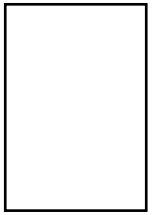
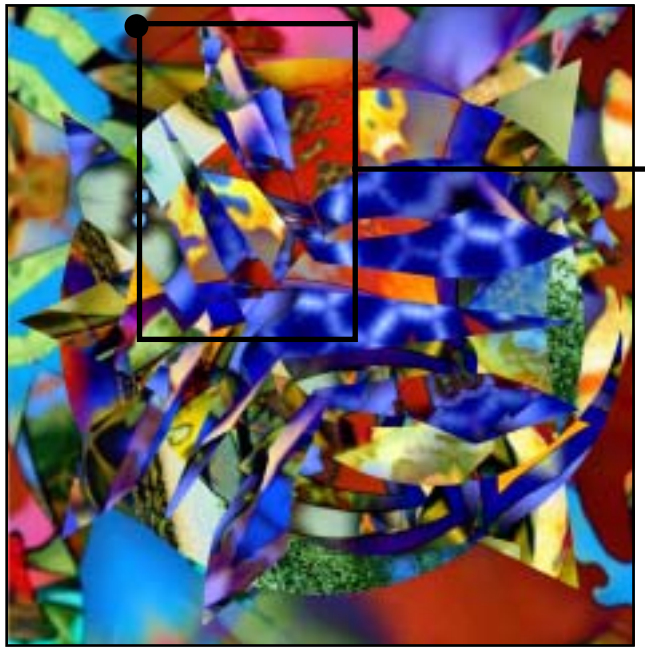
Draft V 0.9

Seamless Plane Groups with ImageMagick/PerlMagick: 1.1) pmm_tile

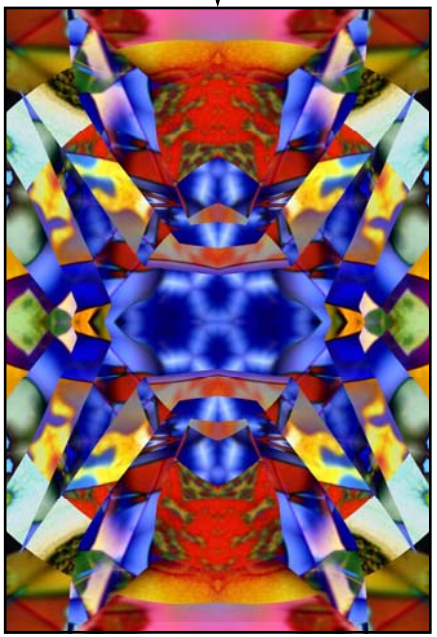
Günter Bachelier
www.aroshu.de AROSHU® Evolutionary Art © Günter Bachelier

pmm basic element:  or 



pmm_G

pmm_tile



given in the deterministic case
a) image.jpg with (image_width, image_height)
    given the example image: image.jpg with
    image_width = image_height: 4000 [pixel]
b) format of the basic element: (rectangle_width, rectangle_height)
    given the example: (1373,1985)
c) Top left point P1 in image.jpg for selection of pmm_G
    given the example: P1 = (839,131)

given in the stochastic case
a) image.jpg with (image_width, image_height)
b) random variable interval for the selection of triangle_width:
    [rectangle_width_min_Faktor, rectangle_width_max_Faktor] = [0.3, 0.7]
    [rectangle_height_min_Faktor, rectangle_height_max_Faktor] = [0.3, 0.7]

pnm-procedure:
Start
    1) Generate G
    2) Generate tile
End

Only relevant for the stochastic case:
1.1) Calculate scalare $rectangle_width, $rectangle_height, x1, y1

```
$rectangle_width_min_Faktor = 0.3;
$rectangle_width_max_Faktor = 0.7;
$rectangle_height_min_Faktor = 0.3;
$rectangle_height_max_Faktor = 0.7;
if ($image_width < $image_height) {$image_min = $image_width} else {$image_min = $image_height};
$rectangle_width_min = $rectangle_width_min_Faktor * $image_min;
$rectangle_width_max = $rectangle_width_max_Faktor * $image_min;
$rectangle_height_min = $rectangle_height_min_Faktor * $image_min;
$rectangle_height_max = $rectangle_height_max_Faktor * $image_min;
$rectangle_width = $rectangle_width_min + (int(rand($rectangle_width_max - $rectangle_width_min)) + 1);
$rectangle_height = $rectangle_height_min + (int(rand($rectangle_height_max - $rectangle_height_min)) + 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;
```

