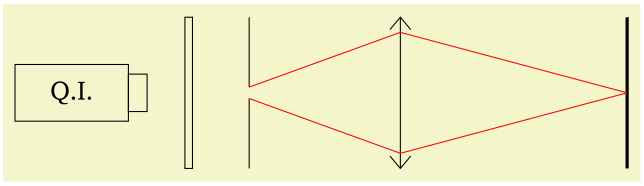
Let us draw a simple schematic : the image of a slit on a screen.

The first step consists in positioning all elements. We start with the light source, located in `(0,0)`. Each further element is added to the right of the previous ones. For instance, the heat filter is located at `0.5cm` of the halogen lamp output (anchor `aperture east` of the node `node quartz iode`) with the code `right=0.5cm of (quartz iode.aperture east)`. I chose to position the screen and lens with respect to the slit, but other choices are equally possible.



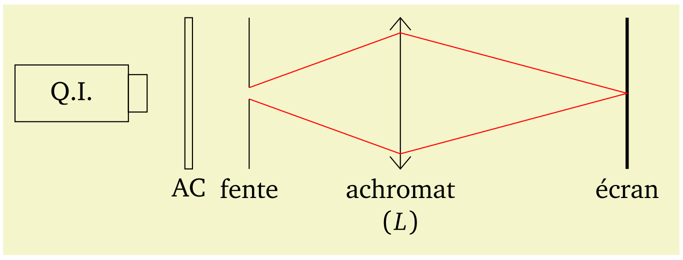
```
\begin{tikzpicture}[use optics]
  \node[halogen lamp] (quartz iode) at (0,0) {Q.I.};
  \node[heat filter,right=0.5cm of quartz iode.aperture east] (AC) {};
  \node[slit,right=0.75cm of AC] (fente) {};
  \node[lens,right=2cm of fente] (L) {};
  \node[screen,right=5cm of fente] (screen) {};
\end{tikzpicture}
```

We then wish to draw the light rays. Here, we are in luck : all the needed positions are predefined anchors of the existing `nodes`. For instance, `(L.lens south)` is a bit before the very end of the lens `L` to make the picture nicer (the very end can still be accessed with the anchor `(L.south)`). Let us then connect the relevant anchors with `--` in a `\draw` command.



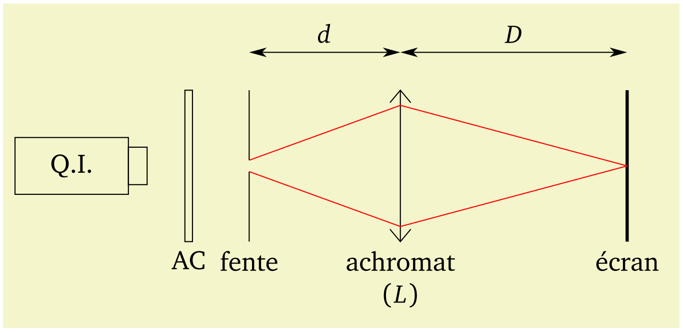
```
\begin{tikzpicture}[use optics]
  \node[halogen lamp] (quartz iode) at (0,0) {Q.I.};
  \node[heat filter,right=0.5cm of quartz iode.aperture east] (AC) {} ;
  \node[slit,right=0.75cm of AC] (fente) {};
  \node[lens,right=2cm of fente] (L) {};
  \node[screen,right=5cm of fente] (screen) {};

  \draw[red] (fente.slit north) -- (L.lens north) -- (screen.center)
  (fente.slit south) -- (L.lens south) -- (screen.center);
\end{tikzpicture}
```

The next step consists in adding labels. Surprisingly, this is done with the `label` key, with a code like `label=`⟨*texte*⟩, or more generally `label=[`⟨*opts*⟩`]`⟨*pos*⟩`:`⟨*text*⟩. As an example, `align=center` will be required for multiline labels.



```
\begin{tikzpicture}[use optics]
  \node[halogen lamp] (quartz iode) at (0,0) {Q.I.};
  \node[heat filter,right=0.5cm of quartz iode.aperture east,label={below:AC}] (AC) {} ;
  \node[slit,right=0.75cm of AC,label={below:fente}] (fente) {};
  \node[lens,right=2cm of fente,label={[align=center]below:achromat \\ $(L)$}] (L) {};
  \node[screen,right=5cm of fente,label={below:\'ecran}] (screen) {};

  \draw[red] (fente.slit north) -- (L.lens north) -- (screen.center)
  (fente.slit south) -- (L.lens south) -- (screen.center);
\end{tikzpicture}
```

Finally, we add arrows to indicate the different lengths. To do so, we'll define a `coordinate` (a particular `node` that is not drawn, and only has one anchor). The vertical position of this `coordinate` (point 1) and the horizontal coordinates of the (centers of the) various objects (point 2) define new positions through the `tikz` syntax `(point 1 -| point 2)`.

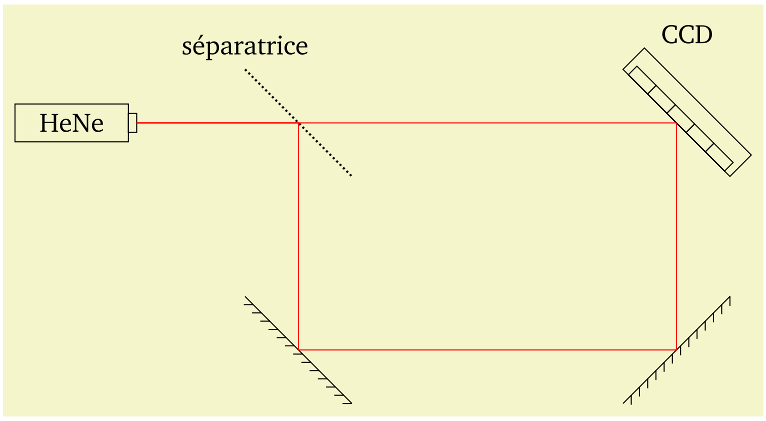


```
\begin{tikzpicture}[use optics]
  \node[halogen lamp] (quartz iode) at (0,0) {Q.I.};
  \node[heat filter,right=0.5cm of quartz iode.aperture east,label={below:AC}] (AC) {} ;
  \node[slit,right=0.75cm of AC,label={below:fente}] (fente) {};
  \node[lens,right=2cm of fente,label={[align=center]below:achromat \\ $(L)$}] (L) {};
  \node[screen,right=5cm of fente,label={below:\'ecran}] (screen) {};

  \draw[red] (fente.slit north) -- (L.lens north) -- (screen.center)
  (fente.slit south) -- (L.lens south) -- (screen.center);

  \coordinate (arrow origin) at (0,1.5cm);

  \draw[>=technical,<->] (arrow origin -| fente) -- (arrow origin -| L) node[midway,above] {$d$};
  \draw[>=technical,<->] (arrow origin -| L) -- (arrow origin -| screen) node[midway,above] {$D$};
\end{tikzpicture}
```

It is also possible to use the style `dim arrow` instead.

2.2 Interferences

Here, we compute the nodes' position instead of using `right=of` and friends.

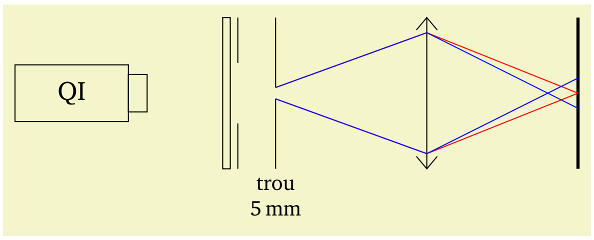


```
\begin{tikzpicture}[use optics]
  \node[laser] (L) at (0,0) {\ce{HeNe}};
  \node[semi-transparent mirror,rotate=45] (ST) at ($(L)+(3cm,0)$) {};
  \node[above] at (ST.north) {s\'eparatrice};
  \node[mirror,rotate=-135] (M1) at ($(ST)+(0,-3cm)$) {};
  \node[mirror,rotate=-45] (M2) at ($(M1)+(5cm,0)$) {};
  \node[sensor line,rotate=45,anchor=pixel 3 west,label={[label distance=0.5cm]above right:CCD}]
    (CCD) at ($(ST)+(5cm,0)$) {};
  \draw[red] (L.aperture east) -- (ST.center) -- (M1.center) -- (M2.center) -- (CCD.pixel 3 west);
  \draw[red] (L.aperture east) -- (ST.center) -- (CCD.pixel 3 west);
\end{tikzpicture}
```

The code `anchor=pixel 3 west` is used so that the center of the receptor is located at `($(ST)+(5cm,0)$)`.
We also showcase various ways of adding a label : creating a `node` at the right place with the relevant text (here *séparatrice*), using the `label` key, etc. (see the pgf/tikz manual).

2.3 Dispersion

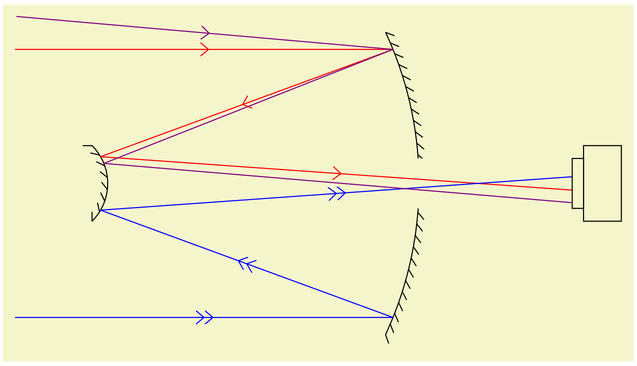
The syntax `($(A)!0.6!(B)$)` (*partway modifiers*, read the tikz manual for details) allows to create a point located at the position `0.6` between the points `(A)` et `(B)` (`0` correspond à `(A)` et `1` à `(B)`).



```
\begin{tikzpicture}[use optics]
  % [align=center] permet les labels multiligne
  % [font=\footnotesize] fait un texte plus petit
  \tikzset{every label/.style={align=center,font=\footnotesize}}

  \node[halogen lamp] (S) at (0,0) {QI};
  \node[heat filter, right=of S] (AC) {};
  \node[diaphragm, right=0.1cm of AC] (diaphragme) {};
  \node[slit, right=0.5cm of diaphragme,label={south:{trou \\ \SI{5}{\milli\meter}}}] (T) {};
  \node[lens,right=2cm of T] (L) {};
  \node[screen,right=2cm of L] (ecran) {};

  \draw[red]
  (T.slit north) -- (L.lens north) -- (ecran.center)
  (T.slit south) -- (L.lens south) -- (ecran.center);

  \draw[blue]
  (T.slit north) -- (L.lens north) -- ($(ecran.north)!0.6!(ecran.south)$)
  (T.slit south) -- (L.lens south) -- ($(ecran.north)!0.4!(ecran.south)$);
\end{tikzpicture}
```

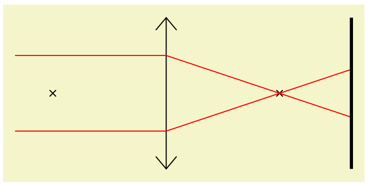
2.4 A Cassegrain telescope

In order to draw a mirror with a hole, we use the tikz command `\clip` (placed in a `scope`, so that only the content of the scope is affected by the `\clip`) We then draw the light rays manually.



