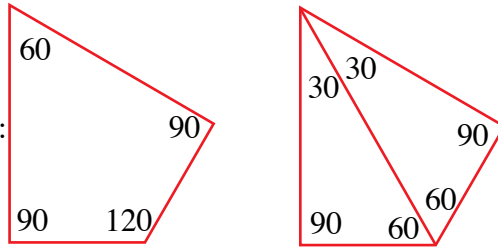


Plane Groups with ImageMagick/PerlMagick: p31m_tile

Günter Bachelier

www.aroshu.de AROSHU® Evolutionary Art © Günter Bachelier

p31m basic element:

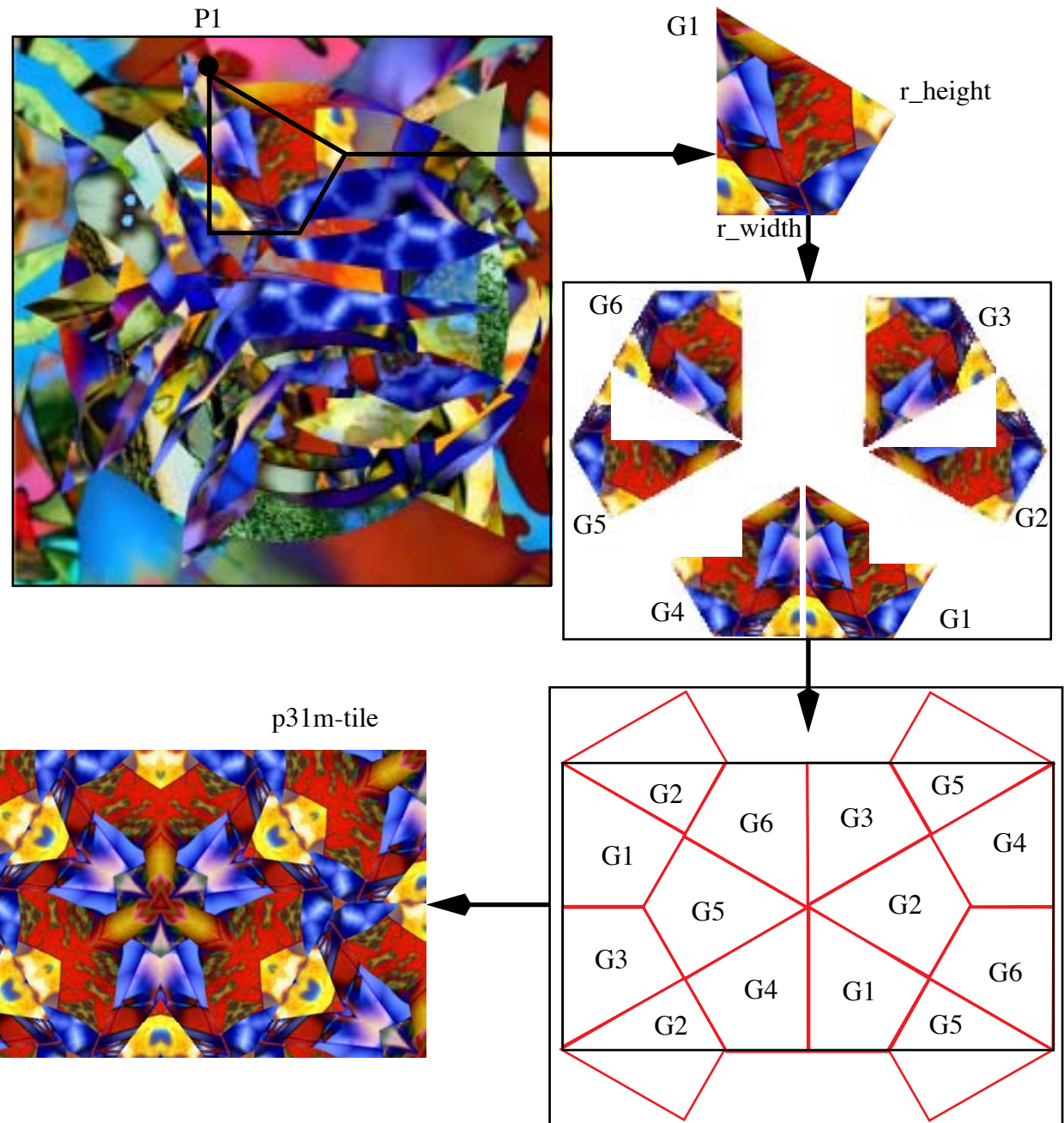


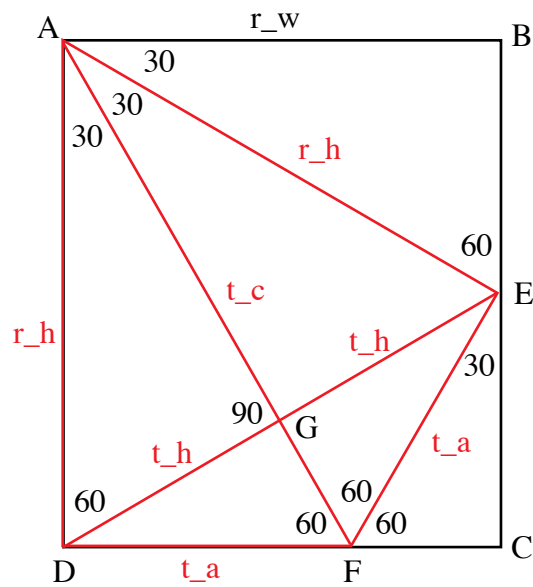
p31m-procedure:

Start

- 1) Generate G1
- 2) Generate G2, G3
- 3) Generate tile_center_right: G1+G2+G3
- 4) Generate tile_center_left
- 5) Generate tile_part_2: $G6 + 1/2 G5$
- 6) Generate tile_part_3: $1/2 G5 + G4$
- 7) Generate tile_right: center_right+part_2+part_3
- 8) Generate tile

End



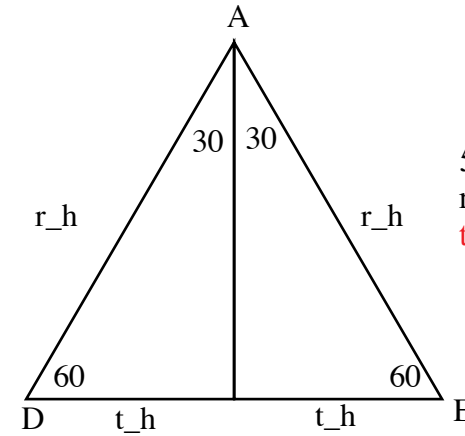


geg.: r_h
 ges.: r_w , BE , t_a

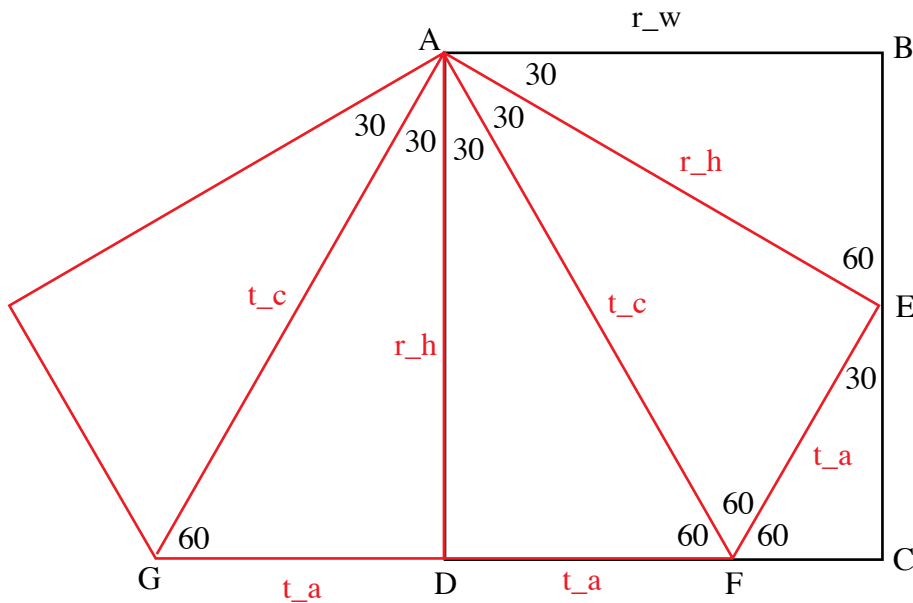
1) ABE: $r_w = 1/2 * \sqrt{3} * r_h$

2) ABE: $r_h^2 = r_w^2 + BE^2 \Leftrightarrow$
 $BE = 1/2 * r_h$

4) ADF:
 $t_c^2 = r_h^2 + t_a^2 \Leftrightarrow$
 $t_c = \sqrt{(r_h^2 + t_a^2)}$
 or insert $t_a = r_h / \sqrt{3} \Rightarrow$
 $t_c = 2/\sqrt{3} * r_h$



5) ADE is an equilateral triangle:
 $r_h = 2 * t_h \Leftrightarrow$
 $t_h = 1/2 * r_h$



idea: reduce p3 1m to p6m
 $AGD = \text{flop}(ADF)$
 AGF is an equilateral triangle ($3 * 60^\circ$)
 $\Rightarrow AG = AF = GF = t_c = 2 * t_a$

3) ADF:
 $t_c^2 = r_h^2 + t_a^2$
 $\Leftrightarrow 4 * t_a^2 = r_h^2 + t_a^2$
 $\Leftrightarrow t_a = r_h / \sqrt{3}$

Only relevant for the stochastic case

```
$r_height_min_Faktor = 0.4;
```

```
$r_height_max_Faktor = 0.6;
```

```
if ($image_width < $image_height) {$image_min = $image_width} else {$image_min = $image_height};
```

```
$r_height_min = $r_height_min_Faktor * $image_min;
```

```
$r_height_max = $r_height_max_Faktor * $image_min;
```

```
$r_height = $r_height_min + (int(rand($r_height_max - $r_height_min)) + 1);
```

```
$r_width = 1/2 * sqrt(3)* $r_height;
```

```
$x_allowed = $image_width - $r_width;
```

```
$y_allowed = $image_hight - $r_height;
```

```
$x1 = int(rand($x_allowed)) + 1;
```

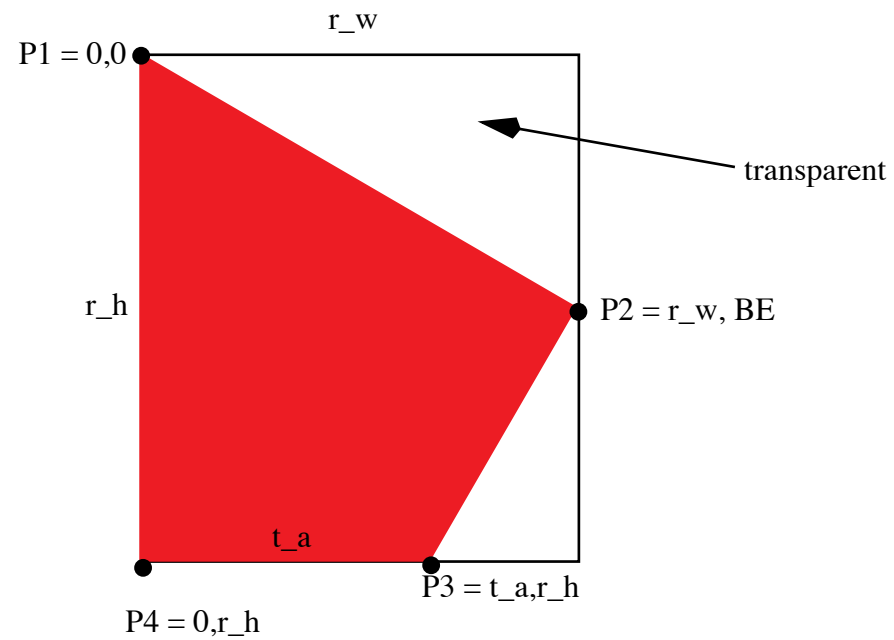
```
$y1 = int(rand($y_allowed)) + 1;
```

