Report on Image warping

Xuan Nie, Dec. 20, 2004

This document summarized the algorithms of our image warping solution for further study, and there is a detailed description about the implementation of these algorithms.

CONTENTS

ALGORITHMS

Image warping is the process of deforming a digital image geometrically. The problem of image warping given a set of scattered translation vectors is essentially the problem of interpolating two functions simultaneously. One function is for the translation in the direction of the *x*-axis and the other for the translation in the direction of the *y*-axis.

Inverse distance weighted interpolation

Inverse distance-weighted interpolation methods were originally proposed by Shepard and improved by a number of other authors, notably Franke and Nielson.

Calculate weight at every point

For each data point P_i, a local approximation $f_i(P): R^2 \to R$ with

$f_i(P_i) = y_i i = 1,2,....,n$ is determined. The interpolation function is a weighted average of these local approximations, with weights dependent on the distance of the observed point from the given data points,

$$f(p) = \sum_{i=1}^{n} w_i(p) f_i(p)$$

where $f(p) = y_i, i = 1,.....,n.w_i : R^2 \to R$ is the weight function, which must satisfy the condition:

$$w_i(p_i) = 1, \sum_{i=1}^{n} w_i(p) = 1, \text{ and } w_i(p) \geq 0, i = 1,........,n.$$

These conditions guarantee the property of interpolation. Shepard proposed the following simple weight function:

$$w_i(p) = \frac{\sigma_i(p)}{\sum_{j=1}^{n} \sigma_j(p)} \text{ with } \sigma_j = \frac{1}{d(p, p_i)^{\mu}}$$

Where $\mathrm{d(p, p_i)}$ is the distance between p and p_i.

Radial basis function transform image warping

Transformations based on radial basis functions have proven to be a powerful tool in image warping. Images are regarded as *2D* objects. In this respect, an image is a finite domain of a plane with a grey level (or color) associated with each point. A warping of an image is then primarily a transformation of the plane to itself, and the grey level values are transformed according to the transformation of their associated coordinates.

Our main concern is the construction of a mapping of images (planes) that are determined by the mapping of some anchor points - points whose mapping is predetermined. This requirement leads us to interpolation theory.

.

● *Whole description*

In two-dimensional interpolation theory we deal with computing a function $T(\overline{x_i}) = \overline{y_i} : R^2 \to R^2$ satisfying $T(\overline{x_k}) = \overline{y_k} (k = 1,2,....,N)$ in the anchor points.

Radial basis functions have proven to be an effective tool in multivariate interpolation problems of scattered data. Given scattered points $\{\mathrm{x_i}\} \in R^2$, and corresponding data values $\{S_i\} \in R^2$, we look for an interpolatory function *T*(*x*) of the form:

$$T(x) = \sum_{i=1}^{N} a_i g(\|x - x_i\|) + Aq + \alpha_3$$

where $A = (\alpha_1, \alpha_2)^T, \alpha_i \in R^2, 1 \le i \le N, \alpha_i = \{\alpha_i^x, \alpha_i^y\} \in R^2$. Here $\|.\|$ denotes the usual Euclidean norm on R^2 and $g : R^+ \to R$. $T(x_i) = F_i, 1 \le i \le N$. A function of this form is usually referred to as radial sum with a linear item. Thus S is determined by *N*+3 coefficients in R^2. The computation of those coefficients involves the solution of two square linear systems of size *N+3* each, where *N* conditions are derived by the interpolation requirements.

To solve the equation, we can add an additional item and solve the linear system defined by following. The system of equations for the vectors of unknowns is: $u_k = (a_1^k, ..., a_N^k)^T$ and $v_k = (\alpha_1^k, \alpha_2^k, \alpha_3^k)^T$, $k = 1,2$. The linear system is described as:

$$Gu_k + Hv_k = b_k$$
$$H^T u_k = 0$$

Here $b_k = (y_1^k, ..., y_N^k)$, $G = \{g(\|q_i - q_j\|)\}_{i,j=1}^N$, and H is an $N \times 3$ matrix with ith row $\{q_i^1, q_i^2, 1\}, 1 \le i \le N$.

● *Choose the basis function*

Well-known radial basis functions are multi-quadrics, originally proposed by R.Hardy :

$$R(d) = (d^2 + r^2)^{\mu/2} \quad \text{with} \quad r > 0 \text{ and } \mu \ne 0$$

Here, $\mu = \pm 1$ has been used successfully.

General method for fold-over free image warping

In image warping, an additional problem to that of interpolation is that of maintaining the one-to-one property of the warping transformation. Each point in the transformed image should correspond to only one point in the original image and vice-versa. If the one-to one property is not satisfied the warped function will have the appearance of being folded, rather than only stretched.

.

● *Whole description*

The one-to-one property of the change of variables can be preserved using two facts. First, the concatenation of two one-to-one maps must be one-to-one, i.e. given two one-to-one mappings:

$(x, y) \to (f, g)$

With Jacobian $J_1 > 0$ and

$(f, g) \to (u, v)$

With Jacobian $J_2 > 0$, then the combined mapping:

$(x, y) \to (u, v)$

will have Jacobian $J_3 = J_1 J_2 > 0$ (by the chain rule of differentiation),and so be one-to-one.

