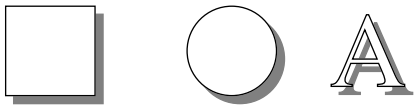
pst-blur package
version 2.0

Martin Giese*and Herbert Voss†

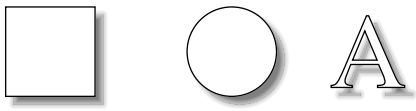
2005/09/08

1 Introduction

The ability to paint shadows on arbitrary shapes is a standard feature of PSTricks. However, these shadows are always 'hard':



The `pst-blur` package provides blurred shadows for closed shapes drawn with PSTricks:



It also provides a new box command `\psblurbox`, which is similar to `\psshadowbox`, but gives the box a blurred shadow.

The new graphics parameters and macros provided by the package are described in section 2 of this document. Section 3, if present, documents the implementation consisting of a generic TEX file and a PostScript header for the `dvi`-to-PostScript converter. You can get section 3 by calling LATEX as follows on most relevant systems:

```
latex '\AtBeginDocument{\AlsoImplementation}\input{pst-blur.dtx}'
```

*giese@ira.uka.de
†voss@pstricks.de

2 Package Usage

To use `pst-blur`, you have to say

```
\usepackage{pst-blur}
```

in the document prologue for LaTeX, and

```
\input pst-blur.tex
```

in "plain" TeX.

`blur` To paint shapes with blurred shadows, set the graphics parameters `shadow` and `blur` to `true`, eg

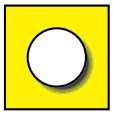
```
\psset{unit=1cm}
\pscircle[shadow=true,blur=true](0,0){0.5}
```

for a circle with a blurred shadow. The parameter `blur` has no influence if `shadow` is `false`.

`shadowsize` `shadowangle` `blurradius` The rendering of blurred shadows is controlled by a number of additional graphics parameters. The offset of the shadow is controlled by the parameters `shadowsize` and `shadowangle`, which are the same as for ordinary shadows.[1] The size of the blurring effect is controlled by the parameter `blurradius`, see Fig 1. The default value for `blurradius` is 1.5pt, which fits nicely with the default `shadowsize` of 3pt.

`shadowcolor` `blurbg` The inner, usually darkest part of the shadow is painted in the colour defined by `shadowcolor`. In the range defined by `blurradius`, the colour gradually fades to the background colour set by `blurbg`. The default value for `blurbg` is white. You should change this parameter when you want to paint shapes over a coloured background, ie

```
\psframe[fillstyle=solid,fillcolor=yellow](-.7,-.7)(.7,.7)
\pscircle[shadow=true,blur=true,blurbg=yellow](0,0){0.4}
```

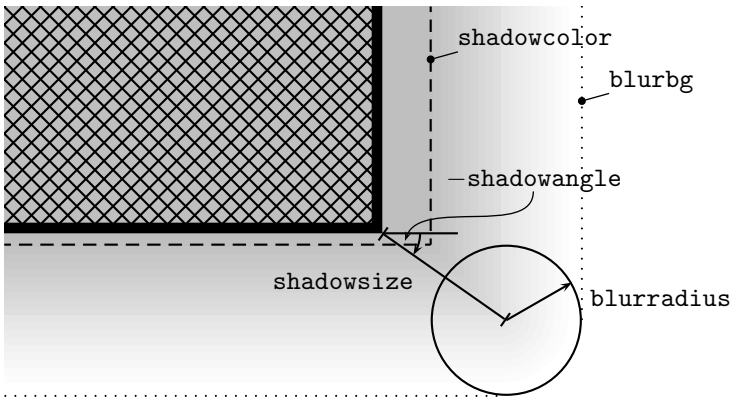


`blursteps` The number of distinct colour steps painted between `shadowcolor` and `blurbg` is controlled by the parameter `blursteps`. The default value for `blursteps` is 20, which is usually more than sufficient. Note, that higher values for `blursteps` result in proportionally slower rendering. This can be very tiresome with complex shapes.

`\psblurbox` Using a `\psframebox` with a blurred shadow in the middle of some text produces poor results, because TeX does not know about the extra space taken

[1] In particular, `shadowangle` has to be negative for the usual placement of shadows below and to the right of shapes.