1.1) Generate mask p31m_mask.png

```
$BE = int(1/2 * $r_height)+1;
```

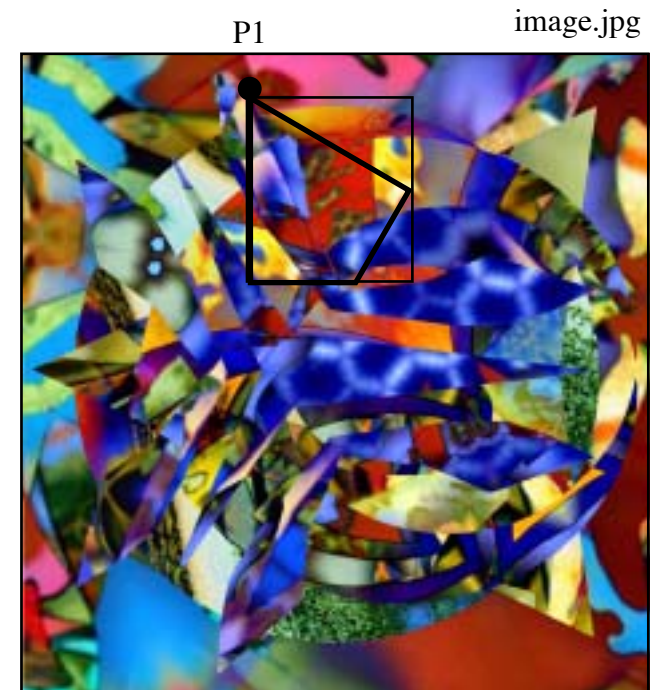
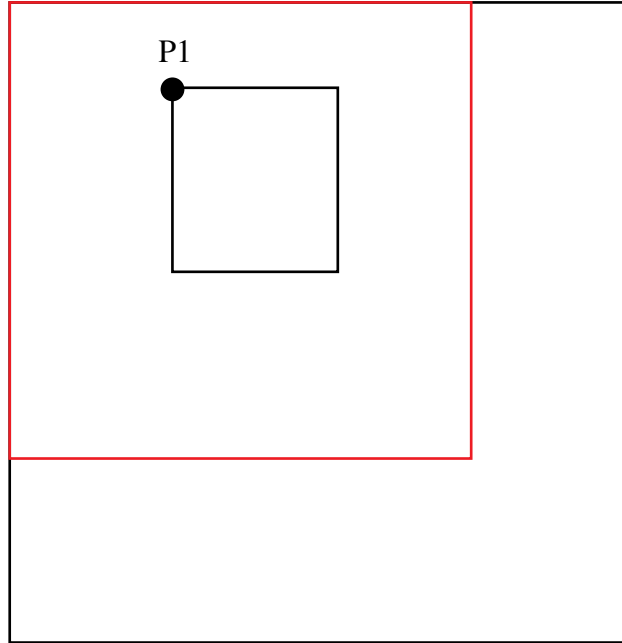
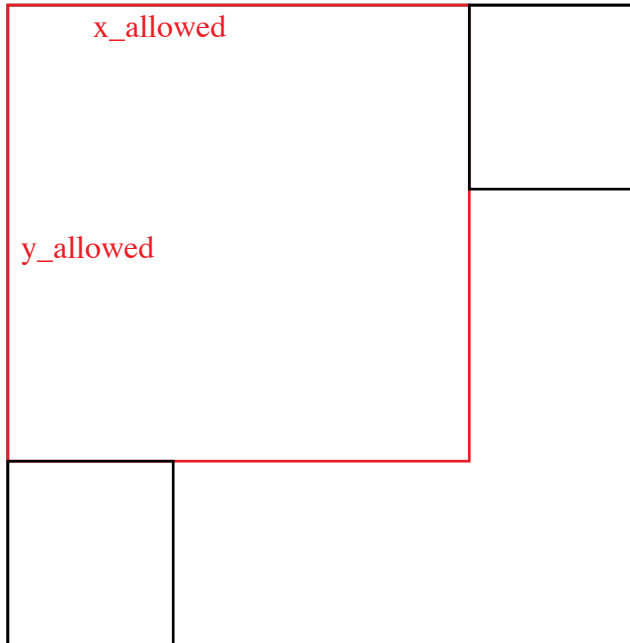
```
$t_a = int(1/sqrt(3) * $r_height)+1;
```

```
convert -size {$r_width}x{$r_height} xc:none -fill red -draw "polyline 0,0 $r_width,$BE $t_a,$r_height 0,$r_height" p31m_mask.png
```



1.2) Copy from image.jpg a (random) rectangle

`convert image.jpg -crop {$r_width}x{$r_height}+$x1+$y1 -background none +repage image_p31m.png`

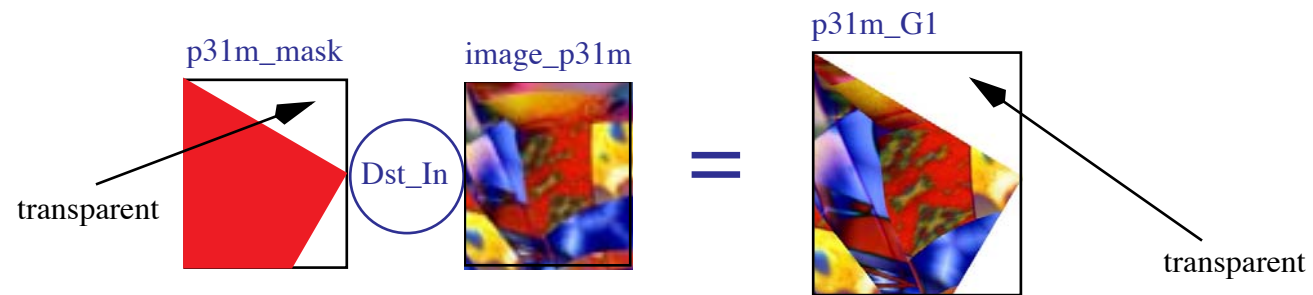


`image_p31m`



1.3) Generate p31m basic element p31m_G1

```
composite p31m_mask.png image_p31m.png -matte -compose Dst_In -background none +repage p31m_G1.png
```



2.1) Generate with G1 the basic element G2

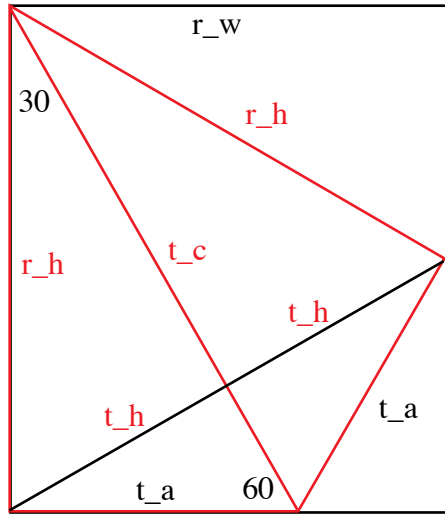
```
convert p31m_G1.png -flop -rotate -120 -background none +repage p31m_G2a.png
```

```
$t_c = int(2/sqrt(3) * $r_height)+1;
```

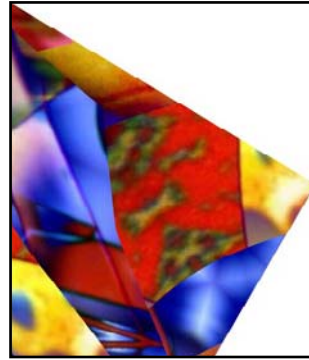
```
$t_height = int(1/2 * $r_height)+1;
```

```
$t_height_double = 2 * $t_height;
```

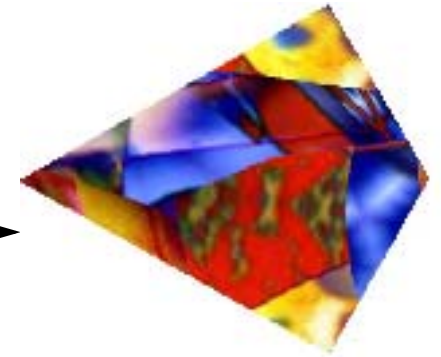
```
convert p31m_G2a.png -crop {$t_c}x{$t_height_double}+0+0 +repage p31m_G2.png
```



p31m_G1

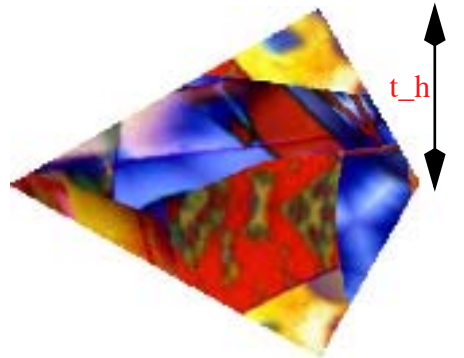


-flop -rotate -120



p31m_G2a

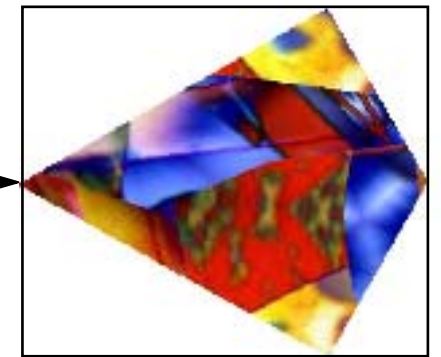
p31m_G2a



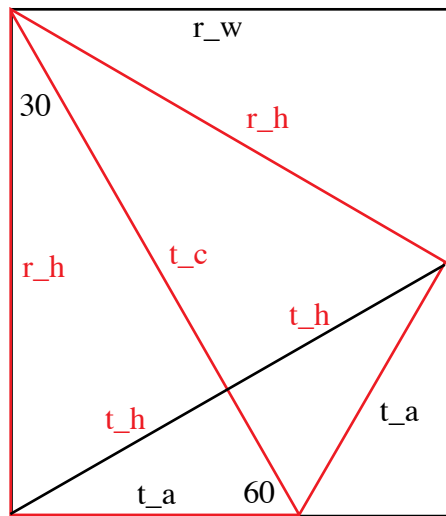
t_h

-crop

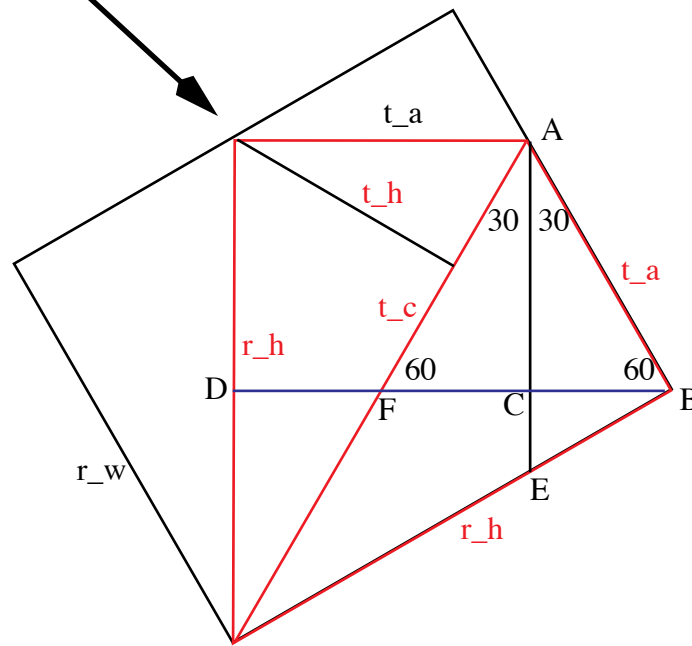
p31m_G2



t_c



-rotate -120



$$DB = t_a + BC$$

ABF is an equilateral triangle:

$$t_a = AF = BF = 2 * BC = 2 * CF$$

$$\Rightarrow DB = t_a + BC = t_a + 1/2 * t_a$$

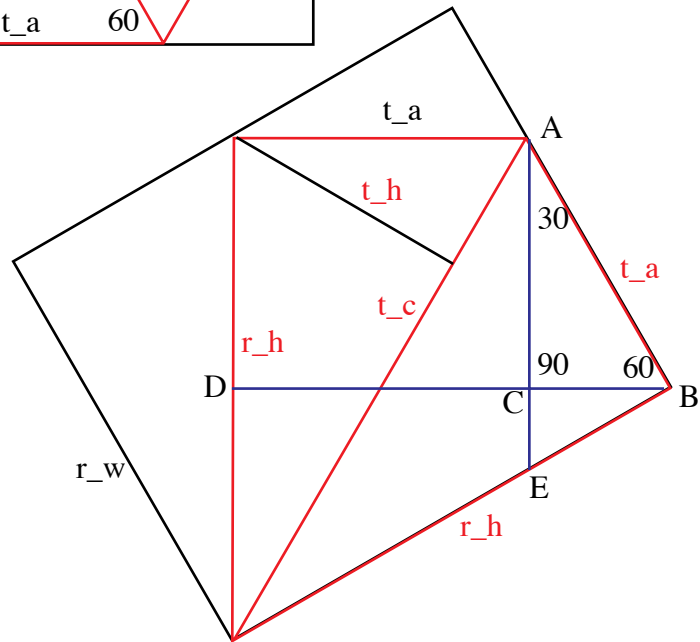
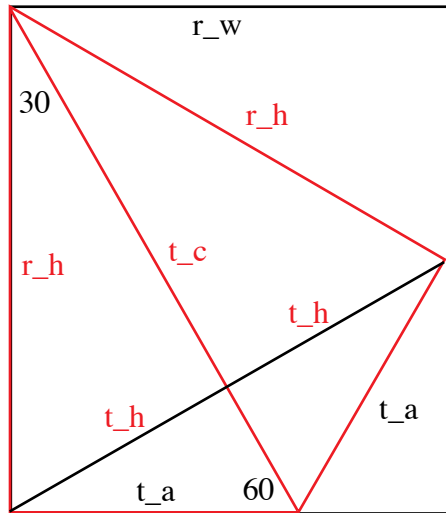
$$\Leftrightarrow DB = 3/2 * t_a$$

or insert $t_a = r_h / \sqrt{3} \Leftrightarrow$

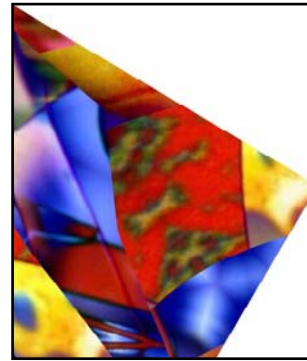
$$DB = 3/(2 * \sqrt{3}) * r_h$$

2.2) Generate with G1 the basic element G3

```
convert p31m_G1.png -rotate -120 p31m_G3a.png  
$p31m_G3a_width = $p31m_G3a->get('columns');  
$p31m_G3a_height = $p31m_G3a->get('rows');  
$DB = int(3/2 * $t_a)+1;  
$d = $p31m_G3a_width - $DB;  
$h = $p31m_G3a_height - $r_height;  
convert p31m_G3a.png -crop {$DB}x{$r_height}+$d+$h +repage p31m_G3.png
```

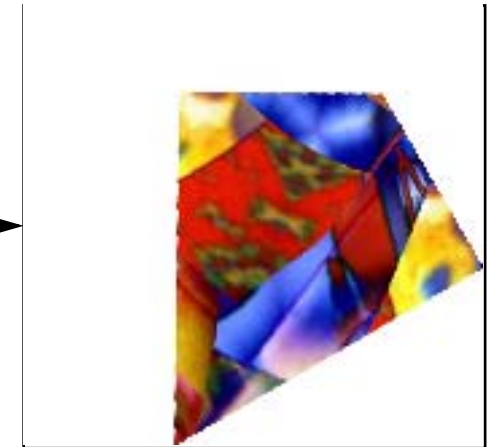


p31m_G1

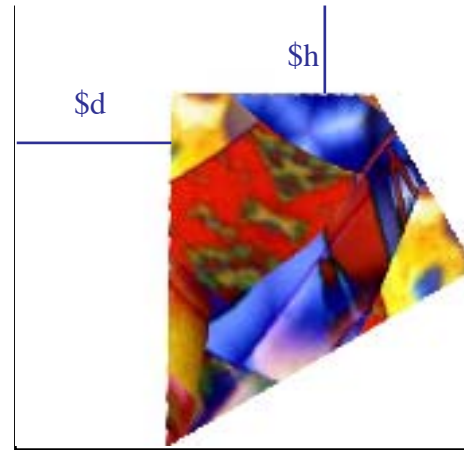


-rotate -120

p31m_G3a

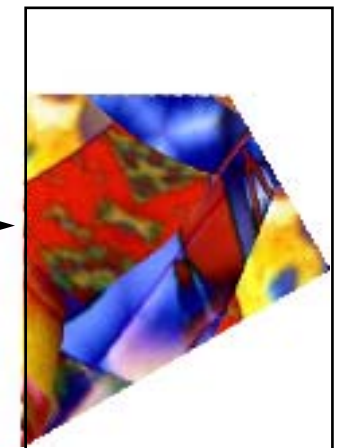


p31m_G3a



-crop

p31m_G3



3) Generate tile_center_right: G1+G2+G3

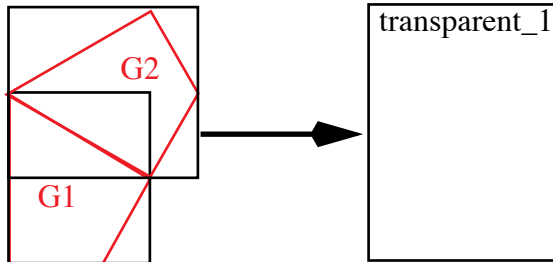
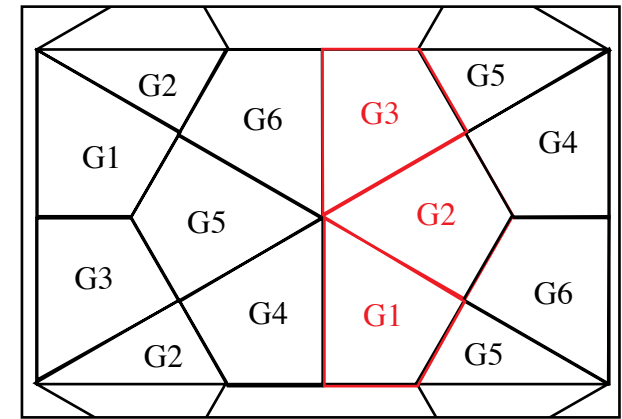
3.1) Generate tile_center_right_1: G1+transparent_1

```
$transparent_1_width = $p31m_G2->Get('columns'); # = $t_c
```

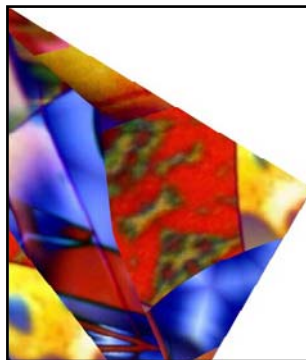
```
$transparent_1_height = $r_height + $t_height;
```

```
convert -size {$transparent_1_width}x{$transparent_1_height} xc:none transparent_1.png
```

```
composite p31m_G1.png transparent_1.png -gravity Southwest -compose Dst_Over -background none +repage p31m_tile_center_right_1.png
```

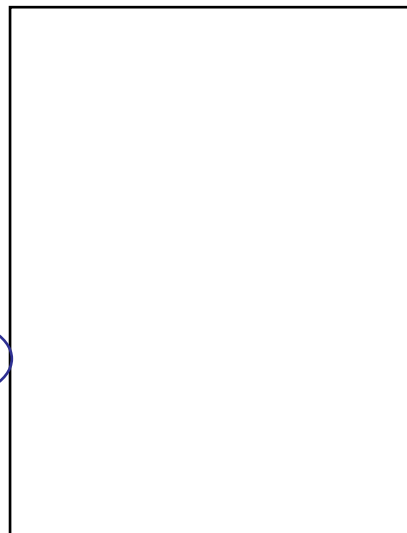


p31m_G1

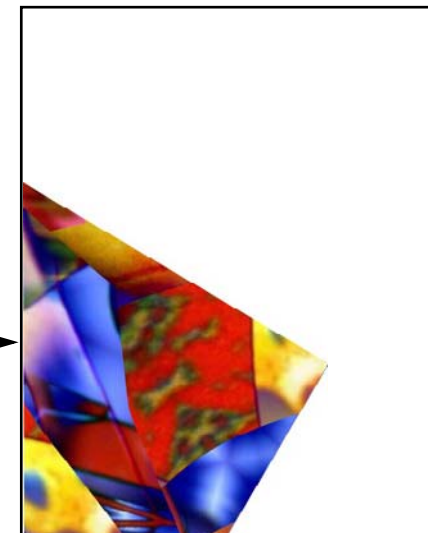


Dst_Over

transparent_1



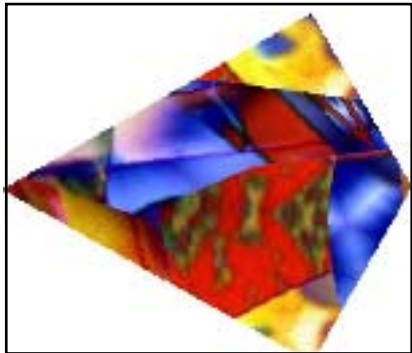
p31m_tile_center_right_1



3.2) Generate tile_center_right_2: tile_center_right_1+G2

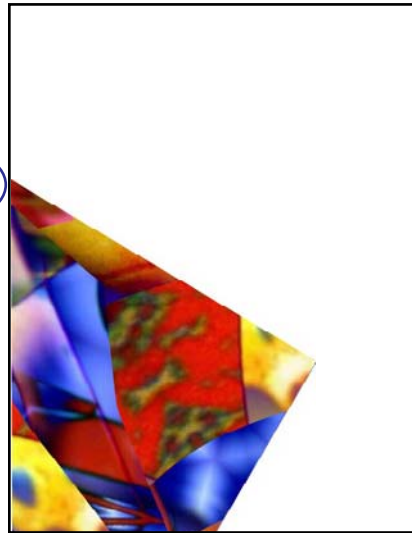
composite p31m_G2.png p31m_tile_center_right_1.png -gravity Northwest -compose Dst_Over -background none +repage p31m_tile_center_right_2.png