1) Generate basic element pmm_G

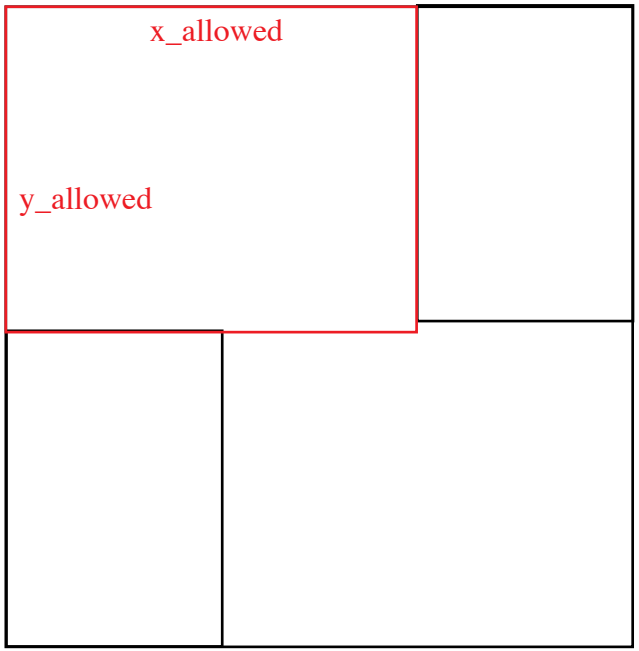
Copy from image.jpg a (random) rectancle with width = rectangle_width and height = rectangle_height

convert image.jpg -crop {$rectangle_width}x{$rectangle_height}+{$x1}+{$y1} +repage pmm_G.png

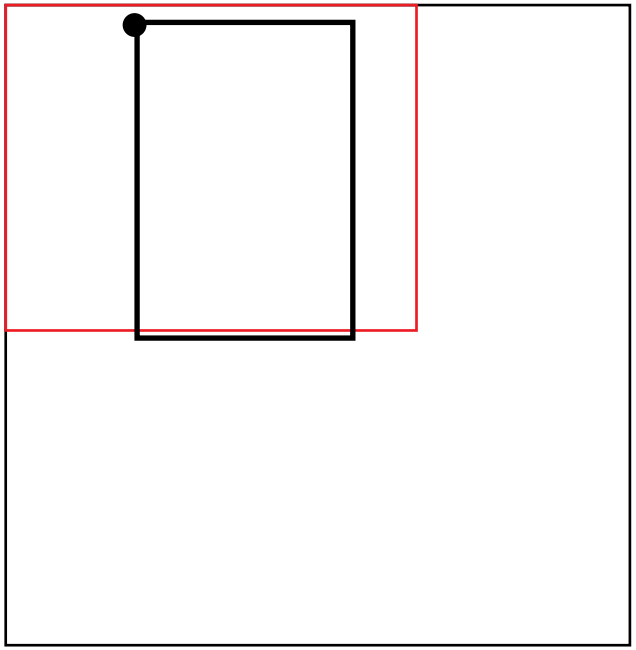
For the example holds:

convert image.jpg -crop 1373x1985+839+131 +repage pmm_G.png

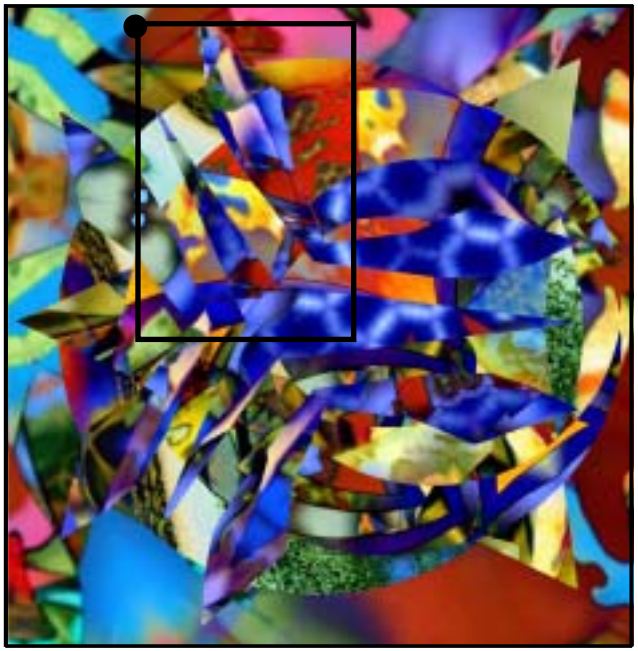
http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_pmm/image.jpg
http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_pmm/pmm_G.png



