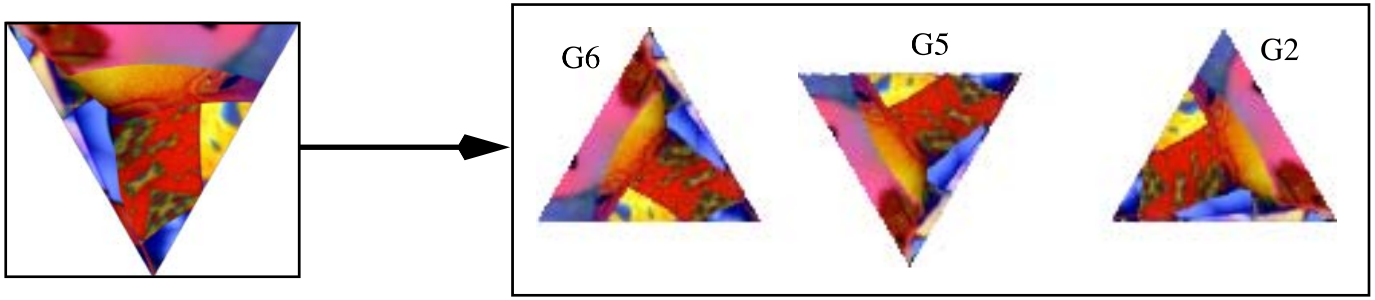
Draft V 0.9

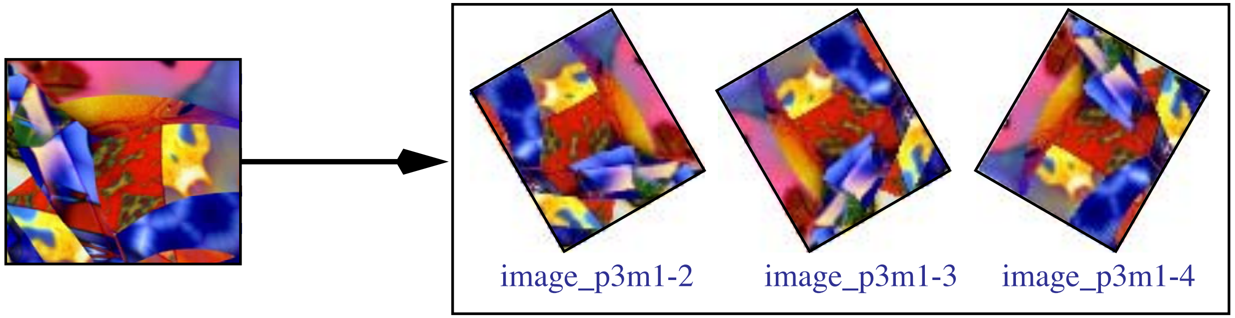
Seamless Plane Groups with ImageMagick/PerlMagick: 1.3) p3m1_tile

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

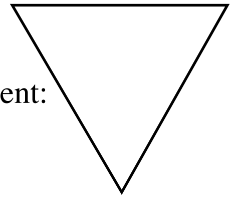
Version 1:
rotation of the selected basic element G1



Version 2:
rotation of the selected image part



p3m1 basic element:



<u>given in the deterministic case</u>

a) image.jpg with (image_width, image_height)
  example: image_width = image_height: 4000 [pixel]

b) format of the basic element: triangle_width
  example: = triangle_width1501 [pixel]

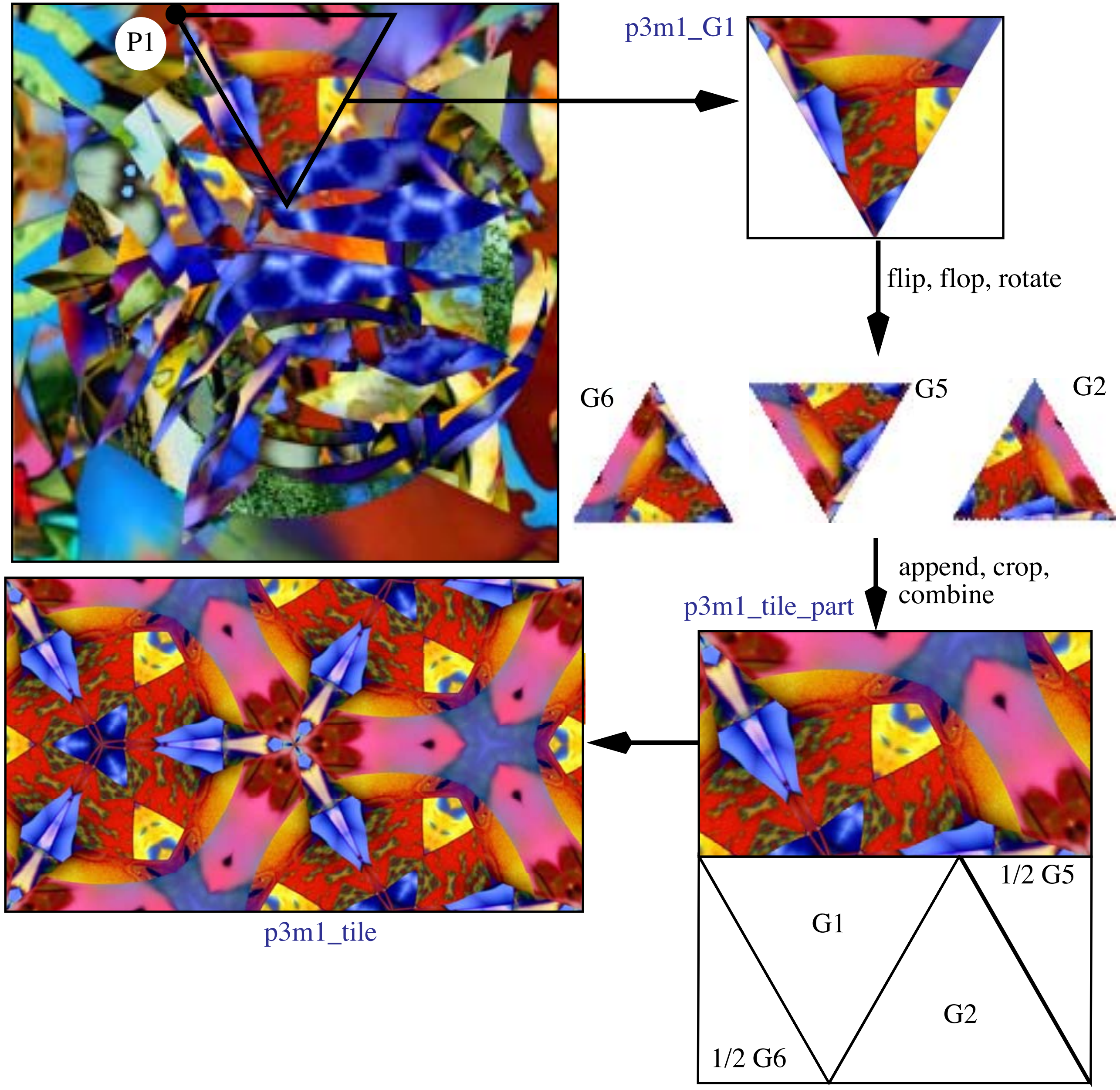
c) Top left point P1 in image.jpg for selection of image_p3m1
  example: P1 = (1241,113)

<u>given in the stochastic case</u>

a) image.jpg with (image_width, image_height)

b) random variable interval for the selection of triangle_width:
  [triangle_min_Faktor, triangle_max_Faktor] = [0.3, 0.7]

```
p3m1-procedure:
Start
    1) Generate G1
    2) Generate G6, G5, G2
    3) Generate p3m1_tile_part
    4) Generate tile
End
```



P1

p3m1_G1

flip, flop, rotate

G6

G5

G2

append, crop, combine

p3m1_tile_part

1/2 G5

G1

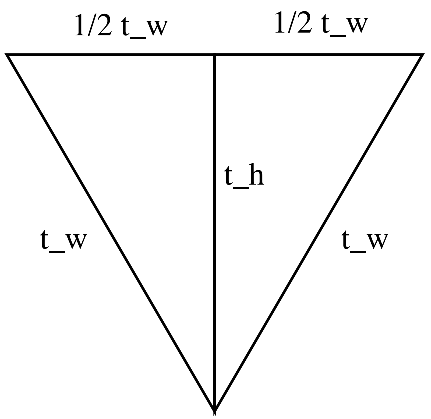
G2

1/2 G6

p3m1_tile

Only relevant for the stochastic case

```
#Calculate scalar triangle_width, triangle_height, x1, y1

$triangle_width_min_Faktor = 0.3;
$triangle_width_max_Faktor =0.7;
if ($image_width < $image_height) {$image_min = $image_width} else {$image_min = $image_height};
$triangle_width_min = $triangle_width_min_Faktor * $image_min;
$triangle_width_max = $triangle_width_max_Faktor * $image_min;
$triangle_width = $triangle_width_min + (int(rand($triangle_width_max - $triangle_width_min)) + 1);
if ($triangle_width % 2 == 0) { $triangle_width++ };
$triangle_height = int(1/ 2 * √3 * $triangle_width) + 1;
$x_allowed = $image_width - $triangle_width;
$y_allowed = $image_hight - $triangle_width;
$x1 = int(rand($x_allowed)) + 1;
$y1 = int(rand($y_allowed)) + 1;
```



$t\_w^2 = 1/4\ t\_w^2 + t\_h^2$

$\Longleftrightarrow t\_h^2 = 3/4 * t\_w^2$

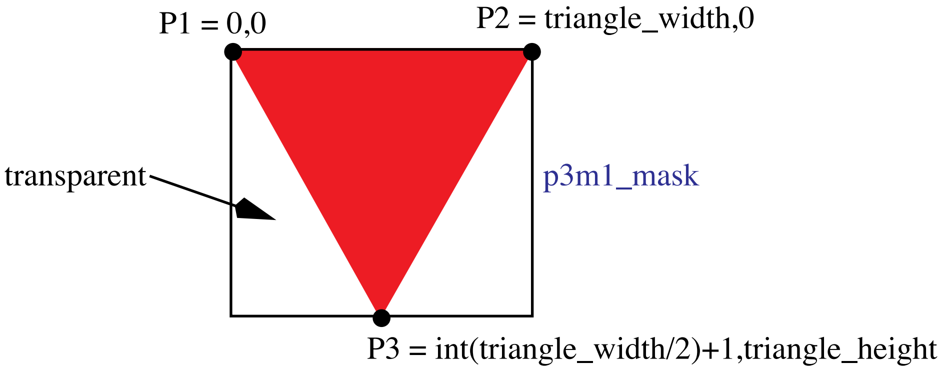
$\Longleftrightarrow t\_h = 1/2 * \sqrt{3} * t\_w$

Generate masks p3m1_mask1 on scratch:

```
        convert -size {$triangle_width}x{$triangle_height} xc:none -fill red -draw "polyline 0,0 $triangle_width,0 int(1/2 * $triangle_width)+1,$triangle_height"
                                                                            p3m1_mask.png
```

For the example holds:

```
        $triangle_height = int(1/ 2 * √3 * 1501) + 1 = 1300
        int(1/ 2 * $triangle_width) + 1 = 751

        convert -size 1501x1300 xc:none -fill red -draw “polyline 0,0 1501,0 751,1300” p3m1_mask.png
```

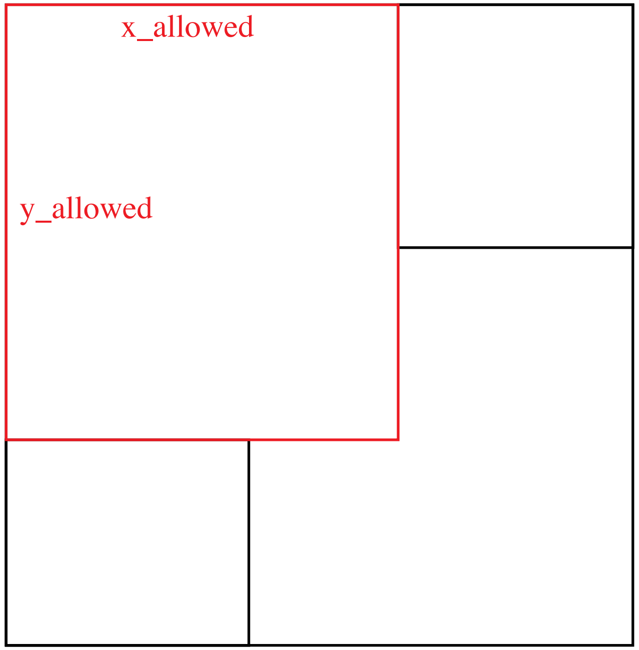


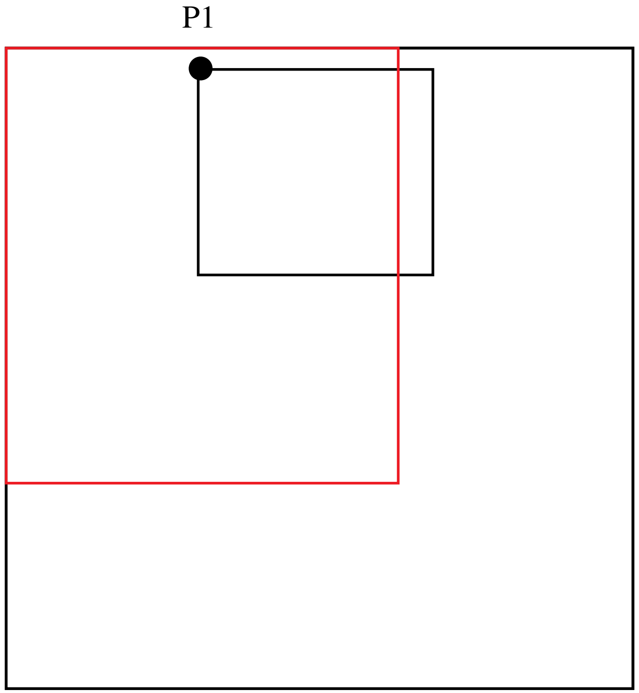
Copy from image.jpg a (random) rectancle with width = triangle_width and height = triangle_width

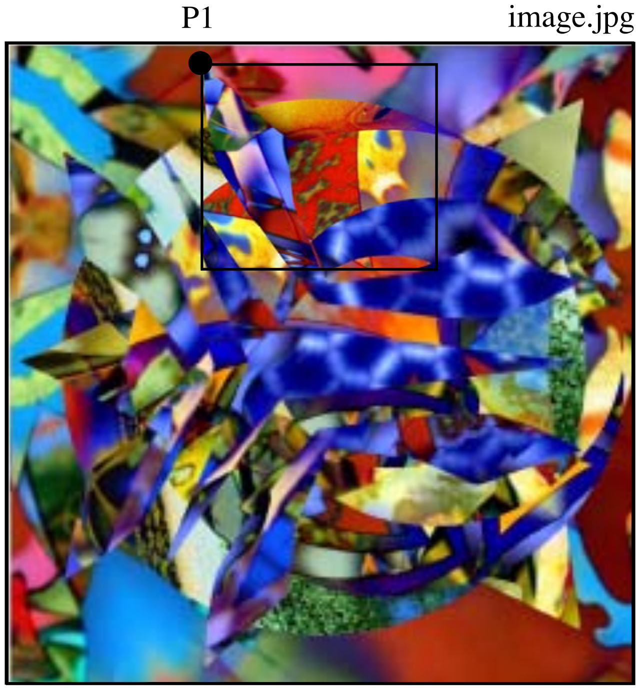
```
convert image.jpg -crop {$triangle_width}x{$triangle_width}+{$x1}+{$y1} +repage image_p3m1.png
```

For the example holds:

```
convert image.jpg -crop 1501x1300+1241+113 +repage image_p3m1.png
```



