

The dichromacy package (v0.3.0)

Johan Larsson, Simon Pfahler

August 10, 2026

Contents

1	Introduction	1
1.1	For the Impatient	1
1.2	Overview	2
2	Usage	2
2.1	Package Options	2
2.2	Commands	3
3	Current Limitations	4
3.1	Engine Requirements	4
3.2	Raster Image Requirements	4
3.3	PDF/Vector Graphics Limitations	4
4	Implementation	4
4.1	Package Dependencies	4
4.2	Engine Check	5
4.3	Load Lua Module	5
4.4	Hook into xcolor	5
4.5	Hook into pgf Shadings	5
4.6	User Commands	6
4.7	Package Configuration	7

1 Introduction

In colorblind-safe documents, the contents are presented so that the same information is conveyed to readers regardless of a potential color vision deficiency. By simulating color vision deficiencies (CVDs), this package provides the tools to check for colorblind-safeness without the need to consult a person affected by it.

1.1 For the Impatient

To check your document for deuteranopia (the most common form of color blindness), simply load the package like this:

```
\usepackage[deuteranopia]{dichromacy}
```

All colors in your document will then be transformed to appear as they would to someone with deuteranopia.

1.2 Overview

This package simulates various types of color vision deficiency (CVD) in $\text{\LaTeX}$ documents, allowing authors to check whether their color choices are accessible to readers with various kinds and severities of CVDs. By transforming all colors according to CVD models, you can verify that information conveyed through color remains distinguishable even for colorblind readers.

CVD simulation is useful for scientific papers with color-coded data, educational materials, presentations, maps, charts, and any document where color plays a role in conveying information.

Figure 1 shows a test image in its original colors alongside three CVD types at different severities.

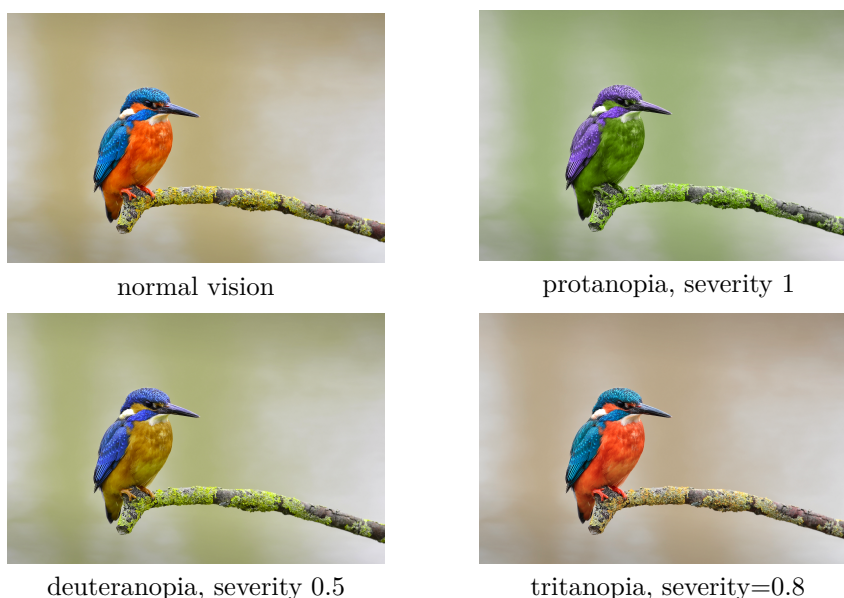


Figure 1: Example of the effect of different color vision deficiencies for a test image.

2 Usage

2.1 Package Options

The behavior of the CVD simulation can be set in the package options. These options apply to the entire document or until they are changed later using the commands from section 2.2.

type *protanopia, deuteranopia, tritanopia*..... (none specified)

Sets the type of color-vision deficiency.