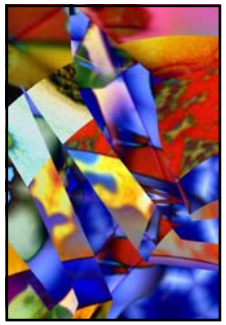
P1



P1



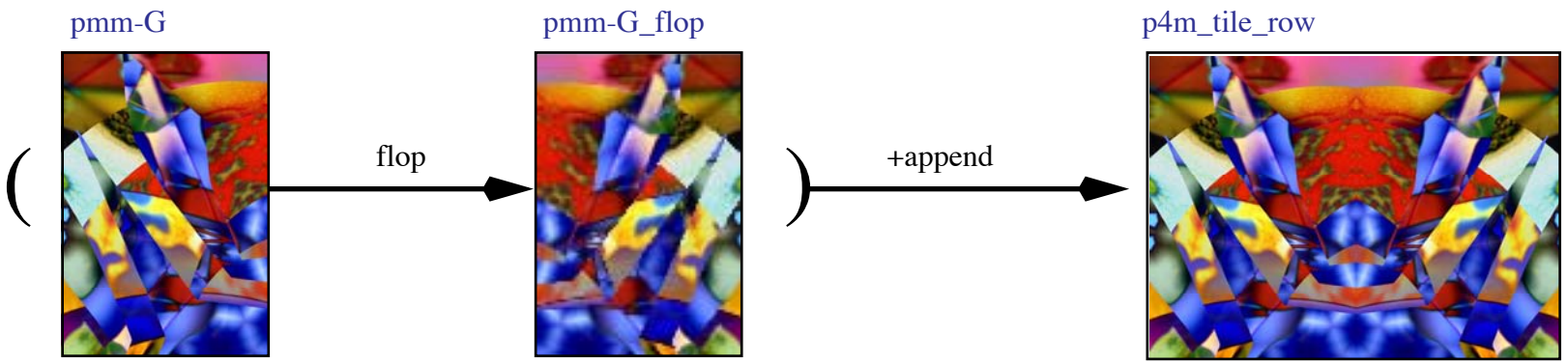
pmm_G.png



2) Generate tile

```
convert pmm-G.png -flop pmm-G_flop.png
convert pmm-G.png pmm-G_flop.png +append pmm-tile_row.png
```

http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_pmm/pmm_G_flop.png
http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_pmm/pmm_tile_row.png

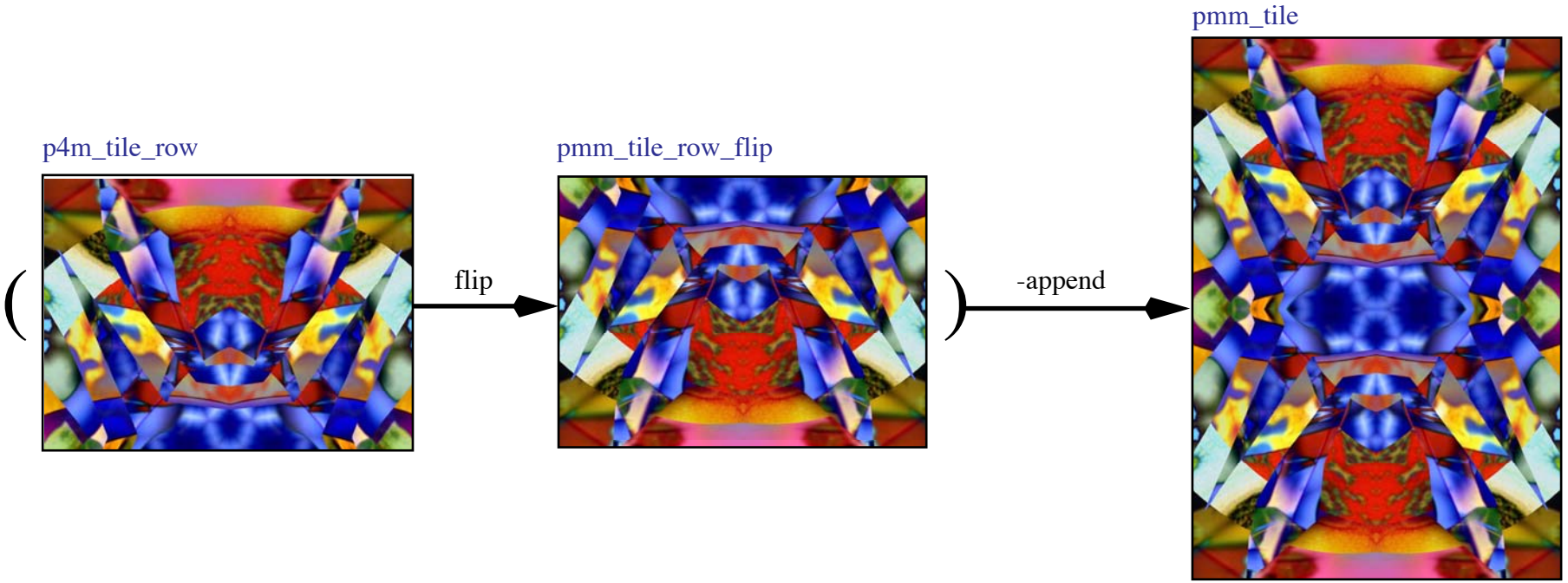


```
convert pmm_tile_row.png -flip pmm_tile_row_flip.png
convert pmm_tile_row.png pmm_tile_row_flip.png -append pmm_tile.png
```

