Model implementation and image data compression in a decision support system

by Xiaobei Wang

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science
Montana State University

Abstract:
The PAYSAGE (landscape in French) model was modified to provide much improved efficiency and flexibility. The bound setting algorithm was used to replace the exhaustive algorithm in the previous implementation for generating dispersed population distribution. A new module was included to output JPEG format in addition to the default SVF format, which provides much more ways for information displaying and exchanging.

Generally it only took less than 50% time for the bound setting algorithm to generate a equivalent dispersed distribution, in terms of dispersion index, as those from exhaustive algorithm. With little sacrifice of dispersion, the bound setting algorithm could save even more time.

JPEG compression technology wisely removes high frequency details in an image, and then compresses the gross part of the image. That makes JPEG compression can get dramatically compression ratio, and to achieve one of the hottest compression technologies.

Vll

MODEL IMPLEMENTATION AND IMAGE DATA COMPRESSION

IN A DECISION SUPPORT SYSTEM

by

Xiaobei Wang

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 1997

APPROVAL

of a thesis submitted by

Xiaobei Wang

This thesis has been read by each member of the committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

11/20/97
Date

Co-Chairperson, Graduate Committee

11/20/97
Date

Co-Chairperson, Graduate Committee

Approved for the Major Department

11/20/97
Date

Head, Major Department

Approved for the College of Graduate Studies

12/16/97
Date

Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U. S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Signature Xiaobei Wang

Date Nov. 21, 1997

ACKNOWLEDGMENTS

First of all I would like to express my gratitude to my academic and research advisor Dr. Denbigh Starkey for his guidance, encouragement and support that enabled me to achieve my goal. I would like to thank my research co-advisor Dr. Andrew Hansen for his guidance and admittance to his research group. Also, I would like to thank my committee member Anne Defrance. I have learned so much in such a short time that I would like to thank everybody in Computer Science Department for his/her friendship and sharing of his/her expertise.

I especially acknowledge my family (my husband Xiao Chen), my father and my mother, and my younger brother Huinan Wang, with my love and gratitude for their care for all time.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The PAYSAGE (landscape in French) model was modified to provide much improved efficiency and flexibility. The bound setting algorithm was used to replace the exhaustive algorithm in the previous implementation for generating dispersed population distribution. A new module was included to output JPEG format in addition to the default SVF format, which provides much more ways for information displaying and exchanging.

Generally it only took less than 50% time for the bound setting algorithm to generate a equivalent dispersed distribution, in terms of dispersion index, as those from exhaustive algorithm. With little sacrifice of dispersion, the bound setting algorithm could save even more time.

JPEG compression technology wisely removes high frequency details in an image, and then compresses the gross part of the image. That makes JPEG compression can get dramatically compression ratio, and to achieve one of the hottest compression technologies.

INTRODUCTION

Timber harvesting affects both composition and structure of the landscape and thus it has important consequences for organisms using forest habitats. For example, forest interior conditions are thought to be important for many forest interior species [6], and the distribution, size and shape of harvest operations such as clear-cut obviously impact the amount and location of the remaining forest interior [4].

Many studies have usually been conducted by first describing spatial patterns and then inferring the characteristics of the processed that produced them. It has merely been possible to directly manipulate the disturbance process to examine the effect on spatial pattern and consequences for forest habitats.

Resource planners and managers rarely examine the long-term ecological consequence of specific management actions or compare the impacts of alternative management approaches in both spatial and temporal context because the lack the analytical tools and basic scientific information to do so.

Franklin and Forman [3] developed one of the first landscape management simulation models. Their simple geometric model considered just two stages of vegetation: forest and clear-cut. Li [8] developed the LSPA model based on the original Franklin and Forman model. LSPA simulated several forest age classes and modified the landscape at each time step according to a user-specified harvest regime [9]. The model also calculated several landscape spatial metrics at each time step. FRAGSTATS (McGarrigal and Marks 1995) offered a more comprehensive set of landscape metrics the

LSPA. And Wallin's CASCADE model [15] has improved upon the timber harvest subroutines relative to LSPA. Hansen [5] introduced a landscape model PAYSAGE (landscape in French). PAYSAGE integrates the GIS ARC/INFO (Environmental systems Research institute 1990), CASCADE, FRAGSTATES, a vegetation subroutine, and a vertebrate habitat subroutine [8]. PAYSAGE was designed to quantify landscape pattern, classify habitat suitability for vertebrate species, and simulate change in vegetation and habitat in the future under different management scenarios[6].