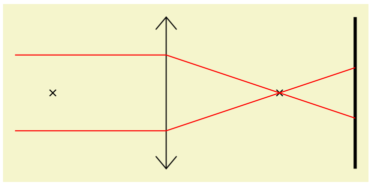
```
\begin{tikzpicture}[use optics]
% mirror with hole
\begin{scope}
  \clip (-0.75cm,-2.2cm) rectangle (1cm,0-0.33cm) (-0.75cm,2.2cm) rectangle (1cm,0+0.33cm);
  \node[spherical mirror, object height=4cm, spherical mirror angle=50] (M1) at (0cm,0) {};
\end{scope}

% small mirror
\node[convex mirror, spherical mirror orientation=rtl,
    object height=1cm, spherical mirror angle=90] (M2) at (-4cm,0) {};

% convergence point
\coordinate (F) at (1cm,0);

% red ray
\begin{scope}[red]
  \draw[-<-] (M1.22) coordinate (P1) -- +(-5cm,0);
  \draw[->-] (P1) -- (M2.30) coordinate (Q1);
  \draw[->-] (Q1) -- ($(Q1)!1.25!(F)$) coordinate (R1);
\end{scope}

% blue ray
\begin{scope}[blue]
  \draw[-<<-] (M1.-22) coordinate (P2) -- +(-5cm,0);
  \draw[->>-] (P2) -- (M2.-30) coordinate (Q2);
  \draw[->>-] (Q2) -- ($(Q2)!1.25!(F)$) coordinate (R2);
\end{scope}

% violet ray
\begin{scope}[violet]
  \draw[-<-] (M1.22) coordinate (P3) -- +(175:5cm);
  \draw (P3) -- (M2.22) coordinate (Q3);
  \draw (Q3) -- ($(Q3)!1.25!($(F)+(0,-0.15cm)$)$) coordinate (R3);
\end{scope}

% sensor
\node[generic sensor, anchor=aperture west] at ($(R1)!0.5!(R2)$) {};
\end{tikzpicture}
```

2.5 Ray optics and computations



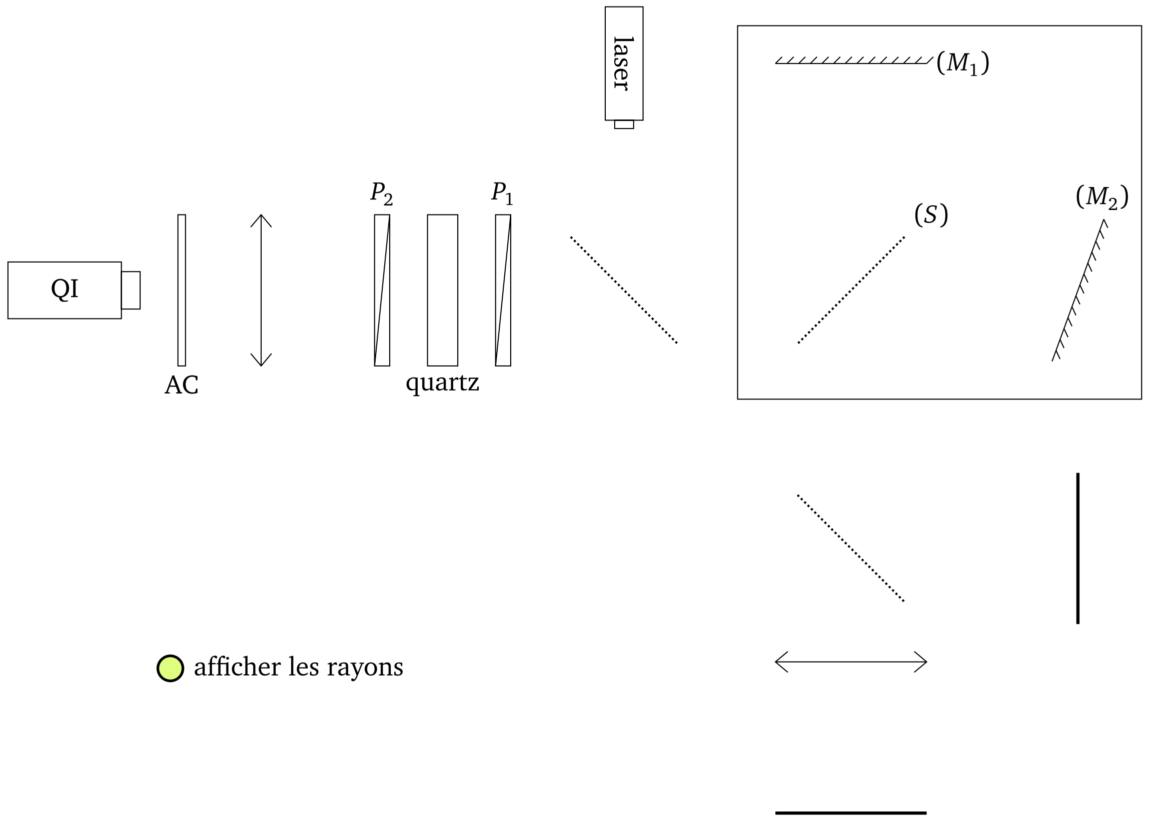
```
\begin{tikzpicture}[use optics]
  \node[lens,draw focal points,focal length=1.5cm,object height=2cm] (L) at (0,0) {};
  \coordinate (P) at (-2cm,0.5cm);
  \coordinate (Q) at (-2cm,-0.5cm);
  \draw[red,shorten >=-1cm] (P) -- ($(L.north)!(P)!(L.south)$) -- (L.east focus);
  \draw[red,shorten >=-1cm] (Q) -- ($(L.north)!(Q)!(L.south)$) -- (L.east focus);
  \node[screen] at (2.45cm,0) {};
\end{tikzpicture}
```

The preceding example is not very clean, because I had to manually enlarge the rays up to the screen. The next example where the intersection is first computed with a macro `\toVerticalProjection` is way nicer.



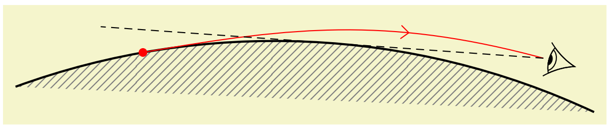
```
\begin{tikzpicture}[use optics]
  \node[lens,draw focal points,focal length=1.5cm,object height=2cm] (L) at (0,0) {} ;
  \coordinate (P) at (-2cm,0.5cm) ;
  \coordinate (Q) at (-2cm,-0.5cm) ;
  \node[screen] (S) at (2.5cm,0) {};

  \def\toVerticalProjection#1#2#3{let \p{1} = #1, \p{2} = #2, \p{3} = #3 in
   -- (\x{3},{\y{1}+(\y{2}-\y{1})/(\x{2}-\x{1})*(\x{3}-\x{1})})}

  \draw[red] (P) -- ($(L.north)!(P)!(L.south)$) coordinate (Plens)
  \toVerticalProjection{(Plens)}{(L.east focus)}{(S)};

  \draw[red] (Q) -- ($(L.north)!(Q)!(L.south)$) coordinate (Qlens)
  \toVerticalProjection{(Qlens)}{(L.east focus)}{(S)};

\end{tikzpicture}
```

2.6 Birefringence with a Michelson interferometer



This example uses intersections to draw light rays : the syntax is not the most short and fun to use, but it works. It is also a good occasion to show how to use the `ocgx` package with `tikz` (this only works with select PDF readers). The code of the figure is attached in the file birefringence_michelson.pgf.

2.7 A mirage



```
\begin{tikzpicture}[use optics]
  \pgfmathsetmacro\thetaA{110}
  \pgfmathsetmacro\thetaB{65}
  \pgfmathsetmacro\thetaP{100}
  \pgfmathsetmacro\thetaQ{70}
  \pgfmathsetmacro\deltaTheta{5}
  \newdimen\radius
  \pgfmathsetlength\radius{10cm}

  \newdimen\height
  \pgfmathsetlength\height{0.4cm}

  \draw[thick, pattern=north east lines, pattern color=gray] (0,0)
    arc [start angle=\thetaA, end angle=\thetaB, radius=\radius];
  \path (0,0) arc [start angle=\thetaA, end angle=\thetaP, radius=\radius] coordinate (P);
  \path (0,0) arc [start angle=\thetaA, end angle=\thetaQ, radius=\radius] coordinate (Q);
  \coordinate (QE) at ($(Q)+(\thetaQ:\height)$);

  \node[circle, draw,fill,red, inner sep=0, minimum size=0.1cm] at (P) {};
  \draw[red,->-={at=0.65}] (P) to[out={\thetaP-90}, in={\thetaQ+\deltaTheta+90}] (QE);
  \pic[scale=0.75,rotate={\thetaQ+\deltaTheta-90}] (eye) at (QE) {optics eye};

  \pgfmathsetmacro\thetaTangential{86}
  \path (0,0) arc [start angle=\thetaA, end angle=\thetaTangential, radius=\radius] coordinate (H);
  \draw[densely dashed,shorten >=-3cm] (eye-in) -- (H);
\end{tikzpicture}
```

3 Reference

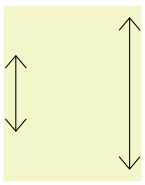
3.1 General considerations

3.1.1 Common options

Some options are shared by a lot of `shapes` (with the important exception of light sources and receptors)

`/tikz/optics/object height`=⟨*length*⟩ (no default, initially `2cm`)

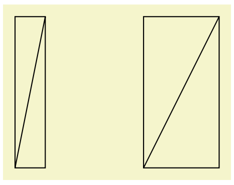
The key `object height` controls the height of most objects (it is the case if nothing is specified).



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens, object height=1cm] (L1) at (0,0) {};
  \node[lens, object height=2cm] (L2) at (3cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/object aspect ratio`=⟨*number or length*⟩ (no default, initially `0.2`)

The key `object aspect ratio` controls the aspect ratio of most objects having with a width. When ⟨*number*⟩ is `1`, the width of the object is equal to its height. When for instance ⟨*number*⟩=`1/2`, the width is half of the height. When ⟨*number or length*⟩ is a dimensionless number (such as `0.5`), it is interpreted as an aspect ratio (relative to the height). When it is a dimensioned length (such as `1cm`), it is directly interpreted as the width of the object.



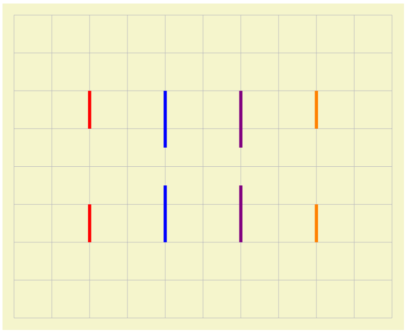
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[polarizer, object aspect ratio=0.2] (L1) at (0,0) {};
  \node[polarizer, object aspect ratio=0.5] (L2) at (4cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/object width` (style, no value)

The key `object width` is an alias for `object aspect ratio`.

3.1.2 Absolute lengths and relative lengths

Several elements have more than one adjustable length. Most of the time, it is enough to specify only one length (usually the height of the object) in an absolute way, i.e. with a length unit (`cm`, `pt`, `em`, etc.). The other lengths can be specified as a multiple of this absolute lengths scale (to do so, they should be inputed as a dimensionless number). Sometimes, it is more convenient to specify them as absolute length (a unit must then be used). In the following example, `slit height` is first specified as a multiple of `object height`, then as an absolute length.



```
\begin{tikzpicture}[use optics]
  \draw[style=help lines,gray!50]
    (-3cm,-2cm) grid[step=0.5cm] (2cm,2cm);
  \node[slit,object height=2cm,slit height=0.5,red,very thick]
    at (-2cm,0) {};
  \node[slit,object height=2cm,slit height=0.25,blue,very thick]
    at (-1cm,0) {};
  \node[slit,object height=2cm,slit height=0.5cm,violet,very
thick]
    at (0cm,0) {};
  \node[slit,object height=2cm,slit height=1cm,orange,very thick]
    at (1cm,0) {};
\end{tikzpicture}
```

Using relative lengths allows to rescale the object without modifying its global shape by changing the single absolute length only, instead of having to change all lengths.

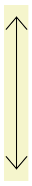
Optical elements have several « heights ». The total height is always called `objet height`. Other heights depend on the element. For instance, the size of a slit is `slit height`, the size of a lens is `lens height` (it does not change the drawing, but moves the anchors), etc.

3.2 Optical elements

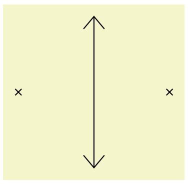
3.2.1 Lens

Shape `lens`

Draws a lens.