p31m_G2

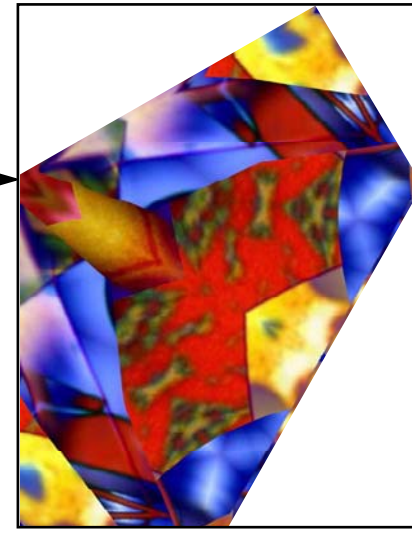


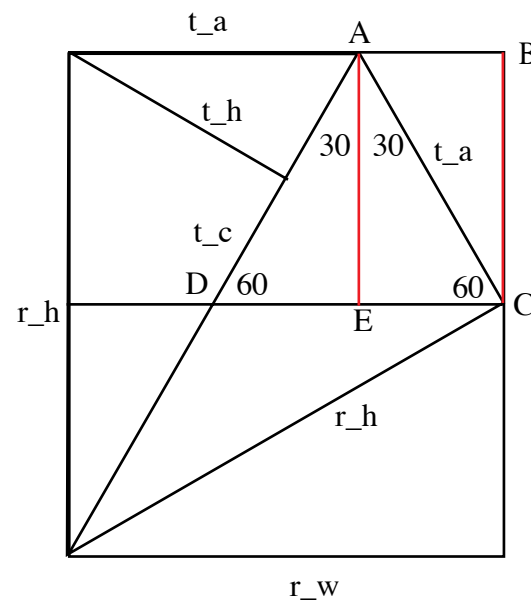
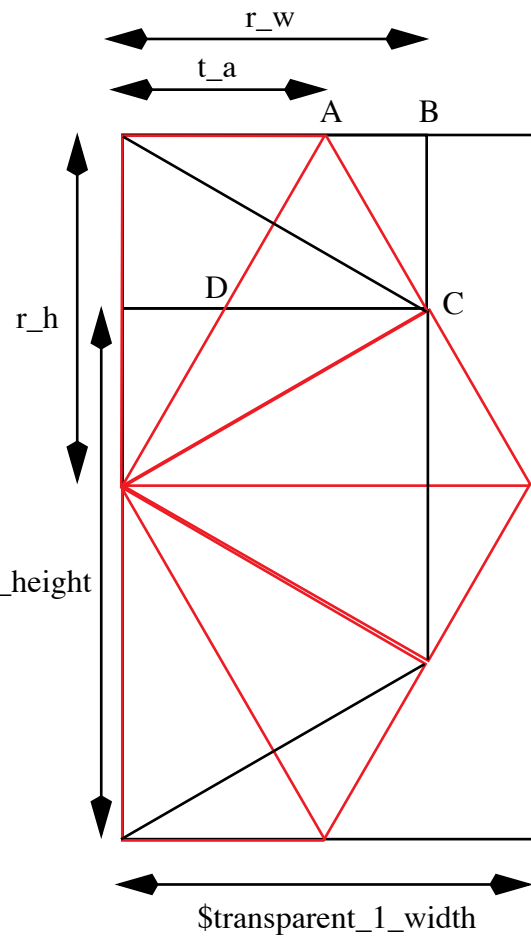
Dst_Over

p31m_tile_center_right_1



p31m_tile_center_right_2





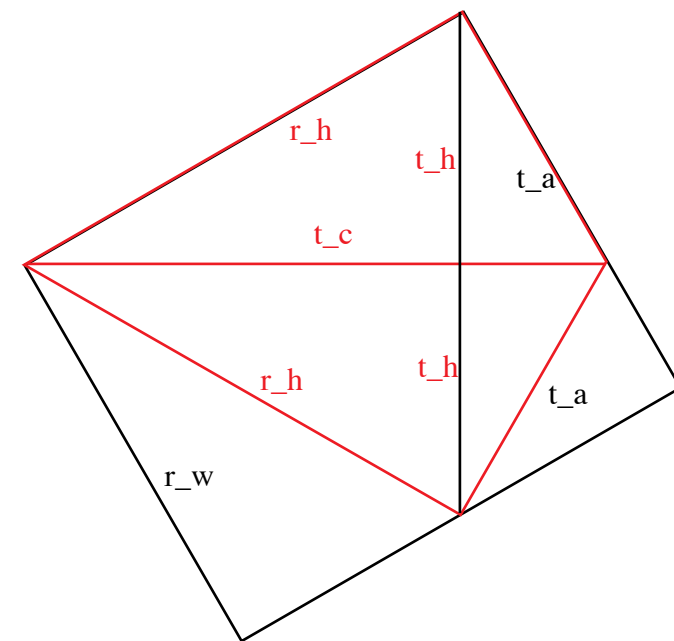
ges.: BC

ACD is an equilateral triangle:
 $t_a = AD = CD = 2 * CE = 2 * DE$

ABC:
 $t_a^2 = AB^2 + BC^2 \Leftrightarrow$
 $BC = \sqrt{t_a^2 - AB^2}$

insert $AB = EC = 1/2 * t_a \Rightarrow$
 $BC = \sqrt{t_a^2 - t_a^2/4}$
 $BC = \sqrt{3/4 * t_a^2}$
 $BC = \sqrt{3}/2 * t_a$

insert $t_a = r_h / \sqrt{3} \Rightarrow$
 $BC = \sqrt{3}/2 * r_h / \sqrt{3}$
 $BC = 1/2 * r_h$



3.3) Generate tile_center_right_3: tile_center_right_2+transparent_2

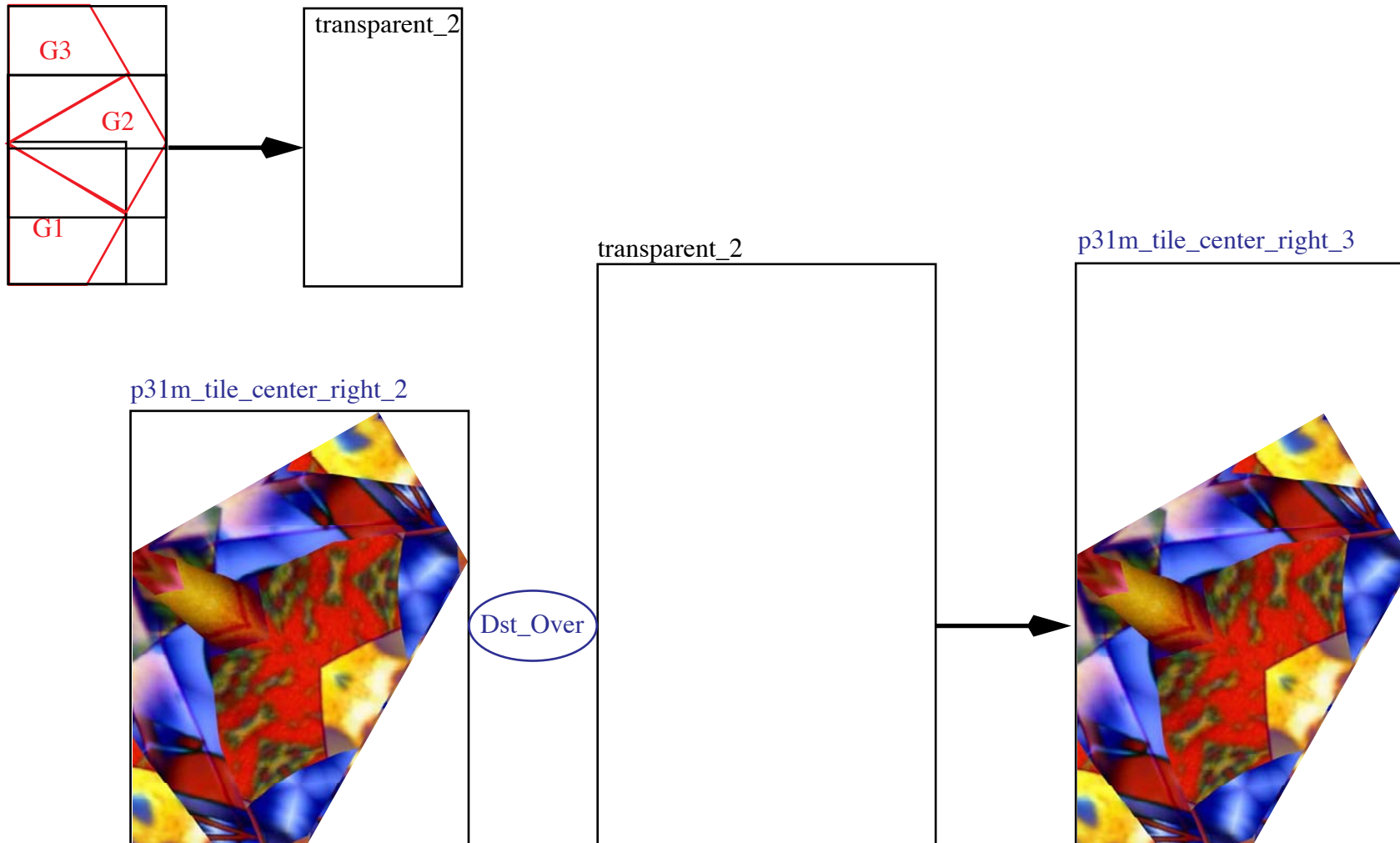
```
$BC = int(1/2 * $r_height)+1;
```

```
$transparent_2_width = $transparent_1_width;
```

```
$transparent_2_height = $transparent_1_height + $BC;
```

```
convert -size {$transparent_2_width}x{$transparent_2_height} xc:none transparent_2.png
```

```
composite p31m_tile_center_right_2.png transparent_2.png -gravity Southwest -compose Dst_Over -background none +repage p31m_tile_center_right_3.png
```



3.4) Generate tile_center_right: tile_center_right_3+G3

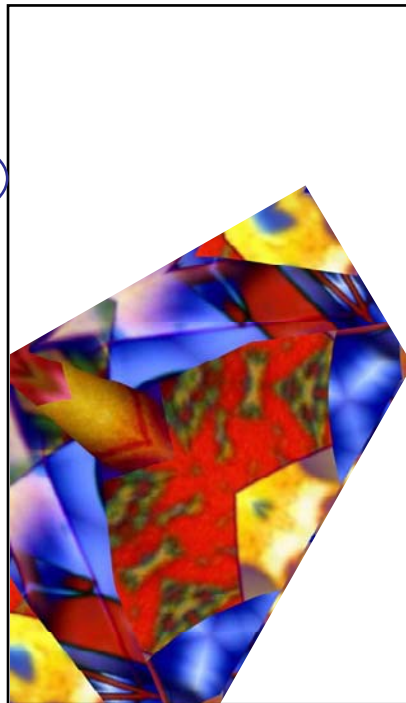
composite p31m_G3.png p31m_tile_center_right_3.png -gravity Northwest -compose Dst_Over -background none +repage p31m_tile_center_right.png