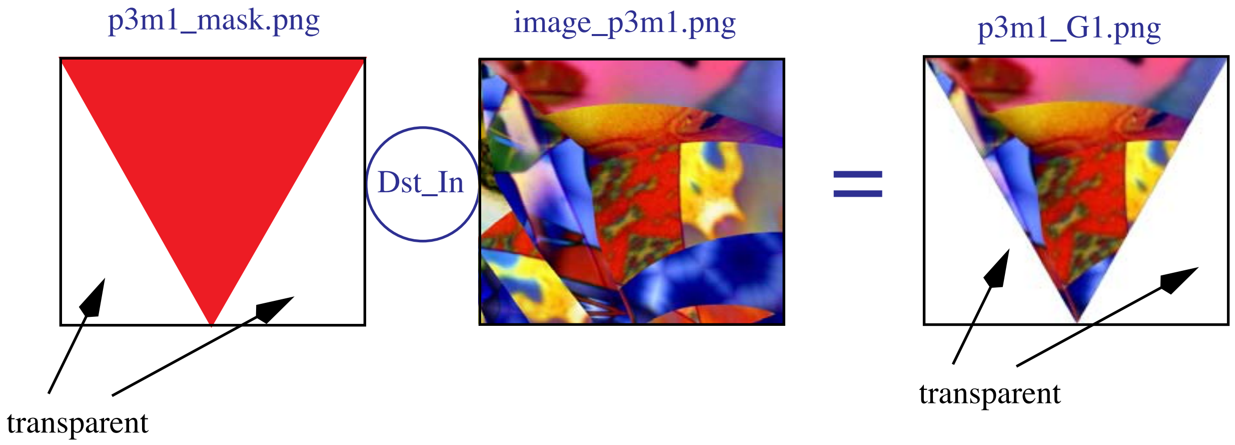




image_p3m1.png

Generate p3m1 basic element p3m1_G1

```
composite p3m1_mask.png image_p3m1.png -matte -compose Dst_In p3m1_G1.png
```



http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_p3m1/p3m1_G1.png

The crop command of the original image can be skiped and the composing with the mask can directly by -geometry with negative x and y coordinates:

```
composite image.jpg p3m1_mask.png -geometry -{$x1}-{$y1} -compose In p3m1_G1.png
```

For the example holds:

```
composite image.jpg p3m1_mask.png -geometry -1241-113 -compose In -background none +repage p3m1_G1.png
```

2) Generate G6, G5, G2

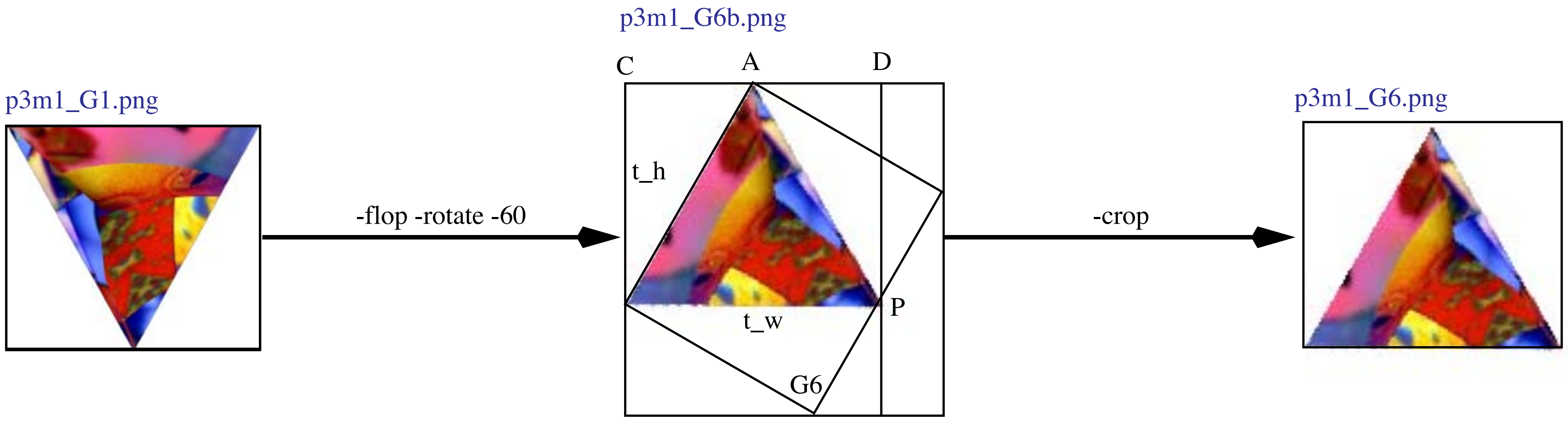
2.1) Generate with G1 the basic element G6

G6 is generated from G1 by a “-flop -rotate -60” command but rotation generates a larger image therefore an additional “-crop” command is necessary. The width $\overline{CD}$ of the rectangle for the crop command is triangle_width and $\overline{DP}$ is triangle_height. The rotation “-60” sets the triangle at the left side and at the top of the canvas therefore the point for the crop command is (0,0):

```
        convert p3m1_G1.png -flop -rotate -60 p3m1_G6b.png
        convert p3m1_G6b.png -crop {$triangle_width}x{$triangle_height}+0+0 +repage p3m1_G6.png
```

For the example holds:

```
        convert p3m1_G1.png -flop -rotate -60 p3m1_G6b.png
        convert -crop p3m1_G6b.png 1500x1300+0+0 +repage p3m1_G6.png
```



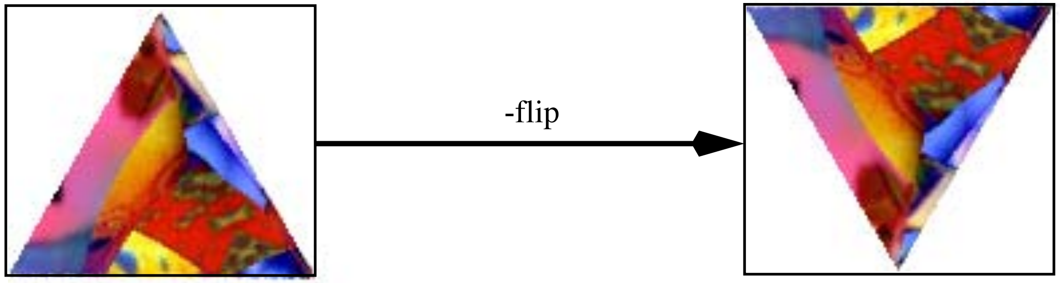
2.2) Generate with G6 the basic element G5

G5 is generated from G6 by a “-flip” command:

```
        convert p3m1_G6.png -flip  p3m1_G5.png
```

p3m1_G6.png

p3m1_G5.png



-flip

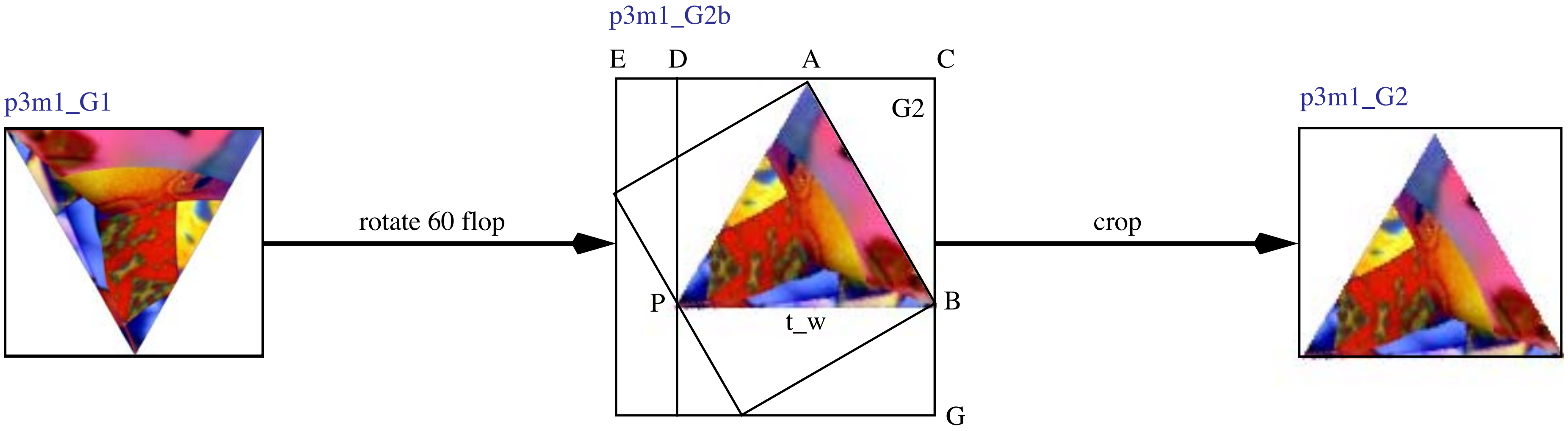
2.3) Generate with G1 the basic element G2

G2 is generated from G1 by a “-flop -rotate 60” command and an additional “-crop” command. The width $\overline{CD}$ of the rectangle for the crop command is triangle_width and $\overline{DP}$ is triangle_height. The rotation “60” sets the triangle at the right side and at the top of the canvas therefore the point for the crop command is (d = $\overline{DE}$,0) and the distance $d must be evaluated. The width p3m1_G2b_width of the rotated canvas can be directly evaluated by the PerlMagick command “Get(‘columns’)”, then $d can be calculated by p3m1_G2b_width - t_w:

```
convert p3m1_G1.png -flop -rotate 60 p3m1_G2b.png
$p3m1_G2b_width = $p3m1_G2b->Get(‘columns’);
$d = $p3m1_G2b_width - $triangle_width;
convert p3m1_G2b.png -crop {$triangle_width}x{$triangle_height}+$d+0 +repage p3m1_G2.png
```

For the example holds:

```
convert p3m1_G1.png -flop -rotate 60 p3m1_G2b.png
p3m1_G2b_width = 1876; t_w = 1501: $d = 1876 - 1501 = 375
convert p3m1_G2b.png -crop 1501x1300+375+0 p3m1_G2.png
```



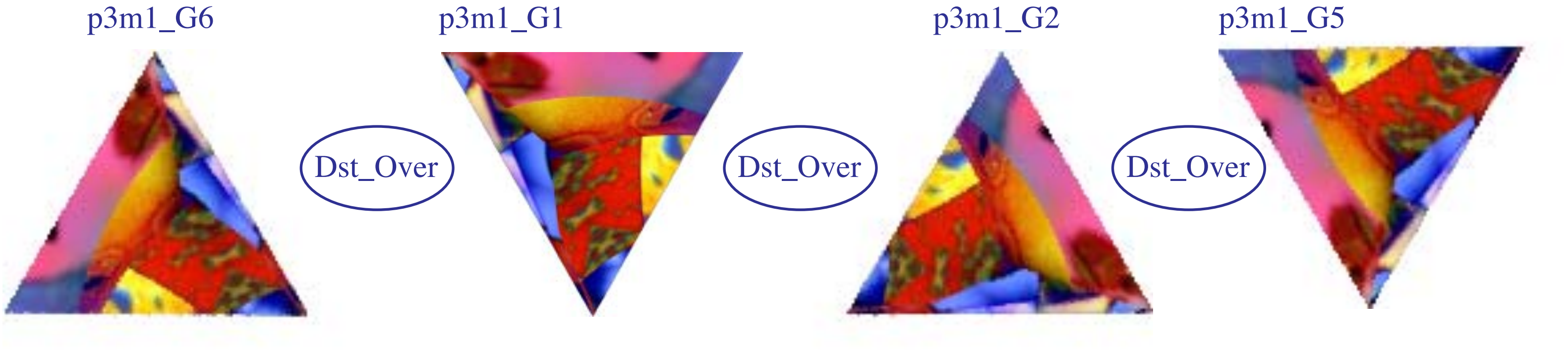
3) Generate p3m1_tile_part

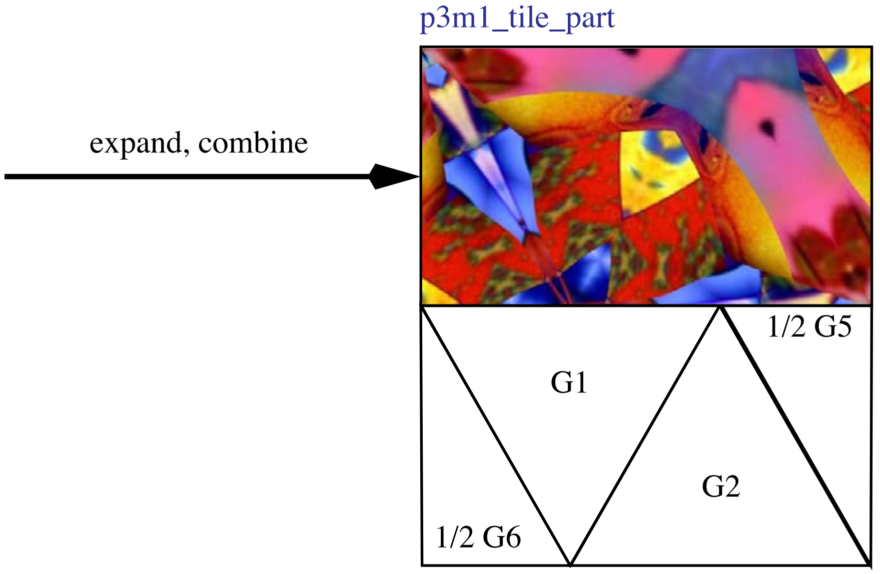
by successive insertion of basic elements G6,G1,G2;G5 with

a) crop G6 and insert 1/2 G6 in G1 at the left side: p3m1_tile_part

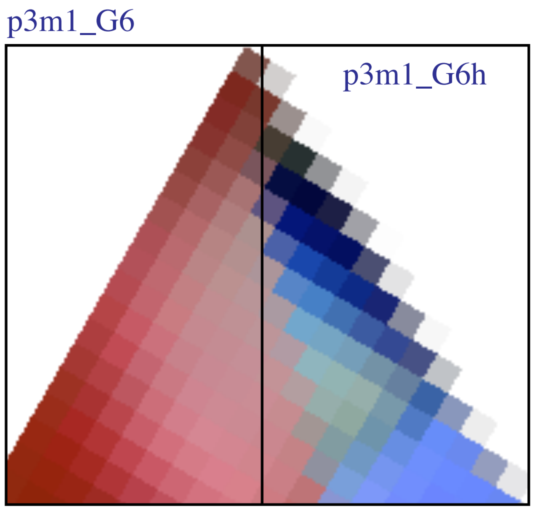
b) expand p3m1_tile_part right and insert G2 at the right side

c) crop G5, extend p3m1_tile_part right and insert 1/2 G5 in p3m1_tile_part





a) crop G6 and insert 1/2 G6 in G1 at the left side: p3m1_tile_part



triangle_width is an odd integer, so it must be decided what to do with the middle row of p3m1_G1. If the middle row should be part of p3m1_tile_part then the width of p3m1_G6 is after the crop command int(1/2*$triangle_width)+1 pixel (see image on the right side):

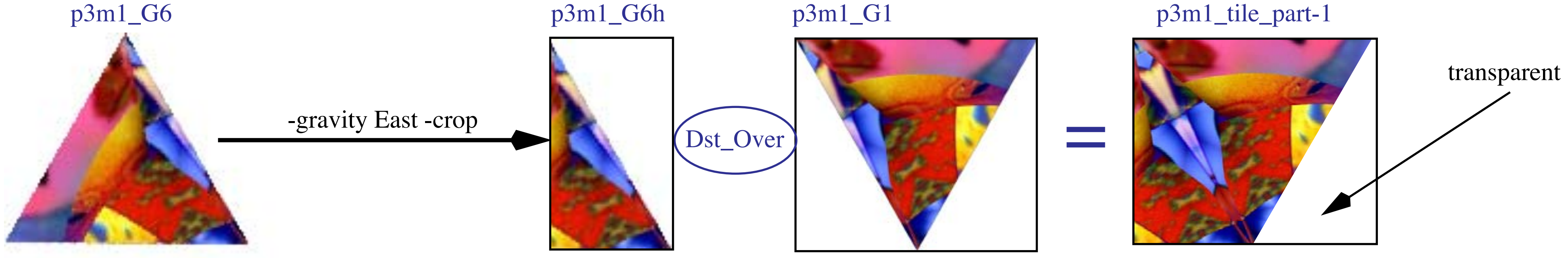
```
convert p3m1_G6.png -gravity East -crop (int(1/2 * $triangle_width))x{$triangle_height}+0+0 -background none
                                                    +repage p3m1_G6h.png
```