Figure 1: Parameters for blurred shadows



by the shadow. For normal shadows, this problem is solved by the `\psshadowbox` macro, which adds the extra space around the box for the shadow. For blurred shadows, this is not sufficient: an extra `\blurradius` has to be added. This is done by the macro `\psblurbox`, which is otherwise identical to `\psshadowbox`. Note, that `\psblurbox` shares a deficiency of `\psshadowbox`: It only works correctly with `shadowangle` $= -45$, because TEX does not provide trigonometric operations.

3 The Code

3.1 The `pst-blur.sty` file

The `pst-blur.sty` file is very simple. It just loads the generic `pst-blur.tex` file.

```
⟨*stylefile⟩
\RequirePackage{pstricks}
\ProvidesPackage{pst-blur}[2005/09/08 package wrapper for
  pst-blur.tex (hv)]
\input{pst-blur.tex}
\ProvidesFile{pst-blur.tex}
  [\filedate\space v\fileversion\space 'PST-blur' (hv)]
⟨/stylefile⟩
```

3.2 The `pst-blur.tex` file

`pst-blur.tex` contains the TEX-side of things. We begin by identifying ourselves and setting things up, the same as in other PSTricks packages.

```
⟨*texfile⟩
\csname PstBlurLoaded\endcsname
\let\PstBlurLoaded\endinput
\ifx\PSTricksLoaded\endinput\else
  \def\next{\input pstricks.tex }\expandafter\next
\fi
```

`pst-blur` uses the extended version of the keyvalue interface.

```
\ifx\PSTXKeyLoaded\endinput\else\input pst-xkey \fi
```

```
\def\fileversion{2.0}
\def\filedate{2005/09/08}
\message{ v\fileversion, \filedate}
\edef\TheAtCode{\the\catcode`\@}
\catcode`\@=11
```

Add the package name to the list of family names of the keyvalue list.

```
\pst@addfams{pst-blur}
\pstheader{pst-blur.pro}
```

3.2.1 New graphics parameters

`blur`
`blurradius`
`blursteps`
`blurbg`

The definitions of the new graphics parameters follow the definitions for parameters of the same types found in `pstricks.tex`.

```
\newif\ifpsblur
\define@key[psset]{pst-blur}{blur}[true]{\@nameuse{psblur#1}\pst@setrepeatarrowsflag}
\psset{blur=false}
%%
\define@key[psset]{pst-blur}{blurradius}{\pst@getlength{#1}\psx@blurradius}
\psset{blurradius=1.5pt}
%%
\define@key[psset]{pst-blur}{blursteps}{\pst@getint{#1}\psx@blursteps}
\psset{blursteps=20}
%%
\define@key[psset]{pst-blur}{blurbg}{\pst@getcolor{#1}\psx@blurbg}
\psset{blurbg=white}
```

3.3 Hooking into the PSTricks shadow macros

`\pst@closedshadow`

The macro `\pst@closedshadow` is usually called internally by PSTricks to paint a shadow in the shape of the current path. This macro has been renamed `\pst@sharpclosedshadow`. The new `\pst@closedshadow` jumps to either of `\pst@sharpclosedshadow` or `\pst@blurclosedshadow`, depending on `\ifpsblur`, which is directly related to the graphics parameter `blur`.

```
\def\pst@closedshadow{%
\ifpsblur\pst@blurclosedshadow\else\pst@sharpclosedshadow\fi
}
\def\pst@sharpclosedshadow{%
  \addto@pscode{%
    gsave
```

```
    \psk@shadowsize \psk@shadowangle \tx@PtoC
    \tx@Shadow
    \pst@usecolor\psshadowcolor
    gsave fill grestore
    stroke
    grestore
    gsave
    \pst@usecolor\psfillcolor
    gsave fill grestore
    stroke
    grestore}}
```

`\pst@blurclosedshadow` The PostScript code for blurred shadows is produced by the following macro. It pushes the diverse parameters (`\tx@PtoC` does polar to cartesian coordinate transformation for the shadow offset) and calls `BlurShadow`. Afterwards, it fills and strokes the current path, same as the original `\pst@closedshadow`.

```
\def\pst@blurclosedshadow{%
  \addto@pscode{%
    gsave
    gsave \pst@usecolor\psshadowcolor currentrgbcolor grestore
    gsave \pst@usecolor\psx@blurbg currentrgbcolor grestore
    \psx@blurradius\space
    \psx@blursteps\space
    \psk@shadowsize \psk@shadowangle \tx@PtoC
    tx@PstBlurDict begin BlurShadow end
    grestore
    gsave
    \pst@usecolor\psfillcolor
    gsave fill grestore
    stroke
    grestore}}
```