p31m_G3

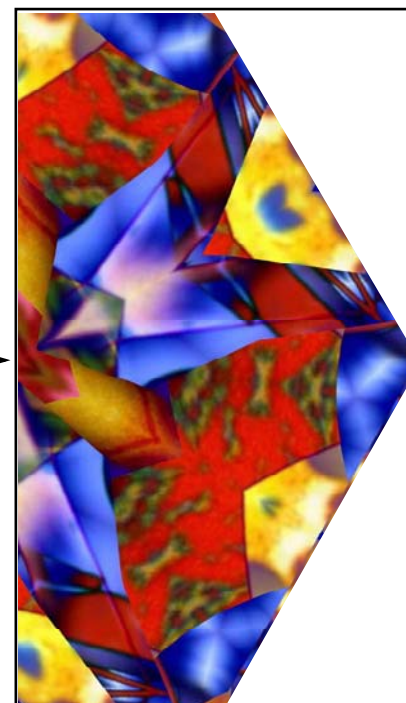


Dst_Over

p31m_tile_center_right_3

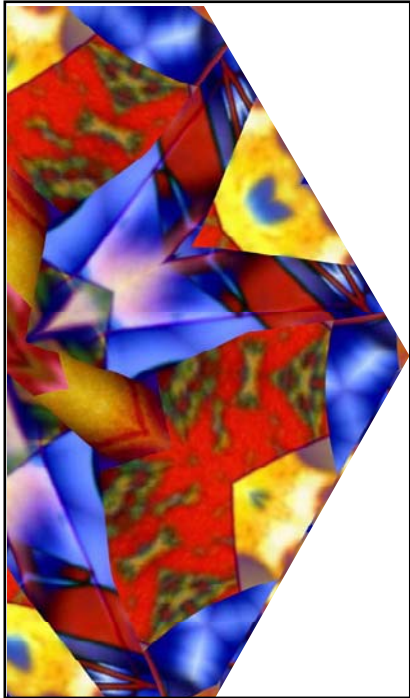


p31m_tile_center_right



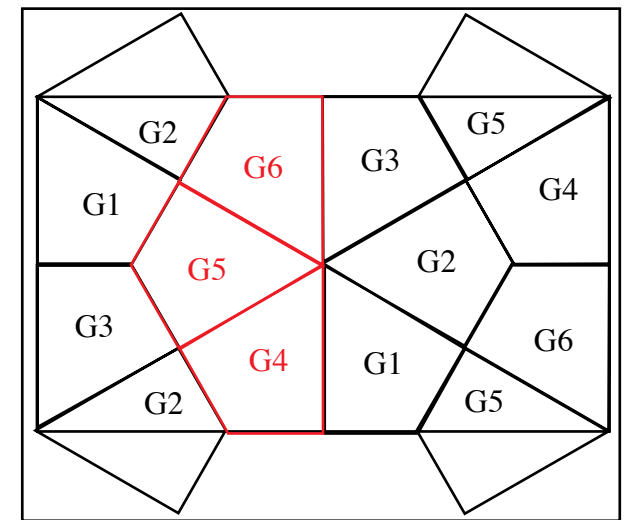
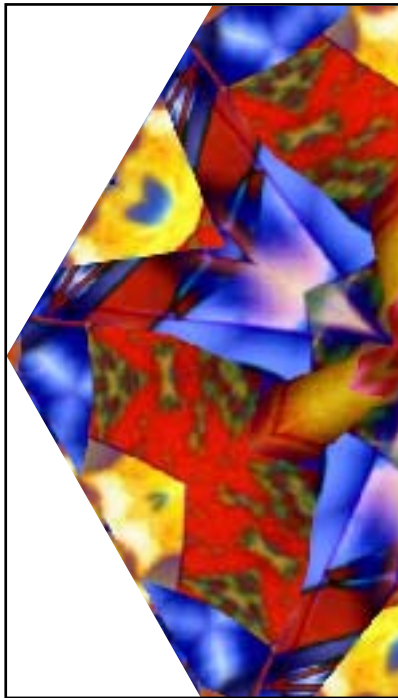
4) Generate tile_center_left
convert p31m_tile_center_right.png -flop p31m_tile_center_left.png

p31m_tile_center_right



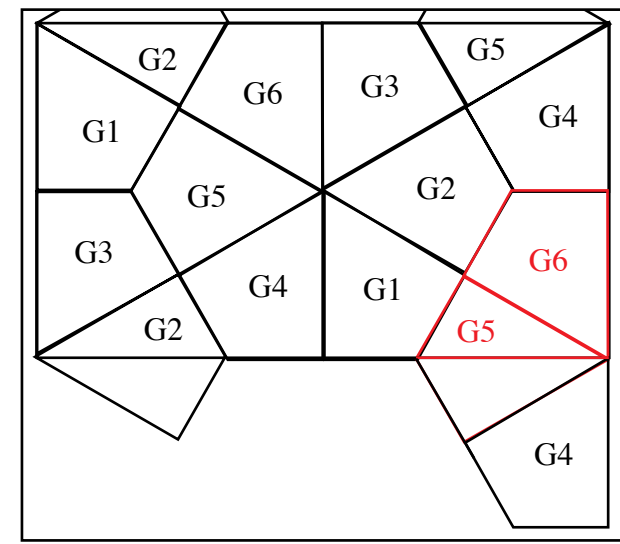
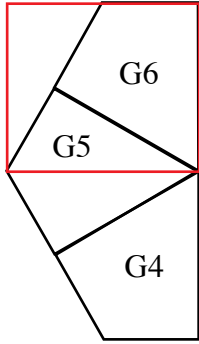
-flop

p31m_tile_center_left

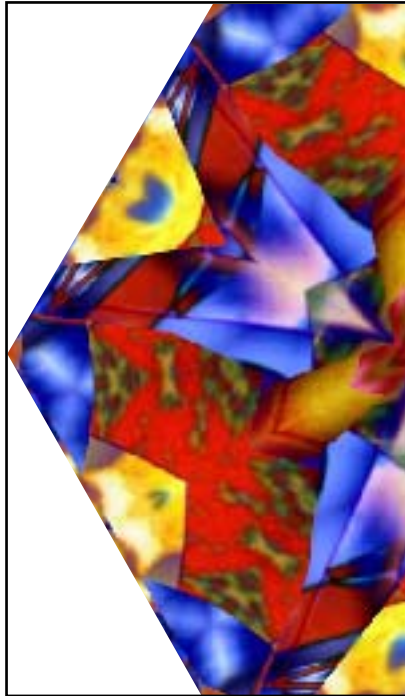


5) Generate tile_part_2: $G6 + 1/2 G5$

convert p31m_tile_center_left.png -crop $\{t_c\} \times \{r_{height}\} + 0 + 0$
-background none +repage p31m_tile_part_2.png

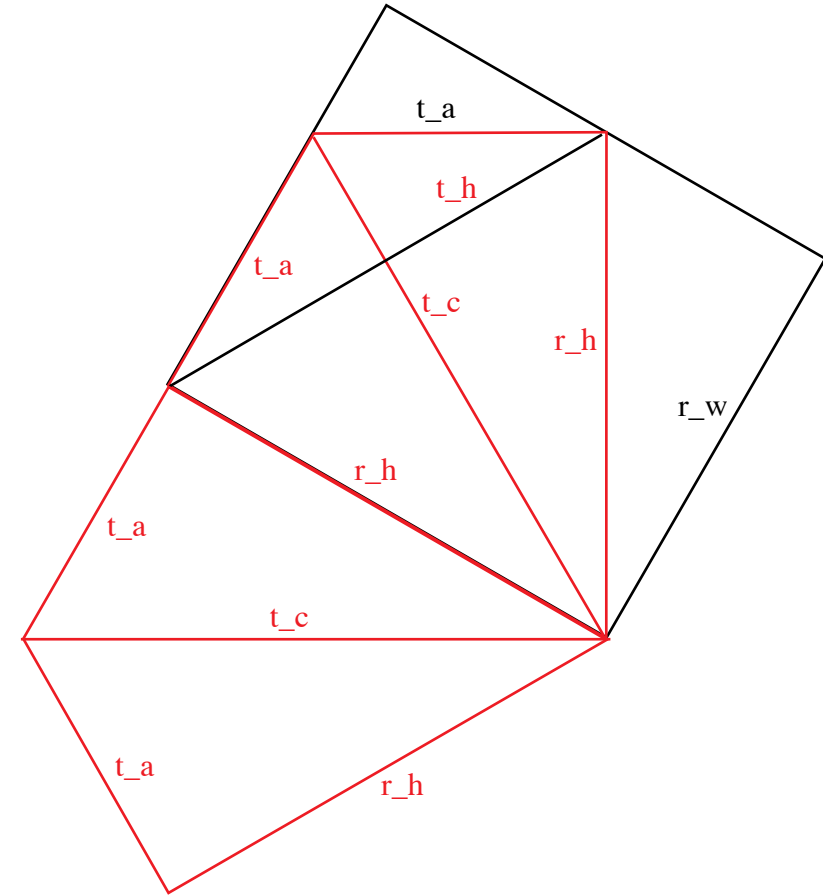
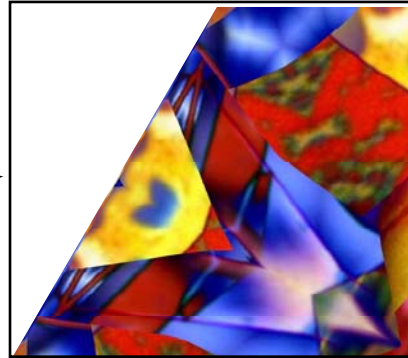


p31m_tile_center_left

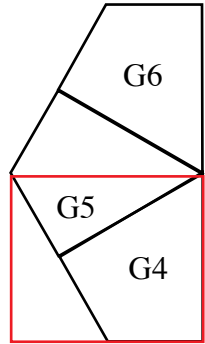
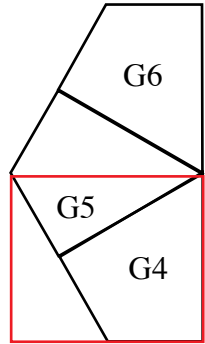


-crop

p31m_tile_part_2



The diagram shows a hexagon divided into six triangles labeled G1 through G6. A red rectangle is drawn over the bottom portion of the hexagon, enclosing triangles G4 and G5.



7) Generate tile_right: center_right+part_2+part_3

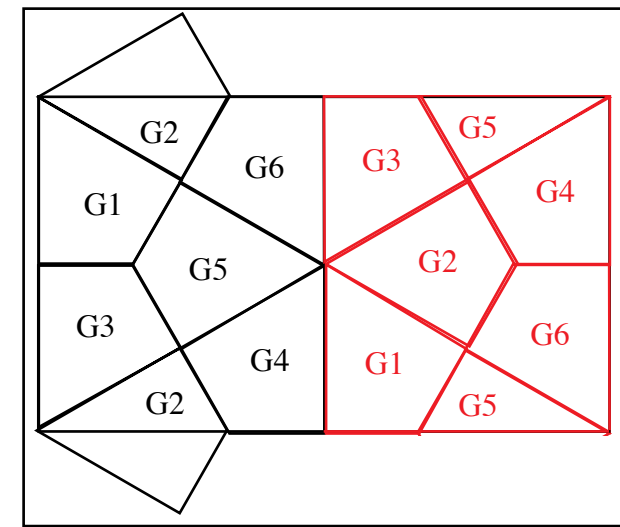
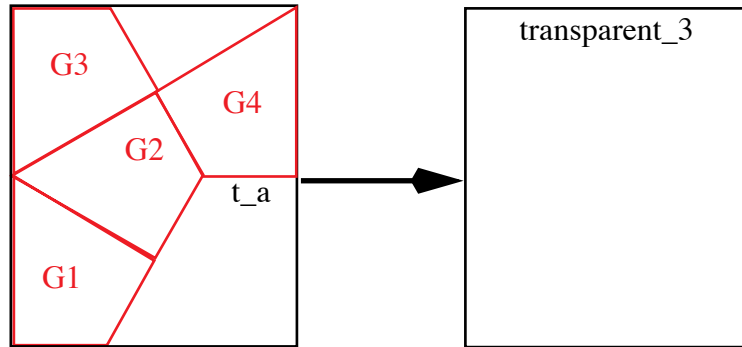
7.1) Generate tile_right_1: tile_center_right+transparent_3