Combining the right half of p3m1_G6 over the left side of p3m1_G1:

```
composite p3m1_G6h.png p3m1_G1.png -gravity west -compose Dst_Over -background none +repage p3m1_tile_part-1.png
```

For the example holds:

```
convert p3m1_G6.png -gravity East -crop 750x1300+0+0 -background none +repage p3m1_G6half.png
composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
```

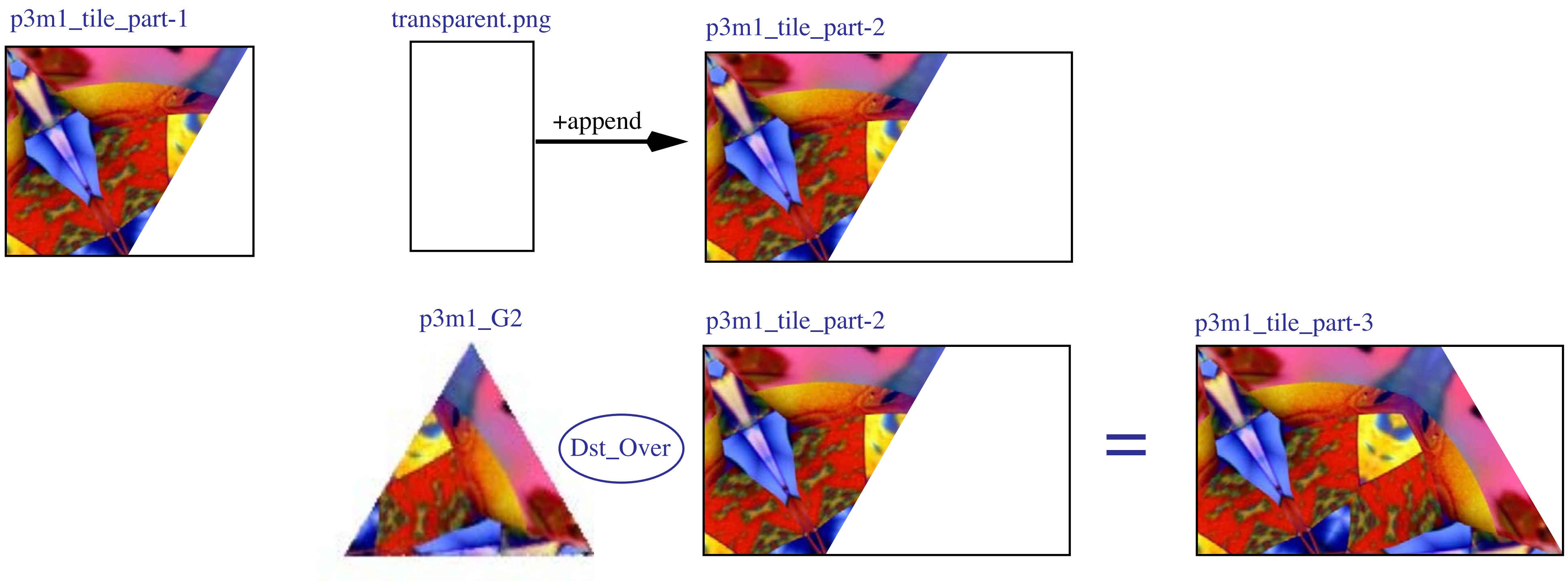


b) enlarge p3m1_tile_part on the right side with a transparent region (don’t use splice or extend because of problems with the PerlMagick environment) and insert G2 on the right side

```
convert -size (int(1/2 * $triangle_width) + 1)x$triangle_height xc:none transparent.png
convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over p3m1_tile_part-3.png
```

For the example holds:

```
convert -size 751x1300 xc:none transparent.png
convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over p3m1_tile_part-3.png
```

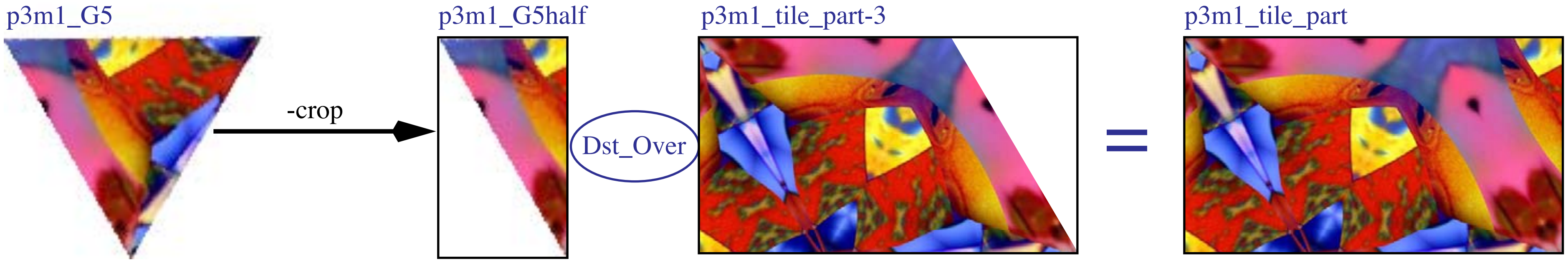


c) crop G5 and insert 1/2 G5 in p3m1_tile_part-3

```
convert p3m1_G5.png -crop {int(1/2 * $triangle_width) + 1}x{$triangle_height}+0+0 +repage p3m1_G5half.png
composite p3m1_G5half.png p3m1_tile_part-3.png -gravity East -compose Dst_Over p3m1_tile_part.png
```

For the example holds:

```
convert p3m1_G5.png -crop 751x1300+0+0 +repage p3m1_G5half.png
composite p3m1_G5half.png p3m1_tile_part-3.png -gravity East -compose Dst_Over p3m1_tile_part.png
```



Alternative for 3) Generate p3m1_tile_part

```
convert -size 2250x1300 xc: \
            -page -750+0  p3m1_G6.png \
            -page +0+0    p3m1_G1.png \
            -page +750+0  p3m1_G2.png \
            -page +1500+0 p3m1_G5.png \
            -flatten p3m1_tile_part.png
```

```
convert -size 2250x1300 xc: -page -750+0  p3m1_G6.png -page +0+0    p3m1_G1.png -page +750+0  p3m1_G2.png -page +1500+0 p3m1_G5.png -flatten
p3m1_tile_part.png
```

in the general case:

```
convert -size 2250x1300 xc: \
            -page -750+0  p3m1_G6.png \
            -page +0+0    p3m1_G1.png \
            -page +750+0  p3m1_G2.png \
            -page +1500+0 p3m1_G5.png \
            -flatten p3m1_tile_part.png
```

```
convert -size 2250x1300 xc: -page -750+0  p3m1_G6.png -page +0+0    p3m1_G1.png -page +750+0  p3m1_G2.png -page +1500+0 p3m1_G5.png -flatten
p3m1_tile_part.png
```

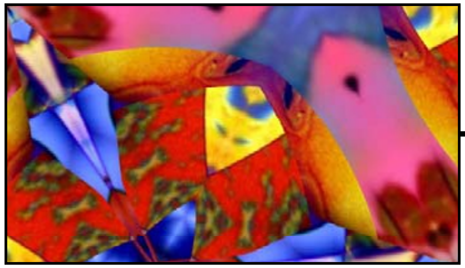
```
         convert p3m1_G5.png -crop {int(1/2 * $triangle_width) + 1}x{$triangle_height}+{int(1/2 * triangle_width)}+0 +repage p3m1_G5half.png
         composite p3m1_G5half.png p3m1_tile_part-3.png -gravity east -compose Dst_Over p3m1_tile_part.png
```

4) Generate p4m_tile

4.1) Generate p3m1_tile_part_flip.png: convert p3m1_tile_part.png -flip p3m1_tile_part_flip.png

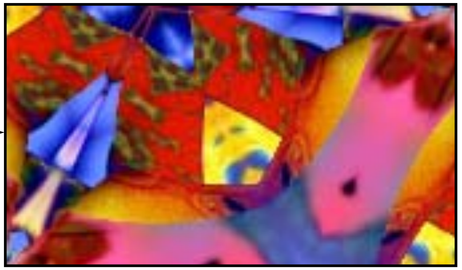
4.2) Generate tile_row: convert p3m1_tile_part.png p3m1_tile_part_flip.png +append p3m1_tile_row.png

p3m1_tile_part



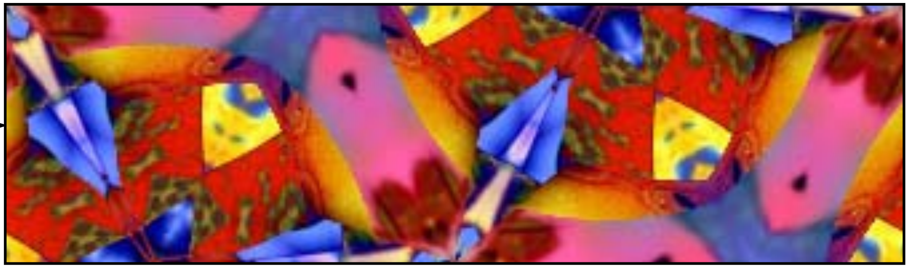
flip

p3m1_tile_part_flip



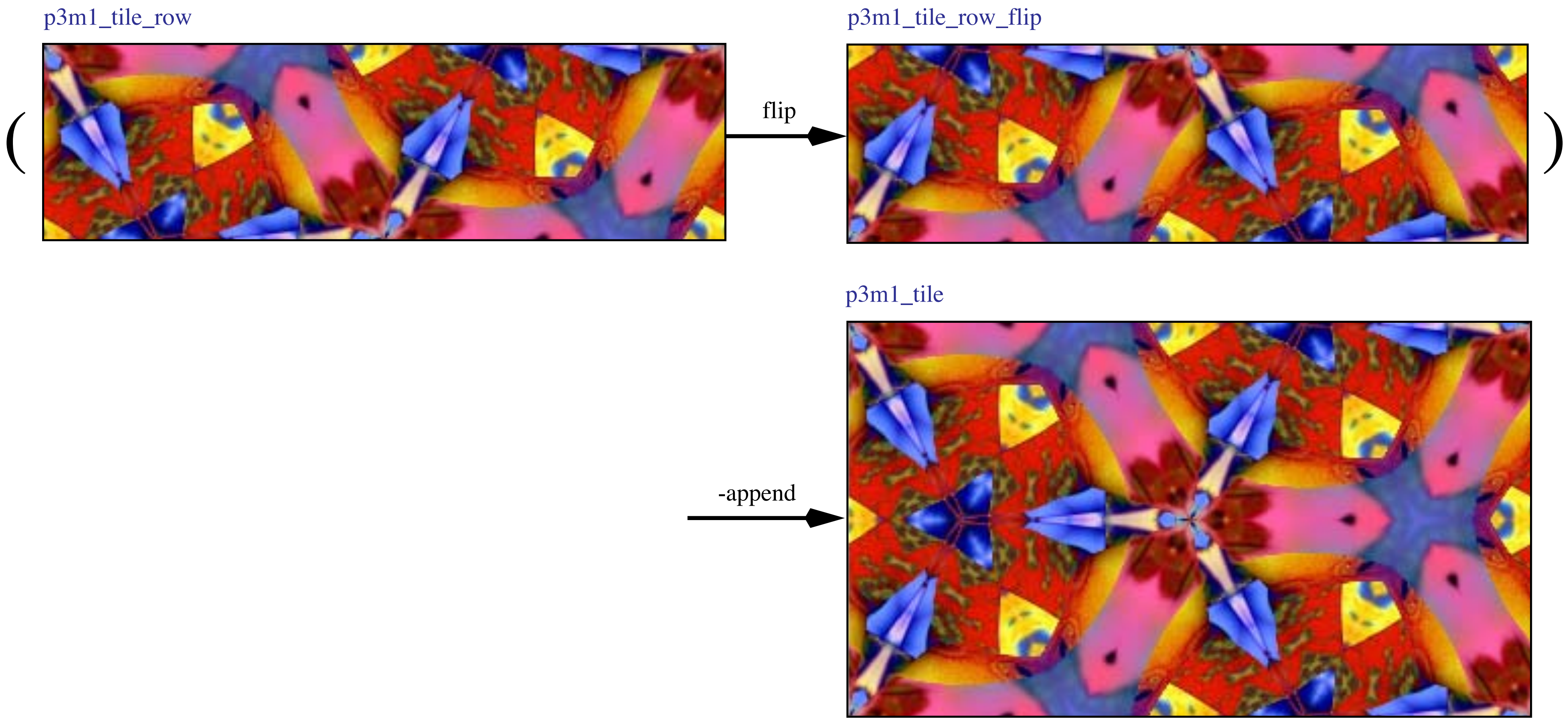
+append

p3m1_tile_row



4.3) Generate tile_row_flop: convert p3m1_tile_row.png -flip p3m1_tile_row_flip.png