Second, an overlapping function of the form $(x, y) \to (x + f, y + g)$ can be made one-to-one by scaling the functions *f* and *g* by an appropriate constant α satisfying $0 < \alpha \le \beta(x, y)$, so that the mapping $(x, y) \to (x + a \times f, y \to a \times g)$ is one-to-one. The values $\beta(x, y)$ are chosen at each point so that the Jacobian, *J*, of the rescaled mapping $(x + \beta \times f(x, y), y + \beta \times g(x, y))$, given by the equations $J = (\beta f_x + 1)(\beta g_y + 1) - \beta^2 f_y g_x$ (where the subscripts denote partial derivatives), is equal to zero. These quadratic equations can be solved at each point in the warping functions for β using “finite difference approximations” for the partial derivatives. We are only interested in solutions for which $0 < \beta \le 1$, because we require the interpolated function to translate the pixels a fraction of the desired shift $\beta > 0$ without overshooting the target ($\beta > 1$). The quadratic equation always passes through *J*=1 for $\beta = 0$, which leaves only three possible cases:

(1) No real roots between 0 and 1. In this case the Jacobian is positive for all $\alpha \in (0,1]$ so a is not restricted by this point.

(2) One real root b between 0 and 1. In this case *J* is positive for $0 < \alpha < \beta$.

(3) Two real roots β_0 and β_1 between 0 and 1. In this case a must be less than the smaller root or larger than the larger root for positive *J*.

Therefore, we set each $\beta(x, y)$ equal to the smallest positive real root on $(0,1]$ or 1 if there are no roots on $(0,1]$. The scaling parameter α is chosen to be less than or equal to the smallest value of b in the warping function.

- ***Implement steps:***

(1), Calculate the partial warping by a chosen method.

(2), Calculate the scaling factor for specified partial warp.

(3), Scale the partial warp

(4), Add partial warp to the total warp

(5), Check for convergence condition, if scaling factor is equal to 1, turn (7), if not, turn (6).

(6), Reposition start Points for next iteration.

(7), End this procedure, return total warp and apply it to whole image;

- ***How to calculate the scaling factor:***

For every point in an image, calculate a scaling factor by the following method and find the smallest one to return:

(1), Estimate partial derivatives of warp functions, which is f_x, f_y, g_x, g_y

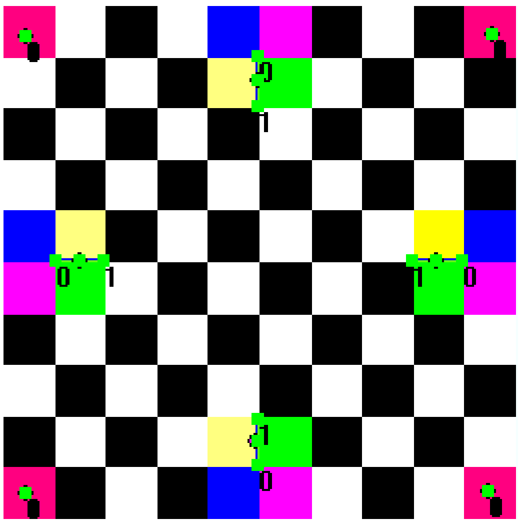
(2), Solve quadratic function ($f_x g_y - f_y g_x, (f_x - g_y), 1 - J_{\min}$) and returns the two real roots.

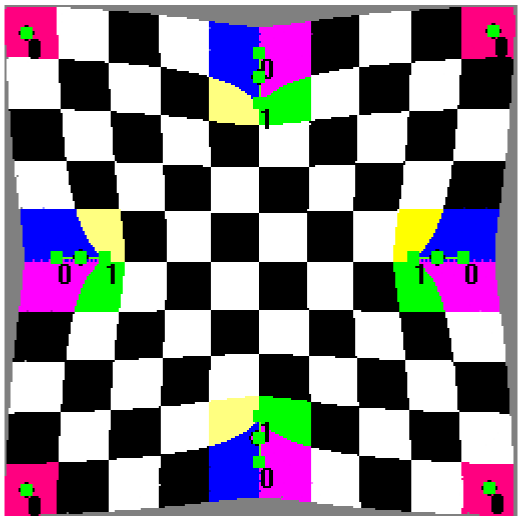
(3), Check all the cases to acquire the proper root.

RESULTS

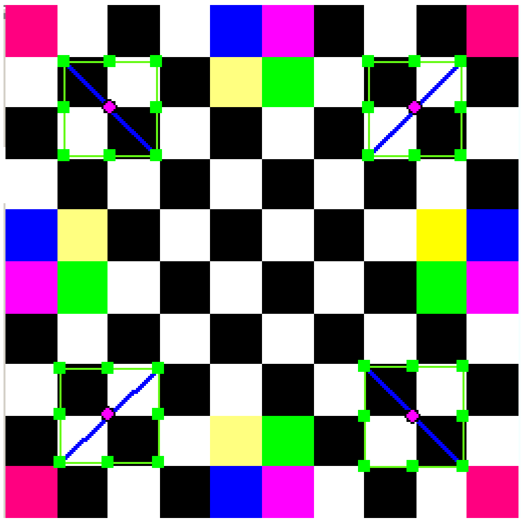
- **Inverse distance weighted interpolation methods**

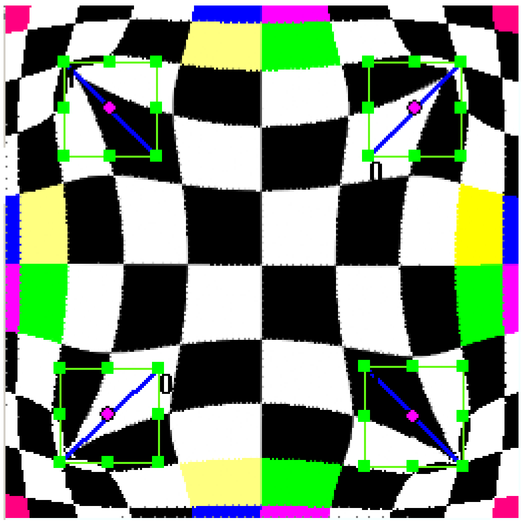
- **No fold-over free control(One Step)**