`\pst@blurclosedshadow` This one looks very impressing. In fact, it is a verbatim copy of `\psshadowbox`, with only the line `\advance\pst@dimh\psx@blurradius\p@` added!

```
\def\psblurbox{%
\def\pst@par{}\pst@object{psblurbox}}
\def\psblurbox@i{\pst@makebox\psblurbox@ii}
\def\psblurbox@ii{%
  \begingroup
  \pst@useboxpar
  \psblurtrue
  \psshadowtrue
  \psboxseptrue
  \setbox\pst@hbox=\hbox{\psframebox@ii}%
  \pst@dimh=\psk@shadowsize\p@
  \pst@dimh=.7071\pst@dimh
  \advance\pst@dimh\psx@blurradius\p@
  \pst@dimg=\dp\pst@hbox
  \advance\pst@dimg\pst@dimh
```

```
  \dp\pst@hbox=\pst@dimg
  \pst@dimg=\wd\pst@hbox
  \advance\pst@dimg\pst@dimh
  \wd\pst@hbox=\pst@dimg
  \leavevmode
  \box\pst@hbox
\endgroup}
%%
\catcode`\@=\TheAtCode\relax
```
⟨/texfile⟩

3.4 The `pst-blur.pro` file

The file `pst-blur.pro` contains PostScript definitions to be included in the PostScript output by the `dvi`-to-PostScript converter, eg `dvips`. This is all rather similar to `pst-slpe.pro`, and I just don't feel like explaining it, so you'll have to work through it yourself, if you want to know what happens. The trick is basically to draw the outline repeatedly with varying line widths. The procedure `Shadow` called in `BlurShadow` is defined in `pstricks.pro` and translates the current path based on an x- and y-displacement taken from the stack.

⟨∗prolog⟩
```
/tx@PstBlurDict 60 dict def
tx@PstBlurDict begin
/Iterate {
  /SegLines ED
  /ThisB ED /ThisG ED /ThisR ED
  /NextB ED /NextG ED /NextR ED
  /W 2.0 BlurRadius mul def
  /WDec W SegLines div def
  /RInc NextR ThisR sub SegLines div def
  /GInc NextG ThisG sub SegLines div def
  /BInc NextB ThisB sub SegLines div def
  /R ThisR def
  /G ThisG def
  /B ThisB def
  SegLines {
    R G B
    sqrt 3 1 roll sqrt 3 1 roll sqrt 3 1 roll
    setrgbcolor
    gsave W setlinewidth
    stroke grestore
    /W W WDec sub def
    /R R RInc add def
    /G G GInc add def
    /B B BInc add def
  } bind repeat
} def
/BlurShadow {
  Shadow
```

```
121  /BlurSteps ED
122  /BlurRadius ED
123  dup mul /BEnd ED dup mul /GEnd ED dup mul /REnd ED
124  dup mul /BBeg ED dup mul /GBeg ED dup mul /RBeg ED
125  RBeg REnd add 0.5 mul /RMid ED
126  GBeg GEnd add 0.5 mul /GMid ED
127  BBeg BEnd add 0.5 mul /BMid ED
128  /OuterSteps BlurSteps 2 div cvi def
129  /InnerSteps BlurSteps OuterSteps sub def
130  1 setlinejoin
131  RMid GMid BMid REnd GEnd BEnd OuterSteps Iterate
132  gsave RBeg sqrt GBeg sqrt BBeg sqrt setrgbcolor fill grestore
133  clip
134  0 setlinejoin
135  RMid GMid BMid RBeg GBeg BBeg InnerSteps Iterate
136 } def
137 end
```
138 ⟨/prolog⟩

Change History

v1.80

General: First public release. (mg) 1

v2.00

General: using the extended pstxkey instead of the old pst-key package; creating a dtx file; new LaTeX wrapper file (hv) 1

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.

S

T