SVF format is used in GIS systems. But it is not PC based and can't be directly viewed by Web browser. A more popular file format is needed to seek, like JPEG, which is widely supported and can be very easy to share on the Web.

In recently years, more and more computer techniques are used in almost every area of science. PAYSAGE [7] is widely used in research and teaching. Although computer is getting more powerful than ever, a new version of PAYSAGE is demanded for more efficiency and flexibility.

The objectives of this study were to :

1. Find more efficient algorithms for timber harvest.

2. Embed some data compression techniques to output popular image file formats.

METHODS

PAYSAGE simulates vegetation and habitat dynamics across complex landscapes under different land use scenarios. Designed as a decision-support system for forest managers, this model is flexible in the quality of input data, spatial and temporal scale of application, and the geographic location of application.

PAYSAGE is composed of five primary components. The ACR/INFO GIS (Environmental Systems Research Institute 1990) is used to store, manipulate and display spatial data. Change in vegetation is simulated as a function of stand age, geomorphic setting, and management history. Vegetation patterns across the landscape are quantified with the landscape metrics model FRAGSTATS (McGarrigal and Marks 1995). The subroutine HABCLASS classified habitat suitability of individual vertebrate species based on vegetation type, landscape patterning, and geomorphic setting. Finally, timber harvest and other silvicultural manipulations are simulated with the disturbance subroutine from CASCADE (Wallin et al. 1994). Thus, the model can be used to quantify landscape patterns and vertebrate habitat suitability for a current landscape. Additionally, the model can be used to simulate changes in these variables over time under one or more landscape management scenarios[5].

The main program allocates memory, read input files, and calls the subroutines. The disturbance subroutine of CASCADE, FRAGSTATS and HABCLASS are programs that input data at each time, perform their functions and output result data.

When the disturbance subroutine is called, silvicalutral treatments are specified in terms of harvest patch size, harvest unit dispersion over the landscape (e.g., aggregated, dispersed) and rotation length. Timber harvest can be constrained in PAYSAGE based on stand type, stand age, adjacency rules, proximity to streams, elevation and other factors[5].

Disturbance (Timber harvest) model

The disturbance (timber harvest) model operates on a simple grid landscape and simulates landscape pattern dynamics in response to forest cutting and subsequent regrowth. FRAGSTATS and HABCLASS will go on to do further analysis based on the new landscape pattern after timber harvest.

The effects of alternative future forest cutting scenarios on landscape patterns were examined by using the disturbance model. Forest regrowth is indexed as time since disturbance. the user specifies a cutting rate (rotation length), minimum stand age eligible for harvest, adjacency constraints on cutting and either an aggregated or dispersed distribution of cuts.

Besides 4 layers of the landscape map (age, region, type, history), the model requires a map that defines "Logical Harvest Units" (LHUs) for the study area. The size and position of the LHUs were defined by a forest engineer, based on regulatory, engineering and other logistical constraints on log removal, road placement and environmental protection. Topography and the position of the stream network are the primary basis for defining these constraints. As cutting proceeds, individual LHUs are selected and cut completely; cutting only a portion of a unit is not permitted.

In the simulations, the initial unit to be cut was selected at random or user specified. To select each subsequent unit, all units eligible for cutting are examined and the unit that either has maximum sum distance (dispersed cutting) or has minimum sum distance (aggregated cutting) to all other cut units within the current considering region is selected as the next unit to be cut.

Studying pattern by distance measurement

The main function of the disturbance model is to decide the next cut unit in a region under specified harvest scenarios (dispersed cutting or aggregated cutting). That requires calculation and comparison of the distance between two harvest units (LHUs) all the time. To avoid repeatedly calculation distance between two harvest units, at the initial stage, a global distance look-up table dist[n][n] (n is the number of LHUs in the considering landscape) is defined to maintain the distance from each LHU centroid to every other LHU centroid.

Test for randomness based on distance measurements

Combinations of distance measurements and density estimates have been widely used to construct tests for randomness and indices of aggregation. Two such procedures are described here.

a) The test due to Hopkins and Skellam (1954) hinges on the fact that if, and only if, a pattern is random, the distribution of the distance from random point to its nearest plant is identical with the distribution of the distance from a random plant to its nearest neighbor.