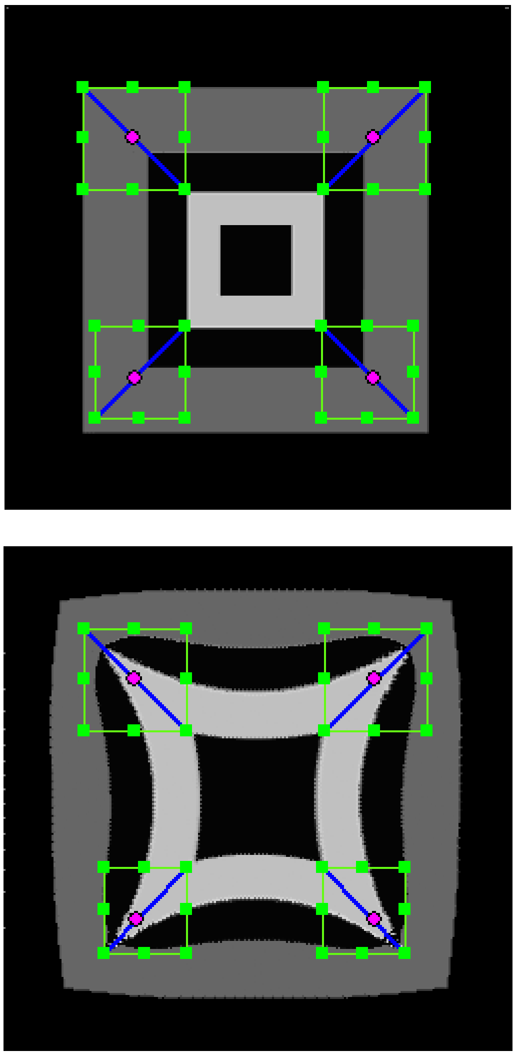


Example 1: μ =1





Example 2: μ =1

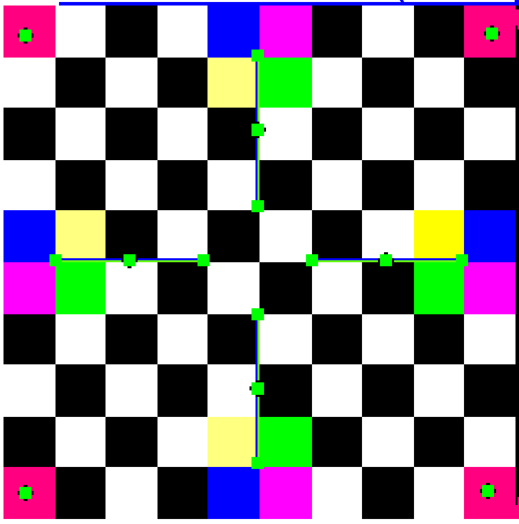


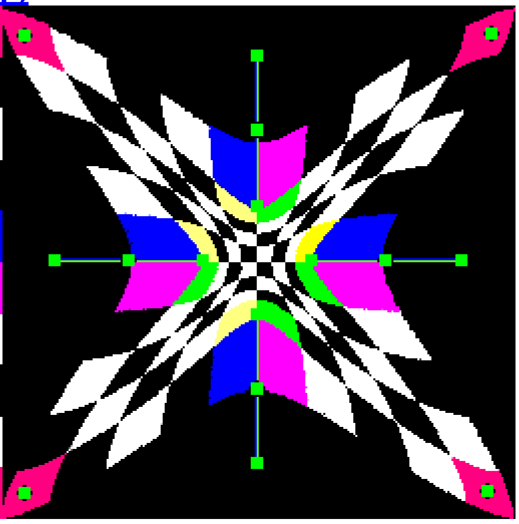
Example 3: μ =1

Radial basis function transform image warping

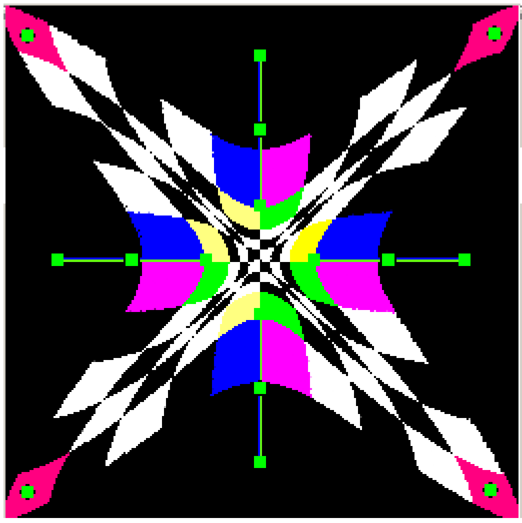
Basis function: $g(d) = (d^2 + r^2)^{u/2} \quad \mu = \pm 1$,

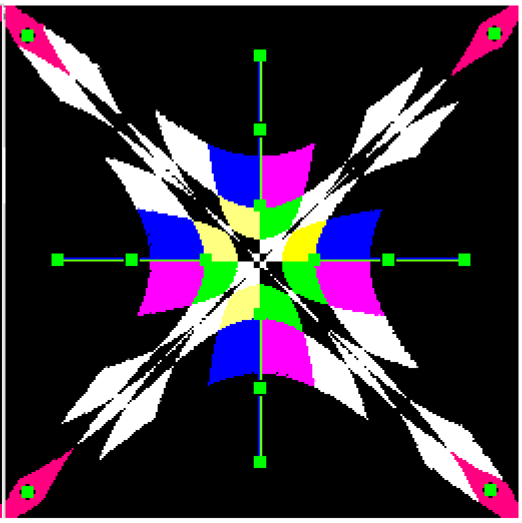
➢ **No fold-over free control(One Step)**