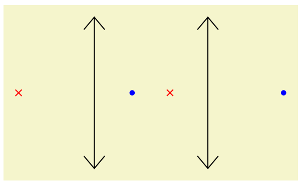
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens] (L) at (0,0) {};
\end{tikzpicture}
```



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens,draw focal points] (L) at (0,0) {};
\end{tikzpicture}
```

`/tikz/optics/draw focal points=⟨style⟩` (style, default `empty`)

The focal points of the lens can be marked with the key `draw focal points`, and the argument ⟨*style*⟩ of the key determines how they are draxs. For instance, `draw focal points=red` produces red markers at the focal points, and draw focal points=circle,draw=none,fill=blue blue circles at the focal points.



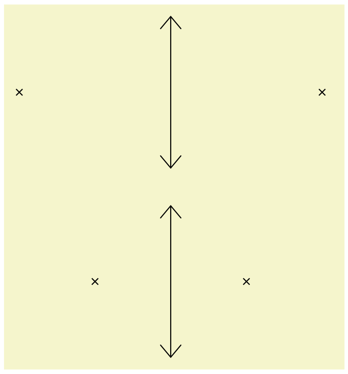
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens,draw focal points={red}]
  (L1) at (0,0) {};
  \node[lens,draw focal points={circle,draw=none,fill=blue}]
  (L2) at (3cm,0) {};
\end{tikzpicture}
```

/tikz/optics/object height=⟨*length*⟩ (no default, initially 2cm)

The key object height applies.

/tikz/optics/focal length=⟨*length*⟩ (no default, initially 1cm)

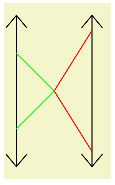
The key focal length determines the focal length of the lens.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens, focal length=1cm] (L1) at (0,0) {};
  \node[lens, focal length=2cm] (L2) at (0,5cm) {};
\end{tikzpicture}
```

/tikz/optics/lens height=⟨*number or length*⟩ (no default, initially 0.8)

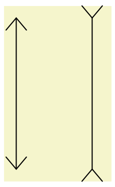
The key lens height determines the height of the lens. It can either be an absolute length (with a unit) or a relative length measured in units of the total height of the lens.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens] (L1) at (1cm,0) {};
  \node[lens, lens height=0.5] (L2) at (-1cm,0) {};
  \draw[red] (0,0) -- (L1.lens north) (0,0) -- (L1.lens south);
  \draw[green] (0,0) -- (L2.lens north) (0,0) -- (L2.lens south);
\end{tikzpicture}
```

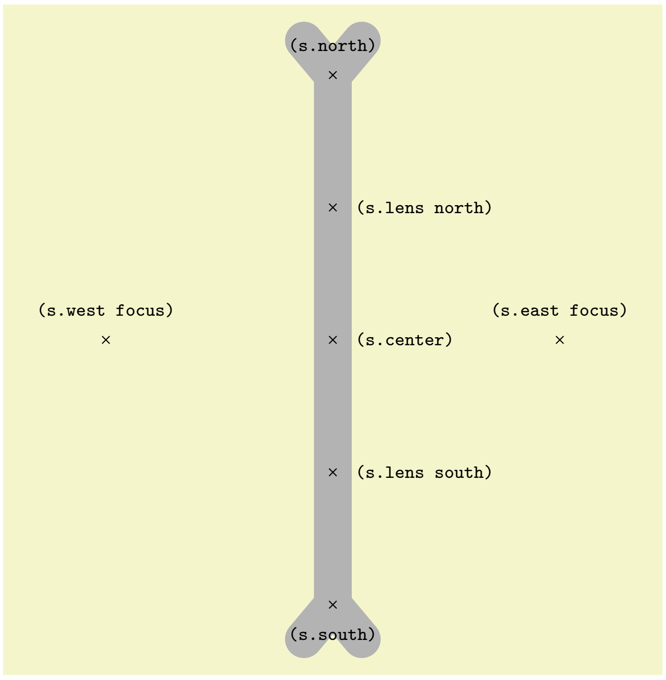
/tikz/optics/lens type (no value)

The key lens type controls the lens type; namely, lens type=converging draws a converging lens (default) while lens type=diverging draws a diverging lens.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens,lens type=converging] (L1) at (-1cm,0) {};
  \node[lens,lens type=diverging] (L2) at (1cm,0) {};
\end{tikzpicture}
```

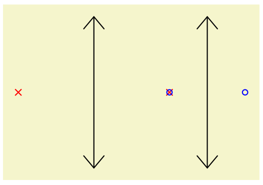
The following figure summarizes the anchors defined by lens.



```
\Huge
\begin{tikzpicture}[use optics]
\node[name=s,lens,object height=7cm,focal length=3cm,
lens height=0.5,line shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,lens north/right,lens south/right,center/right,
east focus/above,west focus/above}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

The keys `east focus` and `west focus` are respectively alias to `east focal point` and `west focal point`.

The `tikz` key `anchor=` can be used to position lenses with respect to each other.

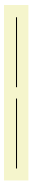


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[lens,draw focal points={red}]
  (L1) at (0,0) {};
  \node[lens,draw focal points={circle,draw=blue},
  focal length=0.5cm,anchor=west focus]
  (L2) at (L1.east focus) {};
\end{tikzpicture}
```

3.2.2 Slit

Shape `slit`

Draws a slit.



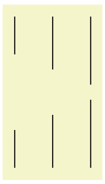
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[slit] (S) at (0,0) {};
\end{tikzpicture}
```

`/tikz/optics/object height`=⟨*length*⟩ (no default, initially `2cm`)

The key `object height` applies.

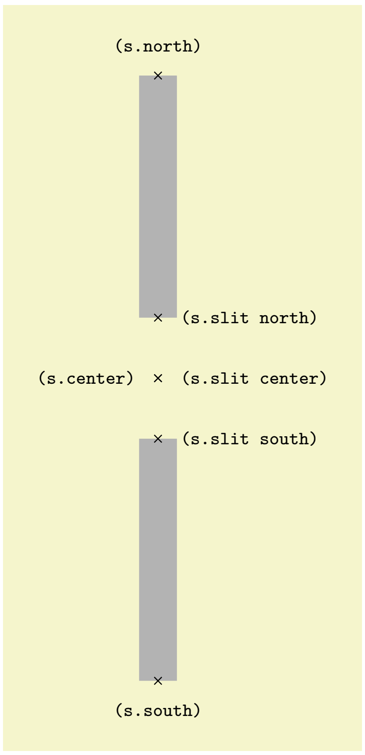
`/tikz/optics/slit height`=⟨*number or length*⟩ (no default, initially `0.075`)

The key `slit height` determines the height of the slit aperture (relative lengths are in units of the total object height). For instance, ⟨*number*⟩=`0.5` produces a slit half as large as the support. Similarly ⟨*length*⟩=`1cm` produces a slit with height `1cm`. The value of ⟨*number*⟩ should be smaller than one, and the value of ⟨*length*⟩ should be smaller than the value of `object height`. The results are unspecified (and probably terrible) when this is not the case.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[slit, slit height=0.5] (S) at (0,0) {};
  \node[slit, slit height=0.3] (S) at (1cm,0) {};
  \node[slit, slit height=0.1] (S) at (2cm,0) {};
\end{tikzpicture}
```

The following figure summarizes the anchors defined by `slit`.

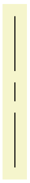


```
\Huge
\begin{tikzpicture}[use optics]
\node[name=s,slit,object height=8cm,
slit height=0.2,line shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,slit north/right,slit south/right,center/left,
slit center/right}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.3 Double slit

Shape `double slit`

Draws a double slit.



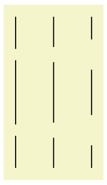
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[double slit] (S) at (0,0) {};
\end{tikzpicture}
```

/tikz/optics/object height=⟨*length*⟩ (no default, initially 2cm)

The key object height applies.

/tikz/optics/slit height=⟨*number or length*⟩ (no default, initially 0.075)

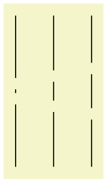
The key slit height determines with height of each slit (relative lengths are in units of the object height). Each slit will have a height ⟨*number or length*⟩.



```
\begin{tikzpicture}[use optics,scale=.5,optics/slit separation=0.5]
  \node[double slit, slit height=0.075] (S) at (0,0) {};
  \node[double slit, slit height=0.1] (S) at (1cm,0) {};
  \node[double slit, slit height=0.2] (S) at (2cm,0) {};
\end{tikzpicture}
```

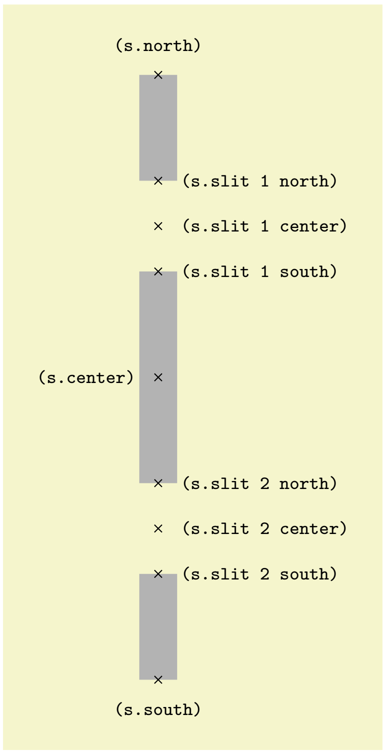
/tikz/optics/slit separation=⟨*number or length*⟩ (no default, initially 0.2)

The key slit separation determines the distance between the two slits (relative lengths are in units of the object height).



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[double slit, slit separation=0.1] (S) at (0,0) {};
  \node[double slit, slit separation=0.2] (S) at (1cm,0) {};
  \node[double slit, slit separation=0.3] (S) at (2cm,0) {};
\end{tikzpicture}
```

The following figure summarizes the anchors defined by double slit.

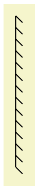


```
\Huge
\begin{tikzpicture}[use optics]
\node[double slit,name=s,object height=8cm, slit height=0.15,
slit separation=0.5, line shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,center/left,
slit 1 north/right,slit 1 south/right,slit 1 center/right,
slit 2 north/right,slit 2 south/right,slit 2 center/right}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.4 Mirror

Shape mirror

Draws a plane mirror.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[mirror] (S) at (0,0) {};
\end{tikzpicture}
```

/tikz/optics/object height=⟨*length*⟩ (no default, initially 2cm)

The key object height applies.

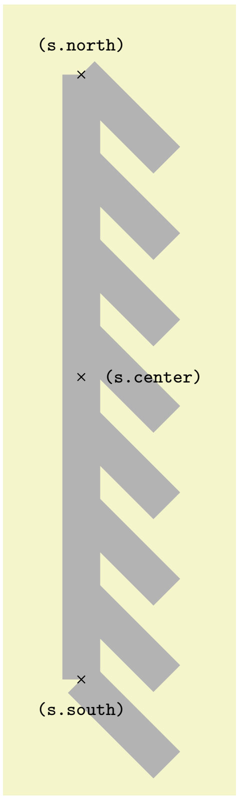
/tikz/optics/mirror decoration separation=⟨*number or length*⟩ (no default, initially 0.15cm)

Sets the key /pgf/decoration/segment length of the border decoration used to draw the mirror (see the tikz manual for details). The value us /pgf/decoration/segment length used when the argument is dimensionless is obtained by multiplying ⟨*number*⟩ by the height of the mirror.

/tikz/optics/mirror decoration amplitude=⟨*number or length*⟩ (no default, initially 0.125cm)

Sets the key /pgf/decoration/amplitude of the border decoration used to draw the mirror (see the tikz manual for details). The value of /pgf/decoration/amplitude used when the argument is dimensionless is obtained by multiplying ⟨*number*⟩ by the height of the mirror.

The following figure summarizes the anchors defined by mirror.



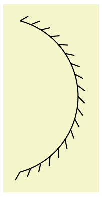
```
\Huge
\begin{tikzpicture}[use optics]
\node[mirror,name=s,object height=8cm,line shape example,
mirror decoration separation=0.141, mirror decoration amplitude=0.2] {};
\foreach \anchor/\placement in
{north/above,south/below,center/right}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.5 Spherical mirror

Shape spherical mirror

Draws a spherical mirror (convex or concave)

⚠ This part is not finished and can change without warning.