4.4) Generate p4m_tile: convert p3m1_tile_row.png p3m1_tile_row_flip.png -append p3m1_tile.png



p3m1_tile_row

flip

p3m1_tile_row_flip

p3m1_tile

-append

IM command sequence for p3m1 in general

```
# 1) Generate mask p3m1_mask
        convert -size {$triangle_width}x{$triangle_height} xc:none -fill red
              -draw "polyline 0,0 $triangle_width,0 int(1/2 * $triangle_width)+1,$triangle_height" p3m1_mask.png

# 2) Generate image_p3m1
        convert image.jpg -crop {$triangle_width}x{$triangle_height}+{$x1}+{$y1} -background none +repage image_p3m1-1.png

# 3) Generate G1, G2, G6, G5
# 3.1) Generate  p3m1_G1
        composite p3m1_mask.png image_p3m1.png -matte -compose Dst_In -background none +repage p3m1_G1.png
# 3.2) Generation of p3m1_G2
        convert p3m1_G1.png -flop -rotate -60 p3m1_G6b.png
        convert p3m1_G2b.png -gravity NorthEast -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G2.png
# 3.3) Generation of p3m1_G6
        convert p3m1_G1.png -flop -rotate -60 p3m1_G6b.png
        convert p3m1_G6b.png -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G6.png
# 3.4) Generation of p3m1_G5
        convert p3m1_G6.png -flip  p3m1_G5.png

# 4) Generate p3m1_tile_part
# 4.1) crop 1/2 G6: p3m1_G6h
        convert p3m1_G6.png -gravity East -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G6h.png
# 4.2) insert p3m1_G6h in G1 at the left side: p3m1_tile_part-1
        composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
# 4.3) extend p3m1_tile_part on the right with transparency: p3m1_tile_part-2
        convert -size {int(1/2 * $triangle_width) + 1}x{$triangle_height} xc:none transparent.png
        convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
# 4.4) insert G2 at the right side: p3m1_tile_part-3
        composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part-3.png
# 4.5) crop1/2 G5: p3m1_G5h
        convert p3m1_G5.png -gravity West -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G5h.png
# 4.6) insert p3m1_G5h right in p3m1_tile_part-3: p3m1_tile_part
        composite p3m1_G5h.png p3m1_tile_part-3.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part.png
```

```
# 5) Generate p3m1_tile
# 5.1) Generate p3m1_tile_part_flip.png
        convert p3m1_tile_part.png -flip -background none +repage p3m1_tile_part_flip.png
# 5.2) Generate tile_row
        convert p3m1_tile_part.png p3m1_tile_part_flip.png +append -background none +repage p3m1_tile_row.png
# 5.3) Generate tile_row_flop
        convert p3m1_tile_row.png -flip -background none +repage p3m1_tile_row_flip.png
# 5.4) Generate p3m1_tile
        convert p3m1_tile_row.png p3m1_tile_row_flip.png -append -background none +repage p3m1_tile.png
```

```
use Image::Magick;


# 0)Input
# 0.1) counting variable, folders, file
$p = '1'; #counting variable
$sourcefolder_path = "C:/ART/IM_PlaneCovering/1_Sourceimages/";
$sourcefile_name = 'image.jpg';
$p3m1_resultfolder_path = "C:/ART/IM_PlaneCovering/2_Tiles/2_03_p3m1-Tiles/";


# 0.2) definition of lokal parameters
$triangle_width_min_Faktor = 0.4;
$triangle_width_max_Faktor = 0.6;


# 1) definition of PerlMagick objects
$image_p3m1 = new Image::Magick;
$p3m1_mask = new Image::Magick;
$transparent_extention = new Image::Magick;


# 2) inilisation of random number generator
srand;


# 3) open the source image
$sourcefile_fullname = $sourcefolder_path . $sourcefile_name;
$image_p3m1->Read("$sourcefile_fullname");
```

```
# 4) local parameter calculations: $triangle_width = $triangle_height, $x1, $y1
$image_width = $image_p3m1->Get('columns');
$image_height = $image_p3m1->Get('rows');
if ($image_width < $image_height) {$image_min = $image_width} else {$image_min = $image_height};
$triangle_width_min = $triangle_width_min_Faktor * $image_min;
$triangle_width_max = $triangle_width_max_Faktor * $image_min;
$triangle_width = $triangle_width_min + (int(rand($triangle_width_max - $triangle_width_min)) + 1);
if ($triangle_width % 2 == 0) {$triangle_width++};
$triangle_height = int(1/2 * sqrt(3) * $triangle_width)+1;
$x_allowed = $image_width - $triangle_width;
$y_allowed = $image_height - $triangle_height;
$x1 = int(rand($x_allowed))+1;
$y1 = int(rand($y_allowed))+1;


# 5) Generate mask p3m1_mask
#convert -size {$triangle_width}x{$triangle_height} xc:none -fill red -draw "polyline 0,0 $triangle_width,0 int(1/2 *
$triangle_width)+1,$triangle_height" p3m1_mask.png
$p3m1_mask->Set(size=>"$triangle_width x $triangle_height");
$p3m1_mask->Read('xc:none');
$triangle_halfwidth = int(1/2 * $triangle_width)+1;
$p3m1_mask->Draw(primitive=>'polyline', points=>"0,0 $triangle_width,0 $triangle_halfwidth,$triangle_height", fill=>'red');


# 6) Generate image_p3m1
#convert image.jpg -crop {$triangle_width}x{$triangle_height}+{$x1}+{$y1} -background none +repage image_p3m1.png
$image_p3m1->Crop(geometry=>"$triangle_width x $triangle_height+$x1+$y1");
```

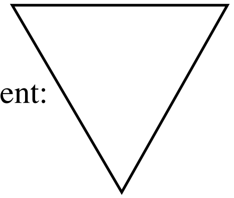
```
# 7) Generate G1, G2, G6, G5
# 7.1) Generate  p3m1_G1
#composite p3m1_mask.png image_p3m1.png -matte -compose Dst_In -background none +repage p3m1_G1.png
$p3m1_G1 = $p3m1_mask->Clone();
$p3m1_G1->Composite(image=>$image_p3m1, compose=>in, color=>'transparent', matte=>'true');


# 7.2) Generation of p3m1_G2
#convert p3m1_G1.png -flop -rotate -60 p3m1_G6b.png
#convert p3m1_G2b.png -gravity NorthEast -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G2.png
$p3m1_G2 = $p3m1_G1->Clone();
$p3m1_G2->Flop();
$p3m1_G2->Rotate(degrees=>60, color=>'transparent');
$p3m1_G2_width_rotated = $p3m1_G2->Get('columns');
$p3m1_G2_cropwidth = $p3m1_G2_width_rotated - $triangle_width;
$p3m1_G2->Crop(geometry=>"$triangle_width x $triangle_height + $p3m1_G2_cropwidth+0", background=>'transparent');


# 7.3) Generation of p3m1_G6
#convert p3m1_G1.png -flop -rotate -60 p3m1_G6b.png
#convert p3m1_G6b.png -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G6.png
$p3m1_G6 = $p3m1_G1->Clone();
$p3m1_G6->Flop();
$p3m1_G6->Rotate(degrees=>-60, color=>'transparent');
$p3m1_G6->Crop(geometry=>"$triangle_width x $triangle_height+0+0", background=>'transparent');


# 7.4) Generation of p3m1_G5
#convert p3m1_G6.png -flip  p3m1_G5.png
$p3m1_G5 = $p3m1_G6->Clone();
$p3m1_G5->Flip();
```

```
# 8) Generate p3m1_tile_part
# 8.1) crop 1/2 G6 left
#convert p3m1_G6.png -gravity East -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G6h.png
$p3m1_G6h = $p3m1_G6->Clone();
$p3m1_G6h->Crop(geometry=>"$triangle_halfwidth x $triangle_height + $triangle_halfwidth+0", background=>'transparent');


# 8.2) insert p3m1_G6h in G1 at the left side
#composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
$p3m1_tile_part = $p3m1_G1->Clone();
$p3m1_tile_part->Composite(image=>$p3m1_G6h, gravity=>West, compose=>over, color=>'transparent', matte=>'true');


# 8.3) extend p3m1_tile_part on the right with transparency
#convert -size {int(1/2 * $triangle_width) + 1}x{$triangle_height} xc:none transparent_extention.png
#convert p3m1_tile_part-1.png transparent_extention.png +append -background none +repage p3m1_tile_part-2.png
$transparent_extention->Set(size=>"$triangle_halfwidth x $triangle_height");
$transparent_extention->Read('xc:none');
$q = $p3m1_tile_part->Clone();
push(@$q, $transparent_extention);
$p3m1_tile_part = $q->Append(stack=>'false');


# 8.4) insert G2 at the right side
#composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part-3.png
$p3m1_tile_part->Composite(image=>$p3m1_G2, gravity=>East, compose=>over, color=>'transparent', matte=>'true');


# 8.5) crop1/2 G5
# convert p3m1_G5.png -gravity West -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G5h.png
$p3m1_G5h = $p3m1_G5->Clone();
$p3m1_G5h->Crop(geometry=>"$triangle_halfwidth x $triangle_height+0+0",background=>'transparent');
```

```
# 8.6) insert p3m1_G5h right in p3m1_tile_part
#composite p3m1_G5h.png p3m1_tile_part-3.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part.png
$p3m1_tile_part->Composite(image=>$p3m1_G5h, gravity=>East, compose=>over, color=>'transparent', matte=>'true');


# 9) Generate p3m1_tile
# 9.1) Generate p3m1_tile_part_flip
#convert p3m1_tile_part.png -flip -background none +repage p3m1_tile_part_flip.png
$p3m1_tile_part_flip = $p3m1_tile_part->Clone();
$p3m1_tile_part_flip->Flip();


# 9.2) Generate tile_row
#convert p3m1_tile_part.png p3m1_tile_part_flip.png +append -background none +repage p3m1_tile_row.png
$q = ();
$q = $p3m1_tile_part->Clone();
push(@$q, $p3m1_tile_part_flip);
$p3m1_tile_row = $q->Append(stack=>'false');


# 9.3) Generate tile_row_flip
#convert p3m1_tile_row.png -flip -background none +repage p3m1_tile_row_flip.png
$p3m1_tile_row_flip = $p3m1_tile_row->Clone();
$p3m1_tile_row_flip->Flip();


# 9.4) Generate p3m1_tile
#convert p3m1_tile_row.png p3m1_tile_row_flip.png -append -background none +repage p3m1_tile.png
@$q = ();
$q = $p3m1_tile_row->Clone();
push(@$q, $p3m1_tile_row_flip);
$p3m1_tile = $q->Append(stack=>'true');
```

```
# 10) generate name of result image
chop $sourcefile_name;
chop $sourcefile_name;
chop $sourcefile_name;
chop $sourcefile_name;

if ($p <= 9) {$resultimage_name = "p3m1_tile_" . $sourcefile_name . "-0" . $p . ".jpg"}
else {$resultimage_name = "p3m1_tile_" . $sourcefile_name . "-"  . $p . ".jpg"};

$resultimagepath_name = $p3m1_resultfolder_path . $resultimage_name;


#11) save result image
$p3m1_tile->Write(filename=>"$resultimagepath_name", compression=>'JPEG', quality=>'95');
```

p3m1 basic element:



given in the deterministic case

a) image.jpg with (image_width, image_height)
    example: image_width = image_height: 4000 [pixel]

b) format of the basic element: triangle_width
    example: = triangle_width1501 [pixel]

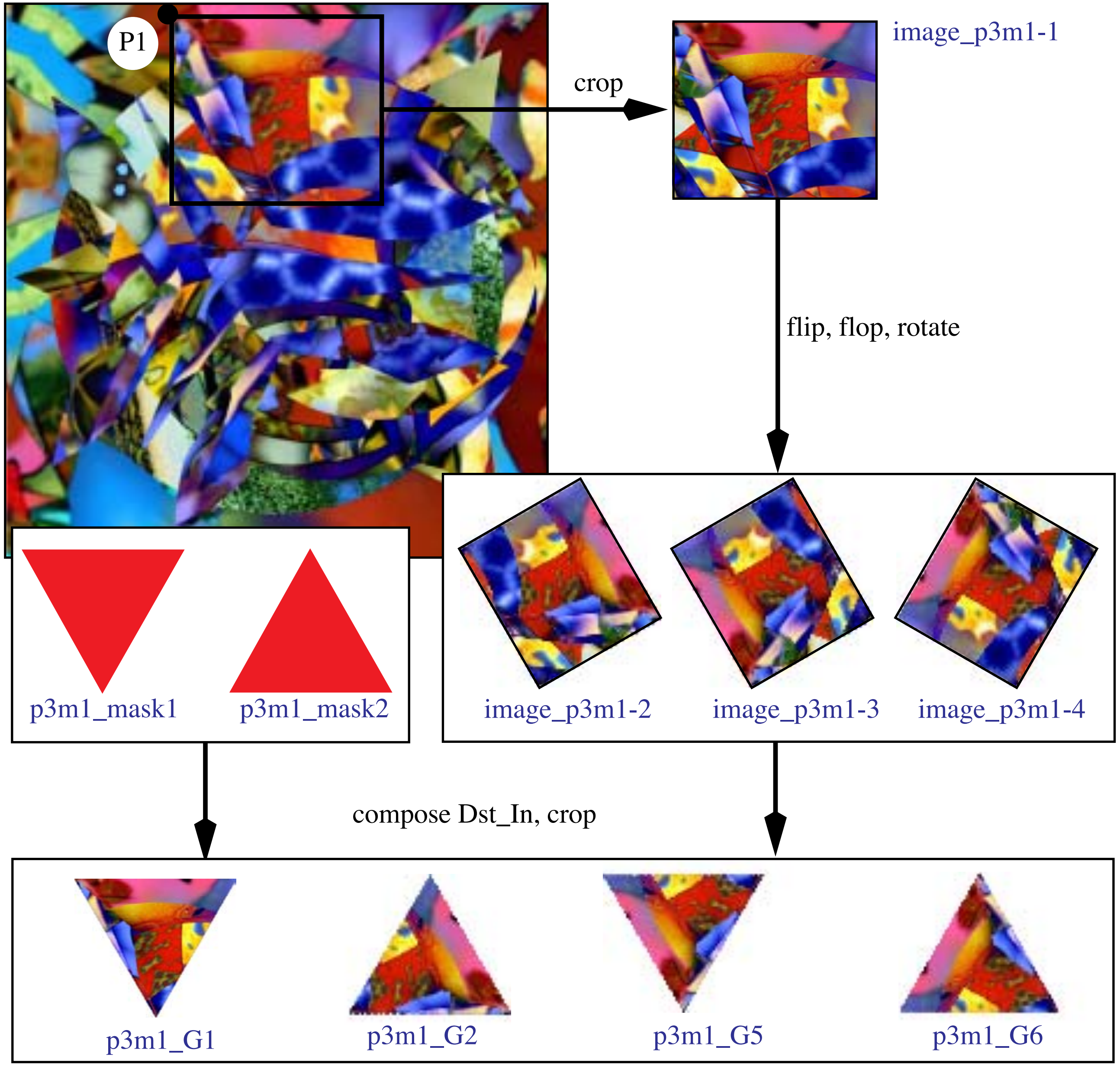
c) Top left point P1 in image.jpg for selection of image_p3m1
    example: P1 = (1241,113)

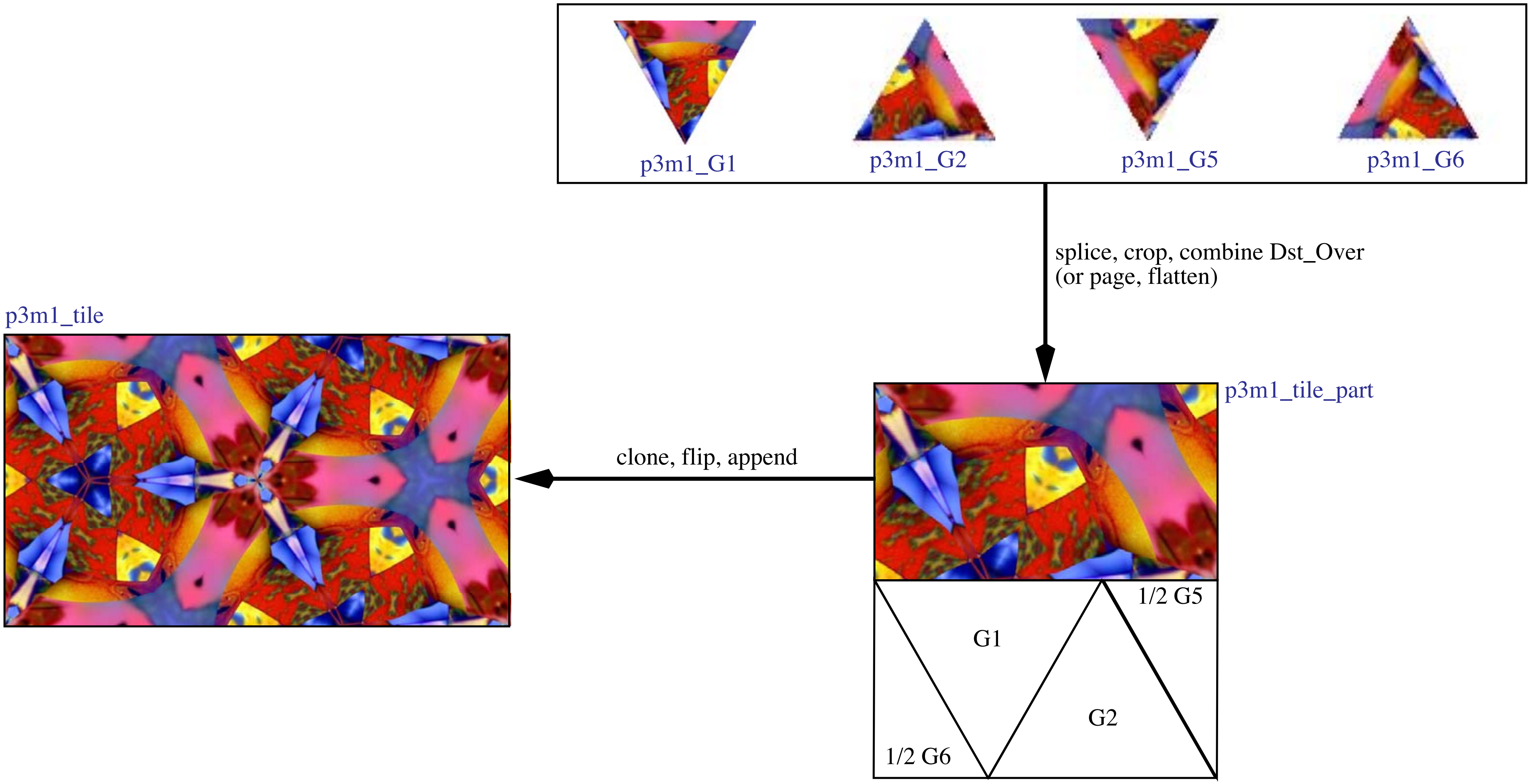
given in the stochastic case

a) image.jpg with (image_width, image_height)