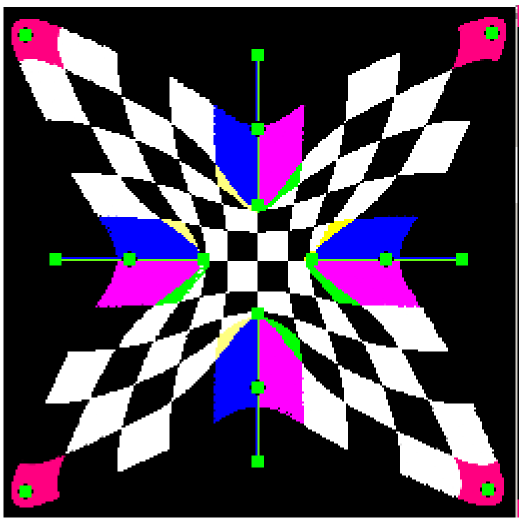


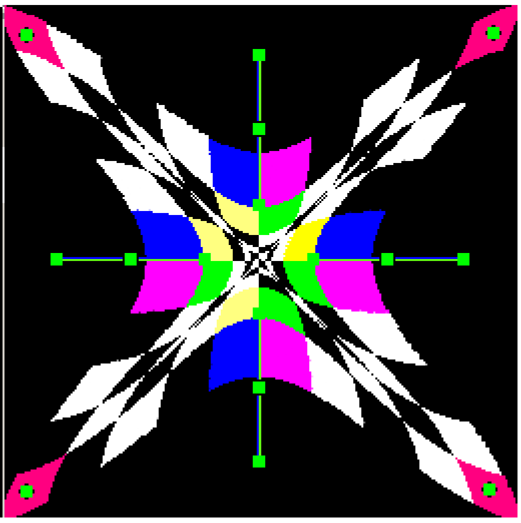
r=10, μ=1





r=25, μ=1 **r=80, μ=1**

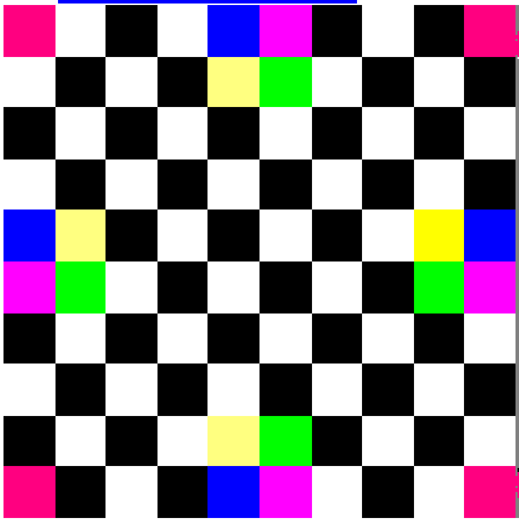


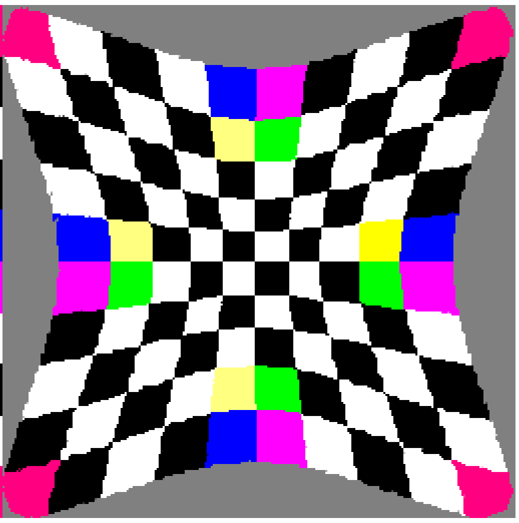


r=25, μ=-1 **r=80, μ=-1**

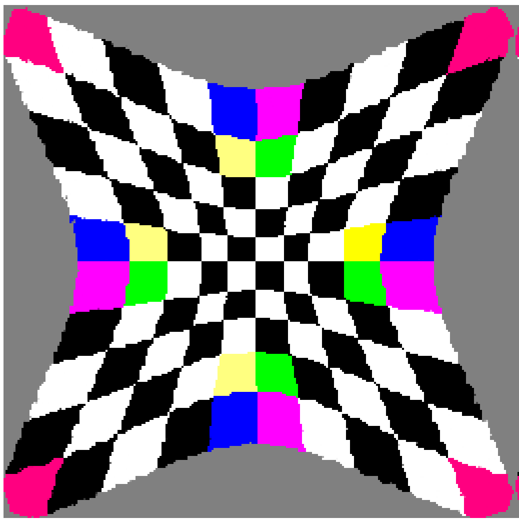
Example 1 Compare the result by different parameters in the case of no overlap free control