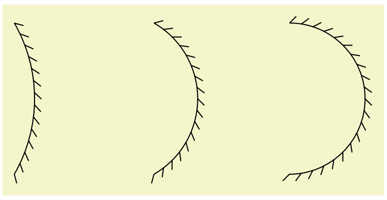
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[spherical mirror] (M) at (0,0) {};
\end{tikzpicture}
```

`/tikz/optics/object height`=⟨*length*⟩ (no default, initially `2cm`)

The key `object height` applies.

`/tikz/optics/spherical mirror angle`=⟨*angle*⟩ (no default, initially `150`)

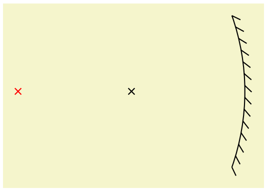
The key `spherical mirror angle` determines the aperture angle of the spherical mirror (an arc with height determined by `object height` and with angular aperture ⟨*angle*⟩ is drawn). Do not use ⟨*angle*⟩ = 0 ! Instead, use a plane `mirror`.



```
\begin{tikzpicture}[use optics]
  \node[spherical mirror, spherical mirror angle=60] at (0,0) {};
  \node[spherical mirror, spherical mirror angle=120] at (2cm,0) {};
  \node[spherical mirror, spherical mirror angle=180] at (4cm,0) {};
\end{tikzpicture}
```

⚠ The function `from_radius` is experimental.

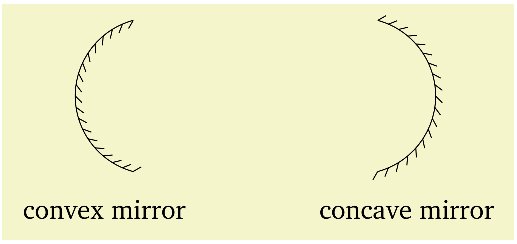
It can be useful to specify the radius of curvature of the mirror instead of the aperture angle. To do so, a function `from_radius(R)` computes the aperture radius corresponding to the radius `R`, with fixed `/tikz/optics/object height`. It is indeed not possible to have a height greater than twice the radius.



```
\begin{tikzpicture}[use optics]
  \node[spherical mirror, object height=2cm,
  spherical mirror angle=from_radius(3cm),
  draw mirror focus, draw mirror center={red}] (M) {};
\end{tikzpicture}
```

`/tikz/optics/spherical mirror type` (no value)

The key `spherical mirror type` determines the type of mirror : `spherical mirror type=concave` produces a concave mirror (default), and `spherical mirror type=convex` produces a convex mirror The styles `convex mirror` and `concave mirror` are shortcuts for `spherical mirror` along with this key.



```
\begin{tikzpicture}[use optics]
  \node[convex mirror, label={[label distance=0.25cm]south:convex mirror}] at (0cm,0) {};
  \node[concave mirror, label={[label distance=0.25cm]south:concave mirror}] at (4cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/concave mirror` (style, no value)

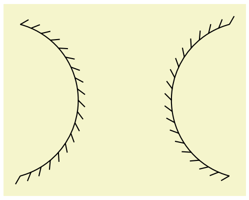
The style `concave mirror` is equivalent to `spherical mirror, spherical mirror type=concave` and draws a concave spherical mirror. See `spherical mirror type` for an example.

`/tikz/optics/convex mirror` (style, no value)

The style `convex mirror` is equivalent to `spherical mirror, spherical mirror type=convex` and draws a convex spherical mirror. See `spherical mirror type` for an example.

`/tikz/optics/spherical mirror orientation=`⟨*type*⟩ (no default, initially `ltr`)

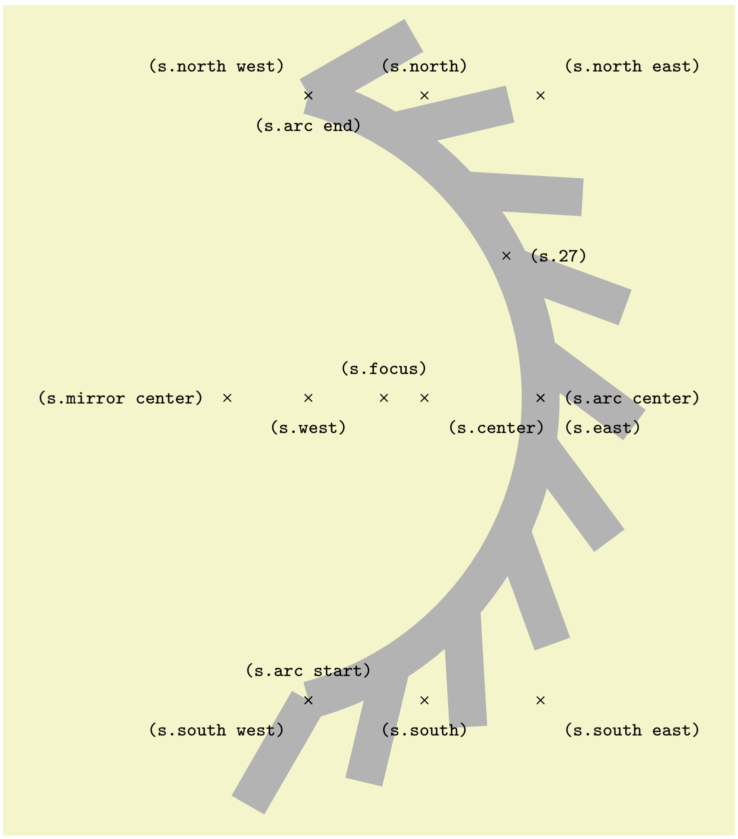
The key `spherical mirror orientation` determines the orientation of the mirror (namely, the assumed direction of propagation of light). The allowed values are `ltr` (« left to right ») and `rtl` (« right to left »).



```
\begin{tikzpicture}[use optics]
  \node[spherical mirror, spherical mirror orientation=ltr] at (0cm,0) {};
  \node[spherical mirror, spherical mirror orientation=rtl] at (2cm,0) {};
\end{tikzpicture}
```

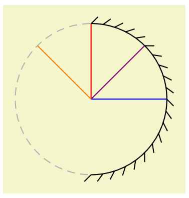
The crosshatch mirror decoration is controlled with the same keys as for `mirror` : `mirror decoration separation` et `mirror decoration amplitude`.

The following figure summarizes some of the anchors defined by `spherical mirror`.



```
\Huge
\begin{tikzpicture}[use optics]
\node[spherical mirror,name=s,object height=8cm,line shape example,
mirror decoration separation=0.141, mirror decoration amplitude=0.2] {};
\foreach \anchor/\placement in
{north/above,south/below,center/below right,
east/below right,
west/below,
north east/above right,
north west/above left,
south east/below right,
south west/below left,
mirror center/left,
arc start/above,
arc end/below,
arc center/right,
27/right,
focus/above}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

With spherical mirrors, tikz « border anchor » are particularly convenient. Numerical anchors (such as `node.27`) allow to access the boundary of the shape in the corresponding angular direction (here at 27°). Recall that in tikz, angles are expressed in degrees and increase counterclockwise from the x-axis. In the following example, the underlying circle has been drawn for clarity.

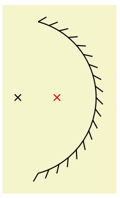


```
\begin{tikzpicture}[use optics,scale=.5]]
\coordinate (O) at (0,0);
\node[circle,draw, inner sep=0, outer sep=0,minimum height=2cm, densely
dashed, gray!60] (C) at (O) {};
\node[spherical mirror,draw,object height=2cm,anchor=mirror center,
    spherical mirror angle=180] (M) at (O) {};
\draw[blue] (O) -- (M.0);
\draw[violet] (O) -- (M.45);
\draw[red] (O) -- (M.90);
\draw[orange] (O) -- (M.135);
\end{tikzpicture}
```

⚠ The positions of the anchors with angles greater than ±`spherical mirror angle`/2 is not specified. Currently, they are treated as if the mirror was a full circle, but this might change in the future.

`/tikz/optics/draw mirror center`=⟨*style*⟩ (style, no default)

The style `draw mirror center` marks the center of the mirror (with the style ⟨*style*⟩ if specified).



```
\begin{tikzpicture}[use optics]
\node[concave mirror,draw mirror center, draw mirror focus={red}] {};
\end{tikzpicture}
```

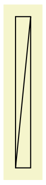
`/tikz/optics/draw mirror focus`=⟨*style*⟩ (style, no default)

The style `draw mirror focus` marks the focal point of the mirror (with the style ⟨*style*⟩ if specified). See `draw mirror center` for an example.

3.2.6 Polarizer

Shape `polarizer`

Draws a polarizer



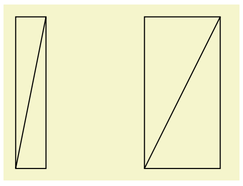
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[polarizer] (S) at (0,0) {};
\end{tikzpicture}
```

/tikz/optics/object height=⟨*length*⟩ (no default, initially 2cm)

The key `object height` applies.

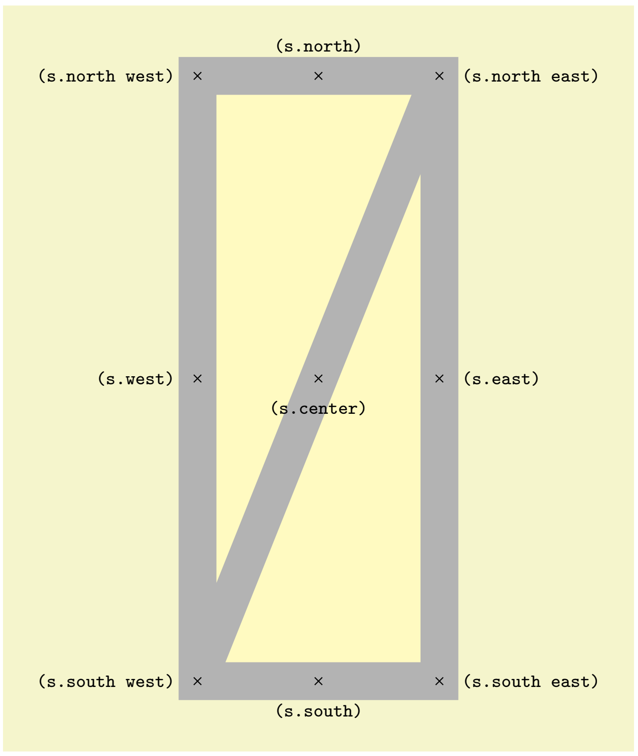
/tikz/optics/object aspect ratio=⟨*number*⟩ (no default, initially 0.2)

The key `object aspect ratio` determines the aspect ratio of the polarizer. When ⟨*number*⟩ is 1, the with of the polarizer is equal to its height. When ⟨*number*⟩=1/2, the with of the polarizer is equal to half its height.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[polarizer, object aspect ratio=0.2] (L1) at (0,0) {};
  \node[polarizer, object aspect ratio=0.5] (L2) at (4cm,0) {};
\end{tikzpicture}
```

The following figure summarizes the anchors defined by `polarizer`.

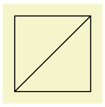


```
\Huge
\begin{tikzpicture}[use optics]
\node[polarizer,name=s,object height=8cm,object aspect ratio=0.4,shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,east/right,west/left,center/below,
north east/right,north west/left,south east/right,south west/left}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.7 Beam splitter

/tikz/optics/beam splitter (style, no value)

Draws a beam splitter. Keys and anchors are the same as polarizer.