b) random variable interval for the selection of triangle_width:
    [triangle_min_Faktor, triangle_max_Faktor] = [0.3, 0.7]

```
p3m1-procedure:
Start
    1) Generate p3m1_mask1 and p3m1_mask2
    2) Generate image_p3m1-1, ..., image_p3m1-4
    3) Generate G1, G2, G5, G6
    4) Generate p3m1_tile_part
    5) Generate p3m1_tile
End
```





p3m1_G1

p3m1_G2

p3m1_G5

p3m1_G6

splice, crop, combine Dst_Over
(or page, flatten)

p3m1_tile_part

1/2 G5

G1

G2

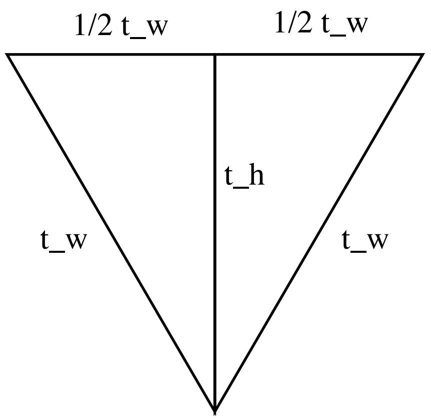
1/2 G6

clone, flip, append

p3m1_tile

Only relevant for the stochastic case

```
#Calculate scalar triangle_width, triangle_height, x1, y1

$triangle_width_min_Faktor = 0.3;
$triangle_width_max_Faktor =0.7;
if ($image_width < $image_height) {$image_min = $image_width} else {$image_min = $image_height};
$triangle_width_min = $triangle_width_min_Faktor * $image_min;
$triangle_width_max = $triangle_width_max_Faktor * $image_min;
$triangle_width = $triangle_width_min + (int(rand($triangle_width_max - $triangle_width_min)) + 1);
if ($triangle_width % 2 == 0) { $triangle_width++ };
$triangle_height = int(1/ 2 * √3 * $triangle_width) + 1;
$x_allowed = $image_width - $triangle_width;
$y_allowed = $image_hight - $triangle_width;
$x1 = int(rand($x_allowed)) + 1;
$y1 = int(rand($y_allowed)) + 1;
```



$t\_w^2 = 1/4\ t\_w^2 + t\_h^2$

$\Leftrightarrow t\_h^2 = 3/4 * t\_w^2$

$\Leftrightarrow t\_h = 1/2 * \sqrt{3} * t\_w$

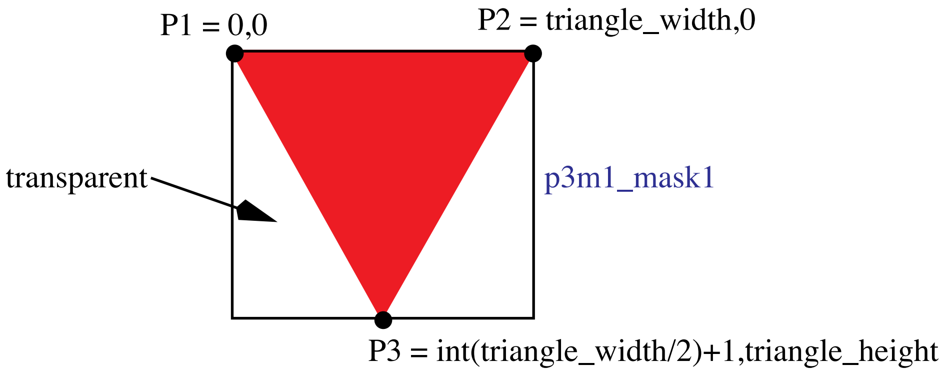
1) Generate masks p3m1_mask1 and p3m1_mask2 on scratch:

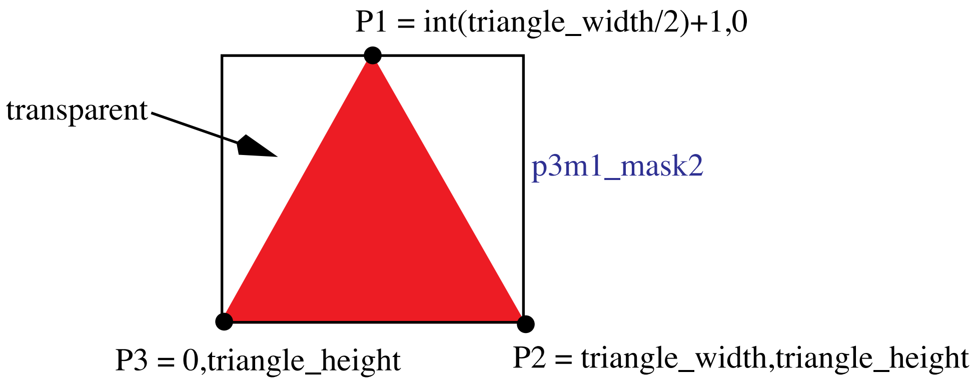
```
convert -size {$triangle_width}x{$triangle_height} xc:none -fill red -draw "polyline 0,0 $triangle_width,0 int(1/2 * $triangle_width)+1,$triangle_height" p3m1_mask1.png

convert -size {$triangle_width}x{$triangle_height} xc:none -fill red -draw "polyline int(1/2 * $triangle_width)+1,0 $triangle_width,$triangle_height 0,$triangle_height" p3m1_mask2.png
```

For the example holds:

$triangle_height = int(1/ 2 * √3 * 1501) + 1 = 1300
int(1/ 2 * $triangle_width) + 1 = 751

```
convert -size 1501x1300 xc:none -fill red -draw “polyline 0,0 1501,0 751,1300” p3m1_mask1.png
convert -size 1501x1300 xc:none -fill red -draw “polyline 751,0 1501,1300 0,1300” p3m1_mask2.png
```



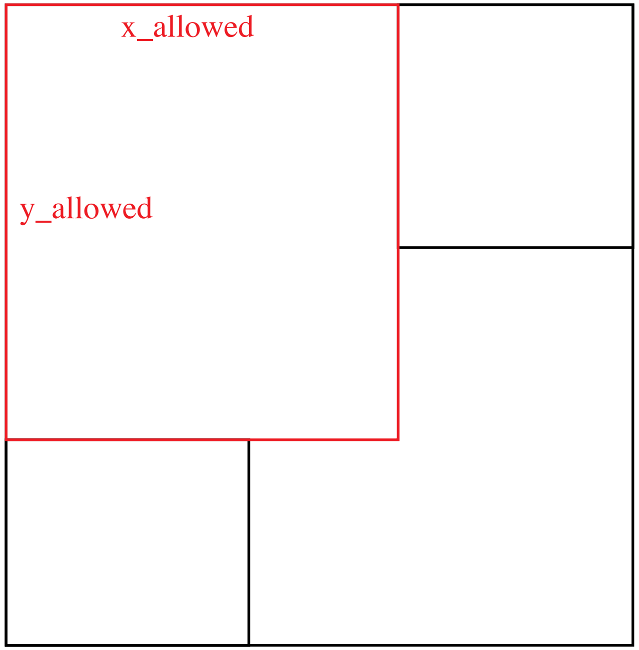

P1 = int(triangle_width/2)+1,0
transparent
p3m1_mask2
P3 = 0,triangle_height
P2 = triangle_width,triangle_height

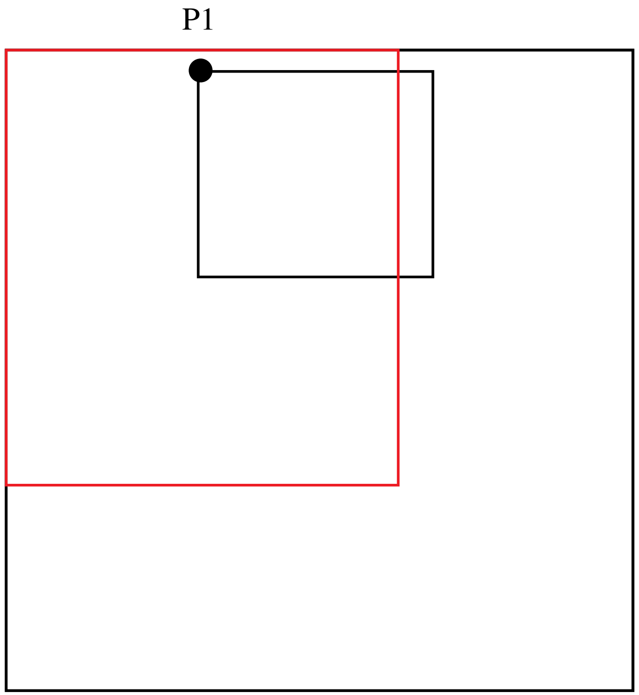
2.1) Generate image_p3m1-1

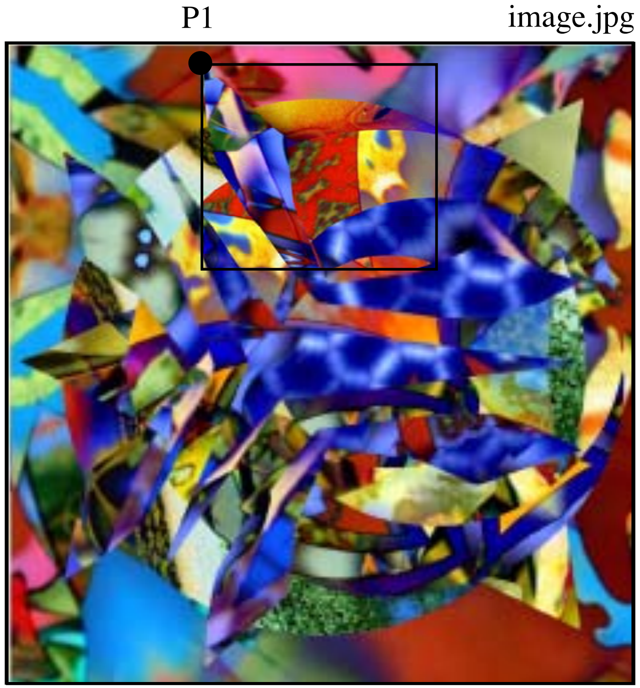
```
convert image.jpg -crop {$triangle_width}x{$triangle_width}+{$x1}+{$y1} -background none +repage image_p3m1-1.png
```

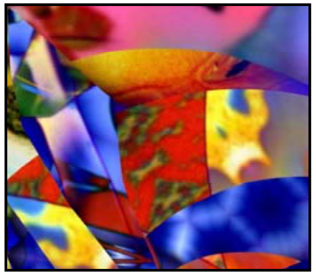
For the example holds:

```
convert image.jpg -crop 1501x1300+1241+113 -background none +repage image_p3m1-1.png
```



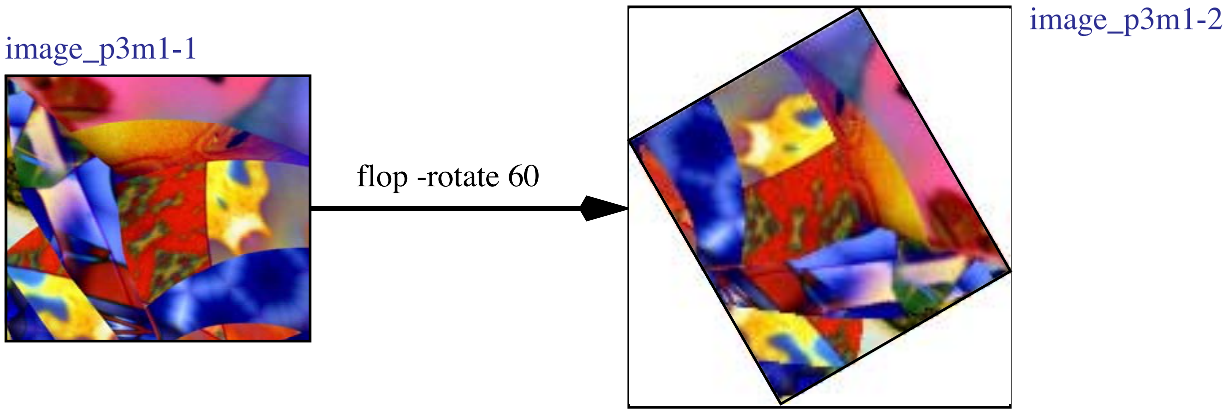






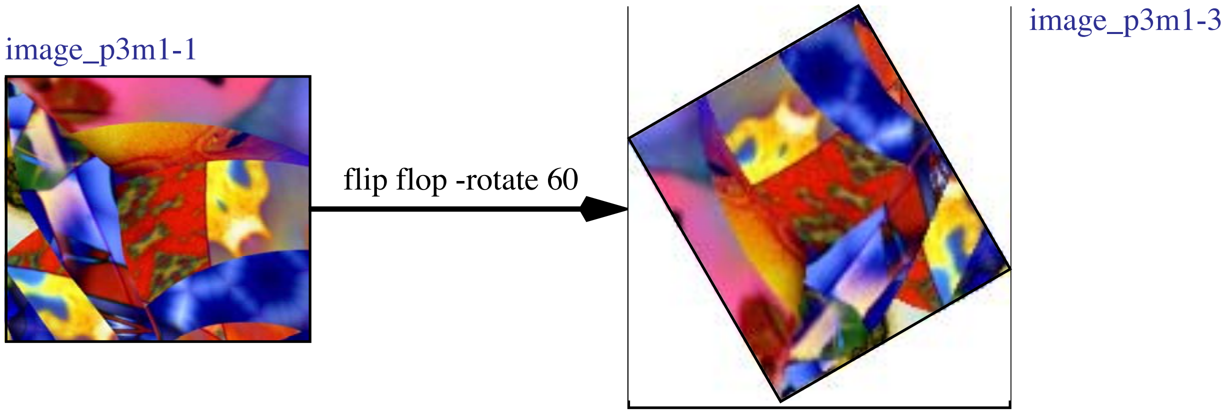
2.2) Generate image_p3m1-2

```
convert  image_p3m1-1.png -flop -background none -rotate 60 +repage image_p3m1-2.png
```