<code>severity</code>	<i><dimension></i> (none specified)
	Sets the severity level of the simulation, on a scale from 0 to 1. 1 means maximum severity (i.e. full colorblindness of a given type), while 0 means completely normal vision.
<code>graphics hook</code>	<i><boolean></i> true
	Enable or disable the simulation for included PDF images.
<code>graphics convert</code>	<i><boolean></i> false
	Enable or disable ImageMagick conversion for raster images (PNG/JPG).
<code>protanopia</code>	<i><flag></i> (not specified)
	Preset for protanopia (red-blind). This is equivalent to providing <code>type=protanopia</code> and <code>severity=1.0</code> .
<code>deutanopia</code>	<i><flag></i> (not specified)
	Preset for deutanopia (green-blind). This is equivalent to providing <code>type=deutanopia</code> and <code>severity=1.0</code> .
<code>tritanopia</code>	<i><flag></i> (not specified)
	Preset for tritanopia (blue-blind). This is equivalent to providing <code>type=tritanopia</code> and <code>severity=1.0</code> .
<code>protanomaly</code>	<i><flag></i> (not specified)
	Preset for protanomaly (red-weak). This is equivalent to providing <code>type=protanopia</code> and <code>severity=0.5</code> .
<code>deutanomaly</code>	<i><flag></i> (not specified)
	Preset for deutanomaly (green-weak). This is equivalent to providing <code>type=deutanopia</code> and <code>severity=0.5</code> .
<code>tritanomaly</code>	<i><flag></i> (not specified)
	Preset for tritanomaly (blue-weak). This is equivalent to providing <code>type=tritanopia</code> and <code>severity=0.5</code> .

2.2 Commands

At any point in the document, the behavior of the CVD simulation can be changed using the following commands. There are some technical limitations to this, see section 3.

<code>\cvdtype{<type>}</code>	Set the type of color-vision deficiency. Possible options are <code>protanopia</code> , <code>deutanopia</code> and <code>tritanopia</code> .
<code>\cvdseverity{<dimension>}</code>	Set the severity level on a scale from 0 to 1. 1 means maximum severity (i.e. full colorblindness of a given type), while 0 means completely normal vision.
<code>\cvdenable</code>	Enable color vision deficiency simulation.
<code>\cvddisable</code>	Disable color vision deficiency simulation. Note that the type and severity are stored in the background, so a subsequent <code>\cvdenable</code> restores the original CVD simulation behavior.
<code>\cvdincludegraphics[...]</code>	Include a raster image (PNG/JPG) with CVD transformation applied. This macro
<code>{<path>}</code>	is mostly for internal use, as <code>\cvdincludegraphics</code> and <code>\includegraphics</code> behave identical when CVD simulation is enabled. However, when CVD simulation is disabled, this macro can be used to apply the CVD transformation to a single raster

image. Note that this macro has no effect for PDF images, these are transformed according to the current state of the CVD simulation.

`\cvddefinecolor[<options>]{<source color>{<target color>}` Define a new color named *<target color>* by applying the current CVD transformation to the existing color *<source color>*. The optional *<options>* argument can be used to override the current CVD settings (e.g., `type`, `severity`) for this specific color definition, with the same syntax as `\cvdset`. For example, `\cvddefinecolor[protanopia,severity=0.5]{blue}{blue-cvd}` defines a color `blue-cvd` that appears as `blue` would to someone with protanopia of severity 50%.

`\cvdset{<options>}` Set one or more CVD options at once using key-value syntax. All options available at package load time can be used here, for a list see section 2.1.

3 Current Limitations

3.1 Engine Requirements

Only LuaTeX is currently supported. Support for other engines is under development.

3.2 Raster Image Requirements

Raster image (PNG/JPG) transformation requires compiling with `-shell-escape` and ImageMagick to be installed on the system. Without these, raster images pass through untransformed.

3.3 PDF/Vector Graphics Limitations

Functional shadings (`\pgfdeclarefunctionalshading`, `ShadingType 1`) pass through unchanged as their colors are embedded in PostScript Type 4 functions.

Cached duplicate shadings pgf caches shadings, so changing CVD settings between identical `\shade` calls has no effect. Workaround: use `\cvddefinecolor` to create unique color names for each CVD variant.

scn/SCN color operators Spot/separation colors, ICCBased colors, and pattern colors are not transformed. Only device-color operators `rg/RG` (RGB) and `k/K` (CMYK) are supported.

PDF stream length When transforming PDF content streams, significant growth may cause truncation or rendering issues in some PDF viewers. Mitigation: use raster formats for complex PDFs, or split into multiple documents. When a significant growth of the PDF stream is detected, a warning for this is issued.