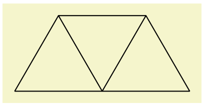


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[beam splitter] at (0,0) {};
\end{tikzpicture}
```

3.2.8 Double Amici prism

Shape double amici prism

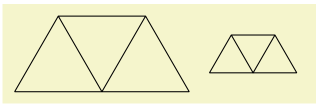
Draws a double Amici prism.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[double amici prism] (PVD) at (0,0) {};
\end{tikzpicture}
```

/tikz/optics/prism height=⟨*length*⟩ (no default, initially 1.5cm)

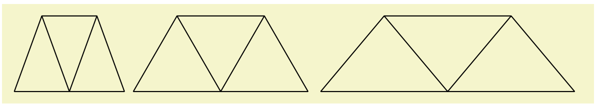
The key prism height determines the height of the three identical prisms that form the double Amici prism (the length of a side is therefore $2/\sqrt{3}$ times this height).



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[double amici prism, prism height=1cm] (PVD1) at (0,0) {};
  \node[double amici prism, prism height=0.5cm] (PVD2) at (4cm,0) {};
\end{tikzpicture}
```

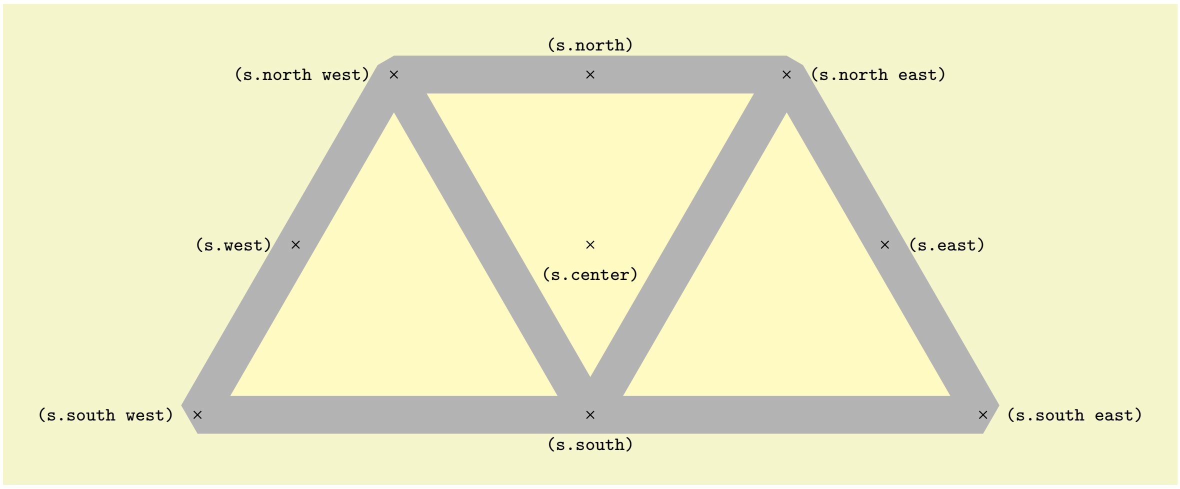
/tikz/optics/prism apex angle=⟨*angle*⟩ (no default, initially 60)

The key prism apex angle determines the apex angle of the three identical prisms. ⟨*angle*⟩ is expressed in degrees. When ⟨*angle*⟩ is 60, the prisms are equilateral triangles.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[double amici prism, prism apex angle=40] (PVD1) at (0,0) {};
  \node[double amici prism, prism apex angle=60] (PVD2) at (4cm,0) {};
  \node[double amici prism, prism apex angle=80] (PVD3) at (10cm,0) {};
\end{tikzpicture}
```

The following figure summarizes the anchors defined by double amici prism.

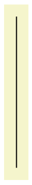


```
\Huge
\begin{tikzpicture}[use optics]
\node[double amici prism,name=s,prism height=4.5cm,shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,east/right,west/left,center/below,
north east/right,north west/left,south east/right,south west/left}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.9 Thin generic element

Shape `thin optics element`

Draws a generic optical element (used to build more concrete ones). This shape can be useful to draw an element not present in this library, by adding some relevant style.

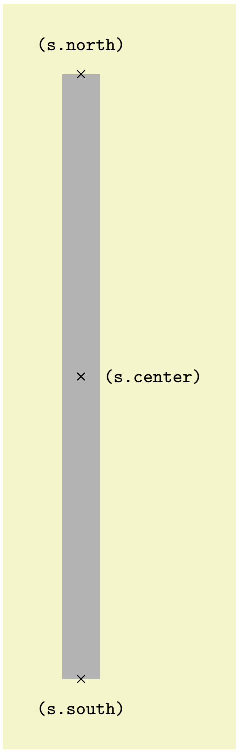


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[thin optics element] (S) at (0,0) {};
\end{tikzpicture}
```

`/tikz/optics/object height`=⟨*length*⟩ (no default, initially 2cm)

The key `object height` applies.

The following figure summarizes the anchors defined by `thin optics element`.

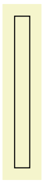


```
\Huge
\begin{tikzpicture}[use optics]
\node[thin optics element,name=s,object height=8cm,line shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,center/right}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.10 Thick generic element

Shape `thick optics element`

Draws a thick generic optical element.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[thick optics element] at (0,0) {};
\end{tikzpicture}
```

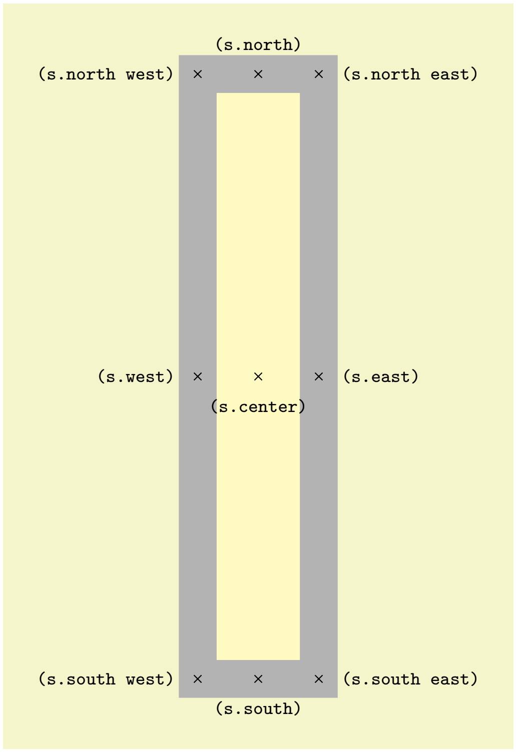
/tikz/optics/object height=⟨*length*⟩ (no default, initially 2cm)

The key `object height` applies.

/tikz/optics/object aspect ratio=⟨*number or length*⟩ (no default, initially 0.05)

The key `object aspect ratio` determines the aspect ratio of the object.

The following figure summarizes the anchors defined by `thick optics element`.

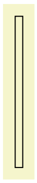


```
\Huge
\begin{tikzpicture}[use optics]
\node[thick optics element,name=s,object height=8cm,shape example,object aspect ratio=0.2] {};
\foreach \anchor/\placement in
{north/above,south/below,east/right,west/left,center/below,
north east/right,north west/left,south east/right,south west/left}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.2.11 Heat filter

/tikz/optics/heat filter (style, no value)

The style `heat filter` draws a heat filter. The options and anchors of `thick optics element` apply.

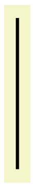


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[heat filter] (S) at (0,0) {};
\end{tikzpicture}
```

3.2.12 Screen

/tikz/optics/screen (style, no value)

The style screen draws a screen. The options and anchors of thin optics element apply.

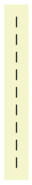


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[screen] (S) at (0,0) {};
\end{tikzpicture}
```

3.2.13 Diffraction grating

/tikz/optics/diffraction grating (style, no value)

The style diffraction grating draws a diffraction grating. The options and anchors of thin optics element apply.

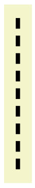


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[diffraction grating] (S) at (0,0) {};
\end{tikzpicture}
```

3.2.14 Grid

/tikz/optics/grid (style, no value)

The style grid draws a grid. The options and anchors of thin optics element apply.

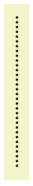


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[grid] (S) at (0,0) {};
\end{tikzpicture}
```

3.2.15 Semi-transparent mirror

/tikz/optics/semi-transparent mirror (style, no value)

The style semi-transparent mirror draws a semi-transparent mirror. The options and anchors of thin optics element apply.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[semi-transparent mirror] (S) at (0,0) {};
\end{tikzpicture}
```

3.2.16 Diaphragm

/tikz/optics/diaphragm (style, no value)

The style diaphragm draws a diaphragm (a slit element with a large slit). The options and anchors of slit apply.

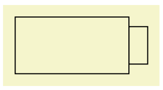
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[diaphragm] (S) at (0,0) {};
\end{tikzpicture}
```

3.3 Light sources and receptors

3.3.1 Generic optical input/output

Shape `generic optics io`

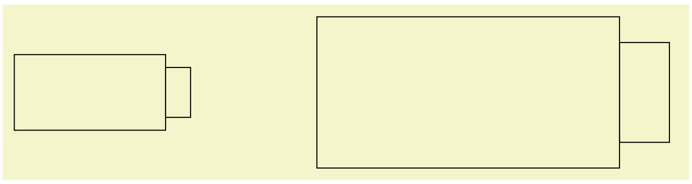
Draws a generic input/output system. We will use this to build more concrete elements.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io] (S) at (0,0) {};
\end{tikzpicture}
```

`/tikz/optics/io body height`=⟨*length*⟩ (no default, initially `0.75cm`)

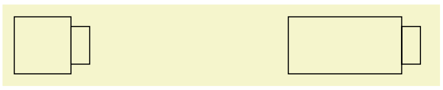
The key `io body height` determines the size of the light source or receptor. More precisely, ⟨*length*⟩ determines the height of the element body, while the other lengths are specified relatively to this height. By changing ⟨*length*⟩, the light source/receptor is then resized without deformation.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io, io body height=1cm] (L1) at (0,0) {};
  \node[generic optics io, io body height=2cm] (L2) at (10cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/io body aspect ratio`=⟨*number or length*⟩ (no default, initially 2)

The key `io body aspect ratio` determines the aspect ratio of the light source/receptor. When ⟨*number*⟩ is `1`, the width of the light source/receptor is equal to its height. When ⟨*number*⟩=`1/2`, the width of the light source/receptor is equal to half its height.



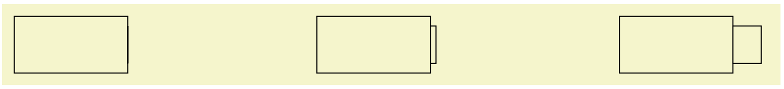
```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io, io body aspect ratio=1] (L1) at (0,0) {};
  \node[generic optics io, io body aspect ratio=2] (L2) at (8cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/io body width`=⟨*number or length*⟩ (style, no default)

Alias for `io body aspect ratio`.

`/tikz/optics/io aperture width`=⟨*number or length*⟩ (no default, initially `0.33`)

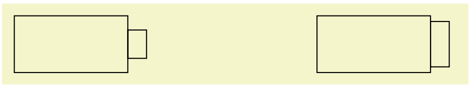
The key `io aperture width` determines the height of the aperture of the element (e.g. a condenser, a collimating lens, etc.), relative to the body height. When ⟨*number*⟩=`0`, the aperture is not drawn.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io, io aperture width=0] at (0,0) {};
  \node[generic optics io, io aperture width=0.1] at (8cm,0) {};
  \node[generic optics io, io aperture width=0.5] at (16cm,0) {};
\end{tikzpicture}
```

/tikz/optics/io aperture height=⟨*number or length*⟩ (no default, initially 0.66)

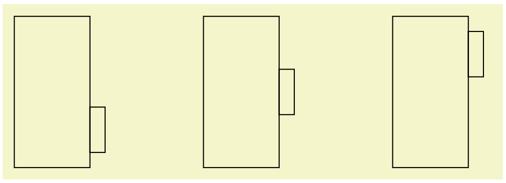
The key `io aperture width` determines the height of the aperture relative to the body height.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io, io aperture height=0.5] at (0,0) {};
  \node[generic optics io, io aperture height=0.8] at (8cm,0) {};
\end{tikzpicture}
```

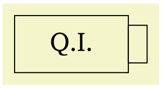
/tikz/optics/io aperture shift=⟨*number or length*⟩ (no default, initially 0)

The key `io aperture shift` determines how much the aperture should be shifted from the center of the body (again, the length is relative to the height of the body).



