Revision August 14, 1999

# *Notes on Inlay Software*

**Evgeny Kusikov**

## *Content*

# 1•Introduction

The problem to solve is formulated at http://inlay.com/cnc/software/software.htm in large. Now I'll reformulate it in more regular way to separate different tasks.

As supposed, there are the following **input information**:

**a)** photo of the artwork (JPEG or other format [1,2]), geometrical parameters of the outgoing inlay (width, height, etc.);

**b)** the basic tiles of material, which cuttings are supposed to make up the outgoing inlay - their photos and geometrical parameters (width, height, etc.). Example - the photos of tiles at http://www.adcdommel.com/tiles.htm photos;

**c)** parameters of back-end software and hardware (ex. http://www.seanet.com/~dmauch/, http://inlay.com/cnc).

**So, in large the problem to deal with is:**

1. to choose appropriate regions of the photo;
2. to reorganize the regions (split and/or merge) and to choose appropriate tile for every region to make up the photo;
3. to construct output inlay path information as needed by back-end software.

**Requirements and rationale** in the text are marked at the beginning of the paragraphs in the following manner:

- those, that concern hardware - by **(HR)**;
- those, that concern back-end software - by **(SR)**;
- those, that concern tiles - by **(TR)**;
- those, that concern resulting inlay - by **(IR)**.

# 2•Basic considerations. Overall structure of the tool.

**important moments**

! • The tiles we have, are essentially different;

**(TR)** • The tile (see http://www.adcdommel.com/tiles.htm photos) represents a range of colors in the color cube (spectrum, RGB-cube), e.g. a color class;

**(TR)** • Different tiles represent different color classes. Different color classes do not intersect;

**(TR)** • There are (approximately) 30 different tiles - different color classes, representing colors in the resulting inlay. So, there is no use choosing regions in the photo (artwork), representing more than 30 different colors.

☞ • The tiles quantize the color cube and that information does not depend on any input photo. That features are put in the database (See “Database” on page 3.).

**<u>The tool consists of 3 parts</u>** (as supposed for now):

1. the database;
2. the core: converters and viewers;
3. the output generator.

# *3•Database*

## *3.1•Tile information*

NoT - the number of the actual (real) tiles;
NoLT - the number of the virtual (logical) tiles (See 3.4.2 on page 4).

### *3.1.1•Tile geometric information*

Such information consists of the shape and measures for actual (real) tiles

| Shape | Measures | Comments |
|---|---|---|
| square | $a\,('') \times a\,('')$ | $a = [\mathbf{3/8''}, \ldots, \mathbf{8''}]$ |
| rectangular | $a\,(mm) \times b\,(mm)$ | $a = \mathbf{10}mm, b = \mathbf{200}mm$ |
| triangular | $a\,(mm) \times b\,(mm) \times c\,(mm)$ | $a = \mathbf{50}mm, b = \mathbf{50}mm, c = \mathbf{70}mm$ |

and the shape, measures, actual tile and disposition for virtual (logical) tiles (See 3.4.2 on page 4).

### *3.1.2•Tile color information*

Tile color information is:

$$TQC = \{N_T = NoLT, Tc_{\mathbf{1}}, \ldots, Tc_{N_T}, TcR_{\mathbf{1}}, \ldots, TcR_{N_T}, Indx^T\}$$

where:

$N_T$ - number of classes, equals to the number of the virtual tiles NoLT;

$Tc_{\mathbf{1}}, \ldots, Tc_{N_T}$ - colors, representing the color classes;

$TcR_{\mathbf{1}}, \ldots, TcR_{N_T}$ - regions of the color classes;

$Indx^{T} \circ RGB\text{–}color \rightarrow \{Tc_1, \ldots, Tc_{N_T}\}$ - color mapping function.

### 3.2•Hardware information

1. **D** - router bit diameter. It may range rather significantly: from 0.1 mm to 5 sm.;
2. **mmd** - minimum moving distance of the router bit: ranges from 0.00025″ to 0.001″;

### 3.3•Inlay information

Inlay information consists of geometric and color information.

#### 3.3.1•Inlay geometric information

We deal with two spaces: pixel and lengthy. Pixel coordinates are denoted by the regular font letters, but spatial coordinates are denoted by the *italic font letters.*

- In pixel space we have: $w_I$ - width and $h_I$ - height of the artwork photo in pixels ($w_I \times h_I$).
- In spatial space we have: $\mathit{w_I}$ - width and $h_I$ - height of the artwork photo in inches/mm/sm.
- And two coefficients: $R_I^w = \frac{w_i}{w_I}$ - width extension and $R_I^h = \frac{h_i}{h_I}$ - height extension.

#### 3.3.2•Inlay color information

It is just like the tile color information and is described in the following algorithms (See “initial photo analysis:” on page 5.).

### 3.4•Basic algorithms

#### 3.4.1•Algorithm to obtain tile color information.

**ALGORITHM 1:**

1° for every tile choose its representing color $Tc_i$ according to class choosing algorithm [6] to get one class. After that, we get $\{Tc_1, \ldots, Tc_{N_T}\}$ set for $N_T$ tiles;

2° construct $TcR_1, \ldots, TcR_{N_T}$ regions by algorithm in [7]. As the first version simple algorithm to construct 3-D Voronoi’ diagram for $\{Tc_1, \ldots, Tc_{N_T}\}$ could be used;

3° As the first version of $Indx^T$ function, the function choosing the minimum distance in a straight forward manner could be used.

### *3.4.2•Algorithm to obtain virtual (logical) tiles*

As a tile represents a texture, but not the only one range of colors, we can split it into several, say $N^t$, virtual tiles, e.g. color ranges. It is done by the following algorithm.