4 Implementation

4.1 Package Dependencies

Load required packages.

```
1 \RequirePackage{iftex}
```

```

2 \RequirePackage{xcolor}
3 \RequirePackage{graphicx}

```

4.2 Engine Check

Currently only LuaTeX is fully supported.

```

4 \sys_if_engine luatex:F
5 {
6   \msg_error:nn { dichromacy } { luatex-required }
7 }
8 \msg_new:nnn { dichromacy } { luatex-required }
9 {
10  LuaTeX-required.\
11  This-package-currently-only-works-with-LuaLaTeX.\
12  pdfLaTeX-support-is-under-development.
13 }

```

4.3 Load Lua Module

Load the Lua module that implements the CVD transformations. The `install_pdf_image_hook` function registers a callback that transforms colors in embedded PDF pages (vector graphics only). We also load the Lua File System module for file timestamp checking.

```

14 \directlua{lfs = require("lfs"); dichromacy = require("dichromacy"); dichromacy.install_pdf_image_hook}

```

4.4 Hook into xcolor

Use `xcolor`'s `\XC@bcolor` hook to transform RGB values before display. This handles text colors, color boxes, and other `xcolor`-based content.

```

15 \cs_set:Npn \XC@bcolor
16 {
17   \directlua
18   {
19     token.set_macro("current@color",~
20     dichromacy.transform_current_color("\luaescapestring{\current@color}"),~
21     "global")
22   }
23 }

```

4.5 Hook into pgf Shadings

TikZ/pgf shadings (linear and radial gradients) store their colors in PDF Shading dictionaries whose `Function` carries `/C0` and `/C1` color arrays. These objects sit in the page resources rather than in any content stream, so none of `dichromacy`'s other transform paths reach them and they need a dedicated hook.

Patch pgf's leaf tuple emitters so the RGB or CMYK tuple is filtered through `dichromacy.transform` first. From one transform each emitter sets both the space-separated macro (`\pgf@rgb/\pgf@cmyk`) that ends up in `/C0` and `/C1` for the pdf/luatex driver, and the brace-grouped system-layer record (`\pgf@sys@rgb/\pgf@sys@cmyk`) used by the dvisvgm driver, so the two never disagree. Guarded with `\ifundefined` so loading `dichromacy` without `pgf` or `tikz`

is a no-op. `ShadingType 1` (functional) shadings and `DeviceGray` shadings are intentionally not handled.

```

24 \def \__dichromacy_patch_pgf_shadings:
25 {
26   \ifundefined { pgf@getrgb@@ } { } {
27     \def \pgf@getrgb@@ ##1,##2,##3!
28     {
29       \directlua { dichromacy.set_pgf_rgb("##1",~"##2",~"##3") }
30     }
31   }
32   \ifundefined { pgf@getcmyk@@ } { } {
33     \def \pgf@getcmyk@@ ##1,##2,##3,##4!
34     {
35       \directlua { dichromacy.set_pgf_cmyk("##1",~"##2",~"##3",~"##4") }
36     }
37   }
38 }
39 \__dichromacy_patch_pgf_shadings:
40 \AtBeginDocument { \__dichromacy_patch_pgf_shadings: }

```

4.6 User Commands

`\cvdtype` Set the type of color vision deficiency to simulate.

```

41 \NewDocumentCommand \cvdtype { m }
42 {
43   \directlua { dichromacy.set_type("#1") }
44 }

```

`\cvdseverity` Set the severity of the simulation (0.0 to 1.0).

```

45 \NewDocumentCommand \cvdseverity { m }
46 {
47   \directlua { dichromacy.set_severity(#1) }
48 }

```

`\cvdenable` Enable CVD simulation.

```

49 \NewDocumentCommand \cvdenable { }
50 {
51   \directlua { dichromacy.enable() }
52 }

```

`\cvddisable` Disable CVD simulation.

```

53 \NewDocumentCommand \cvddisable { }
54 {
55   \directlua { dichromacy.disable() }
56 }

```

`\cvdincludegraphics` Include a graphics file with CVD transformation applied to raster images.

```

57 \tl_new:N \l__dichromacy_imgpath_tl
58 \NewDocumentCommand \cvdincludegraphics { O{} m }
59 {

```

```

60 \tl_set:Nx \l__dichromacy_imgpath_tl
61 {
62   \directlua
63   { tex.sprint(dichromacy.get_image_path("\luaescapestring{#2}")) }
64 }
65 \__dichromacy_orig_includegraphics[#1]{\tl_use:N \l__dichromacy_imgpath_tl}
66 }

```

`\cvddefinecolor` Define a new color by applying CVD transformation to an existing color. Usage:

`\cvddefinecolor[<options>]{<source color>}{<target color>}`

```

67 \tl_new:N \l__dichromacy_model_tl
68 \tl_new:N \l__dichromacy_values_tl
69
70 \NewDocumentCommand \cvddefinecolor { 0{ } m m }
71 {
72   % Extract the original color
73   \extractcolorspecs{#2}{\l__dichromacy_model_tl}{\l__dichromacy_values_tl}
74
75   % Apply CVD transformation with specified settings
76   \keys_set:nn { dichromacy } { #1 }
77   \cvdenable
78
79   % Transform the RGB values directly via Lua
80   \directlua{
81     local~values~~~"\luaescapestring{\l__dichromacy_values_tl}"
82     local~r,~g,~b~~~values:match("([^\,]+),([^\,]+),([^\,]+)")
83     r,~g,~b~~~tonumber(r),~tonumber(g),~tonumber(b)
84     r,~g,~b~~~dichromacy.transform("rgb",~r,~g,~b)
85     token.set_macro("l__dichromacy_values_tl",~string.format("\csstring\%.6f,\csstring\%.6f,\csstring\%.6f"))
86   }
87
88   % Define the color with transformed values
89   \use:x { \definecolor {#3} { \exp_not:V \l__dichromacy_model_tl } { \exp_not:V \l__dichromacy_values_tl } }
90
91   \cvddisable
92 }

```

4.7 Package Configuration

Define keys for package configuration using `l3keys`. Keys are available both as package load-time options and via the `\cvdset` command.

```

93 \bool_new:N \l__dichromacy_graphics_hook_bool
94 \bool_new:N \l__dichromacy_graphics_convert_bool
95 \cs_new_eq:NN \__dichromacy_orig_includegraphics \includegraphics
96
97 \keys_define:nn { dichromacy }
98 {
99   type .code:n = { \cvdtype{#1} } ,
100   severity .code:n = { \cvdseverity{#1} } ,
101   graphics~hook .bool_set:N = \l__dichromacy_graphics_hook_bool ,
102   graphics~hook .initial:n = true ,
103   graphics~hook .default:n = true ,
104   graphics~hook / true .code:n = { \directlua { dichromacy.enable_graphics_hook() } } ,

```

```

105     graphics~hook / false .code:n = { \directlua { dichromacy.disable_graphics_hook() } } ,
106     graphics~convert .bool_set:N = \l__dichromacy_graphics_convert_bool ,
107     graphics~convert .initial:n = false ,
108     graphics~convert .default:n = true ,
109     graphics~convert / true .code:n = { \__dichromacy_patch_includegraphics: } ,
110     graphics~convert / false .code:n = { \__dichromacy_unpatch_includegraphics: } ,
111     protanopia .code:n = { \cvdtype{protanopia} \cvdseverity{1.0} } ,
112     deuteranopia .code:n = { \cvdtype{deuteranopia} \cvdseverity{1.0} } ,
113     tritanopia .code:n = { \cvdtype{tritanopia} \cvdseverity{1.0} } ,
114     protanomaly .code:n = { \cvdtype{protanopia} \cvdseverity{0.5} } ,
115     deuteranomaly .code:n = { \cvdtype{deuteranopia} \cvdseverity{0.5} } ,
116     tritanomaly .code:n = { \cvdtype{tritanopia} \cvdseverity{0.5} } ,
117     unknown .code:n =
118         { \msg_warning:nxx { dichromacy } { unknown-option } { \l_keys_key_str } }
119 }
120 \msg_new:nnn { dichromacy } { unknown-option }
121 { Unknown~option~'#1'. }
122
123 \cs_new:Npn \__dichromacy_patch_includegraphics:
124 {
125     \RenewDocumentCommand \includegraphics { 0{ } m }
126     {
127         \cvdincludegraphics[##1]{##2}
128     }
129     \directlua { dichromacy.enable_graphics_convert() }
130 }
131
132 \cs_new:Npn \__dichromacy_unpatch_includegraphics:
133 {
134     \cs_set_eq:NN \includegraphics \__dichromacy_orig_includegraphics
135     \directlua { dichromacy.disable_graphics_convert() }
136 }
137
138 \NewDocumentCommand \cvdset { m }
139 {
140     \keys_set:nn { dichromacy } { #1 }
141 }
142 \ProcessKeyOptions [ dichromacy ]

```