- **Fold-over free control**

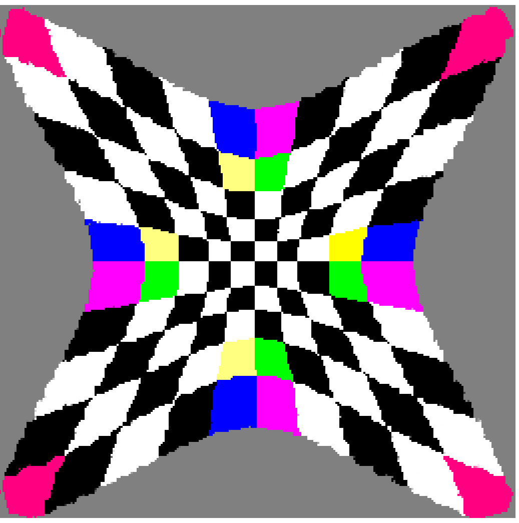




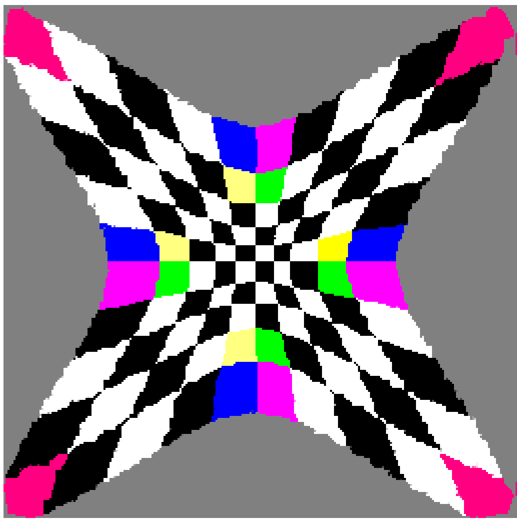
a = 0.147547

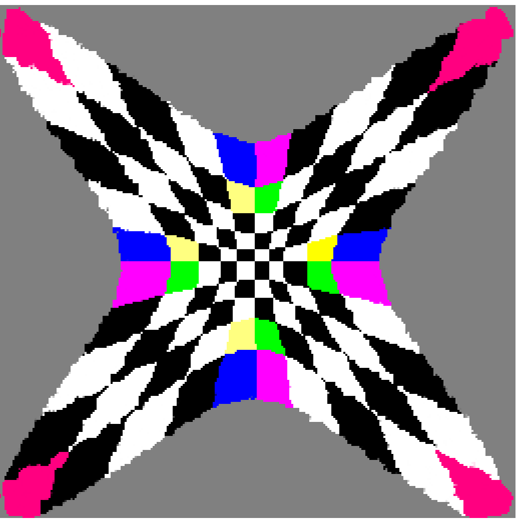


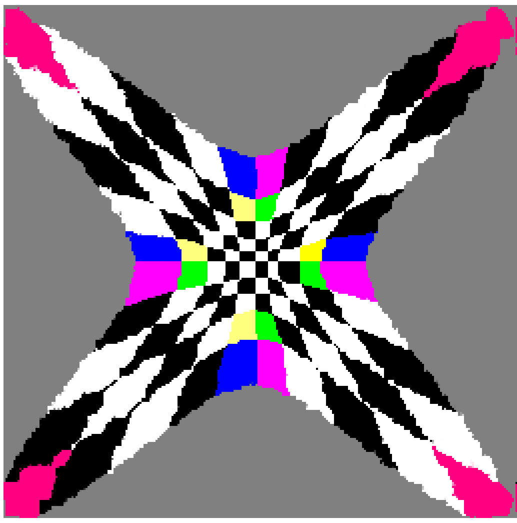
a = 0.156282

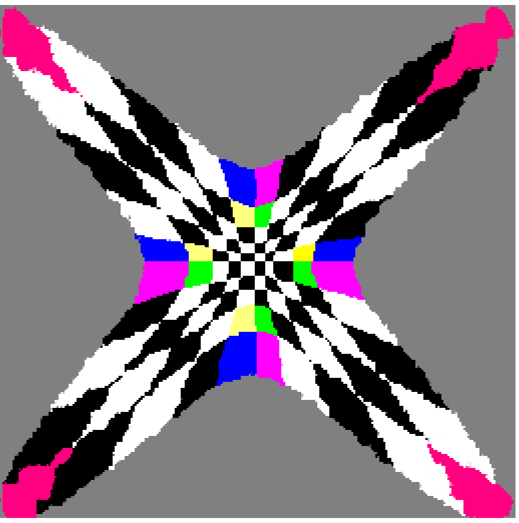


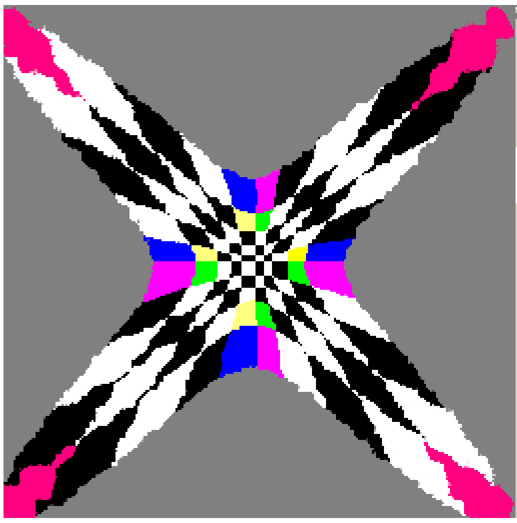
a = 0.172062

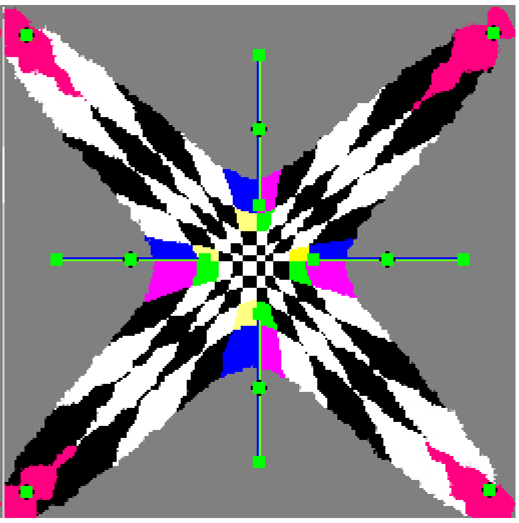

a = 0.206629


a = 0.239406

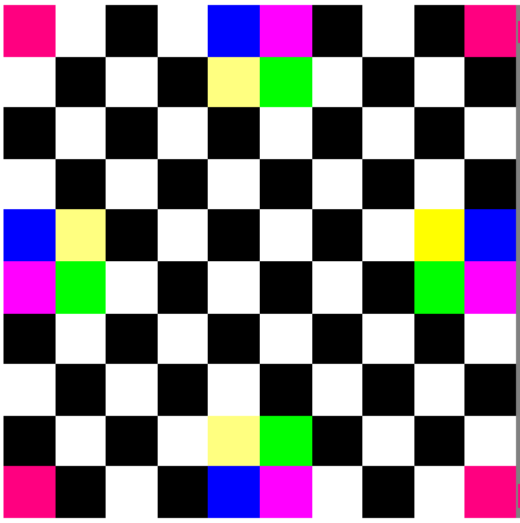

a = 0.268611