**ALGORITHM 2:**

1° Let us take the tile photo as a inlay photo.
Initial number $N^t$ could be taken from histogram of the tile photo, or simply taken ad hoc;

2° carry out steps 1.i and 2.i-2.ii of the basic steps on page 5 with $N_I = N^t$;

3° steps dealing with geometry are dropped by now!

## *4•Core*

**The goal of the core is**:

1. to choose appropriate regions of the photo;
2. to reorganize the regions (split and/or merge) and to choose appropriate tile for every region to make up the photo;

**important moments**

(IR) • if there are two adjacent regions A and B, than the contour line separating that regions, the corresponding bounding line segment for region A and the corresponding bounding line segment for region B are just the same line.

<u>This goal of the core is achieved by the following basic steps:</u>

1. initial photo analysis:
   i. obtain the inlay color information:

$$IQC = \{N_I = N_T, Ic_{\mathbf{1}}, \ldots, Ic_{N_I}, IcR_{\mathbf{1}}, \ldots, IcR_{N_I}, Indx^I\}$$

by applying class choosing algorithm [6] to get $N_I$ classes and constructing $IcR_{\mathbf{1}}, \ldots, IcR_{N_I}$ regions by algorithm [7];

ii. estimate the correspondence IQC to TQC:
“richly” or “poorly” colors of IQC could be represented by TQC - if “many” colors of IQC belong to one class of TQC, it’s “poorly”;

iii. visualize the results and if “not so bad” go to the next;

2. initial regions construction [8,9,10]:

i. transform the photo by substituting its colors by their TQC representing colors (call it mphoto);

ii. separate mphoto into $N_I$ layers according to each color class (so, each layer could be thought of as bitmap (black&white) image).
For every region:

- attribute internal and border pixels;
- estimate: $\underline{d}_R$ - internal diameter, $\bar{d}_R$ - external diameter and $S_R$ - area of the region.

Sort regions due to region area in decreasing order; throw off the regions which are “too small”;

iii. construct **region adjacency graph** (RAG). Analyze it: construct the corresponding **full connected component graph -** the splitting of the RAG according to the fully connected subgraphs;

iv. visualize the results and if “not so bad” go to the next;

3. initial contour line construction:

i. for every region obtain the representation of it as a segmented (basic) curves by one of the method of [11,12,13].
Transform it to the representation of the region by minimum basic points and curves.
Going that way the next step (4.i) turns out to be very mathematically difficult;

ii. **Another variant** is to use thinning technique [14]:

- construct border line region - it possesses only adjacent border pixels of the regions borders;
- apply thinning transformation to that region;
- obtain the representation of the segment by the basic points and segmented curves [12];
- **the nice moment** - the “unpleasant” step 4.i is dropped;

iii. visualize the results and if “not so bad” go to the next;

4. contour lines adjustment:

i. for two different adjacent regions we should adjust border line segments to the only separating line (See "Important Moments" on page 5):

- find out segments of the two border lines that are “closer” (?) to each other than other segments and construct the “joint” line segment;
- correct the initial border lines;

ii. check the bifurcation point;

iii. visualize the results and if “not so bad” go to the next;

After that step the basic segmentation of the photo is found. The basic means that other region manipulations are done on the requirement of the back-end software only.

5. Basic region analysis to meet the requirements of the back-end software through splitting, triangularization, etc. ----- TO BE DETAILED!

## 5•Output generator

## 6•Reference.


1. C. Wayne Brown, Barry J. Shepherd
   Graphics file formats: reference and guide. Prentice Hall, 1995
2. J.Levine
   Programming for Graphics Files in C and C++.John Wiley&Sons, Inc., 1994
3. Hearn D, Baker M.P.
   Computer Graphics, 2nd ed., Prentice Hall, 1994
4. Russ John C.
   The Image Processing Handbook, 2nd ed., CRC Press, Inc. 1995
5. Chellappa Rama
   Digital Image Processing, IEEE Computer Society Press, 1993
6. L.Velzo, J.Gomes, M.Sobreiro
   Color Image Quantization by Pairwise Clustering, SIBGRAPI 97, Proceedings of the X Brazilian Symposium of Computer Graphics and Image Processing, p.203-210
7. C.A.Kapoutsis, C.P.Vavoulidis, I.Pitas
   Morphological Techniques in the Iterative Closest Point Algorithm, Proceedings of the 1998 International Conference on Image Processing (ICIP98), p.808-812
8. F.Meyer
   Levelings and morphological segmentation SIBGRAPI 98, Proceedings of the International Symposium on Computer Graphics, Image Processing and Vision, p.28-35
9. S.Ji, H.W.Park
   Image segmentation of Color Image Based on Region Coherency Proceedings of the 1998 International Conference on Image Processing (ICIP98), p.80-83
10. H.Tao, T.S.Huang
    Color image edge detection using cluster analysis, Proceedings of the 1997 International Conference on Image Processing (ICIP97), p.834-836
11. G.Melnikov, G.M.Schuster, a.K. Katsaggelos
    Simultaneous optimal boundary encoding and variable-length code selection, Proceedings of the 1998 International Conference on Image Processing (ICIP98), p.256-260
12. M.Mokhtari, R. Bergevin
    Multiscale Segmentation and Approximation for Significant Description of 2D Contours, Proceedings of the 1997

International Conference on Image Processing (ICIP97), p.212-215

**13.** A.E.Fabris, A.R.Forrest
High Quality Rendering of Two-Dimensional Continuous Curves, SIBGRAPI 97, Proceedings of the X Brazilian Symposium of Computer Graphics and Image Processing, p.10-17

**14.** L.Lam, S-W.Lee, C.Y.Suen
Thinning Methodologies - A comprehensive Survey, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, Sept. 1992, pp. 869-885