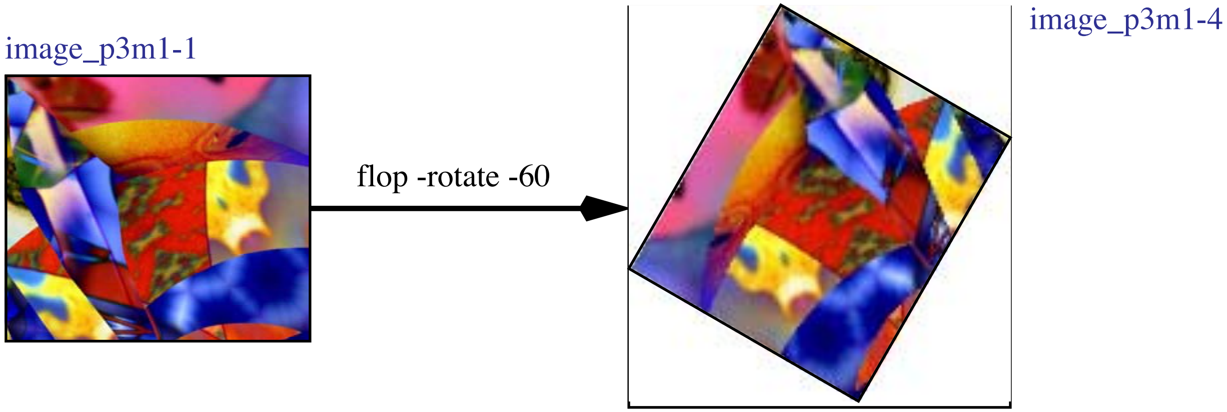
2.3) Generate image_p3m1-3

```
convert  image_p3m1-1.png -flip -flop -background none -rotate 60 +repage image_p3m1-3.png
```



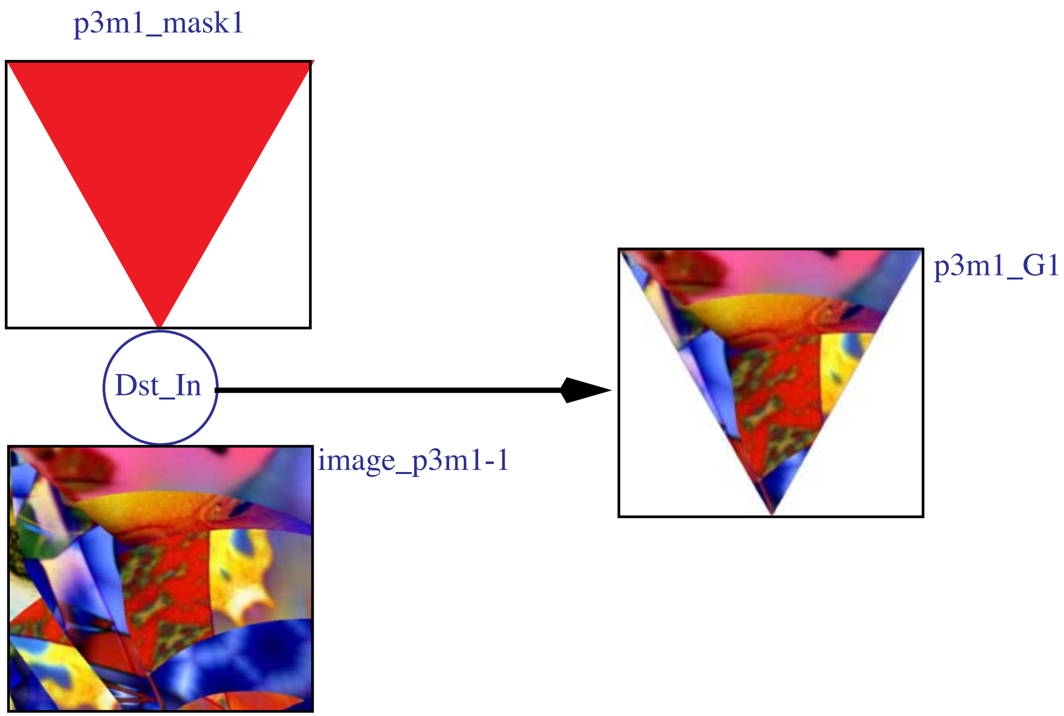
2.4) Generate image_p3m1-4

```
convert  image_p3m1-1.png -flop -background none -rotate -60 +repage image_p3m1-4.png
```



3.1) Generate p3m1_G1

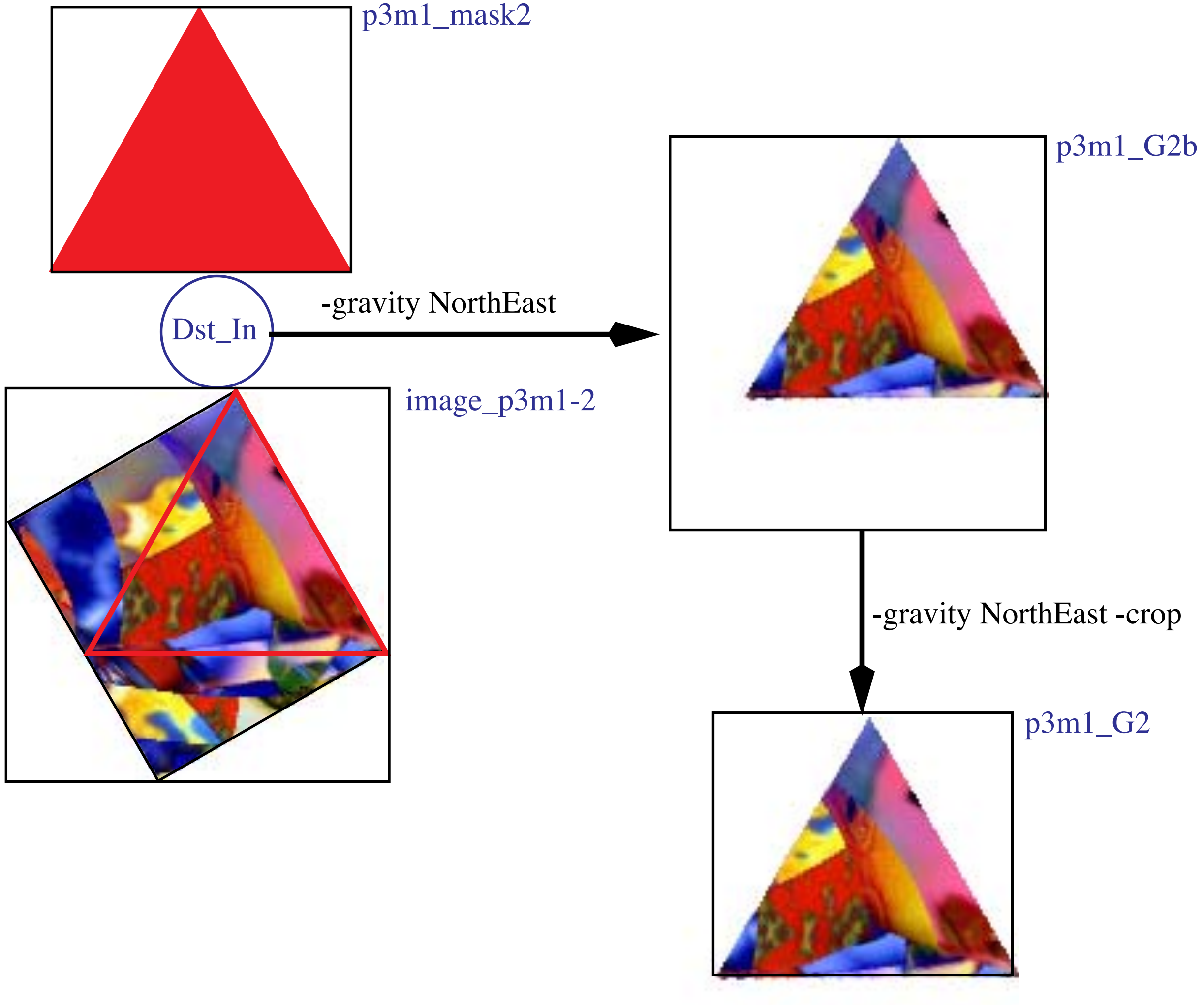
```
composite p3m1_mask1.png image_p3m1-1.png -matte -compose Dst_In p3m1_G1.png
```



3.2) Generation of p3m1_G2

Compose: composite p3m1_mask2.png image_p3m1-2.png -gravity NorthEast -matte -compose Dst_In -background none +repage p3m1_G2b.png

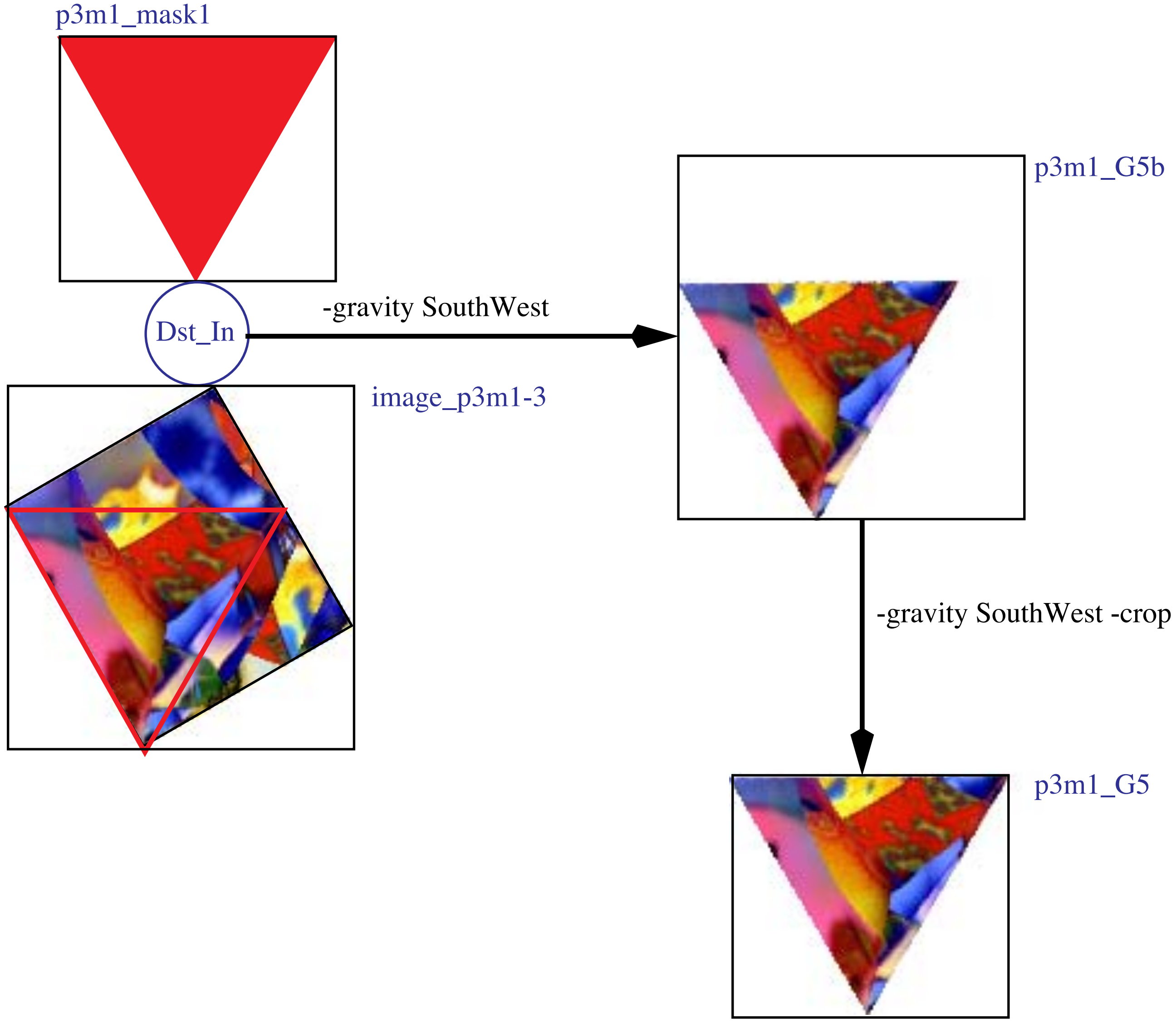
Cut out image part: convert p3m1_G2b.png -gravity NorthEast -crop 1501x1300+0+0 -background none +repage p3m1_G2.png



3.3) Generation of p3m1_G5

Compose: composite p3m1_mask1.png image_p3m1-3.png -gravity SouthWest -matte -compose Dst_In -background none +repage p3m1_G5b.png

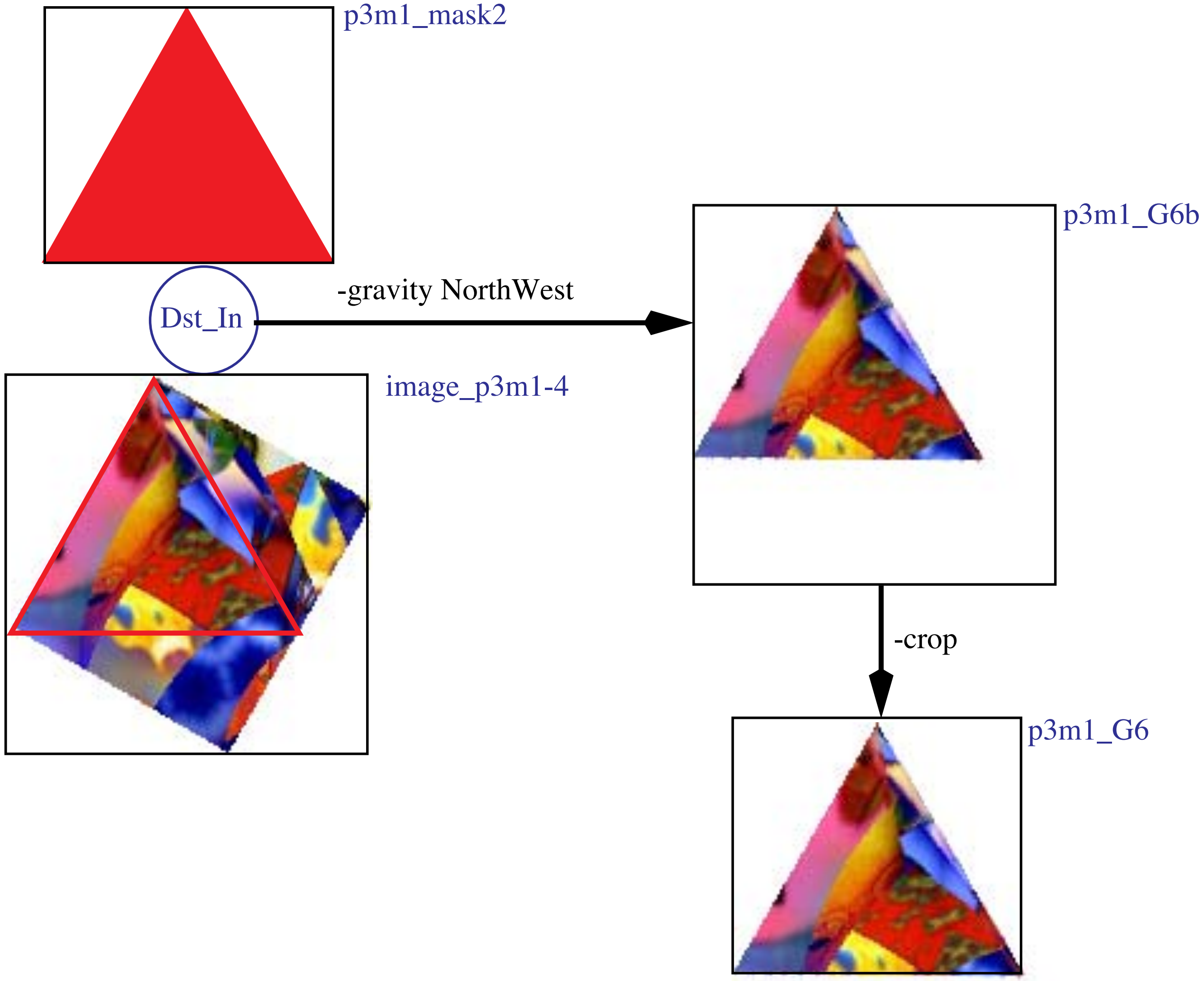
Cut out image part: convert p3m1_G5b.png -gravity SouthWest -crop 1501x1300+0+0 -background none +repage p3m1_G5.png



3.4) Generation of p3m1_G6

Compose: composite p3m1_mask2.png image_p3m1-4.png -gravity NorthWest -matte -compose Dst_In -background none +repage p3m1_G6b.png