```
\begin{tikzpicture}[use optics,scale=.5,
optics,io body height=2cm,io body aspect ratio=0.5,io aperture height=0.3, io aperture width=0.1]
\node[generic optics io,io aperture shift=-0.25] at (-5cm,0) {};
  \node[generic optics io,io aperture shift=0] at (0,0) {};
  \node[generic optics io,io aperture shift=0.25] at (5cm,0) {};
\end{tikzpicture}
```

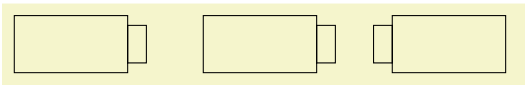
Text can be drawn inside the body of a lamp or receptor through the `node` text.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io] at (-5cm,0) {Q.I.};
\end{tikzpicture}
```

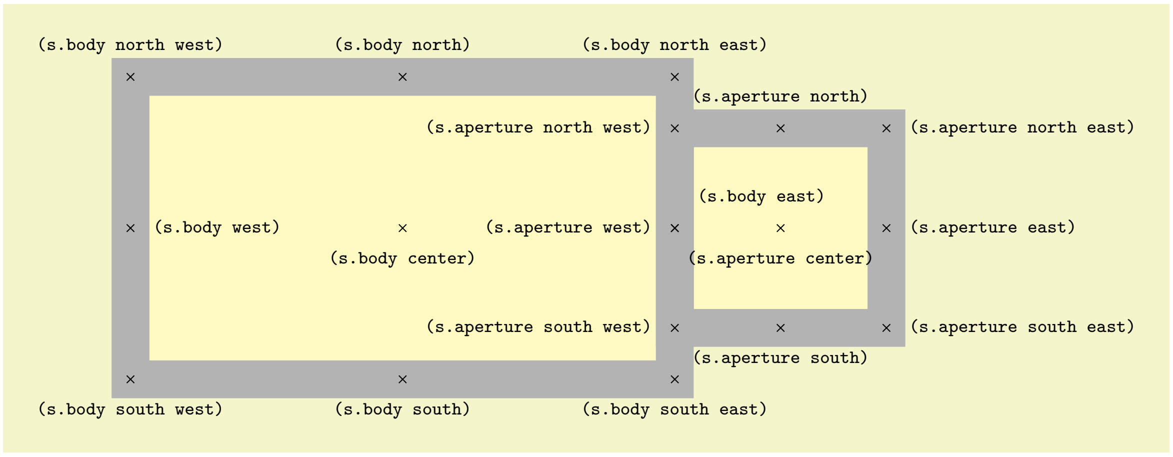
/tikz/optics/io orientation=⟨*type*⟩ (no default, initially ltr)

The key `io orientation` determines the direction in which the object should be drawn (the aperture is on the right [at the `east` anchor] when it is set to `ltr`, and it is on the left [at the `west` anchor] when it is set to `rtl`). The Les noms correspondent à et . The allowed values of this key are `ltr` (for « left to right ») and `rtl` (for « right to left »). Unlike `rotate`, the key `io orientation` modifies the anchors of the shape.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic optics io] at (0,0) {};
  \node[generic optics io,io orientation=ltr] at (5cm,0) {};
  \node[generic optics io,io orientation=rtl] at (10cm,0) {};
\end{tikzpicture}
```

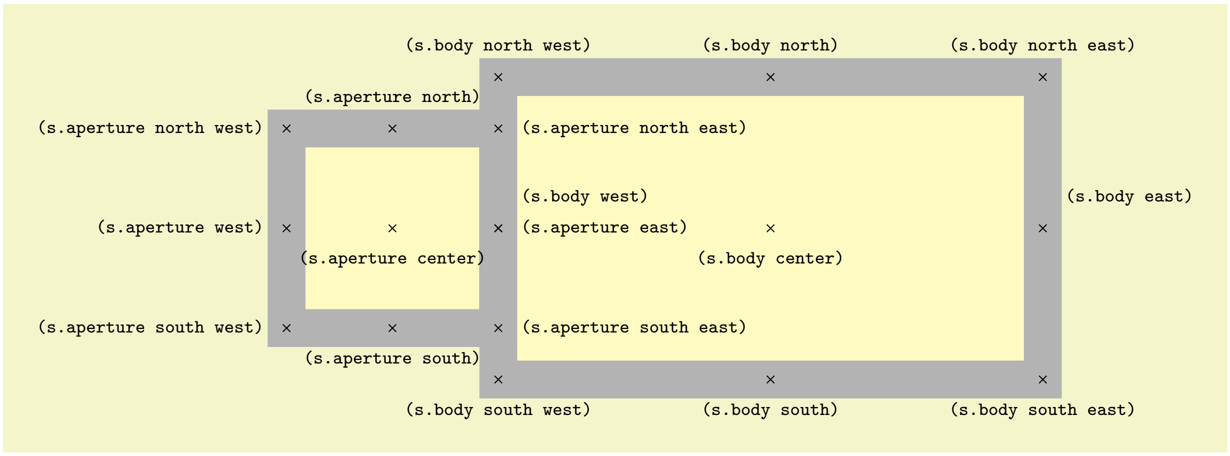
The following figure summarizes the anchors defined by generic optics io.



```
\Huge
\begin{tikzpicture}[use optics]
\node[generic optics io,name=s,io body height=4cm,io aperture width=0.7,io body aspect ratio=1.8,shape
example] {};
\foreach \anchor/\placement in
{body north/above,body south/below,body east/above right,body west/right,body center/below,
body north east/above,body north west/above,body south east/below,body south west/below,
aperture north/above,aperture south/below,aperture east/right,aperture west/left,aperture center/below,
aperture north east/right,aperture north west/left,aperture south east/right,aperture south west/left}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

The anchors body east and aperture west correspond to the same point. Both are defined for consistency. The anchor east is defined either as body east or aperture east depending on the value of io orientation so that east is the easternmost (rightmost) anchor. The same thing is done for the anchor west.

The following figure summarizes the anchors defined by generic optics io, io orientation=rtl.

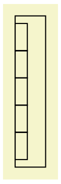


```
\Huge
\begin{tikzpicture}[use optics]
\node[generic optics io,name=s,io body height=4cm,io aperture width=0.7,io body aspect ratio=1.8,
io orientation=rtl,shape example] {};
\foreach \anchor/\placement in
{body north/above,body east/above right,body west/above right,body center/below,
body north east/above,body north west/above,body south east/below,body south west/below,
aperture north/above,aperture south/below,aperture east/right,aperture west/left,aperture center/below,
aperture north east/right,aperture north west/left,aperture south east/right,aperture south west/left}
\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};
\end{tikzpicture}
```

3.3.2 Sensor line

Shape `sensor line`

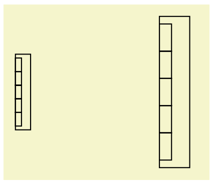
Draws a sensor line (such as a CCD or CMOS sensor).



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[sensor line] (S) at (0,0) {};
\end{tikzpicture}
```

`/tikz/optics/sensor line height`=⟨*length*⟩ (no default, initially `2cm`)

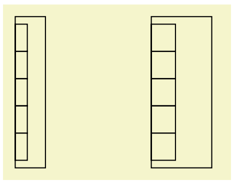
The key `sensor line height` determines the size of the sensor line. More precisely, ⟨*length*⟩ determines the height of the sensor line body and the other lengths can be specified relative to this one.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[sensor line, sensor line height=1cm] (L1) at (0,0) {};
  \node[sensor line, sensor line height=2cm] (L2) at (4cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/sensor line aspect ratio`=⟨*number*⟩ (no default, initially `0.2`)

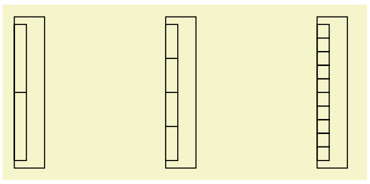
The key `sensor line aspect ratio` determines the aspect ratio of the sensor line.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[sensor line, sensor line aspect ratio=0.2] (L1) at (0,0) {};
  \node[sensor line, sensor line aspect ratio=0.4] (L2) at (4cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/sensor line pixel number`=⟨*number*⟩ (no default, initially `5`)

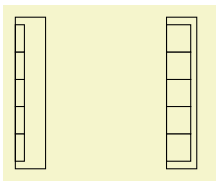
The key `sensor line pixel number` determines the number of pixels ⟨*number*⟩ of the sensor line.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[sensor line, sensor line pixel number=2] at (0,0) {};
  \node[sensor line, sensor line pixel number=4] at (4cm,0) {};
  \node[sensor line, sensor line pixel number=10] at (8cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/sensor line pixel width`=⟨*number or length*⟩ (no default, initially `0.4`)

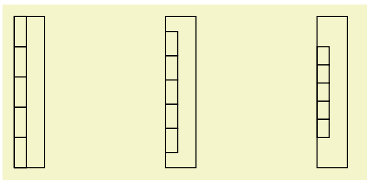
The key `sensor line pixel width` determines with width of each pixel (relative to the sensor width if it is a ⟨*number*⟩).



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[sensor line, sensor line pixel width=0.3] at (0,0) {};
  \node[sensor line, sensor line pixel width=0.8] at (4cm,0) {};
\end{tikzpicture}
```

`/tikz/optics/sensor line inner ysep`=⟨*number or length*⟩ (no default, initially `0.05`)