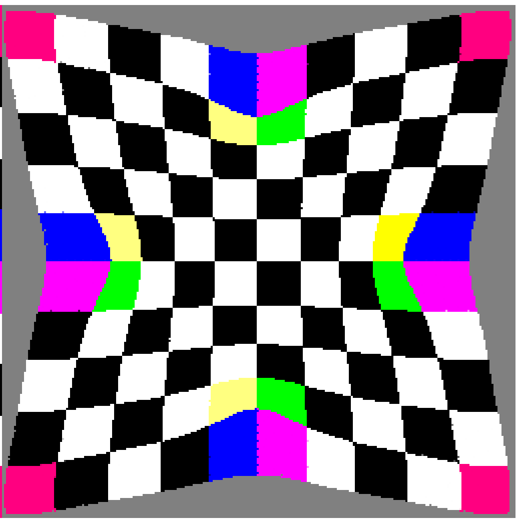

a = 0.339659


a = 0.554736

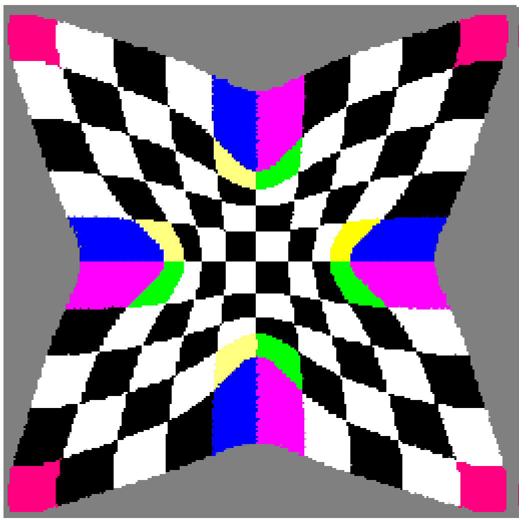

a = 1.000000

Example 1, One result with Overlap free control: r=80, μ =1

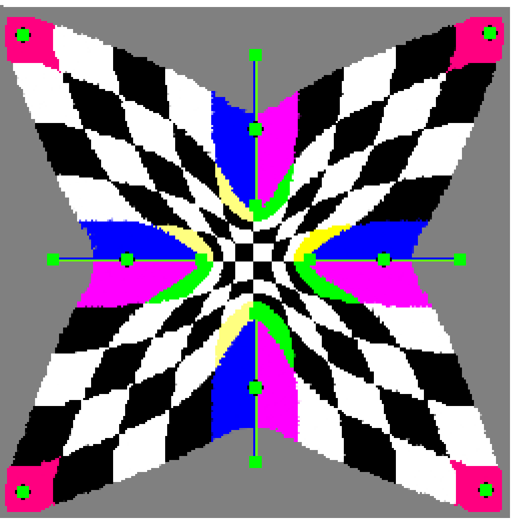




a = 0.391942

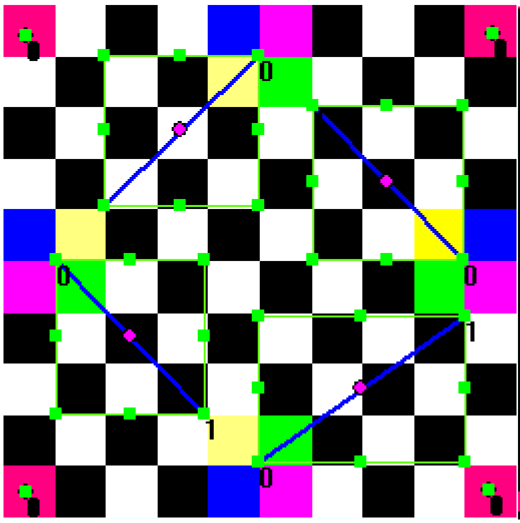


a = 0.589912

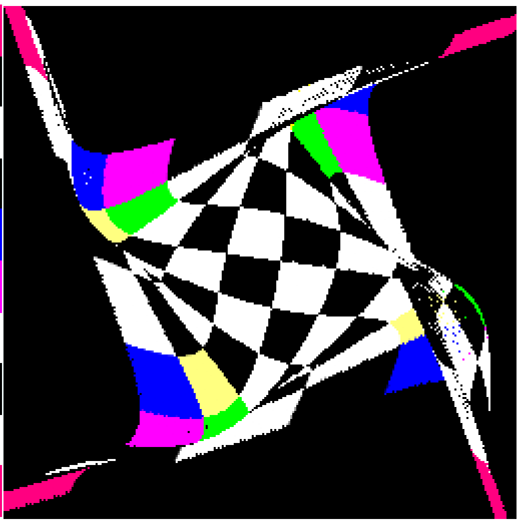


a = 1.000000

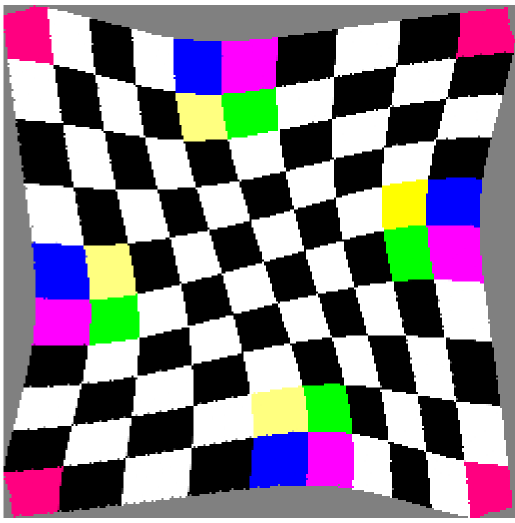
Example 2, One result with Overlap free control: r=25, μ=-1

