# TikZ + graphicx

## Coordinate translation from picture pixel to TikZ

Benjamin Kellermann

July 26, 2011

It happens often, that one wants to annotate a graphic or draw some lines inside a graphic. This package solves the problem, of finding the right coordinates for TikZ by translating the pixel coordinates of the picture into TikZ coordinates. We show this using GIMP, but other image editors may be used to. Additionally, the package offers the possibility to zoom parts of an image.

## 1 Preamble

```
\usepackage{tikzgraphicx}
```

## 2 Drawing in a Picture

Assume we want to include a picture in our document. This is done with a new environment `tikzgraphics`. This environment takes 4 parameters, the width of the picture, the x and y resolution and the filename:

```
\begin{tikzgraphics}[<tikzopts>]{<TeX-width>}
  {<x-pixel-size>}{<y-pixel-size>}{<filename>}
\end{tikzgraphics}
```

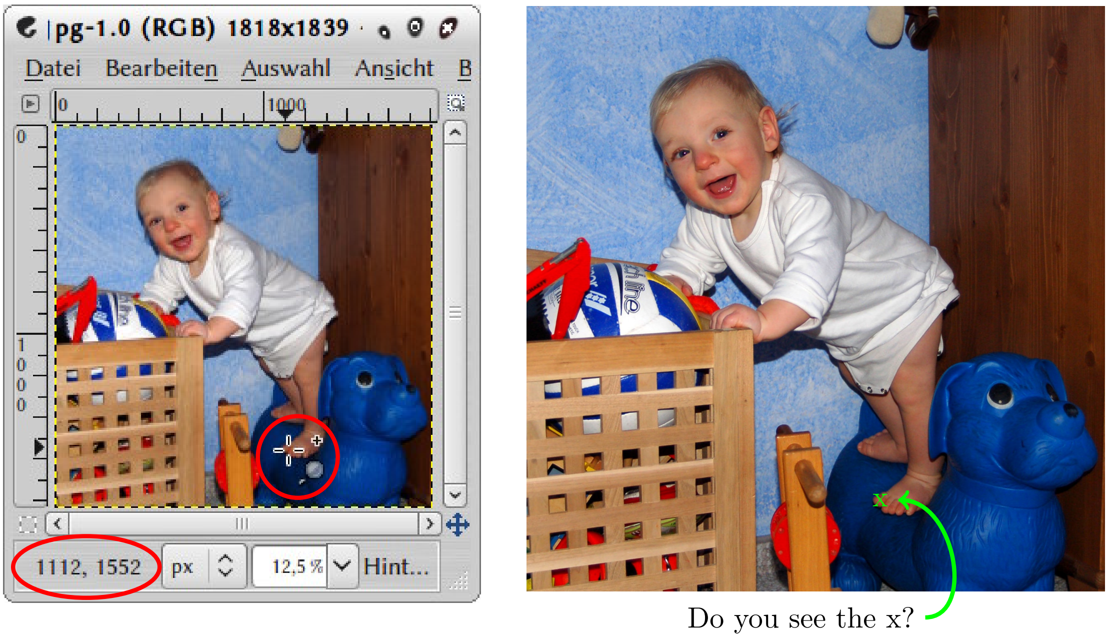
To include a picture one would write something like:

```
\begin{tikzgraphics}{.5\textwidth}{1818}{1839}{arthur}
\end{tikzgraphics}
```

which would produce the same as

```
\includegraphics[width=.5\textwidth]{arthur}
```

with the difference, that one can write TikZ code within the environment. Additionally to the TikZ commands, one can use special commands defined by tikzgraphicx.



```
\begin{tikzgraphics}[remember picture]{5cm}{1818}{1839}{arthur}
  \pxnode[color=green]{1112}{1552}{foot}{x}
\end{tikzgraphics}
```

Figure 1: Placing a TikZ node with GIMP. The code shown at the bottom is for the left picture.