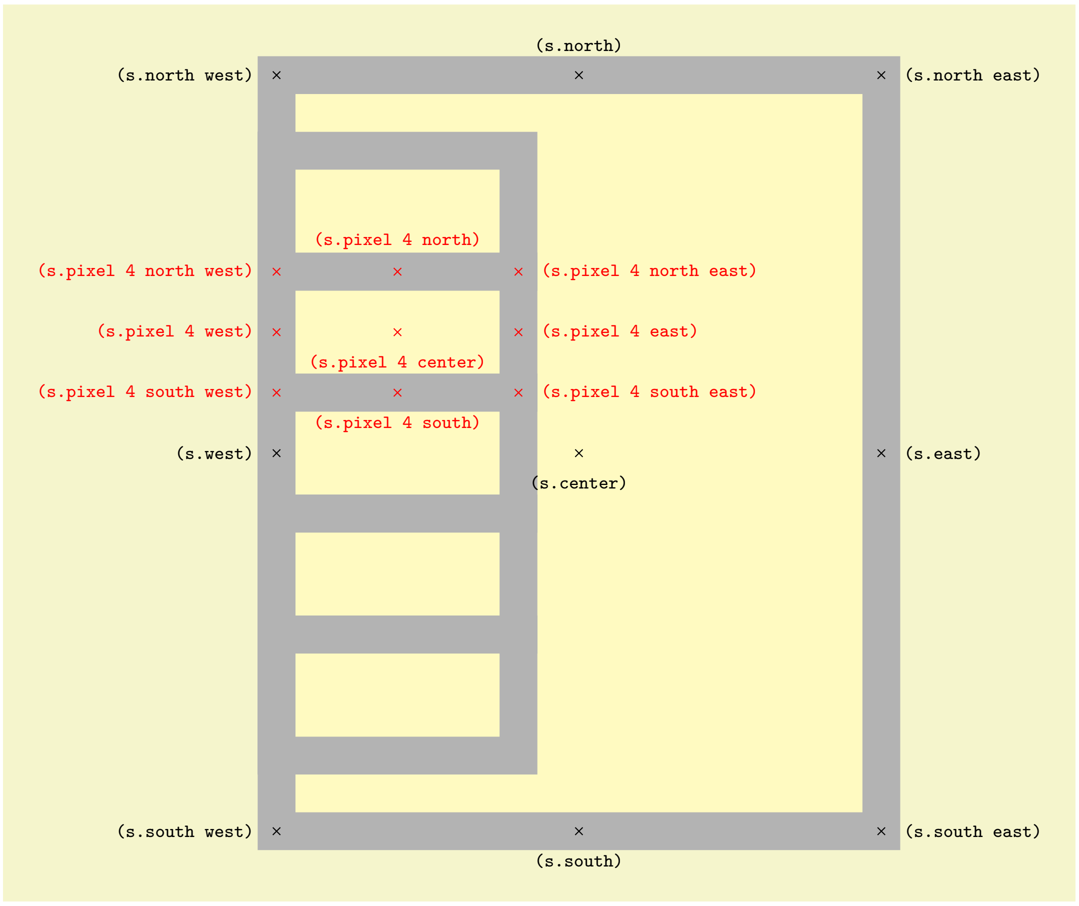
The key `sensor line inner ysep` determines the separation between the pixels and the top and bottom of the sensor line. When ⟨*number or length*⟩ is a relative lengths, it is relative to the height of thr sensor line. The height of each pixel is such that there are `sensor line pixel number` of them, with this separation taken into account.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[sensor line,sensor line inner ysep=0] at (0,0) {};
  \node[sensor line,sensor line inner ysep=0.1] at (4cm,0) {};
  \node[sensor line,sensor line inner ysep=0.2] at (8cm,0) {};
\end{tikzpicture}
```

The anchors `north`, `south`, `east`, `west`, `center`, `north east`, `north west`, `south east`, `south west`, as well as `pixel <i> <anchor>` where `<anchor>` is `north`, `south`, etc. and where `<i>` is the number of the pixel (going from 1 to `sensor line pixel number`) are defined. For instance, one can use `pixel 3 west`.

The following figure summarizes the anchors defined by `sensor line` globally as well as the anchors for the 4th pixel (in rouge).



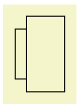
```
\Huge
\begin{tikzpicture}[use optics]
\node[sensor line,name=s,sensor line height=10cm,sensor line aspect ratio=0.8,
sensor line inner ysep=0.1,sensor line pixel number=5, shape example] {};
\foreach \anchor/\placement in
{north/above,south/below,east/right,west/left,center/below,
north east/right,north west/left,south east/right,south west/left}

\draw[shift=(s.\anchor)] plot[mark=x] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};

\foreach \anchor/\placement in
{pixel 4 north/above,pixel 4 south/below,pixel 4 east/right,pixel 4 west/left,pixel 4 center/below,
pixel 4 north east/right,pixel 4 north west/left,pixel 4 south east/right,pixel 4 south west/left}
% manque pixel <i> <subanchor>
\draw[red,shift=(s.\anchor)] plot[mark=x,red] coordinates{(0,0)}
node[\placement] {\scriptsize\texttt{(s.\anchor)}};

\end{tikzpicture}
```

Note : when `east` and `west` are not explicitly defined, they are aliases for `center`. This allows `right=of`... and similar keys to work as expected.

3.3.3 Generic sensor

`/tikz/optics/generic sensor` (style, no value)

The style `generic sensor` draws a sensor. The options of `generic optics io` apply.

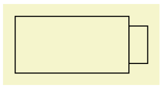


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic sensor] (S) at (0,0) {};
\end{tikzpicture}
```

3.3.4 Generic light source

`/tikz/optics/generic lamp` (style, no value)

The style `generic lamp` draws a light source. The options of `generic optics io` apply.

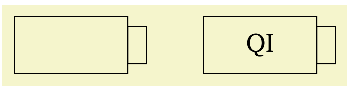


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[generic lamp] (S) at (0,0) {};
\end{tikzpicture}
```

3.3.5 Halogen lamp

`/tikz/optics/halogen lamp` (style, no value)

The style `halogen lamp` draws a halogen lamp. The options of `generic optics io` apply.

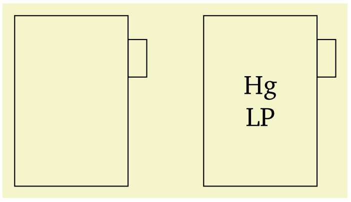


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[halogen lamp] (S) at (0,0) {};
  \node[halogen lamp] (S) at (5cm,0) {QI};
\end{tikzpicture}
```

3.3.6 Spectral lamp

`/tikz/optics/spectral lamp` (style, no value)

The style `spectral lamp` draws a spectral lamp. The options of `generic optics io` apply. In addition, a style is automatically applied to allow multiline text.

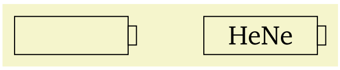


```
\begin{tikzpicture}[use optics,scale=.5]
  \node[spectral lamp] (S) at (0,0) {};
  \node[spectral lamp] (S) at (5cm,0) {\ce{Hg} \\ LP};
\end{tikzpicture}
```

3.3.7 Laser

/tikz/optics/laser (style, no value)

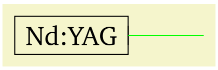
The style laser draws a laser. The options of generic optics io apply.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[laser] (S) at (0,0) {};
  \node[laser] (S) at (5cm,0) {\ce{HeNe}};
\end{tikzpicture}
```

/tikz/optics/laser' (style, no value)

The style laser' draws a laser without output aperture (i.e. with io aperture width=0pt). The options of generic optics io apply.



```
\begin{tikzpicture}[use optics,scale=.5]
  \node[laser'] (S) at (0,0) {\ce{Nd}:YAG};
  \draw[green] (S.aperture east) -- +(2cm,0);
\end{tikzpicture}
```

3.4 Utilities

3.4.1 Drawing rays

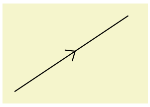
⚠ 2014-12-07 : this part was substantially modified

⚠ This part is not fully stable. Shortcuts such as ->-, should in principle not change.

In optics, it is often useful to mark the rays with arrows to distinguish them. The following styles use an arrow tip designed such that the arrow will be centered at the middle of the path (or at any specified position), instead of being a bit offset (in contrast with what would happen e.g. with put arrow and \arrow{>>}).

/tikz/optics/->- (style, no value)

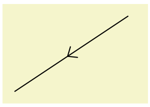
This style adds an arrow in the middle of the path.



```
\begin{tikzpicture}[use optics]
  \draw[->-] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

/tikz/optics/-<- (style, no value)

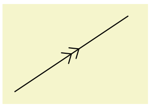
This style adds an arrow in the middle of the path, with a direction opposite of ->-.



```
\begin{tikzpicture}[use optics]
  \draw[-<-] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

/tikz/optics/->>- (style, no value)

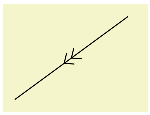
This style adds a double arrow in the middle of the path.



```
\begin{tikzpicture}[use optics]
  \draw[->>-] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

/tikz/optics/-<<- (style, no value)

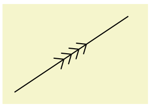
This style adds a double arrow in the middle of the path, with a direction opposite of ->>-.



```
\begin{tikzpicture}[use optics]
  \draw[-<<-] (0,0) -- (1.5cm,1.1cm);
\end{tikzpicture}
```

/tikz/optics/->n-={n=⟨*num*⟩, ⟨*specs*⟩} (style, no default)

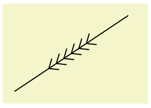
This style adds ⟨*num*⟩ arrows in the middle of the path. The specifications ⟨*specs*⟩ are applied to the arrows following put arrow syntax.



```
\begin{tikzpicture}[use optics]
  \draw[->n-={n=4}] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

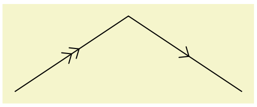
/tikz/optics/-<n-=n=⟨*num*⟩, ⟨*specs*⟩ (style, no default)

This style adds ⟨*num*⟩ arrows in the middle of the path, with a direction opposite of ->n-.



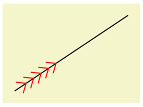
```
\begin{tikzpicture}[use optics]
  \draw[-<n-={n=6}] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

Shortcuts from ->- to ->>>>- (similar for -<-) are available. The styles ->-, -<-, etc. use put arrow with the arrow multiple ray arrow. They can be configured with the syntax of put arrow, for instance



```
\begin{tikzpicture}[use optics]
  \draw[->>-={at=0.25}, ->-={at=0.75}] (0,0) -- (1.5cm,1cm) -- (3cm, 0);
\end{tikzpicture}
```

This also works with ->n- and -<n-.



```
\begin{tikzpicture}[use optics]
  \draw[->n-={n=5, at=0.2, style=red}] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

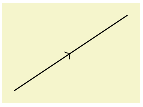
3.4.2 Put things on paths

⚠ 2014-12-07 : this part was substantially modified

⚠ This part is not stable yet. Everything can entirely change without any warning.

/tikz/put arrow (no value)

The key put arrow allows to easily add an arrow on a path.



```
\begin{tikzpicture}[use optics]
  \draw[put arrow] (0,0) -- (1.5cm,1cm);
\end{tikzpicture}
```

By default, the arrow is \arrow{>} and it is added in the middle of the path.

/tikz/put arrow/pos=⟨*pos*⟩ (no default, initially 0.5)

Sets the position of the arrow on the path to ⟨*pos*⟩ (for instance ⟨*pos*⟩=0.5 means that that arrow is in the middle of the path).

/tikz/put arrow/at (no value)

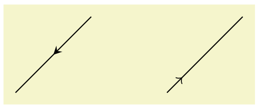
Alias for /tikz/put arrow/pos.

/tikz/put arrow/arrow=⟨*arrow specification*⟩ (no default)

Uses the arrow defined by ⟨*arrow specification*⟩ (for instance stealth of latex).

/tikz/put arrow/arrow'=⟨*arrow specification*⟩ (no default)

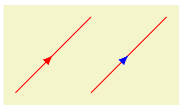
Uses the arrow defined by ⟨*arrow specification*⟩ (for instance stealth or latex), but reversed.



```
\begin{tikzpicture}[use optics]
  \draw[put arrow={arrow'=stealth}] (0,0) -- (1cm,1cm);
  \draw[put arrow={at=0.2}] (2cm,0) -- (3cm,1cm);
\end{tikzpicture}
```

/tikz/put arrow/style=⟨*style*⟩ (no default)

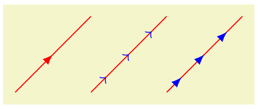
The \meta{style} is passed to \arrow[⟨*style*⟩] to draw the arrows.



```
\begin{tikzpicture}[use optics]
  \draw[red,put arrow={arrow=latex}]
  (0,0) -- (1cm,1cm);
  \draw[red,put arrow={arrow=latex,style={blue}}]
  (1cm,0) -- (2cm,1cm);
\end{tikzpicture}
```

/tikz/put arrow/every arrow (style, no value)

This style is passed to all arrows drawn by put arrow (in the scope of the style). Use every arrow/.style={⟨*style*⟩} (or append style, etc.).