$\$transparent_3_width = \$transparent_2_width + \$t_a;$

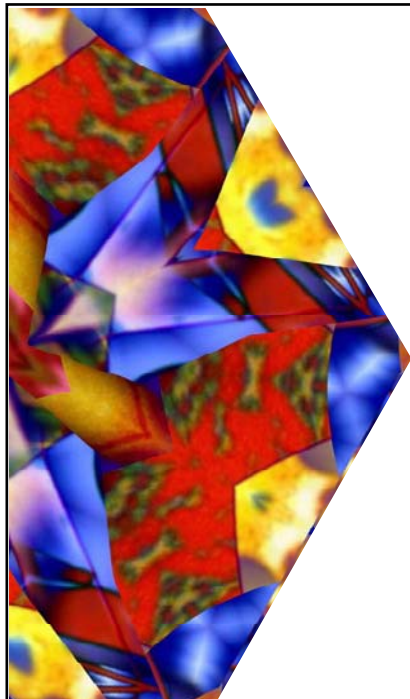
$\$transparent_3_height = \$transparent_2_height;$

convert -size $\{\$transparent_3_width\} \times \{\$transparent_3_height\}$ xc:none transparent_3.png

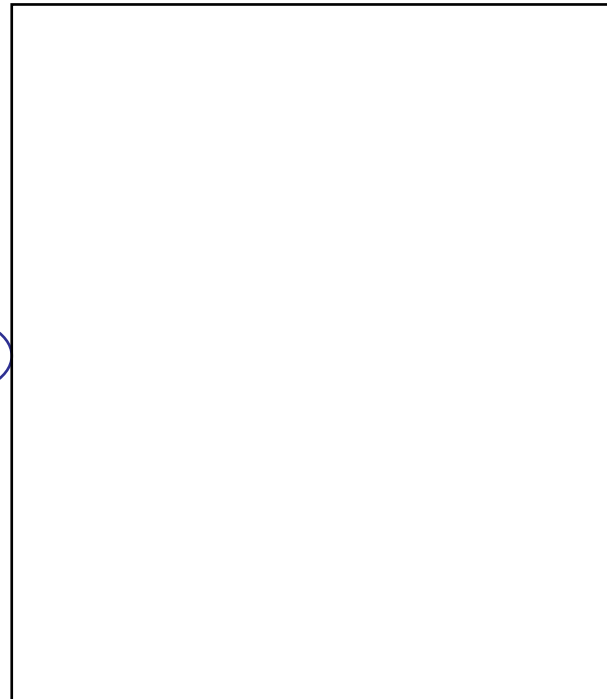
composite p31m_tile_center_right.png transparent_3.png -gravity West -compose Dst_Over
-background none +repage p31m_tile_right_1.png



p31m_tile_center_right

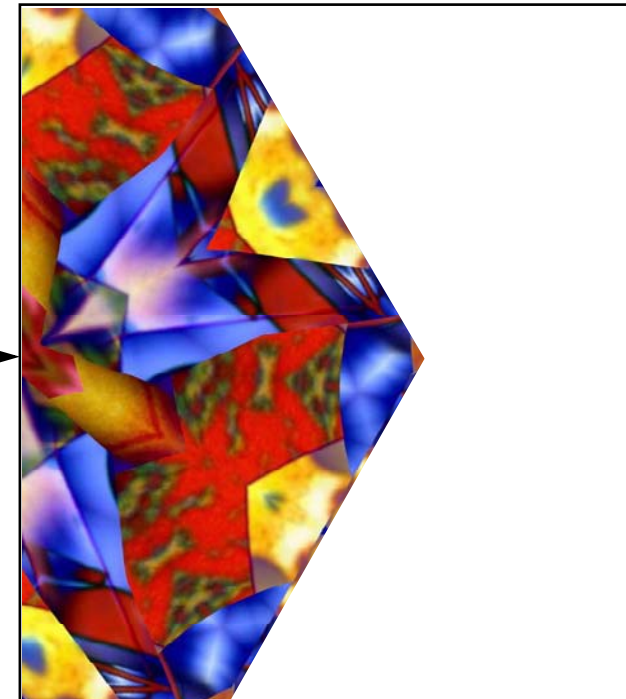


transparent_3



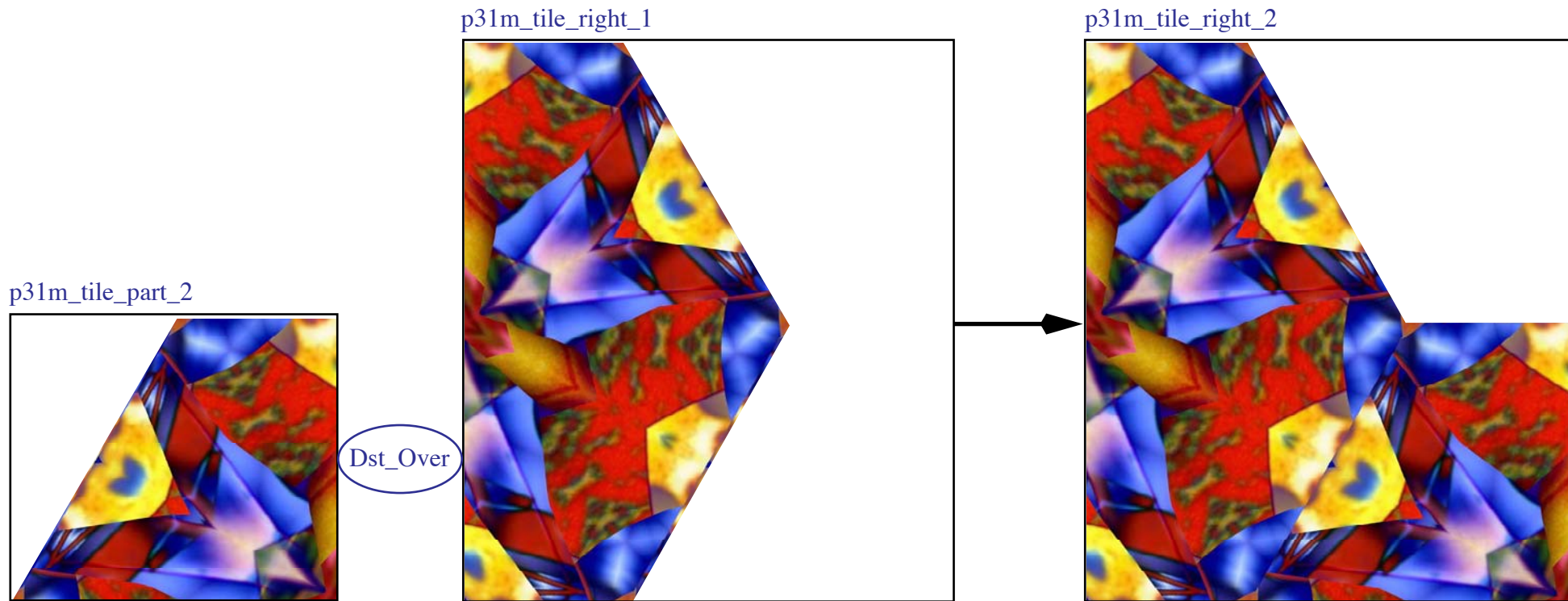
Dst_Over

p31m_tile_right_1



7.2) Generate tile_right_2: tile_part_2+tile_right_1

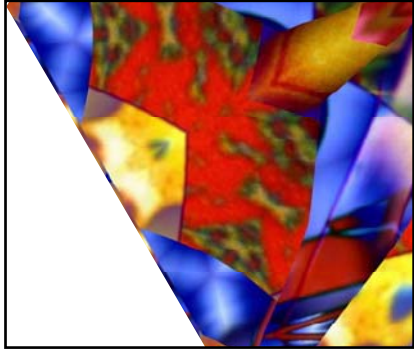
composite p31m_tile_part_2.png p31m_tile_right_1.png -gravity Southeast -compose Dst_Over -background none +repage p31m_tile_right_2.png



7.3) Generate tile_right: tile_part_3+tile_right_2

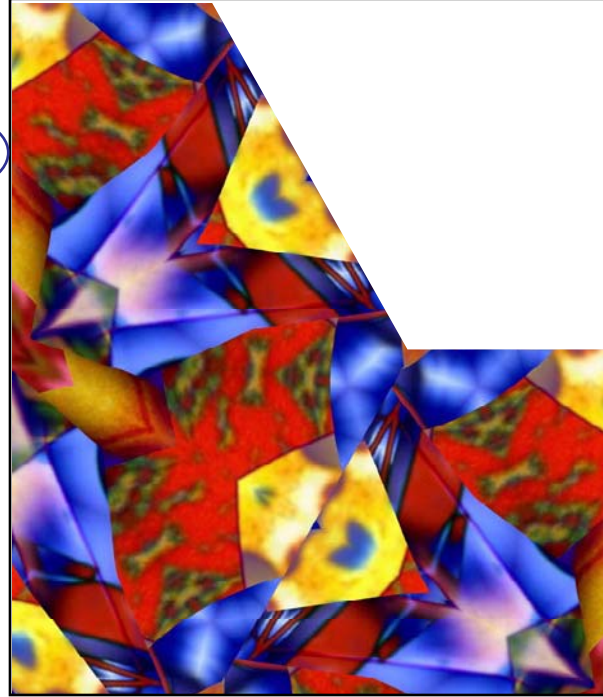
composite p31m_tile_part_3.png p31m_tile_right_2.png -gravity Northeast -compose Dst_Over -background none +repage p31m_tile_right.png