Cut out image part: convert p3m1_G6b.png -crop 1501x1300+0+0 -background none +repage p3m1_G6.png



4) Generate p3m1_tile_part

by successive insertion of basic elements G6,G1,G2;G5 with

4.1) crop 1/2 G6: p3m1_G6h

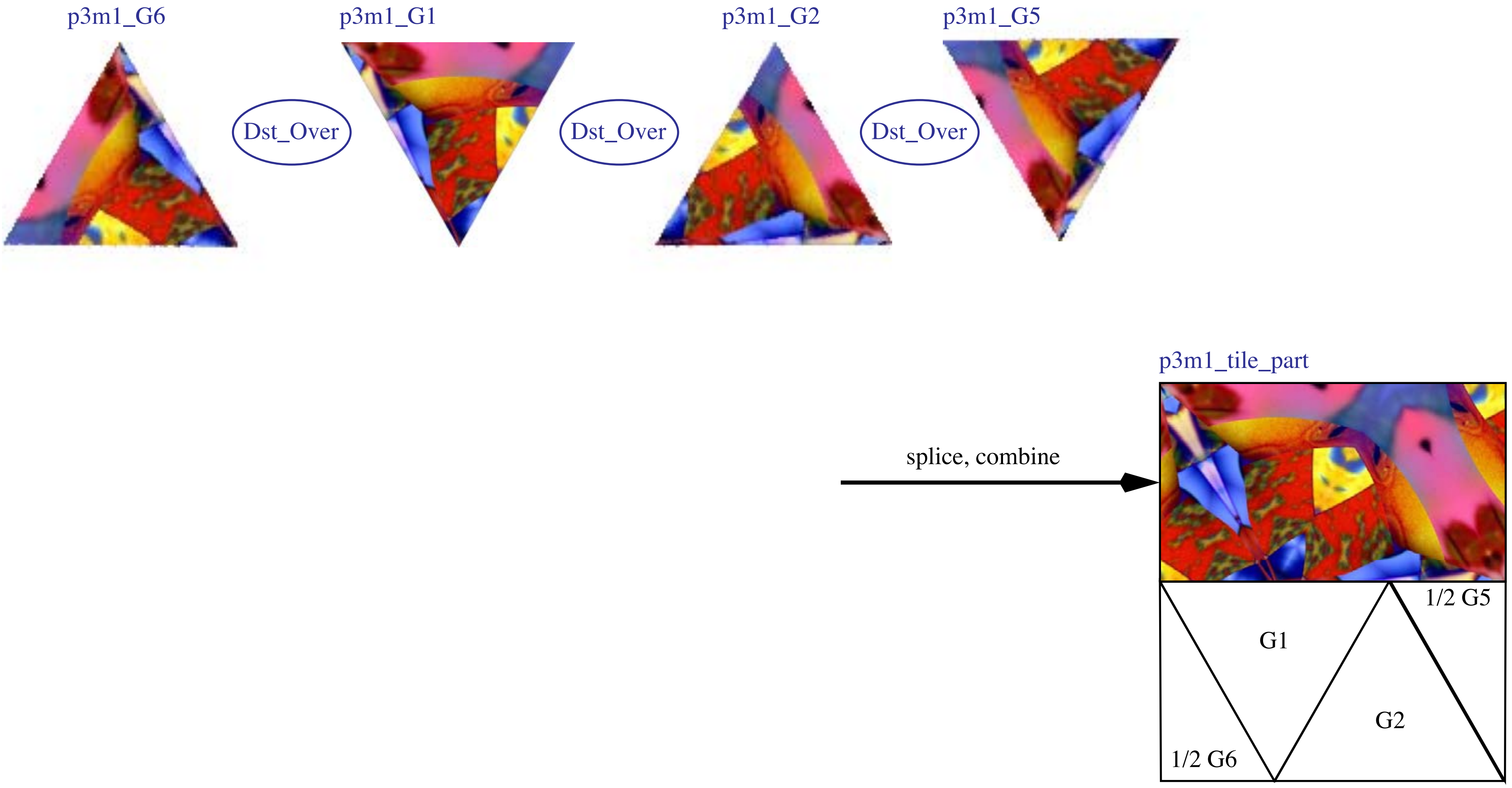
4.2) insert p3m1_G6h in G1 at the left side: p3m1_tile_part-1

4.3) splice p3m1_tile_part right: p3m1_tile_part-2

4.4) insert G2 at the right side: p3m1_tile_part-3

4.5) crop1/2 G5: p3m1_G5h

4.6) insert p3m1_G5h right in p3m1_tile_part-3: p3m1_tile_part



p3m1_G6

Dst_Over

p3m1_G1

Dst_Over

p3m1_G2

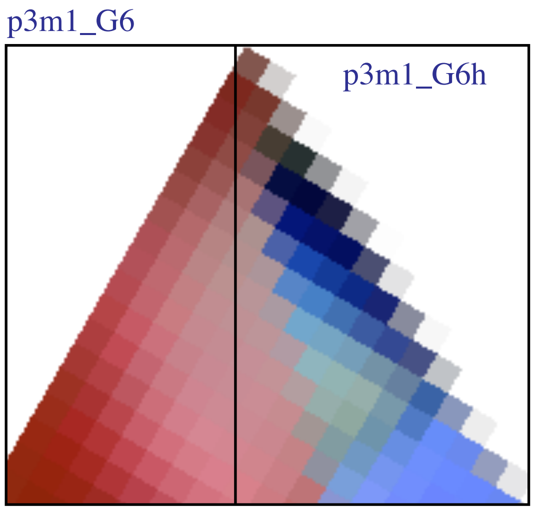
Dst_Over

p3m1_G5

splice, combine

p3m1_tile_part

4.1) crop 1/2 G6: p3m1_G6h

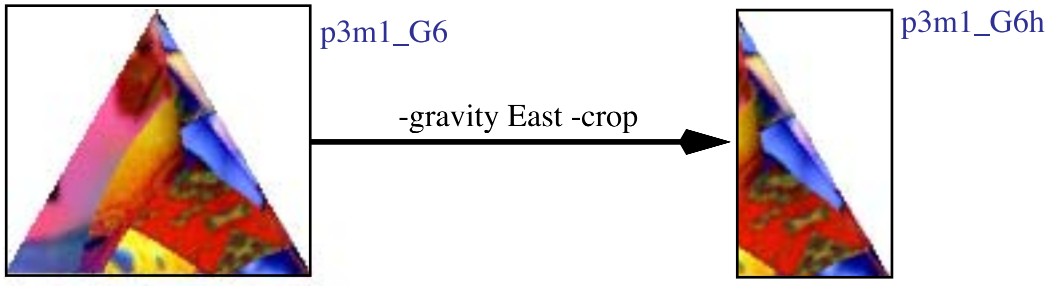


triangle_width is an odd integer, so it must be decided what to do with the middle row of p3m1_G1. If the middle row should be part of p3m1_tile_part then the width of p3m1_G6h is after the crop command int(1/2*$triangle_width)+1 pixel (see image on the right side).

```
convert p3m1_G6.png -gravity East -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G6h.png
```

For the example holds:

```
convert p3m1_G6.png -gravity East -crop 751x1300+0+0 -background none +repage p3m1_G6h.png
```



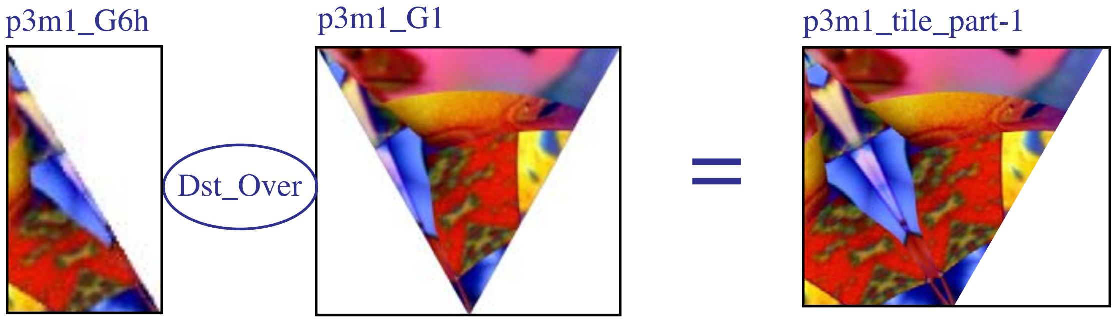
4.2) insert p3m1_G6h in G1 at the left side: p3m1_tile_part-1

Combining the right half of p3m1_G6 over the left side of p3m1_G1:

```
composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
```

For the example holds:

```
composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
```

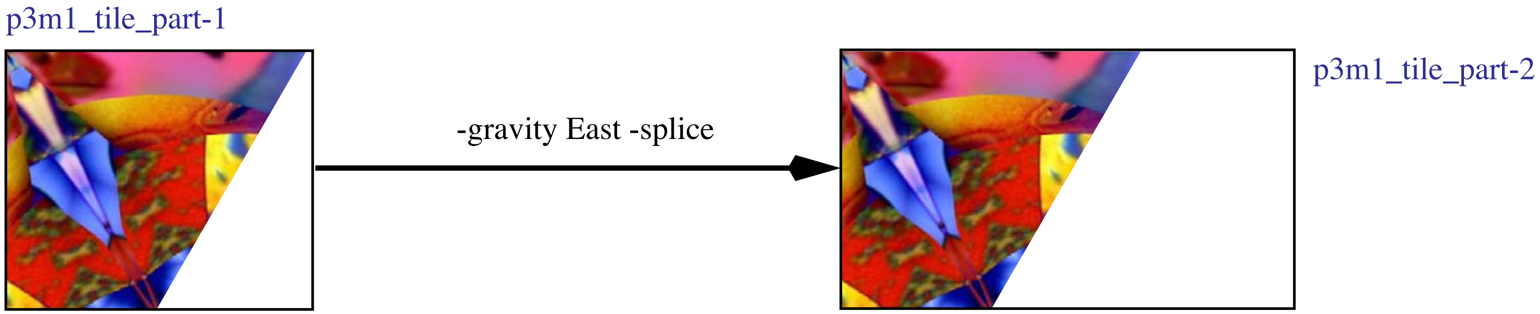


4.3) splice p3m1_tile_part right: p3m1_tile_part-2

```
    convert p3m1_tile_part-1.png -gravity East -background none -splice (int(1/2 * $triangle_width) + 1)x0 -background none +repage p3m1_tile_part-2.png
```

For the example holds:

```
    convert p3m1_tile_part-1.png -gravity East -background none -splice 751x0 -background none +repage p3m1_tile_part-2.png
```

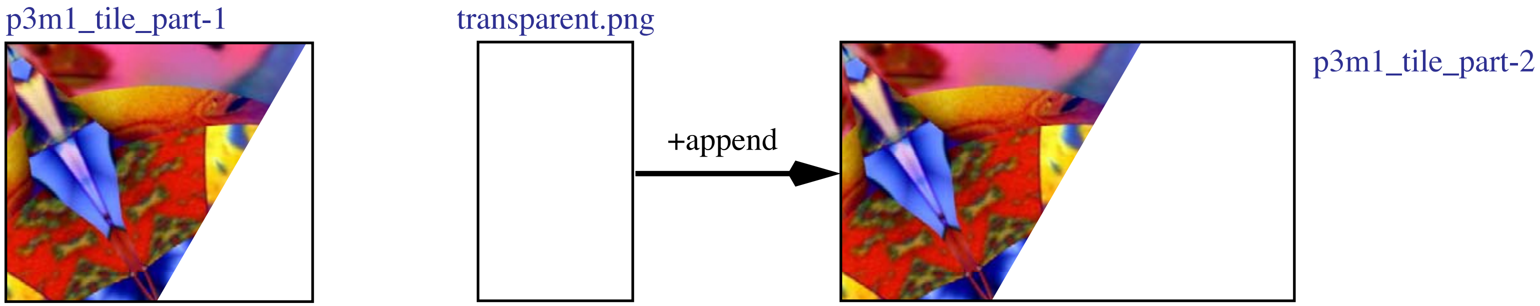


Due to problems with “splice” in the PerlMagick environment an alternative is used by which a transparent image is created and append to the right side of p3m1_tile_part-1:

```
    convert -size (int(1/2 * $triangle_width) + 1)x$triangle_height xc:none transparent.png
    convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
```

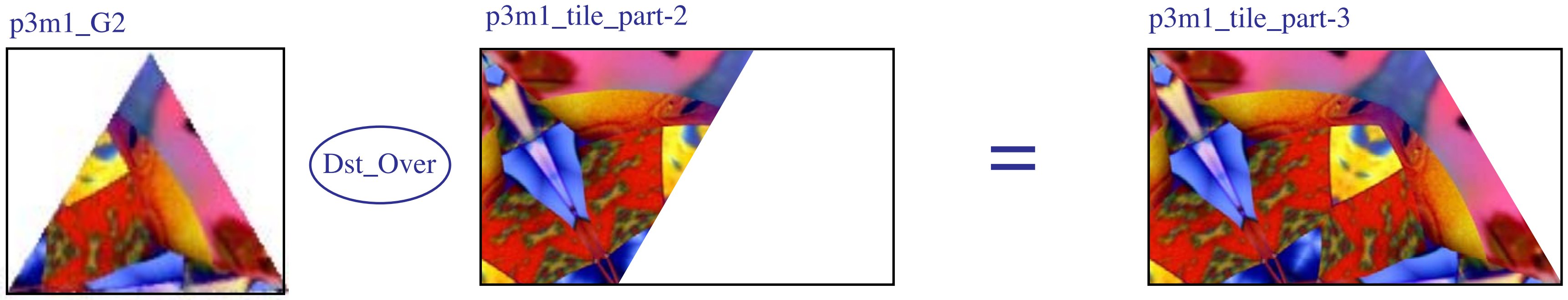
For the example holds:

```
    convert -size 751x1300 xc:none transparent.png
    convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
```



4.4) insert G2 at the right side: p3m1_tile_part-3

```
composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part-3.png
```

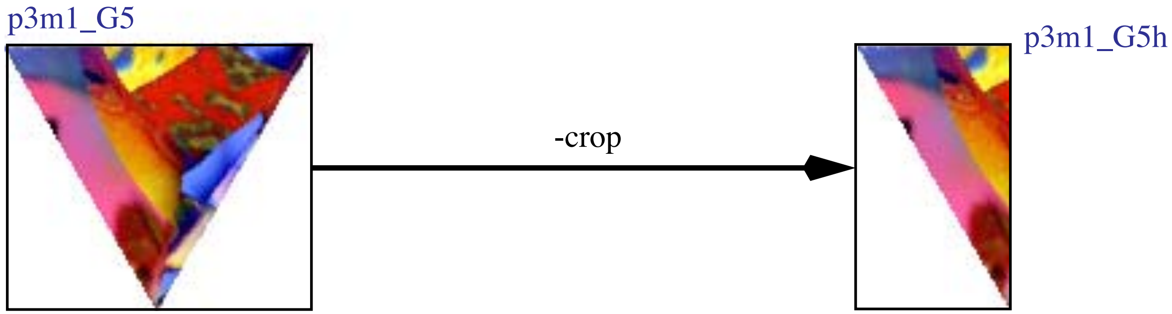


4.5) crop1/2 G5: p3m1_G5h

```
convert p3m1_G5.png -gravity West -crop {int(1/2 * $triangle_width) + 1}x{$triangle_height}+0+0 -background none +repage p3m1_G5h.png
```

For the example holds:

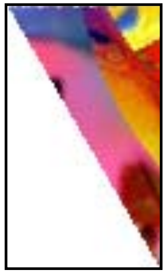
```
convert p3m1_G5.png -gravity West -crop 751x1300+0+0 -background none +repage p3m1_G5h.png
```