```
\begin{tikzpicture}[use optics]
  \draw[red,put arrow={arrow=latex}]
  (0,0) -- (1cm,1cm);
  \draw[red, put arrow/every arrow/.style={blue},
    put arrow={at=0.2}, put arrow={at=0.5}, put arrow={at=0.8}]
  (1cm,0) -- (2cm,1cm);
  \draw[red, >=latex, put arrow/every arrow/.style={blue},
    put arrow={at=0.2}, put arrow={at=0.5}, put arrow={at=0.8}]
  (2cm,0) -- (3cm,1cm);
\end{tikzpicture}
```

The arrows that can be used are discussed in the tikz manual.

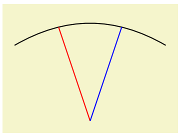
For a finer control of what happens, use the tikz library markings instead.

Specific styles ->-, -<-, ->>- et -<<- allow to label light rays.

3.4.3 Add coordinates on paths

/tikz/put coordinate=⟨*coordinate*⟩ at ⟨*position*⟩ (no default)

The style put coordinate creates a coordinate with name ⟨*coordinate*⟩ at the relative position ⟨*position*⟩ on the path to which the style is applied.

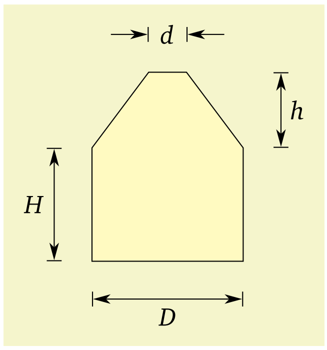


```
\begin{tikzpicture}[use optics]
  \draw[put coordinate=P at 0.3,put coordinate=Q at 0.7] (0,0)
    to[bend left] (2cm,0);
  \draw[red] (P) -- (1cm,-1cm);
  \draw[blue] (Q) -- (1cm,-1cm);
\end{tikzpicture}
```

3.4.4 Label dimensions and distances on figures

⚠ This part is experimental and might change a lot without warning.

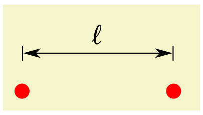
Let us start with an example.



```
\begin{tikzpicture}
\draw[fill=yellow!30]
   (-1cm,0) coordinate (A) -- (1cm,0) coordinate (B)
   -- (1cm,1.5cm) coordinate (B') -- (0.25cm,2.5cm) coordinate (b)
   -- (-0.25cm,2.5cm) coordinate (a) -- (-1cm,1.5cm) coordinate (A')
   -- cycle;

\draw (A) to[dim arrow'={label'=$D$}] (B);
\draw (A) to[dim arrow={label=$H$}] (A');
\draw (a) to[short dim arrow={label=$d$,label near middle}] (b);
\draw (B') to[dim arrow'={label'=$h$}] (b -| B');
\end{tikzpicture}
```

`/tikz/dim arrow`=⟨*sous-clés*⟩ (style, no default)

The style `dim arrow` allows to label distances. It applies to a `to path`. For instance,



```
\begin{tikzpicture}[use optics]
  \node[circle,fill=red,inner sep=2pt] (a) at (0,0) {};
  \node[circle,fill=red,inner sep=2pt] (b) at (2cm,0) {};
  \draw (a.center) to[dim arrow={label=$\ell$}] (b.center);
\end{tikzpicture}
```

The dimension arrow is shifted with respect to the initial and final position in order not to overlap with the labeled object.

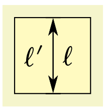
Several subkeys can be used through `dim arrow`={⟨*subkeys*⟩}.

`/tikz/dim arrow/label`=⟨*text*⟩ (no default)

The key `label` determiens the ⟨*text*⟩ to show on the arrow. It is drawn on the left of the arrow with respect to the path orientation (via `/tikz/auto=left`).

`/tikz/dim arrow/label'`=⟨*text*⟩ (no default)

The key `label'` does the same thing as `label`, with ⟨*text*⟩ drawn on the right.



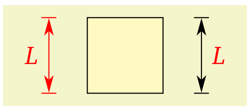
```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=1cm] (R) at (0,0) {};

  \draw (R.north east) to[dim arrow'={label'=$\ell'$}] (R.south east);
  \draw (R.north east) to[dim arrow'={label=$\ell$}] (R.south east);
\end{tikzpicture}
```

`/tikz/dim arrow/label text`=⟨*text*⟩ (no default)

The key `label text` determines the ⟨*text*⟩ to add on the arrow without modifying its position.

`/tikz/dim arrow/label style` (style, no value)

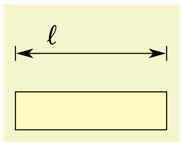
The key `label style` determines the style to use to draw the label. This style should not be replaced (expect on purpose) as it is used to position the label. Instead, styling should be appended with `/.append style`.



```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=1cm] (R) at (0,0) {};

  \draw (R.north east)
    to[dim arrow={label=$L$, label style/.append style=red}] (R.south east);
  \draw (R.north west)
    to[dim arrow'={label'=$L$}, red] (R.south west);
\end{tikzpicture}
```

`/tikz/dim arrow/label pos`=⟨*number*⟩ (no default, initially `0.5`)

The key `label pos` determines the position ⟨*number*⟩ where the label should be drawn. This key does not apply to `short dim arrow`.



```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=2cm, minimum
height=0.5cm] (R) at (0,0) {};

  \draw (R.north west)
    to[dim arrow={label=$\ell$, label pos=0.25}] (R.north east);
\end{tikzpicture}
```

/tikz/dim arrow/label near start (no value)

Equivalent to `label pos`=0 (for `short dim arrow`, see the specific documentation below).

/tikz/dim arrow/label near middle (no value)

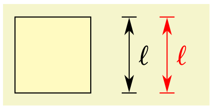
Equivalent to `label pos`=0.5 (for `short dim arrow`, see the specific documentation below).

/tikz/dim arrow/label near end (no value)

Equivalent to `label pos`=1 (for `short dim arrow`, see the specific documentation below).

/tikz/dim arrow/raise=⟨*length*⟩ (no default, initially `0.5cm`)

The key `raise` determines the distance ⟨*length*⟩ between the dimension arrow and the initial path.



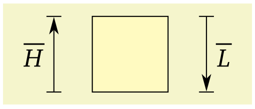
```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=1cm] (R) at (0,0) {};

  \draw (R.north east)
    to[dim arrow={label=$\ell$, raise=0.5cm}, black] (R.south east);
  \draw (R.north east)
    to[dim arrow={label=$\ell$, raise=1cm}, red] (R.south east);
\end{tikzpicture}
```

/tikz/dim arrow/no raise (style, no value)

Equivalent to `raise`=0.

/tikz/dim arrow/-> (no value)

When `->` is used, an arrow is drawn on one side only (selected in the same way as `->` in tikz). This can be used to label an algebraic length.



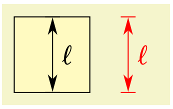
```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=1cm] (R) at (0,0) {};

  \draw (R.north east)
    to[dim arrow={->, label=$\overline{L}$, raise=0.5cm}] (R.south east);
  \draw (R.north west)
    to[dim arrow={<-, label'=$\overline{H}$, raise=-0.5cm}] (R.south west);
\end{tikzpicture}
```

/tikz/dim arrow/<- (no value)

Similar to `/tikz/dim arrow/->`, but in the other direction.

/tikz/dim arrow'=⟨*sous-clés*⟩ (style, no default)

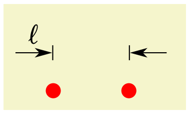
The style `dim arrow'` does the same thing as `dim arrow` with the initial value `/tikz/dim arrow/raise` set to `-0.5cm` instead of `0.5cm`.



```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=1cm] (R) at (0,0) {};

  \draw (R.north east) to[dim arrow'={label=$\ell$}, black] (R.south east);
  \draw (R.north east) to[dim arrow={label=$\ell$}, red] (R.south east);
\end{tikzpicture}
```

/tikz/short dim arrow=⟨*sous-clés*⟩ (style, no default)

The style short dim arrow has the same aim as dim arrow. It is used when the dimension to label is too small, so that the arrows should be on the outside.

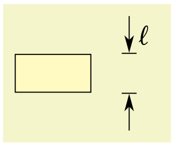


```
\begin{tikzpicture}[use optics]
  \node[circle,fill=red,inner sep=2pt] (a) at (0,0) {};
  \node[circle,fill=red,inner sep=2pt] (b) at (1cm,0) {};
  \draw (a.center)
  to[short dim arrow={label=$\ell$}]
  (b.center);
\end{tikzpicture}
```

The options of /tikz/dim arrow apply, with some differences and additional options discussed below.

/tikz/dim arrow/label near start (no value)

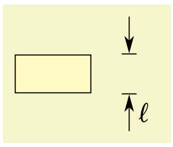
The key label near start positions the label near the beginning of the path. This is the default option.



```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=0.5cm] (R) at (0,0) {};

  \draw (R.north east) to[short dim arrow={label=$\ell$}] (R.south east);
\end{tikzpicture}
```

/tikz/dim arrow/label near end (no value)

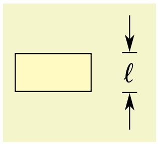
The key label near end positions the label near the end of the path.



```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=0.5cm] (R) at (0,0) {};

  \draw (R.north east)
      to[short dim arrow={label=$\ell$, label near end}] (R.south east);
\end{tikzpicture}
```

/tikz/dim arrow/label near middle (no value)

The key label near middle positions the label near the middle of the path. In this case, the label is not shifted with respect to the arrow (as there is no overlap). This can be restored using /tikz/dim arrow/label style.



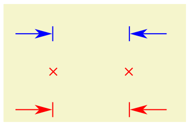
```
\begin{tikzpicture}
  \node[rectangle, draw=black, fill=yellow!30, minimum width=1cm, minimum
height=0.5cm] (R) at (0,0) {};

  \draw (R.north east)
      to[short dim arrow={label=$\ell$, label near middle}] (R.south east);
\end{tikzpicture}
```

/tikz/dim arrow/arrow length=⟨*length*⟩ (no default, initially 5mm)

The key arrow length determines the ⟨*length*⟩ of the external arrows.

/tikz/short dim arrow'=⟨*sous-clés*⟩ (style, no default)

The style short dim arrow' does the same thing as short dim arrow, with the initial value of /tikz/dim arrow/raise set to -0.5cm instead of 0.5cm.

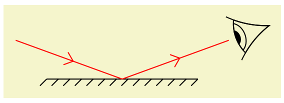


```
\begin{tikzpicture}[use optics]
  \coordinate (a) at (0,0);
  \coordinate (b) at (1cm,0);
  \draw[red,mark=x, draw=none] plot coordinates {(a) (b)};
  \draw (a.center) to[short dim arrow, blue] (b.center);
  \draw (a.center) to[short dim arrow', red] (b.center);
\end{tikzpicture}
```

3.4.5 Eyes

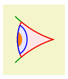
/tikz/pics/optics eye (style, no value)

The pic optics eye draws a stylized eye. Note that optics eye is not a shape, but a pic (see the tikz manual).



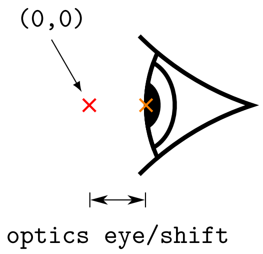
```
\begin{tikzpicture}[use optics]
  \node[mirror,rotate=-90] (M) at (0,0) {};

  \pgfmathsetmacro\rayangle{70}
  \draw[red,->-] ($(M.center)+({90+\rayangle}:1.5cm)$) -- (M.center);
  \draw[red,->-] (M.center) -- ($(M.center)+({90-\rayangle}:1.5cm)$)
  coordinate (out) {};
  \pic[rotate={90-\rayangle}] (eye) at (out) {optics eye};
\end{tikzpicture}
```

How the eye is drawn is controlled through optics eye={\meta{styles}}. In particular, the styles pupil, cornea, iris and contour can be modified (e.g. with append style) to change how the eye is drawn.



```
\begin{tikzpicture}[use optics]
\pic (colorful eye) {optics eye={
cornea/.append style={red, fill=red!10},
iris/.append style={blue, fill=blue!10},
pupil/.append style={orange},
contour/.append style={green!60!black},
}};
\end{tikzpicture}
```

When a optics eye pic is added at position (0,0), it is shifted by a quantity optics eye/shift that can be controlled through optics eye={shift=\meta{length}} so that the light does not collide with the eye. Two coordinates (\meta{name}-in) and (\meta{name}-center) are created; for instance, \pic (eye) at (0,0) {optics eye}; will produce the coordinates (eye-in) [at (0,0)] and (eye-center). They are separated by the length optics eye/shift and their positions are explained in the picture below.



4 Thanks

I thank the colleagues and friends who had the unfortunate privilege of testing the first versions of this library, and in particular the authors of the book *Physique expérimentale* [8] for which the library was developed.

8. http://www.physique-experimentale.com/