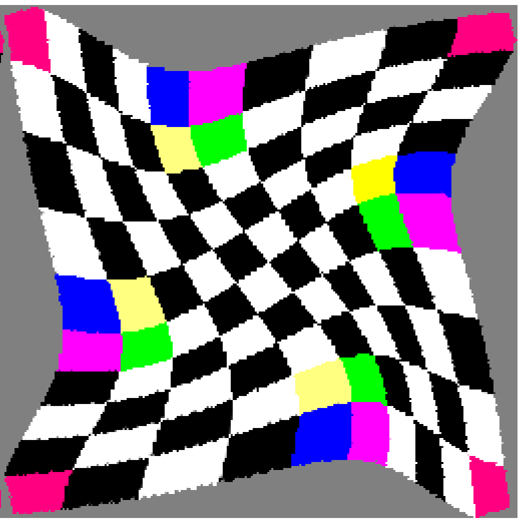


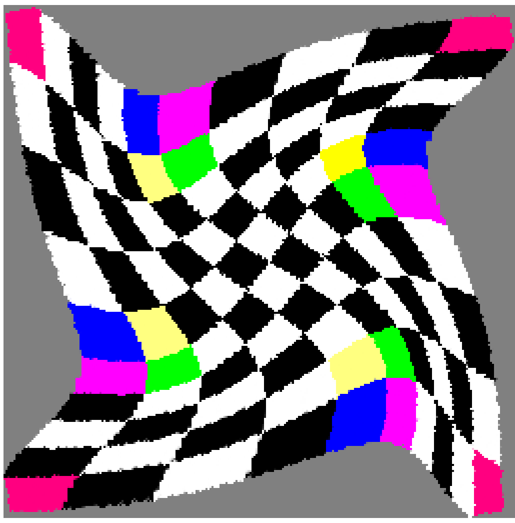
Specification

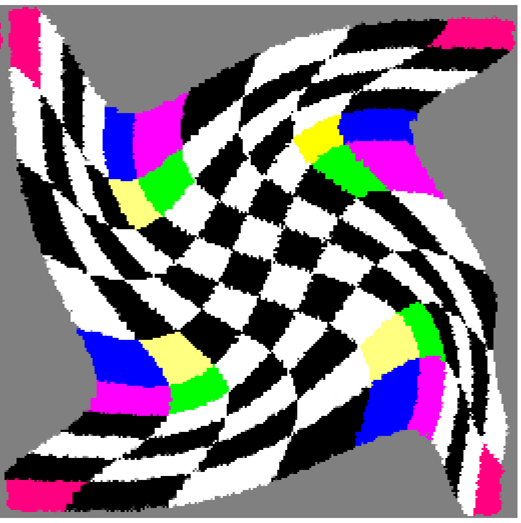


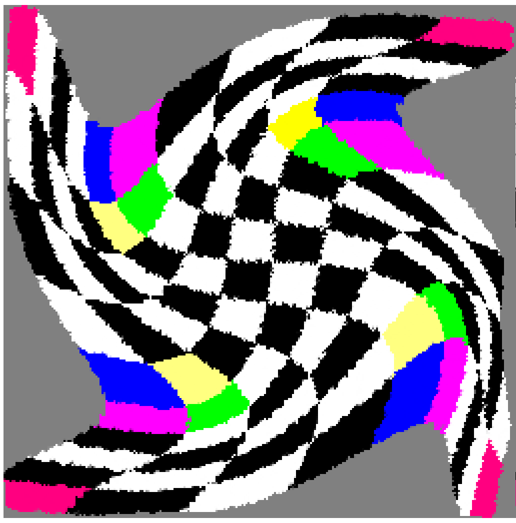
without Overlap free control

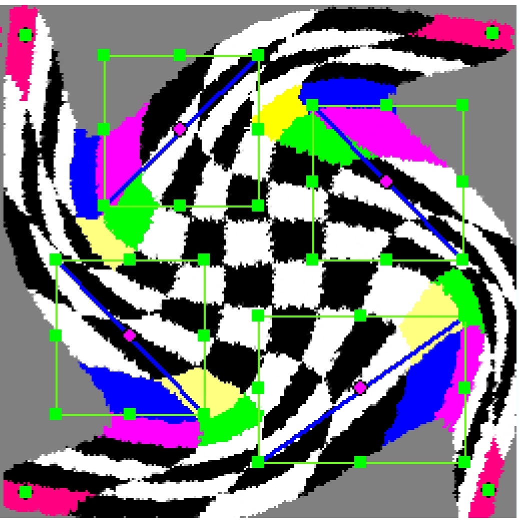

a = 0.241285


a = 0.275535


a = 0.321808

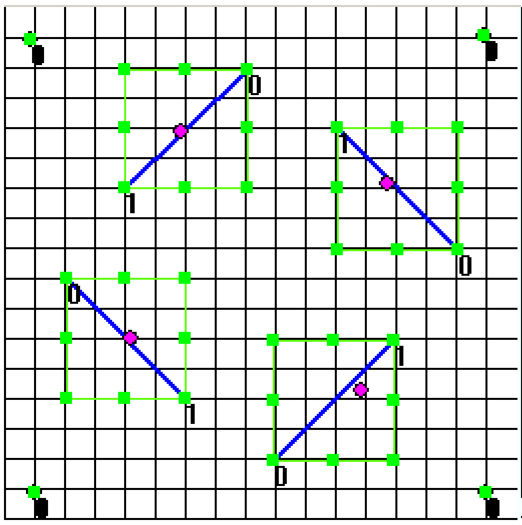

a = 0.414261


a = 0.595294

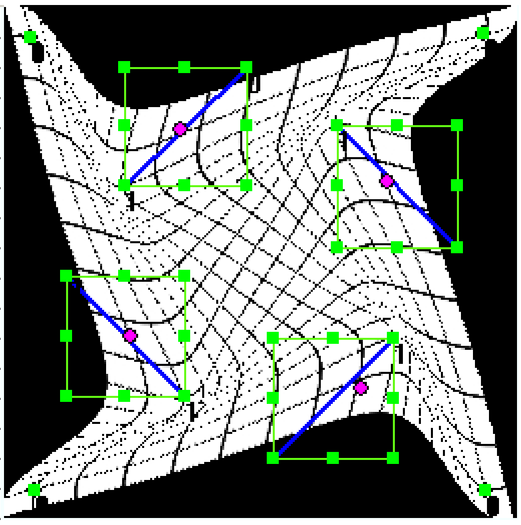

a = 1.000000

With Overlap free control