Denote by w_1 the square of a point-to-plant distance and by w_2 the square of a plant-to-neighbor distance and suppose a sample is obtained of n distances of each kind. The statistic $A=\Sigma w_1/\Sigma w_2$ then has an expected value of 1 if the pattern is random and A may be used as measure of nonrandomness. Clearly, if the plants are aggregated, we shall have $A>1$; conversely, if they are more evenly spaced than in a randomly dispersed population, $A<1$ [7].

b) The test proposed by Clark and Evan (1954) requires knowledge of population density and a sample of n values of r, the distance from a random plant to its nearest neighbor. These distances are not squared.

Let p be the number of plants per unit area; then the randomly disperse population $E(r) = (2 * \text{sqrt}(p))^{-1}$. Write r^- for the mean of the observed distances. we may use the ratio of the observed to the expected mean distance as

$$R = r^- / E(r)$$

Then in a random population R =1; for aggregated populations R <1; for dispersed populations R >1 [2].

Algorithms for dispersed population distribution

Algorithms used in other models

CASCADE is another ecology model that generates either a dispersed or aggregated distribution of cuts based on the use of a dispersion index developed by Clark & Evan (1954).

$R = r / E(r)$

where R is the dispersion index;

r is the observed distance;

E(r) is the dispersed population.

For a random arrangement of objects, R = 1; R > 1 indicates a dispersed or uniform distribution of objects; R < 1 indicates an aggregated distribution of objects. The dispersion index R can be used to verify the harvest result. R will be a maximum after a maximum dispersed cutting. And R will be a minimum after a maximum aggregated cutting.

$E(r) = (2*\mathrm{sqrt}(p))^{-1}$

$p = (N+1)/B$

Where p, is the population density;

N, represents the units which have been cut;

B, represents the area of the considering region of a landscape.

In all simulations, the initial unit to be cut was selected at random. To select each subsequent unit, in CASCADE's early version, it selected a random unit and calculate the disperse index R. If R > 1, this unit is considered as the next suitable cut unit. But in a new version it exams all units eligible for cutting. And the unit that either maximizes (dispersed cutting) or minimizes (aggregated cutting) the dispersion index is selected as the next unit to be cut.

The search algorithm used in PAYSAGE

An exhaustive searching algorithm used in disturbance subroutine of PAYSAGE. To select a LHU for cutting, all LHUs are examined in turn and the sum distance is calculated using a single candidate LHU and all those containing open-canopy forest [Hansen, 1992]. After examine each candidate, the one that has maximum sum distance (dispersed cutting) or has minimum sum distance (aggregated cutting) is selected. This process is repeated until the cutting quota for the time step is reached. At the end of each time step, a map of opened and closed canopy forest is generated and landscape conditions are quantified by using FRAGSTATS.

The result will be definitely correct, because it exam all units in turn, i.e. it exhaustively search the whole image map. Obviously this exhaustive search (exam all units) cost time. Time efficiency will be the problem for this algorithm.

There are two loops in the code to do the exhaustive search. To find each suitable cut unit will cost time complexity as $O(m^2)$, m is the number of all units in the landscape. If a map is big enough, it'll take very long time to find all cuttable units.

Here is the pseudo code:

```
sum = 0;
far_dist = 0;
from every uncut unit , for (i = 0;  i< m ; i++)
to every cut unit, for (j=0; j<number of cut unit; j++)
{
        sum = sum + dist[i][j];
        if ( sum > far_dist ) {
                far_dist = sum;
                far_id = i;
        }
}
```

A better algorithm with distance bound --- Bound setting algorithm

Transferring this optimal problem to a decision problem will reduce the time complexity to linear time complexity [1]. The critical thing is how to find a bound, a "max_distance", each time for a new cuttable unit. Assuming there is a max_dist_bound, we can guess an arbitrary unit in [0 ~ m-1], and check if the sum distance from this guessed unit to every other cut units is bigger than max_dist_bound. If it is, we think this unit is the correct next cuttable unit that we are looking for.