4.6) insert p3m1_G5h right in p3m1_tile_part-3: p3m1_tile_part

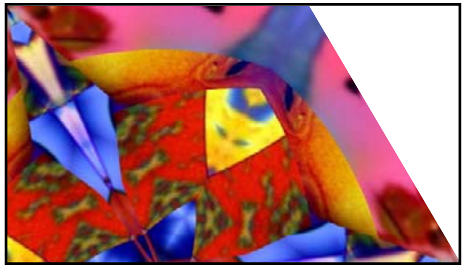
```
composite p3m1_G5h.png p3m1_tile_part-3.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part.png
```

p3m1_G5h



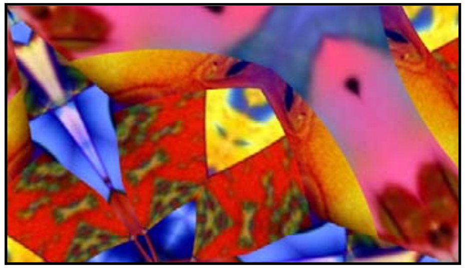
Dst_Over

p3m1_tile_part-3



=

p3m1_tile_part



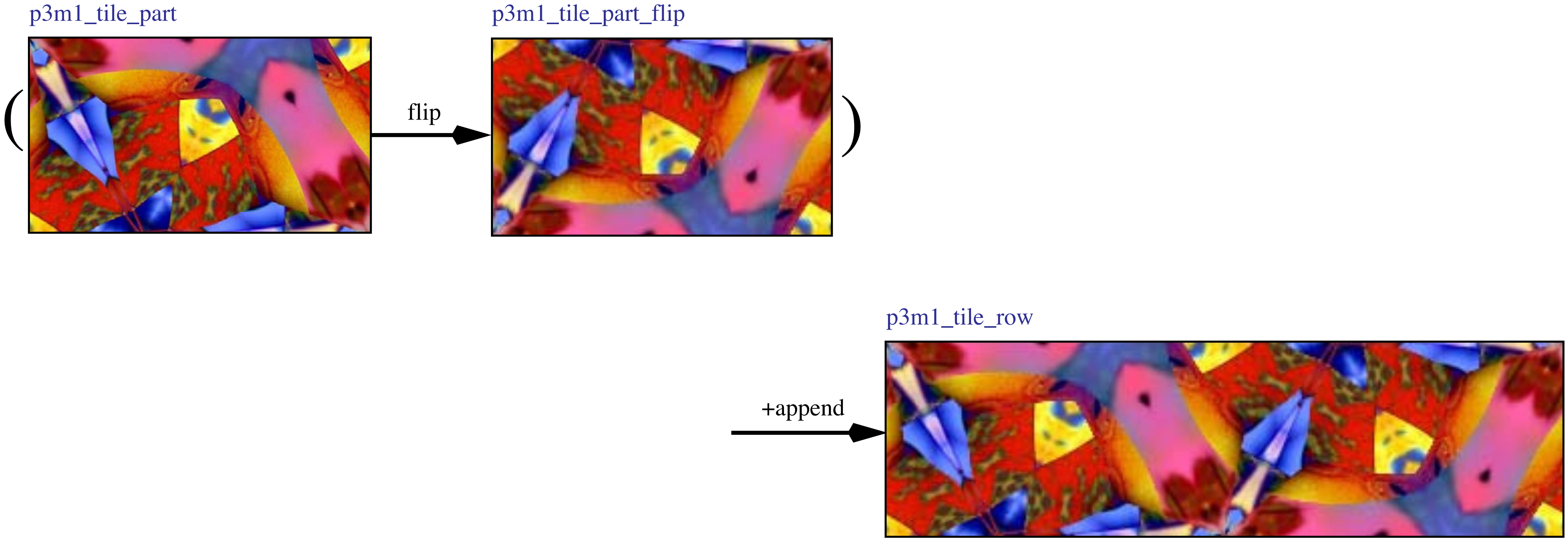
5) Generate p3m1_tile

5.1) Generate p3m1_tile_part_flip.png

```
        convert p3m1_tile_part.png -flip -background none +repage p3m1_tile_part_flip.png
```

5.2) Generate tile_row

```
        convert p3m1_tile_part.png p3m1_tile_part_flip.png +append -background none +repage p3m1_tile_row.png
```



p3m1_tile_part

flip

p3m1_tile_part_flip

+append

p3m1_tile_row

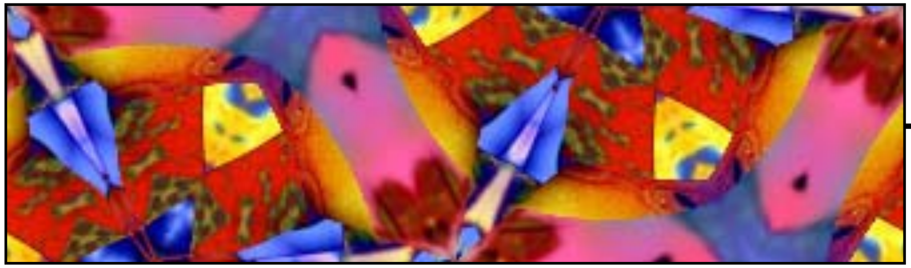
5.3) Generate tile_row_flop

```
convert p3m1_tile_row.png -flip -background none +repage p3m1_tile_row_flip.png
```

5.4) Generate p3m1_tile

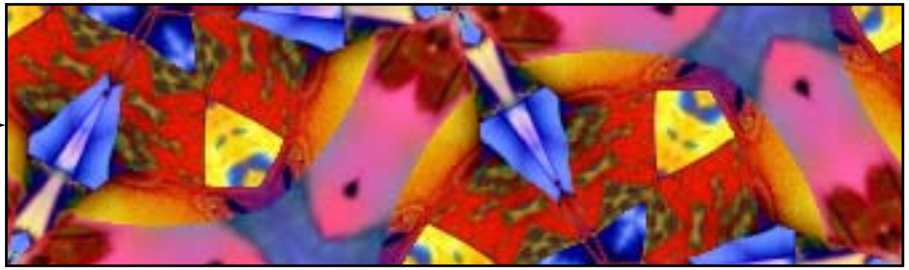
```
convert p3m1_tile_row.png p3m1_tile_row_flip.png -append -background none +repage p3m1_tile.png
```

p3m1_tile_row



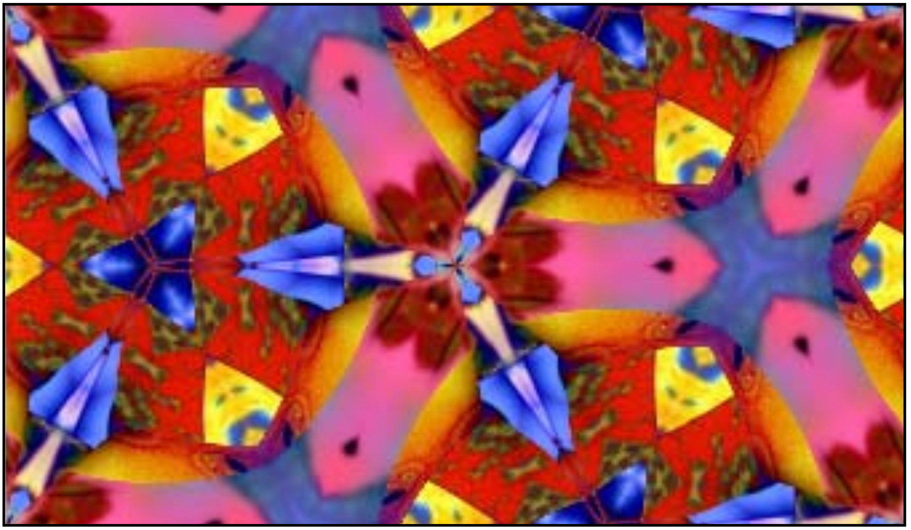
flip

p3m1_tile_row_flip



-append

p3m1_tile



IM command sequence for p3m1 for example

```
# 1) Generate masks p3m1_mask1 and p3m1_mask2
# 1.1) Generate mask p3m1_mask1
        convert -size 1501x1300 xc:none -fill red -draw “polyline 0,0 1501,0 751,1300” p3m1_mask1.png
# 1.2) Generate mask p3m1_mask2
        convert -size 1501x1300 xc:none -fill red -draw “polyline 751,0 1501,1300 0,1300” p3m1_mask2.png

# 2) Generate image_p3m1-1, ..., image_p3m1-4
# 2.1) Generate image_p3m1-1
        convert image.jpg -crop 1501x1300+1241+113 -background none +repage image_p3m1-1.png
# 2.2) Generate image_p3m1-2
        convert  image_p3m1-1.png -flop -background none -rotate 60 +repage image_p3m1-2.png
# 2.3) Generate image_p3m1-3
        convert  image_p3m1-1.png -flip -flop -background none -rotate 60 +repage image_p3m1-3.png
# 2.4) Generate image_p3m1-4
        convert  image_p3m1-1.png -flop -background none -rotate -60 +repage image_p3m1-4.png

# 3) Generate G1, G2, G5, G6
# 3.1) Generate  p3m1_G1
        composite p3m1_mask1.png image_p3m1-1.png -matte -compose Dst_In p3m1_G1.png
# 3.2) Generation of p3m1_G2
        composite p3m1_mask2.png image_p3m1-2.png -gravity NorthEast -matte -compose Dst_In -background none +repage p3m1_G2b.png
        convert p3m1_G2b.png -gravity NorthEast -crop 1501x1300+0+0 -background none +repage p3m1_G2.png
# 3.3) Generation of p3m1_G5
        composite p3m1_mask1.png image_p3m1-3.png -gravity SouthWest -matte -compose Dst_In -background none +repage p3m1_G5b.png
        convert p3m1_G5b.png -gravity SouthWest -crop 1501x1300+0+0 -background none +repage p3m1_G5.png
# 3.4) Generation of p3m1_G6
        composite p3m1_mask2.png image_p3m1-4.png -gravity NorthWest -matte -compose Dst_In -background none +repage p3m1_G6b.png
        convert p3m1_G6b.png -crop 1501x1300+0+0 -background none +repage p3m1_G6.png

# 4) Generate p3m1_tile_part
# 4.1) crop 1/2 G6: p3m1_G6h
        convert p3m1_G6.png -gravity East -crop 751x1300+0+0 -background none +repage p3m1_G6h.png
# 4.2) insert p3m1_G6h in G1 at the left side: p3m1_tile_part-1
        composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
# 4.3) extend p3m1_tile_part on the right with transparency: p3m1_tile_part-2
        convert -size 751x1300 xc:none transparent.png
        convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
```

```
# 4.4) insert G2 at the right side: p3m1_tile_part-3
        composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part-3.png
# 4.5) crop1/2 G5: p3m1_G5h
        convert p3m1_G5.png -gravity West -crop 751x1300+0+0 -background none +repage p3m1_G5h.png
# 4.6) insert p3m1_G5h right in p3m1_tile_part-3: p3m1_tile_part
        composite p3m1_G5h.png p3m1_tile_part-3.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part.png

# 5) Generate p3m1_tile
# 5.1) Generate p3m1_tile_part_flip.png
        convert p3m1_tile_part.png -flip -background none +repage p3m1_tile_part_flip.png
# 5.2) Generate tile_row
        convert p3m1_tile_part.png p3m1_tile_part_flip.png +append -background none +repage p3m1_tile_row.png
# 5.3) Generate tile_row_flop
        convert p3m1_tile_row.png -flip -background none +repage p3m1_tile_row_flip.png
# 5.4) Generate p3m1_tile
        convert p3m1_tile_row.png p3m1_tile_row_flip.png -append -background none +repage p3m1_tile.png
```

IM command sequence for p3m1 in general

```
# 1) Generate masks p3m1_mask1 and p3m1_mask2
# 1.1) Generate mask p3m1_mask1
        convert -size {$triangle_width}x{$triangle_height} xc:none -fill red
              -draw "polyline 0,0 $triangle_width,0 int(1/2 * $triangle_width)+1,$triangle_height" p3m1_mask1.png
# 1.2) Generate mask p3m1_mask2
        convert -size {$triangle_width}x{$triangle_height} xc:none -fill red
              -draw “polyline int(1/2 * $triangle_width)+1,0 $triangle_width,$triangle_height 0,$triangle_height” p3m1_mask2.png

# 2) Generate image_p3m1-1, ..., image_p3m1-4
# 2.1) Generate image_p3m1-1
        convert image.jpg -crop {$triangle_width}x{$triangle_height}+{$x1}+{$y1} -background none +repage image_p3m1-1.png
# 2.2) Generate image_p3m1-2
        convert  image_p3m1-1.png -flop -background none -rotate 60 +repage image_p3m1-2.png
# 2.3) Generate image_p3m1-3
        convert  image_p3m1-1.png -flip -flop -background none -rotate 60 +repage image_p3m1-3.png
# 2.4) Generate image_p3m1-4
        convert  image_p3m1-1.png -flop -background none -rotate -60 +repage image_p3m1-4.png

# 3) Generate G1, G2, G5, G6
# 3.1) Generate  p3m1_G1
        composite p3m1_mask1.png image_p3m1-1.png -matte -compose Dst_In p3m1_G1.png
# 3.2) Generation of p3m1_G2
        composite p3m1_mask2.png image_p3m1-2.png -gravity NorthEast -matte -compose Dst_In -background none +repage p3m1_G2b.png
        convert p3m1_G2b.png -gravity NorthEast -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G2.png
# 3.3) Generation of p3m1_G5
        composite p3m1_mask1.png image_p3m1-3.png -gravity SouthWest -matte -compose Dst_In -background none +repage p3m1_G5b.png
        convert p3m1_G5b.png -gravity SouthWest -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G5.png
# 3.4) Generation of p3m1_G6
        composite p3m1_mask2.png image_p3m1-4.png -gravity NorthWest -matte -compose Dst_In -background none +repage p3m1_G6b.png
        convert p3m1_G6b.png -crop {$triangle_width}x{$triangle_height}+0+0 -background none +repage p3m1_G6.png

# 4) Generate p3m1_tile_part
# 4.1) crop 1/2 G6: p3m1_G6h
        convert p3m1_G6.png -gravity East -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G6h.png
# 4.2) insert p3m1_G6h in G1 at the left side: p3m1_tile_part-1
        composite p3m1_G6h.png p3m1_G1.png -gravity West -compose Dst_Over -background none +repage p3m1_tile_part-1.png
```

```
# 4.3) extend p3m1_tile_part on the right with transparency: p3m1_tile_part-2
        convert -size (int(1/2 * $triangle_width) + 1)x$triangle_height xc:none transparent.png
        convert p3m1_tile_part-1.png transparent.png +append -background none +repage p3m1_tile_part-2.png
# 4.4) insert G2 at the right side: p3m1_tile_part-3
        composite p3m1_G2.png p3m1_tile_part-2.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part-3.png
# 4.5) crop1/2 G5: p3m1_G5h
        convert p3m1_G5.png -gravity West -crop {int(1/2 * $triangle_width)+1}x{$triangle_height}+0+0 -background none +repage p3m1_G5h.png
# 4.6) insert p3m1_G5h right in p3m1_tile_part-3: p3m1_tile_part
        composite p3m1_G5h.png p3m1_tile_part-3.png -gravity East -compose Dst_Over -background none +repage p3m1_tile_part.png

# 5) Generate p3m1_tile
# 5.1) Generate p3m1_tile_part_flip.png
        convert p3m1_tile_part.png -flip -background none +repage p3m1_tile_part_flip.png
# 5.2) Generate tile_row
        convert p3m1_tile_part.png p3m1_tile_part_flip.png +append -background none +repage p3m1_tile_row.png
# 5.3) Generate tile_row_flop
        convert p3m1_tile_row.png -flip -background none +repage p3m1_tile_row_flip.png
# 5.4) Generate p3m1_tile
        convert p3m1_tile_row.png p3m1_tile_row_flip.png -append -background none +repage p3m1_tile.png
```