

pst-blur package

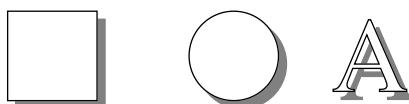
version 2.0

Martin Giese*

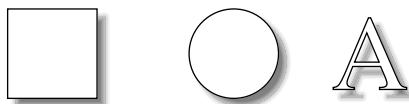
2005/09/08

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}'
```

Martin.Giese@oeaw.ac.at

2 Package Usage

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

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

in the document prologue for L^AT_EX, and

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

in “plain” T_EX.

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}
```

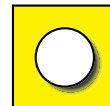


for a circle with a blurred shadow. The parameter **blur** has no influence if **shadow** is **false**.

shadowsize The rendering of blurred shadows is controlled by a number of additional graphics parameters. The offset of the shadow is controlled by the parameters **shadowangle** and **shadowsize**, which are the same as for ordinary shadows.¹
blurradius 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 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}
```

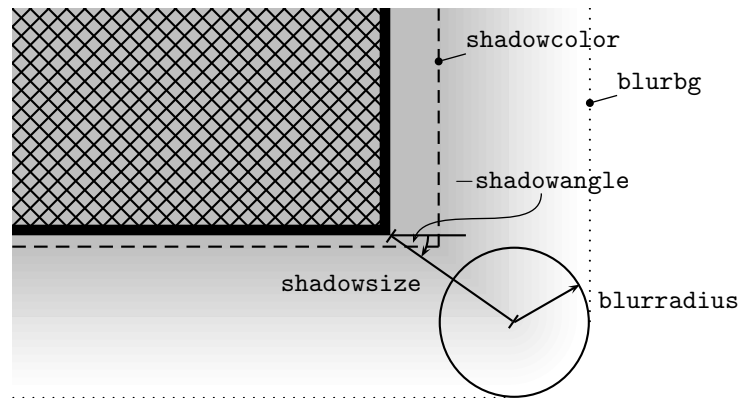


blursteps The number of distinct colour steps painted between **shadowcolor** and **blurbg** is controlled by the parameter **blursteps**. The default value for **blursteps** is 20, which is usually more than sufficient. Note, that higher values for **blursteps** result in proportionally slower rendering. 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 T_EX does not know about the extra space taken

¹In 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.

```

1 <*stylefile>
2 \RequirePackage{pstricks}
3 \ProvidesPackage{pst-blur}[2005/09/08 package wrapper for
4   pst-blur.tex (hv)]
5 \input{pst-blur.tex}
6 \ProvidesFile{pst-blur.tex}
7   [\filedate\space v\fileversion\space 'PST-blur' (hv)]
8 </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.

```

9 <*texfile>
10 \csname PstBlurLoaded\endcsname
11 \let\PstBlurLoaded\endinput
12 \ifx\PSTricksLoaded\endinput\else
13   \def\next{\input pstricks.tex }\expandafter\next
14 \fi

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

15 \ifx\PSTXKeyLoaded\endinput\else\input pst-xkey \fi
16 \def\fileversion{2.0}
17 \def\filedate{2005/09/08}
18 \message{ v\fileversion, \filedate}
19 \edef\TheAtCode{\the\catcode'\@}
20 \catcode'\@=11

```

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

```

21 \pst@addfams{pst-blur}
22 \pstheader{pst-blur.pro}

```

3.2.1 New graphics parameters

<p>blur</p> <p>blurradius</p> <p>blursteps</p> <p>blurbg</p>	<p>The definitions of the new graphics parameters follow the definitions for parameters of the same types found in <code>pstricks.tex</code>.</p> <pre> 23 \newif\ifpsblur 24 \define@key[psset]{pst-blur}{blur}[true]{\@nameuse{psblur#1}\pst@setrepeatarrowsflag} 25 \psset{blur=false} 26 %% 27 \define@key[psset]{pst-blur}{blurradius}{\pst@getlength{#1}\psx@blurradius} 28 \psset{blurradius=1.5pt} 29 %% 30 \define@key[psset]{pst-blur}{blursteps}{\pst@getint{#1}\psx@blursteps} 31 \psset{blursteps=20} 32 %% 33 \define@key[psset]{pst-blur}{blurbg}{\pst@getcolor{#1}\psx@blurbg} 34 \psset{blurbg=white} </pre>
--	---

3.3 Hooking into the PSTricks shadow macros

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

```

35 \def\pst@closedshadow{%
36 \ifpsblur\pst@blurclosedshadow\else\pst@sharpclosedshadow\fi
37 }
38 \def\pst@sharpclosedshadow{%
39   \addto@pscode{%
40     gsave

```

```

41 \psk@shadowsize \psk@shadowangle \tx@PtoC
42 \tx@Shadow
43 \pst@usecolor\psshadowcolor
44 gsave fill grestore
45 stroke
46 grestore
47 gsave
48 \pst@usecolor\psfillcolor
49 gsave fill grestore
50 stroke
51 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`.