Here is the pseudo code:

```
found = 0;
count = (m-1) / 2;
set up the max_dist_bound;
while ( !found && count > 0) {
        guess a unit I in 0~m-1;
        from I to every cut unit j ( 0 ~m-1, m is the worst cast)
                sum = sum + dist[i][j];
        if sum > max_dist_bound
                found = 1;
        else
                count = count -1;
}
if  ( !found)  printf (" there is no suitable dispersed cutting unit in this area! \n");
```

The critical thing remaining is how to set up the distance bound. As discussed before, the calculated dispersed index is used in CASCADE. The resulting dispersed distribution may not be the maximum dispersed distribution. However, Clark & Evan's dispersed population E(r) can help us to set up the max distance bound.

$$E(r) = (2*\mathrm{sqrt}(p))^{-1}$$

$$p = (N+1)/B$$

Where p, is the population density;

N, represents the units which have been cut;

B, represents the area of the considering region of a landscape.

The distance bound as E(r) or more will result in a dispersed landscape pattern. So, we set up our distance-bound = E(r) * K, where K >= 1. When we set this index K big enough, the result will be more close to the optimal result. If we set this index K too big, it may not find enough uncut units to fulfill the cutting percentage requirement. Even with self-adjusting functionality the program will give right output, but it will take long time for the model to adjust the index value K to smaller ones.

The pseudo code listed before can be modified as,

```
found = 0;
count = (m - 1) / 2
set index K to a number >1; // a recommend start_index K = 4
while ( !found && K >1) {
        set max_dist_bound = E(r)*K;
        guess a unit I in 0~m-1;
        from I to every cut unit j
                sum = sum + dist[i][j];

        mean_dist = sum / N        // N is the number of units which have been cut
        if  mean_dist > max_dist_bound
                found = 1;
        else
                if
                    count > 0   count = count -1
                else{
                    K = K - 0.5;
                    count = (m-1) / 2;
                }
}// end of while
if  ( !found)  printf (" there is no suitable dispersed cutting unit in this area! \n");
```

The unit that is decided by this routine is possibly not guaranteed just the one with maximum sum distance, more experiment results will be provided in next section. At this step the time complexity has reduced to at most O(m).

The algorithm used for aggregated cutting

For a given landscape region metric, according to the user specified start_cutting unit, (we call it seed), the aggregated cutting model will find the next suitable unit to cut which has the nearest distance to the seed.

The aggregated cutting model will go on finding the next suitable unit among all uncut candidate units and do cut. Until the user required cutting percentage of the area has been reached.

The implementation of this model is relatively simple comparing to the dispersed model, because the destination of all distance from any candidate is only the seed itself.

The aggregated model will go through every uncut LHU, and look up the distance look-up table dist[][], to get the distance from the considered unit to the seed. At last select the unit with maximum distance to the seed.

The pseudo code is:

```
max_dist = 0;
ID = -1;
for I=0 to m // to consider very candidate unit
        if (dist[i][seed] > max_dist) {
                max_dist = dist[i][seed];
                ID = I;
        };
```

Then do the cut and go on to find the next cut unit until reach the user required cutting percentage.

Image file formats and data compression

PAYSAGE gets information from 5 layer images of the landscape (age, region, type, history and LHU maps), and outputs dynamic landscape pattern. So the image data processing includes sequential, random reading and modifying the values of pixels in an image.

First thing in image manipulation is to choose the right file format to save and represent data of images. Image data is traditionally divided into two classes: Bitmap and Vector. Bitmap data contains an exact pixel by pixel mapping of an image. Vector data usually refers to a means of representing lines, polygons, or curves by numerically specifying key points. Today, most image storage is bitmap-based, and displays are raster-based. While vector files are widely used in CAD and other design systems.

Bitmap image files

The basic components of a simple bitmap file are: Header + Bitmap Data. The only difference among bitmap formats is that they used different and specified compression algorithms. The simplest bitmap file format is PPM format that doesn't compressing its bitmap data.

The header is a section normally found at the beginning of the file, containing information about the bitmap data. All bitmap files have some sort of header, although the format of the header and the information stored in it varies considerable from one format to the others. Typically, a bitmap header is composed of fixed fields.

For a PPM file format header, which includes:

Magic value: p3 for ASCII version, p6 for binary

Image Width: width of image in pixels

Image Height: Height of image in pixels

MaxGrey: Maximum color value

This information is closed to the bare minimum required to describe a bitmap image so it can be need and rendered.

Bitmap data is usually found immediately after the end of the file header. That is a series of lines describing width x height pixels. In PPM format file, each pixel contains three ASCII decimal values between 0 and the specified maximum value, starting at the top-left corner of the bitmap[11].