## 2.1 Coordinates and Nodes

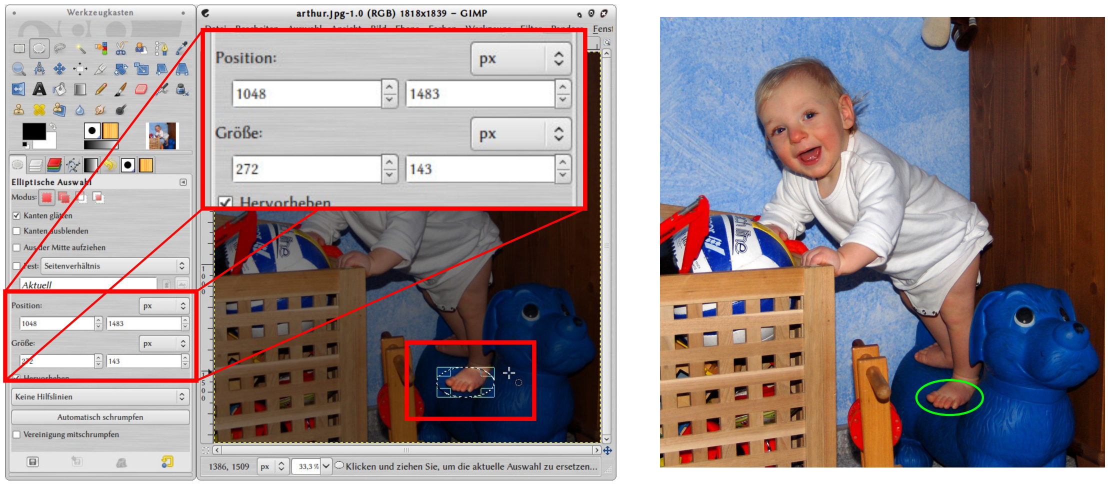
The first ones are \pxcoordinate and \pxnode, which are used to place TikZ coordinates and nodes. As they work similarly, with the only difference that the coordinate does not have a text argument, we will explain only the node usage. To determine the destination, GIMP is used. Open the picture in GIMP and place the cursor at the position you want to have an node as shown in Figure 1. You can read the x,y-coordinates which are given to \pxcoordinate and \pxnode at the bottom left.[1]

```
  \pxcoordinate[<opt>]{<x>}{<y>}{<name>}
  \pxnode[<opt>]{<x>}{<y>}{<name>}{<text>}
```

## 2.2 Ellipses and Rectangles

The next thing we want to do is painting an ellipse. For this, one paints the ellipse in GIMP and reads the coordinates given to \pxellipse. Equivalent to \pxellipse, one can

[1] Note, that GIMP maps the coordinate 0,0 to the upper left corner. This might be different if you use another graphic editor!



arthur.jpg-1.0 (RGB) 1818x1839 – GIMP

Position: 1048 1483 px

Größe: 272 143 px

☑ Hervorheben

```
\begin{tikzgraphics}[ultra thick]{5cm}{1818}{1839}{arthur}
  \pxellipse[color=green]{1048}{1483}{272}{143}
\end{tikzgraphics}
```

Figure 2: Drawing a TikZ ellipse with GIMP. The code shown below is the code for the right picture. The coordinates are taken from GIMP.

use \pxrectangle, which uses the same input variables, but draws a rectangle. The usage is:

```
\pxellipse[<opt>]{<x>}{<y>}{<+x>}{<+y>}
\pxrectangle[<opt>]{<x>}{<y>}{<+x>}{<+y>}
```

An example is shown in Figure 2.