p31m_tile_part_3

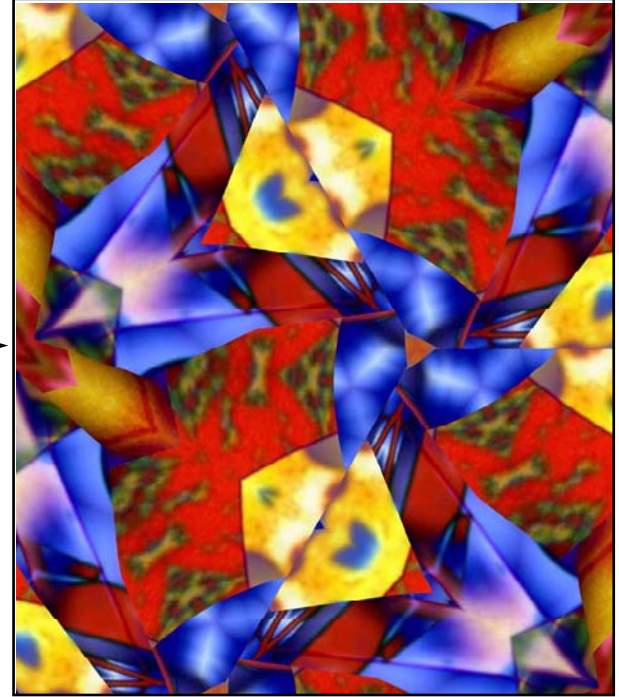


Dst_Over

p31m_tile_right_2



p31m_tile_right

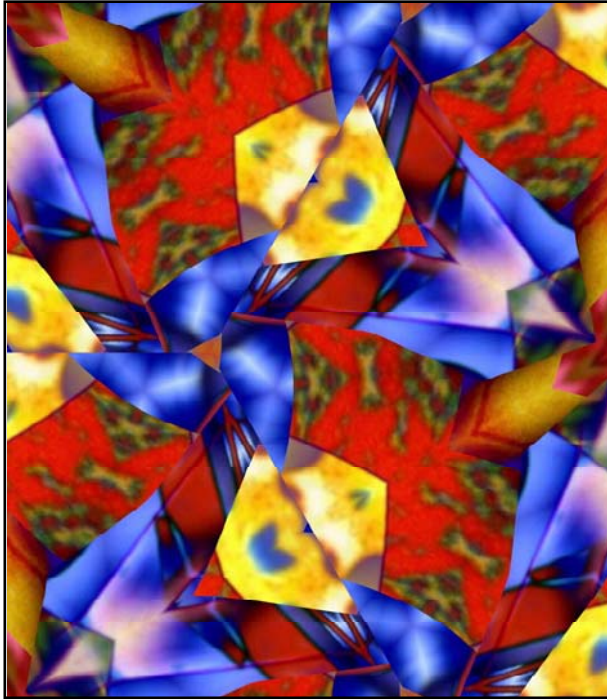


8) Generate tile

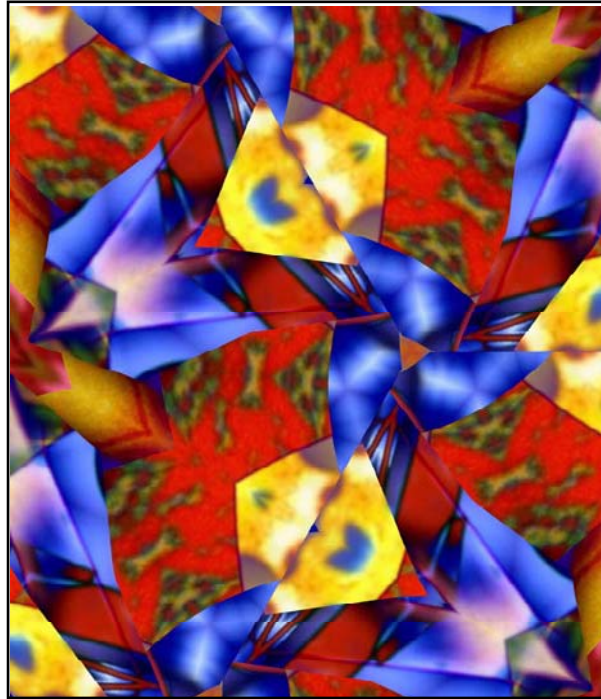
```
convert p31m_tile_right.png -flop p31m_tile_left.png
```

```
convert p31m_tile_left.png p31m_tile_right.png +append p31m_tile.png
```

p31m_tile_left



p31m_tile_right



p31m_tile

+append →



IM command sequence for p31m in general

1) Generate G1

1.1) Generate mask p31m_mask.png

```
$BE = int(1/2 * $r_height)+1;  
$t_a = int(1/sqrt(3) * $r_height)+1;  
convert -size {$r_width}x{$r_height} xc:none -fill red -draw "polyline 0,0 $r_width,$BE $t_a,$r_height 0,$r_height" p31m_mask.png
```

1.2) Copy from image.jpg a (random) rectangle

```
convert image.jpg -crop {$r_width}x{$r_height}+$x1+$y1 -background none +repage image_p31m.png
```

1.3) Generate p31m basic element p31m_G1

```
composite p31m_mask.png image_p31m.png -matte -compose Dst_In -background none +repage p31m_G1.png
```

2) Generate G2, G3

2.1) Generate with G1 the basic element G2

```
convert p31m_G1.png -flop -rotate -120 -background none +repage p31m_G2a.png  
$t_c = int(2/sqrt(3) * $r_height)+1;  
$t_height = int(1/2 * $r_height)+1;  
$t_height_double = 2 * $t_height;  
convert p31m_G2a.png -crop {$t_c}x{$t_height_double}+0+0 +repage p31m_G2.png
```

2.2) Generate with G1 the basic element G3

```
convert p31m_G1.png -rotate -120 p31m_G3a.png  
$p31m_G3a_width = $p31m_G3a->get('columns');  
$p31m_G3a_height = $p31m_G3a->get('rows');  
$DB = int(3/2 * $t_a)+1;  
$d = $p31m_G3a_width - $DB;  
$h = $p31m_G3a_height - $r_height;  
convert p31m_G3a.png -crop {$DB}x{$r_height}+$d+$h +repage p31m_G3.png
```

3) Generate tile_center_right: G1+G2+G3

3.1) Generate tile_center_right_1: G1+transparent_1

```
$transparent_1_width = $p31m_G2->Get('columns');  
$transparent_1_height = $r_height + $t_height;  
convert -size {$transparent_1_width}x{$transparent_1_height} xc:none transparent_1.png  
composite p31m_G1.png transparent_1.png -gravity Southwest -compose Dst_Over -background none +repage p31m_tile_center_right_1.png
```

3.2) Generate tile_center_right_2: tile_center_right_1+G2

```
composite p31m_G2.png p31m_tile_center_right_1.png -gravity Northwest -compose Dst_Over -background none +repage p31m_tile_center_right_2.png
```

3.3) Generate tile_center_right_3: tile_center_right_2+transparent_2
\$BC = int(1/2 * \$r_height)+1;
\$transparent_2_width = \$transparent_1_width;
\$transparent_2_height = \$transparent_1_height + \$BC;
convert -size {\$transparent_2_width}x{\$transparent_2_height} xc:none transparent_2.png
composite p31m_tile_center_right_2.png transparent_2.png -gravity Southwest -compose Dst_Over -background none +repage p31m_tile_center_right_3.png

3.4) Generate tile_center_right: tile_center_right_3+G3
composite p31m_G3.png p31m_tile_center_right_3.png -gravity Northwest -compose Dst_Over -background none +repage p31m_tile_center_right.png

4) Generate tile_center_left
convert p31m_tile_center_right.png -flop p31m_tile_center_left.png

5) Generate tile_part_2: G6+1/2G5
convert p31m_tile_center_left.png -crop {\$t_c}x{\$r_height}+0+0 -background none +repage p31m_tile_part_2.png

6) Generate tile_part_3: 1/2G5+G4
convert p31m_tile_center_left.png -crop {\$t_c}x{\$r_height}+0+{\$r_height} -background none +repage p31m_tile_part_3.png

7) Generate tile_right: center_right+part_2+part_3

7.1) Generate tile_right_1: tile_center_right+transparent_3
\$transparent_3_width = \$transparent_2_width + \$t_a;
\$transparent_3_height = \$transparent_2_height;
convert -size {\$transparent_3_width}x{\$transparent_3_height} xc:none transparent_3.png
composite p31m_tile_center_right.png transparent_3.png -gravity West -compose Dst_Over -background none +repage p31m_tile_right_1.png

7.2) Generate tile_right_2: tile_part_2+tile_right_1
composite p31m_tile_part_2.png p31m_tile_right_1.png -gravity Southeast -compose Dst_Over -background none +repage p31m_tile_right_2.png

7.3) Generate tile_right: tile_part_3+tile_right_2
composite p31m_tile_part_3.png p31m_tile_right_2.png -gravity Northeast -compose Dst_Over -background none +repage p31m_tile_right.png

8) Generate tile
convert p31m_tile_right.png -flop p31m_tile_left.png
convert p31m_tile_left.png p31m_tile_right.png +append p31m_tile.png

```
use Image::Magick;  
use diagnostics;  
use File::Basename;
```

```
# 0)Input
```

```
@suffixes = ('.jpg', '.jp2', '.png');
```

```
# 0.1) counting variable, folders, file  
$p = 1; #counting variable  
$sourcefolder_path = "C:/ART/IM_PlaneCovering/1_Sourceimages/";  
$sourcefile_name = 'image.jpg';  
#$sourcefile_name = 'logo.png';  
$p3m1_resultfolder_path = "C:/ART/IM_PlaneCovering/2_Tiles/p31m-Tiles/";
```

```
# 0.2) definition of lokal parameters  
my $r_height_min_Faktor = 0.3;  
my $r_height_max_Faktor = 0.5;
```

```
# 1) definition of PerlMagick objects  
$image_p31m = new Image::Magick;  
$p31m_mask = new Image::Magick;  
$transparent_1 = new Image::Magick;
```



```
$transparent_2 = new Image::Magick;  
$transparent_3 = new Image::Magick;
```

```
# 2) inilisation of random number generator  
srand;
```

```
# 3) open the source image  
my $sourcefile_fullname = $sourcefolder_path . $sourcefile_name;  
$image_p31m->Read("$sourcefile_fullname");
```

```
# 4) local parameter calculations: $triangle_width, $triangle_height, $x1, $y1  
my $image_width = $image_p31m->Get('columns');  
my $image_height = $image_p31m->Get('rows');
```

```
if ($image_width < $image_height) {my $image_min = $image_width} else {$image_min = $image_height};
```

```
my $r_height_min = $r_height_min_Faktor * $image_min;  
my $r_height_max = $r_height_max_Faktor * $image_min;
```

```
$r_height = $r_height_min + (int(rand($r_height_max - $r_height_min)) + 1);  
$r_width = int(1/2 * sqrt(3) * $r_height)+1;
```

```
my $x_allowed = $image_width - $r_width;  
my $y_allowed = $image_height - $r_height;  
my $x1 = int(rand($x_allowed)) + 1;
```

```
my $y1 = int(rand($y_allowed)) + 1;
```

```
#print "$image_width $image_height $r_height_min $r_height_max $r_height $r_width $x_allowed $y_allowed $x1 $y1";
```

```
# 5) Generate mask p31m_mask
```

```
#$BE = int(1/2 * $r_height)+1;
```

```
#$t_a = int(1/sqrt(3) * $r_height)+1;
```

```
#convert -size {$r_width}x{$r_height} xc:none -fill red -draw "polyline 0,0 $r_width,$BE $t_a,$r_height 0,$r_height" p31m_mask.png
```

```
my $BE = int(1/2 * $r_height)+1;
```

```
my $t_a = int(1/sqrt(3) * $r_height)+1;
```

```
$p31m_mask->Set(size=>"$r_width x $r_height");
```

```
$p31m_mask->Read('xc:none');
```

```
$p31m_mask->Draw(primitive=>'polyline', points=>"0,0 $r_width,$BE $t_a,$r_height 0,$r_height", fill=>'red');
```

```
$mask_width = $p31m_mask->get('columns');
```

```
$mask_height = $p31m_mask->get('rows');
```

```
#$p31m_mask->Write(filename=>'p31m_mask.jpg', compression=>'JPEG', quality=>'95');
```

6) Generate image_p3

#convert image.jpg -crop \${r_width}x\${r_height}+\${x1}+\${y1} -background none +repage image_p31m.png

\$image_p31m->Crop(geometry=>"\$mask_width x \$mask_height+\${x1}+\${y1}");

\$image_p31m->Set(page=>'0x0+0+0');

#\$image_p31m->Write(filename=>'image_p31m.jpg', compression=>'JPEG', quality=>'95');

#7) Generate p31m basic element p31m_G1

#composite p31m_mask.png image_p31m.png -matte -compose Dst_In -background none +repage p31m_G1.png

\$p31m_G1 = \$p31m_mask->Clone();

\$p31m_G1->Composite(image=>\$image_p31m, compose=>in, color=>'transparent', matte=>'true');

#\$p31m_G1->Write(filename=>'p31m_G1.jpg', compression=>'JPEG', quality=>'95');