Summarize tile generation from pmm_G:

```
convert pmm_G.png ( +clone -flop ) + append ( +clone -flip ) -append pmm_tile.png
```

http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_pmm/pmm_tile_row_flip.png
http://www.vi-anec.de/Trance-Art/IM-examples/IM-plane_group_pmm/pmm_tile.png



IM command sequence for pmm in general

```
convert image.jpg -crop {$rectangle_width}x{$rectangle_height}+$x1+$y1 +repage pmm_G.png
convert pmm_G.png -flop pmm_G_flop.png
convert pmm_G.png pmm_G_flop.png +append pmm_tile_row.png
convert pmm_tile_row.png -flip pmm_tile_row_flip.png
convert pmm_tile_row.png pmm_tile_row_flip.png -append pmm_tile.png
```

IM command sequence for pmm for the example

```
convert image.jpg -crop 1373x1985+839+131 +repage pmm_G.png
convert pmm_G.png -flop pmm_G_flop.png
convert pmm_G.png pmm_G_flop.png +append pmm_tile_row.png
convert pmm_tile_row.png -flip pmm_tile_row_flip.png
convert pmm_tile_row.png pmm_tile_row_flip.png -append pmm_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';
$pmm_resultfolder_path = "C:/ART/IM_PlaneCovering/2_Tiles/2_01_pmm-Tiles/";


# 0.2) definition of lokal parameters
$rectangle_width_min_Faktor = 0.4;
$rectangle_width_max_Faktor = 0.6;
$rectangle_height_min_Faktor = 0.4;
$rectangle_height_max_Faktor = 0.6;


# 1) definition of PerlMagick objects
$pmm_G = new Image::Magick;


# 2) inilisation of random number generator
srand;


# 3) read source image
$sourcefile_fullname = $sourcefolder_path . $sourcefile_name;
$pmm_G->Read("$sourcefile_fullname");
```

```
# 4) local variable calculations: $rectangle_width, $rectangle_height, $x1, $y1
$image_width = $pmm_G->Get('columns');
$image_height = $pmm_G->Get('rows');
if ($image_width < $image_height) {$image_min = $image_width} else {$image_min = $image_height};
$rectangle_width_min = $rectangle_width_min_Faktor * $image_min;
$rectangle_width_max = $rectangle_width_max_Faktor * $image_min;
$rectangle_height_min = $rectangle_height_min_Faktor * $image_min;
$rectangle_height_max = $rectangle_height_max_Faktor * $image_min;
$rectangle_width = $rectangle_width_min + (int(rand($rectangle_width_max - $rectangle_width_min)) + 1);
$rectangle_height = $rectangle_height_min + (int(rand($rectangle_height_max - $rectangle_height_min)) + 1);
$x_allowed = $image_width - $rectangle_width;
$y_allowed = $image_height - $rectangle_height;
$x1 = int(rand($x_allowed))+1;
$y1 = int(rand($y_allowed))+1;


# 5) generate pmm_G by cropping the source image
$pmm_G->Crop(geometry => "$rectangle_width x $rectangle_hight+$x1+$y1");


# 6) convert pmm_G.png -flop pmm_G_flop.png
$pmm_G_flop = $pmm_G->Clone();
$pmm_G_flop->Flop();


# 7) convert pmm_G.png pmm_G_flop.png +append pmm_tile_row.png
$q = $pmm_G->Clone();
push(@$q, $pmm_G_flop);
$pmm_tile_row = $q->Append(stack=>'false');
```

```
# 8) convert pmm_tile_row.png -flip pmm_tile_row_flip.png
$pmm_tile_row_flip = $pmm_tile_row->Clone();
$pmm_tile_row_flip->Flip();


# 9) convert pmm_tile_row.png pmm_tile_row_flip.png -append pmm_tile.png
@$q = ();
$q = $pmm_tile_row->Clone();
push(@$q, $pmm_tile_row_flip);
$pmm_tile = $q->Append(stack=>'true');


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

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

$resultimagepath_name = $pmm_resultfolder_path . $resultimage_name;


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