```

52 \def\pst@blurclosedshadow{%
53 \addto@pscode{%
54   gsave
55   gsave \pst@usecolor\psshadowcolor currentrgbcolor grestore
56   gsave \pst@usecolor\psx@blurbg currentrgbcolor grestore
57   \psx@blurradius\space
58   \psx@blursteps\space
59   \psk@shadowsize \psk@shadowangle \tx@PtoC
60   tx@PstBlurDict begin BlurShadow end
61   grestore
62   gsave
63   \pst@usecolor\psfillcolor
64   gsave fill grestore
65   stroke
66   grestore}}

```

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

```

67 \def\psblurbox{%
68 \def\pst@par{}\pst@object{psblurbox}}
69 \def\psblurbox@i{\pst@makebox\psblurbox@ii}
70 \def\psblurbox@ii{%
71 \beginngroup
72 \pst@useboxpar
73 \psblurtrue
74 \psshadowtrue
75 \psboxseptrue
76 \setbox\pst@hbox=\hbox{\psframebox@ii}%
77 \pst@dimh=\psk@shadowsize\p@
78 \pst@dimh=.7071\pst@dimh
79 \advance\pst@dimh\psx@blurradius\p@
80 \pst@dimg=\dp\pst@hbox
81 \advance\pst@dimg\pst@dimh

```

```

82 \dp\pst@hbox=\pst@dimg
83 \pst@dimg=\wd\pst@hbox
84 \advance\pst@dimg\pst@dimh
85 \wd\pst@hbox=\pst@dimg
86 \leavevmode
87 \box\pst@hbox
88 \endgroup}
89 %%
90 \catcode'\@=\TheAtCode\relax
91 </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.

```

92 <*prolog>
93 /tx@PstBlurDict 60 dict def
94 tx@PstBlurDict begin
95 /Iterate {
96   /SegLines ED
97   /ThisB ED /ThisG ED /ThisR ED
98   /NextB ED /NextG ED /NextR ED
99   /W 2.0 BlurRadius mul def
100  /WDec W SegLines div def
101  /RInc NextR ThisR sub SegLines div def
102  /GInc NextG ThisG sub SegLines div def
103  /BInc NextB ThisB sub SegLines div def
104  /R ThisR def
105  /G ThisG def
106  /B ThisB def
107  SegLines {
108    R G B
109    sqrt 3 1 roll sqrt 3 1 roll sqrt 3 1 roll
110    setrgbcolor
111    gsave W setlinewidth
112    stroke grestore
113    /W W WDec sub def
114    /R R RInc add def
115    /G G GInc add def
116    /B B BInc add def
117  } bind repeat
118 } def
119 /BlurShadow {
120   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 pst-	
	xkey instead of the old pst-key package; creating a dtx file; new L ^A T _E X 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.

A		<code>\pst@getcolor</code>	33
<code>\addto@pscode</code>	39, 53	<code>\pst@getint</code>	30
B		<code>\pst@getlength</code>	27
<code>\blur</code>	<i>2</i> , <u>23</u>	<code>\pst@hbox</code>	76, 80, 82, 83, 85, 87
<code>\blurbg</code>	<i>2</i> , <u>23</u>	<code>\pst@makebox</code>	69
<code>\blurradius</code>	<i>2</i> , <u>23</u>	<code>\pst@object</code>	68
<code>\blursteps</code>	<i>2</i> , <u>23</u>	<code>\pst@par</code>	68
I		<code>\pst@setrepeatarrowsflag</code>	24
<code>\ifpsblur</code>	23, 36	<code>\pst@sharpclosedshadow</code>	36, 38
P		<code>\pst@useboxpar</code>	72
<code>\psblurbox</code>	<i>2</i> , 67	<code>\pst@usecolor</code>	43, 48, 55, 56, 63
<code>\psblurbox@i</code>	69	<code>\PstBlurLoaded</code>	11
<code>\psblurbox@ii</code>	69, 70	<code>\pstheader</code>	22
<code>\psblurtrue</code>	73	<code>\PSTXKeyLoaded</code>	15
<code>\psboxseptrue</code>	75	<code>\psx@blurbg</code>	33, 56
<code>\psfillcolor</code>	48, 63	<code>\psx@blurradius</code>	27, 57, 79
<code>\psframebox@ii</code>	76	<code>\psx@blursteps</code>	30, 58
<code>\psk@shadowangle</code>	41, 59	S	
<code>\psk@shadowsize</code>	41, 59, 77	<code>\shadowangle</code>	<i>2</i>
<code>\psshadowcolor</code>	43, 55	<code>\shadowcolor</code>	<i>2</i>
<code>\psshadowtrue</code>	74	<code>\shadowsize</code>	<i>2</i>
<code>\pst@blurclosedshadow</code>	36, <u>52</u> , <u>67</u>	T	
<code>\pst@closedshadow</code>	<u>35</u>	<code>\TheAtCode</code>	19, 90
<code>\pst@dimg</code>	80–85	<code>\tx@PtoC</code>	41, 59
<code>\pst@dimh</code>	77–79, 81, 84	<code>\tx@Shadow</code>	42