## 3 Picture in Picture

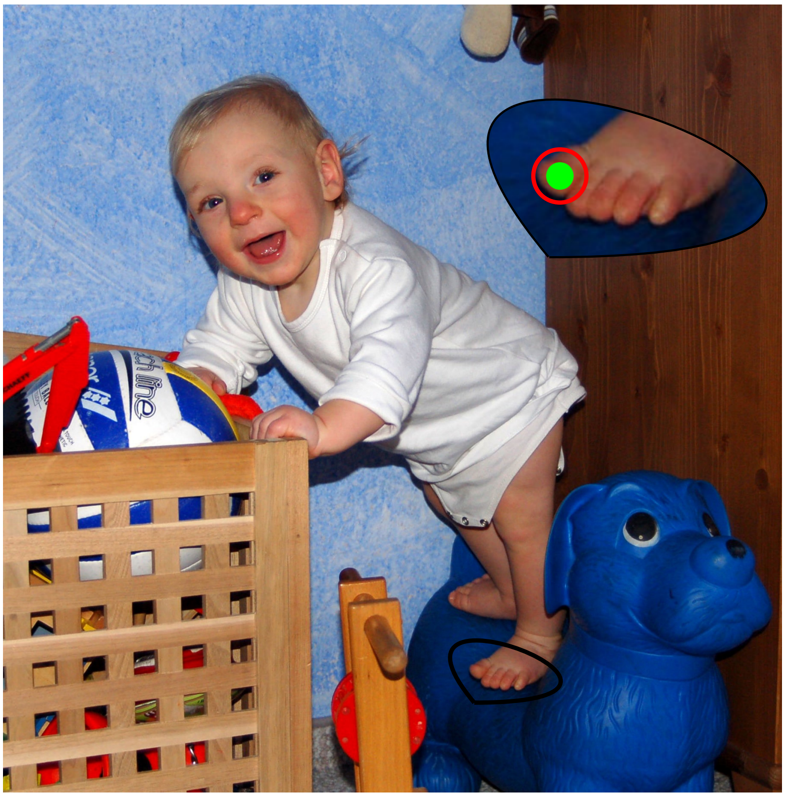
It occurs sometimes, that one wants to zoom parts of a picture. This is possible with the tikzzoomgraphics environment, which can be accessed inside a tikzgraphics environment. It expects 5 parameters. The first, is a factor, by which level the picture is zoomed. E.g., if the tikzgraphics environment was opened with 5 cm and 2.5 is given as zoomfactor, the resulting image would have a size of 12.5 cm. The zoomed image is placed over the original image. Two points have to be given, which are laid on top of each other. The second and third parameters are x,y coordinates of a point in the source image. The last two parameters are x,y coordinates of a point in the zoomed image. These are two points, which will be put on top of each other.

Inside the tikzzoomgraphics environment, one can use the already defined commands. Inside the inner graphic, one has to insert the zoomed picture with \insertpicture manually. This gives one the opportunity to draw a clip beforehand and paint other things afterwards. The usage is:

```
\begin{tikzgraphics}[<tikzopts>]{<TeX-width>}
    {<x-pixel-size>}{<y-pixel-size>}{<filename>}
  \begin{tikzzoomgraphics}
      [<tikzopts>]
      {<zoomfactor>}
      {<xpos_source_pic>}{<ypos_source_pic>}
      {<xpos_zoomed_pic>}{<ypos_zoomed_pic>}
    \path[clip] ...;
    \insertpicture
  \end{tikzzoomgraphics}
\end{tikzgraphics}
```

An example is given in Figure 3 on the following page. The following lines may be helpful if one wants to gray out all of the picture instead of the clipping path. These have to be inserted before the tikzzoomgraphics environment

```
\path[fill, fill opacity=.5,even odd rule]
  (current bounding box.north west) rectangle
  (current bounding box.south east)
  (a).. controls +(130:15) and +(0:20).. (a);
```



```
1  \begin{tikzgraphics}[ultra thick]{5cm}{1818}{1839}{arthur}
2    \pxcoordinate{1100}{1630}{a}
3    \draw (a).. controls +(130:15) and +(0:20)..(a);
4    \begin{tikzzoomgraphics}{2.5}{1300}{400}{1110}{1555}
5      \pxcoordinate{1100}{1630}{b}
6      \draw[clip] (b).. controls +(130:15) and +(0:20)..(b);
7      \insertpicture
8      \pxcoordinate[draw,circle,color=red,minimum width=20pt]
9        {1110}{1555}{}
10   \end{tikzzoomgraphics}
11   \pxcoordinate[circle,fill=green]{1300}{400}{}
12 \end{tikzgraphics}
```

Figure 3: Zooming a Picture. Remark, that inside and outside the tikzzoomgraphics environment, almost the same commands are written. The coordinates are calculated in the proper contexts. I.e., line 2 and 5 are the same (specifying the same point in both pictures). Line 3 and 6 draw the same path in both pictures. The pxcoordinates in line 8–9 and line 11 are the two coordinates, specified in line 4 and show, that the two points specified as parameter of tikzzoomgraphics environment lay on top of each other.