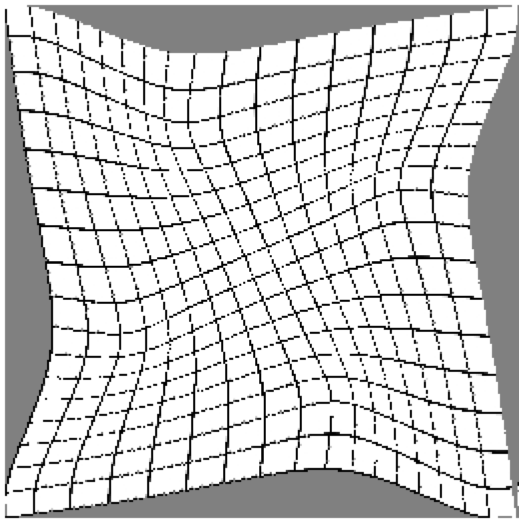
Example 2, Two results without and with Overlap free control: r=40, μ =1



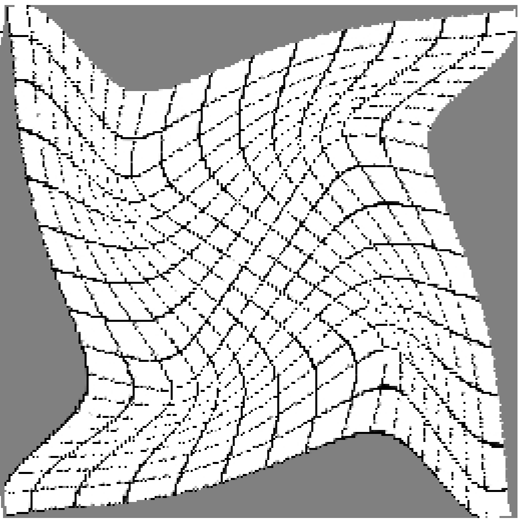
Specification



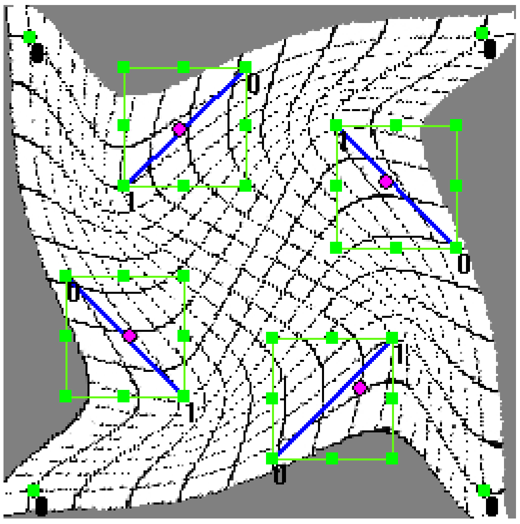
without Overlap free control



a = 0.467757



a = 0.848091



a = 1.000000

With Overlap free control

Example 3, Two results without and with Overlap free control: r=15, μ =1

FUTURE WORK

- Utilize the triangle dissection based method to establish overlap free solution.