#8) Generate G2, G3

#8.1) Generate with G1 the basic element G2

```
#convert p3_G1.png -flop -rotate -120 -background none +repage p31m_G2a.png
#$t_c = int(2/sqrt(3) * $r_height)+1;
#$t_height = int(1/2 * $r_height)+1;
#$t_height_double = 2 * $t_height;
#convert p31m_G2a.png -crop {$t_c}x{$t_height_double}+0+0 +repage p31m_G2.png
```

```
$p31m_G2 = $p31m_G1->Clone();
$p31m_G2->Flop();
$p31m_G2->Rotate(degrees=>-120, color=>'transparent');
```

```
#$p31m_G2->Write(filename=>'p31m_G2a.jpg', compression=>'JPEG', quality=>'95');
```

```
##my $p31m_G2a_width = $p31m_G2->get('columns');
##my $p31m_G2a_height = $p31m_G2->get('rows');
```

```
my $t_c = int(2/sqrt(3) * $r_height)+1;
my $t_height = int(1/2 * $r_height)+1;
my $t_height_double = 2 * $t_height;
```

```
$p31m_G2->Crop(geometry=>"$t_c x $t_height_double +0+0", background=>'transparent');
$p31m_G2->Set(page=>'0x0+0+0');
```

```
#$p31m_G2->Write(filename=>'p31m_G2.jpg', compression=>'JPEG', quality=>'95');
```

```
#8.2) Generate with G1 the basic element G3
#convert p31m_G1.png -rotate -120 p31m_G3a.png
#$p31m_G3a_width = $p31m_G3a->get('columns');
#$p31m_G3a_height = $p31m_G3a->get('rows');
#$DB = int(3/2 * $t_a)+1;
#$d = $p31m_G3a_width - $DB;
#$h = $p31m_G3a_height - $r_height;
#convert p3_G3a.png -crop {$DB}x{$r_height}+$d+$h +repage p3_G3.png
```

```
$p31m_G3 = $p31m_G1->Clone();
$p31m_G3->Rotate(degrees=>-120, color=>'transparent');
```

```
#$p31m_G3->Write(filename=>'p31m_G3a.jpg', compression=>'JPEG', quality=>'95');
```

```
$p31m_G3a_width = $p31m_G3->get('columns');
$p31m_G3a_height = $p31m_G3->get('rows');
```

```
my $DB = int(3/2 * $t_a)+1;
my $d = $p31m_G3a_width - $DB;
my $h = $p31m_G3a_height - $r_height;
```

```
$p31m_G3->Crop(geometry=>"$DB x $r_height +$d+$h", background=>'transparent');
$p31m_G3->Set(page=>'0x0+0+0');
```

```
#$p31m_G3->Write(filename=>'p31m_G3.jpg', compression=>'JPEG', quality=>'95');
```

```
#9) Generate tile_center_right: G1+G2+G3
#9.1) Generate tile_center_right_1: G1+transparent_1
#$transparent_1_width = $p31m_G2->Get('columns');
#$transparent_1_height = $r_height + $t_height;
#convert -size {$transparent_1_width}x{$transparent_1_height} xc:none transparent_1.png
#composite p31m_G1.png transparent_1.png -gravity Southwest -compose Dst_Over -background none +repage p31m_tile_center_right_1.png
```

```
my $transparent_1_width = $p31m_G2->Get('columns');
my $transparent_1_height = $r_height + $t_height;
```

```
$transparent_1->Set(size=>"$transparent_1_width x $transparent_1_height");
$transparent_1->Read('xc:none');
```

```
#$transparent_1->Write(filename=>'transparent_1.jpg', compression=>'JPEG', quality=>'95');
```

```
my $p31m_tile_center_right_1 = $transparent_1->Clone();
$p31m_tile_center_right_1->Composite(image=>$p31m_G1, gravity=>Southwest, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_center_right_1->Write(filename=>'p31m_tile_center_right_1.jpg', compression=>'JPEG', quality=>'95');
```


#9.2) Generate tile_center_right_2: tile_center_right_1+G2

```
#composite p31m_G2.png p31m_tile_center_right_1.png -gravity Northwest -compose Dst_Over -background none +repage  
p31m_tile_center_right_2.png
```

```
$p31m_tile_center_right_2 = $p31m_tile_center_right_1->Clone();
```

```
$p31m_tile_center_right_2->Composite(image=>$p31m_G2, gravity=>Northwest, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_center_right_2->Write(filename=>'p31m_tile_center_right_2.jpg', compression=>'JPEG', quality=>'95');
```

#9.3) Generate tile_center_right_3: tile_center_right_2+transparent_2

```
#$BC = int(1/2 * $r_height)+1;
```

```
#$transparent_2_width = $transparent_1_width;
```

```
#$transparent_2_height = $transparent_1_height + $BC;
```

```
#convert -size {$transparent_2_width}x{$transparent_2_height} xc:none transparent_2.png
```

```
#composite p31m_tile_center_right_2.png transparent_2.png -gravity Southwest -compose Dst_Over -background none +repage
```

```
p31m_tile_center_right_3.png
```

```
my $BC = int(1/2 * $r_height)+1;  
my $transparent_2_width = $transparent_1_width;  
my $transparent_2_height = $transparent_1_height + $BC;  
$transparent_2->Set(size=>"$transparent_2_width x $transparent_2_height");  
$transparent_2->Read('xc:none');
```

```
#$transparent_2->Write(filename=>'transparent_2.jpg', compression=>'JPEG', quality=>'95');
```

```
my $p31m_tile_center_right_3 = $transparent_2->Clone();  
$p31m_tile_center_right_3->Composite(image=>$p31m_tile_center_right_2, gravity=>Southwest, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_center_right_3->Write(filename=>'p31m_tile_center_right_3.jpg', compression=>'JPEG', quality=>'95');
```

#9.4) Generate tile_center_right: tile_center_right_3+G3

```
#composite p31m_G3.png p31m_tile_center_right_3.png -gravity Northwest -compose Dst_Over -background none +repage  
p31m_tile_center_right.png
```

```
my $p31m_tile_center_right = $p31m_tile_center_right_3->Clone();  
$p31m_tile_center_right->Composite(image=>$p31m_G3, gravity=>Northwest, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_center_right->Write(filename=>'p31m_tile_center_right.jpg', compression=>'JPEG', quality=>'95');
```

```
#10) Generate tile_center_left  
# convert p31m_tile_center_right.png -flop p31m_tile_center_left.png
```

```
my $p31m_tile_center_left = $p31m_tile_center_right->Clone();  
$p31m_tile_center_left->Flop();
```

```
#$p31m_tile_center_left->Write(filename=>'p31m_tile_center_left.jpg', compression=>'JPEG', quality=>'95');
```

#11) Generate tile_part_2: $G6 + 1/2G5$

```
#convert p31m_tile_center_left.png -crop ${t_c}x${r_height}+0+0 -background none +repage p31m_tile_part_2.png
```

```
my $p31m_tile_part_2 = $p31m_tile_center_left->Clone();
```

```
my $p31m_tile_part_2_width = $t_c;
```

```
my $p31m_tile_part_2_height = $r_height + 1;
```

```
$p31m_tile_part_2->Crop(geometry=>"$p31m_tile_part_2_width x $p31m_tile_part_2_height +0+0", background=>'transparent');
```

```
$p31m_tile_part_2->Set(page=>'0x0+0+0');
```

```
#$p31m_tile_part_2->Write(filename=>'p31m_tile_part_2.jpg', compression=>'JPEG', quality=>'95');
```

#12) Generate tile_part_3: $1/2G5 + G4$

```
#convert p31m_tile_center_left.png -crop ${t_c}x${r_height}+0+${r_height} -background none +repage p31m_tile_part_3.png
```

```
my $p31m_tile_part_3 = $p31m_tile_center_left->Clone();
```

```
my $p31m_tile_part_3_width = $t_c;
```

```
my $p31m_tile_part_3_height = $r_height;
```

```
my $p31m_tile_part_3_y = $r_height - 1;
```

```
$p31m_tile_part_3->Crop(geometry=>"$p31m_tile_part_3_width x $p31m_tile_part_3_height +0 +$p31m_tile_part_3_y", background=>'transparent');  
$p31m_tile_part_3->Set(page=>'0x0+0+0');  
  
#$p31m_tile_part_3->Write(filename=>'p31m_tile_part_3.jpg', compression=>'JPEG', quality=>'95');
```

```
#13) Generate tile_right: center_right+part_2+part_3  
#13.1) Generate tile_right_1: tile_center_right+transparent_3  
#$transparent_3_width = $transparent_2_width + $t_a;  
#$transparent_3_height = $transparent_2_height;  
#convert -size {$transparent_3_width}x{$transparent_3_height} xc:none transparent_3.png  
#composite p31m_tile_center_right.png transparent_3.png -gravity West -compose Dst_Over -background none +repage p31m_tile_right_1.png
```

```
my $transparent_3_width = $transparent_2_width + $t_a;  
my $transparent_3_height = $transparent_2_height;  
$transparent_3->Set(size=>"$transparent_3_width x $transparent_3_height");  
$transparent_3->Read('xc:none');
```

```
#$transparent_3->Write(filename=>'transparent_3.jpg', compression=>'JPEG', quality=>'95');
```

```
my $p31m_tile_right_1 = $transparent_3->Clone();  
$p31m_tile_right_1->Composite(image=>$p31m_tile_center_right, gravity=>West, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_right_1->Write(filename=>'p31m_tile_right_1.jpg', compression=>'JPEG', quality=>'95');
```

```
#13.2) Generate tile_right_2: tile_part_2+tile_right_1  
#composite p31m_tile_part_2.png p31m_tile_right_1.png -gravity Southeast -compose Dst_Over -background none +repage  
p31m_tile_right_2.png
```

```
my $p31m_tile_right_2 = $p31m_tile_right_1->Clone();  
$p31m_tile_right_2->Composite(image=>$p31m_tile_part_2, gravity=>Southeast, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_right_2->Write(filename=>'p31m_tile_right_2.jpg', compression=>'JPEG', quality=>'95');
```

```
#13.3) Generate tile_right: tile_part_3+tile_right_2  
#composite p31m_tile_part_3.png p31m_tile_right_2.png -gravity Northeast -compose Dst_Over -background none +repage p31m_tile_right.png
```

```
my $p31m_tile_right = $p31m_tile_right_2->Clone();  
$p31m_tile_right->Composite(image=>$p31m_tile_part_3, gravity=>Northeast, compose=>'over', color=>'transparent', matte=>'true');
```

```
#$p31m_tile_right->Write(filename=>'p31m_tile_right.jpg', compression=>'JPEG', quality=>'95');
```

#14) Generate tile

```
#convert p31m_tile_right.png -flop p31m_tile_left.png  
#convert p31m_tile_left.png p31m_tile_right.png +append p31m_tile.png
```

```
my $p31m_tile_left = $p31m_tile_right->Clone();  
$p31m_tile_left->Flop();
```

```
my $q = $p31m_tile_left->Clone();  
push(@$q, $p31m_tile_right);  
my $p31m_tile = $q->Append(stack=>'false');
```

```
#$p31m_tile->Write(filename=>'p31m_tile.jpg', compression=>'JPEG', quality=>'95');
```

15) make comment with parameter values

```
my $comment_string = $sourcefile_name . "*" . $r_height . "*" . $x1 . "*" . $y1 . "*";
```

```
my $p31m_comment = $p31m_tile->Comment($comment_string);
```

16) generate name of result image

```
my($filename) = fileparse($sourcefile_fullname, @suffixes);
```

```
if ($p <= 9) {$resultimage_name = "p31m_tile_" . $filename . "-0" . $p . ".jpg"};
```

```
if ($p > 9 and $p <= 99) {$resultimage_name = "p31m_tile_" . $filename . "-" . $p . ".jpg"};
```

```
$resultimagepath_name = $p3_resultfolder_path . $resultimage_name;
```

16) save result image

```
$p31m_tile->Write(filename=>"$resultimagepath_name", compression=>'JPEG', quality=>'95');
```