Conversions between two bitmap formats are always successful. All bitmap images consist of pixels, and ultimately all bitmap data is converted one pixel value at a time. Bitmap headers and the data contained in them can vary considerable, but the data contained in them can be added or discarded at the discretion of the conversion software to make the best conversion possible.

Image data compression and compression algorithms

Data compression is a type of data encoding, and one that is used to reduce the size of data. For a particular file format, data encoding is used to refer to algorithms that perform compression. Data encoding is actually a general term of data compression.

For bitmap file, only the image data is compressed. (the head, color map, footer are left uncompressed). Compression ratio is used to describe the quality of a compressed image.

Compression algorithms are used to re-encode data into a different, more compact representation conveying the same information. In other words, compression algorithms are used to remove the redundancy that existed in the data. Because graphics images usually require a very large amount of storage space, compression is an important consideration for graphics file formats. Almost every graphics file format uses some compression algorithm, except PPM and raw formats[11]. There are two classes of compression algorithms. The first, called lossless compression, ensures that the data recovered from the compression/ decompression process is exactly the same as the original data. In contrast, lossy compression does not promise that the data received is exactly the same as the data sent.

Run-Length Encoding (RLE)

Run-Length encoding is a data lossless compression algorithm that is supported by most bitmap file formats, such as TIFF, BMP, and PCX. RLE is suited for compressing any type of data regardless of its information content, but the content of the data will affect the compression ratio achieved by RLE. Although most RLE algorithms can not achieve the high compression ratios by using the more advanced compression methods, RLE is both easy to implement and quick to execute, making it a good alternative to either using a complex compression algorithm or leaving your image data uncompressed [11].

RLE works by reducing the physical size of a repeating string of characters. An uncompressed character run of 15 A characters would normally require 15 bytes to store: AAAAAAAAAAAAAAA. The same string after RLE encoding would require only two bytes: 15A. A string like AAAAAAbbbXXXXXt, will be represented as 6A3b5X1t. Another example as string Xtmprsqzntwlfb, will be represented as 1X1t1m1p1r1s1q1z1n1t1n1t1w1l1f1b, which requires more space than the original uncompressed one.

RLE schemes are simple and fast, but this compression efficiency depends on the type of image data being encoded. A black and white image that is mostly white, such are the page of a book , will encode very well, due to the large amount of contiguous data that is all the same color. An image with many colors that is very busy in appearance, however, such as a photograph, will not encode very well. This is because the complexity of the image is expressed as a large number of different colors. And because of this complexity there will be relatively few runs of the same color.

JPEG Compression

One of the hottest topics in image compression technology is JPEG. GIF can store only images with a maximum pixel depth of 8 bit, for a maximum of 256 color. JPEG can handle pixel depth 6~24 bits with reasonable speed and efficiency[11]. User can choose compression ratio according to different situation.

JPEG compression takes place in three phases, as illustrated in Figure 1. On the compression side, the image is fed through these three phases one 8*8 block at a time[12].

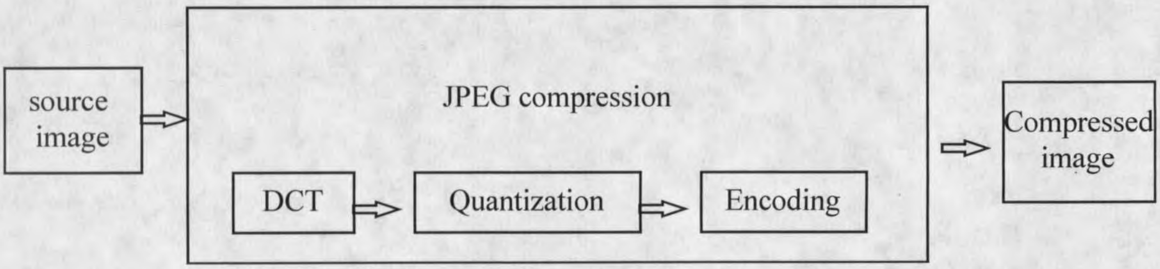

Figure 1. Block diagram of JPEG compression

The first phase applies the Discrete Cosine Transform (DCT) to the block. If you think of the image as a signal in the spatial domain, then DCT transforms this signal into an equivalent signal in the spatial frequency domain. DCT is a transformation closely related to the Fast Furrier Transform (FFT)[14]. It takes an 8*8 matrix of pixel values as input and outputs an 8*8 matrix of frequency coefficients. DCT actually breaks this signal into 64 spatial frequencies. The low frequencies correspond to the gross features of the picture, while the high frequencies correspond to fine detail. The idea behind the DCT is to separate the gross features, which are essential to viewing the image, from the fine detail, which is less essential and, in some cases, might be barely perceived by the eye. These higher-frequencies coefficients are increasingly unimportant to the perceived quality of the image. It is the second phase of JPEG that decides which portion of which coefficients to throw away.

After the DCT, the second phase applies quantification to the resulting signal, which will loss the least significant bits of the frequency coefficients. Rather than using the same quantum for all 64 coefficients, JPEG uses a quantification table that gives the quantum to use for each of the coefficients. In practice, the JPEG standard specifies a set of quantification tables that have proven effective in compressing digital images.

The third phase encodes the final result, which uses a lossless data compression algorithm. Starting with the DC coefficient in position (0, 0), the coefficients are processed in the zigzag sequence. Along this zigzag, a form of run length encoding is used --- RLE is applied to only the 0 coefficients, which is significant because many of the later coefficients are 0. The individual coefficient values are then encoded using a Huffman code[14].

Decompression follows these same three phases, but in reverse order.

SVF format

PAYSAGE included a GIS application ARC/INF, it is using SVF (single value file) format as its input and output image file format. SVF format is one kind of bitmap file format which used RLE (Run Length Encoding) to compress the image data. As mentioned before, RLE compression algorithm is simple but its compression ratio will depend on the image data. Unless the image data has a very high degree of contagious repetition, this algorithm can get high compression ratio.

At this situation, a landscape information is already divided to 5 layers of images. So each image has high repetition indeed and RLE compression can get high compression ratio. This SVF format is also widely used in GIS (Geography Information System).

The components in SVF file format is also followed the two basic components of the typical bitmap format, Head + bitmap data. The head is only composed of two values, the row and the column number of the image. The bitmap data is a string of compressed values of each pixel.

However, SVF file format is not that popular like JPEG or GIF format. SVF format can't directly display on the Web browser to be shared the research results with other scientist easily. A JPEG format output is considered to add to PAYSAGE to make it more convenient.

JPEG format

Except the image header, the JPEG format is essential just the JPEG compression technology including three phases. An external modular is included in PAYSAGE package to allow it output final images in JPEG format and convert other bitmap formats to JPEG format.

RESULTS AND DISCUSSIONS

Applying dispersed algorithms on various landscapes

Two different cases had been tested using both exhaustive search algorithm and bound setting algorithm. All experiments were performed by using COMPAQ PC with 100MHz Pentium CPU and 16MB RAM. To evaluate the results of dispersed cutting algorithms the dispersed index was calculated after each dispersed cutting.

disp_index = mean_dist / E(r)

For a maximum dispersed cutting model, we want our model to have this final distribution, represented as disp_index as big as possible.

Case A

An ideal rectangular landscape with a 40*40 cell matrix, is divided into 64 LHUs (5*5 cells for each LHU). Assuming every LHU is cuttable, 10% cut percentage at each time step, and the start cutting is 20. Following is 5 layers of the considered landscape:

2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33
34	35	36	37	38	39	40	41
42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57
58	59	60	61	62	63	64	65

LHU.svf (40*40 matrix)

Every cell
type = 1
(40*40matrix)

type.svf

Every cell
age = 100 years old
(40*40 matrix)

age.svf

Every cell
history = 1
(40*40 matrix)

history.svf

Every cell
within region=1
(40*40 matrix)

regions.svf

Figure 2. Schematic diagrams of the model image inputs

Using the intuitive exhaustive search algorithm, it took 2.436 seconds for 10% harvest at the 40*40 area. And it took 5.82 seconds for 20% cutting, 10.21 seconds for 30% cutting and 19.28 seconds for 40% cutting [Fig.5]. Since it add to check every unit in the 40*40 area to find the most suitable unit, so the result was the best. For 10% cutting, disp_index is 2.436, 2.171 for 20% cutting, 2.093 for 30% cutting, and 2.007 for 40% cutting [Fig. 6]. The result image of modified landscape patterns are showed in Fig. 3 (units in shadow represent new cut area, with age=0)