2013 IEEE 16th International Conference on Computational Science and Engineering

Edge Detection and Linking Pattern Analysis Using Markov Chains

Haklin Kimm & Neda Abolhassani
Computer Science Department
East Stroudsburg University of Pennsylvania
East Stroudsburg, PA, USA
hkimm@esu.edu

Fenglien Lee
Math & Computer Science Department
University of Guam
Mangilao, GU 96923, USA
drflee@hotmail.com

Abstract— ***There have been many studies on developing more accurate edge detection algorithms and employing them for various applications - especially in geographic information system images and maps, say coastline images. In this paper the coastline images are implemented with an edge detection algorithm using the MATLAB programs; and the edge detected images are analyzed so as to reconnect broken links. In doing so Markov transition probability matrices are generated upon the edge images and processed in order to analyze the edge linking patterns of the coastline images. Furthermore, the image analysis using the Markov Chains shall be extended and acknowledged as another alternative that can be a replacement or supportive to traditional coastline mapping techniques; because the Markov Chain represents the pattern of the coastline images numerically.***

***K*eywords- Edge detection and linking, Markov Chains, Image Processing, Coastline images.**

I. INTRODUCTION

Coastline features such as position, orientation and geometric shape play an important role in coastal erosion monitoring, autonomous navigation and geographical exploration. The information about these boundaries is essential in management of coastal resources [1]. Year-round monitoring of the shoreline along sea sides and islands will establish baseline conditions and will eventually result in development of comprehensive model for erosion and accretion patterns on seasonal and inter-annual time scales [2]. Traditional coastline mapping techniques were based on conventional ground surveys or on the interpretation of aerial photographs. These aerial photographs were manually interpreted using the stereo-plotter process. The traditional methods have high cost due to the periodic over-flights and aerial photograph rectification and analysis. In contrast, image processing on satellite images provides an appropriate tool for updating coastal maps over large areas at relatively low costs. In addition, satellite images are provided with a high repetition rate which has the advantage of getting the appropriate temporal sampling in order to study the dynamic phenomena of determining the coastline shape [3].

In image processing and computer vision, edge detection and segmentation are two conventional approaches for boundary detection. Boundaries in the problem of coastline detection are located between two relatively homogenous areas of land and ocean. There have been a lot of approaches for coastline detection using these conventional approaches. In [4], a coastline extraction technique based on edge detection was applied to a small portion of SEASAT synthetic aperture radar (SAR) images. In [5], an edge detection method with a coarse-fine resolution processing strategy was employed and it was applied to European Remote Sensing Satellite (ERS)-1 SAR images. In [6], an image segmentation based on neural networks was used on several portions of scanned US Geological Survey (USGS) aerial photographs. In [7], an alternative approach to automatically mapping the coastline was proposed which used an image segmentation based on locally adaptive thresh holding algorithm. This method was applied to both ERS-1 SAR and Satellite Probatoire d'Observation de la Terre (SPOT) images to extract ice margins for Greenland.

Edge detection is simpler to implement than segmentation but edge pixels produced by edge detectors are quite discontinuous and post-processing algorithms such as edge linking are needed to enhance the edge detected images. There have been many studies on developing more accurate edge detection algorithms and employing them for various applications. As an example in [8, 9], different edge detection algorithms were studied and compared to find out a better edge detection algorithm for planet images, where the mathematical morphological algorithm has been applied well. Different edge detectors would work well in their own application problems; but they would not work well outside their

978-0-7695-5096-1/13 $31.00 © 2013 IEEE
DOI 10.1109/CSE.2013.170

own applications. The edge detection problem domain considers different conditions as different kinds of images: Infra Red (IR) images, Ultrasound images, SAR images, satellite images, and others, where each image may inherit its own noise before filtering. Detecting coastline changes after implementing different edge detection and edge linking algorithms on the satellite coastline images will lead us to find a better method for the coastline resource management, and help to prepare for future plan on coastline analysis.

In this paper we are interested in investigating the problem of edge detection for the coastline changes and select the most suitable approach for coastline feature detection which results in not only an image with less noise and fewer meaningless edges but also with more continuous edge segments. Due to presence of non-continuous edge detected results, this work implements an edge linking method which is based on Markov transitions matrices - this is generated in order to find the suitable edge terminators which can be connected to each other. The Markov chain has been developed to investigate possible line drawing options to reconnect the broken edges and to verify whether the use of the Markov chain provide a proper pattern for a linking path between two candidate edge terminators. The Bresenham's line drawing algorithm, which is quite useful in computer graphics, is applied upon the candidate edge terminators [10, 11].

The rest of the paper is organized as follows. In section 2, Markov Chain is briefly illustrated; in section 3, previous edge detection studies on coastline images are introduced; in section 4, link pattern analysis using Markov Chain is presented with its implementations. Finally, the paper concludes with future thoughts.

II. APPLING MARKOV CHAINS

In this paper we are interested in applying Markov chain to see a pattern of coastline images numerically and applying the directional tendencies of connected and broken lines between object labels. Furthermore the profile of the coastline image based on the Markov chain is enhanced to detect the coastline formation changes numerically.

Markov chain represents a sequence of random variables, which correspond to the states of a certain system, in such a way that the state at one time epoch depends only on the one in the previous time epoch. Markov chain has a wide range of applications. The PageRank of a webpage as used by Google is based on Markov chain [12]. Markov Chain Monte Carlo methodology, which provides enormous scope for realistic statistical modeling, also uses Markov chain [13]. Markov decision processes and Markov chains are special cases of stochastic games. It describes the dynamics of the states of a stochastic game where each player has a single action in each state [14].

A Markov chain is described as follows: There is a set of states, $S = \{s_1, s_2, ..., s_m\}$. The process starts in one of these states and moves successively from one state to another. Each move is called a step. If the chain is currently in state s_i, and it moves to state s_j at the next step, the probability is denoted by P_{ij}. The probability of remaining in the same state is $P_{ii.}$ The Markov transition probability does not depend upon which states the chain was in before the current state. Thus, the Markov process is with memory-less property. These transition probabilities of each state are stored in a transition matrix. A Markov transition matrix P is a square matrix describing the probabilities of moving from one state to another in a dynamic system. Each row contains the probabilities of moving from the state represented by that row, to the other states. Thus the summation of elements in rows of a Markov transition matrix must be equal to (1) [15].

$$P = \begin{bmatrix} P_{11} & \cdots & P_{m1} \\ \vdots & \ddots & \vdots \\ P_{1m} & \cdots & P_{mm} \end{bmatrix} \qquad (1)$$

To determine the probability that, given the chain is in state s_i in the current step (step 0) it will be in state s_j two steps from now ($P_{ij}^{(2)}$) we need to calculate the disjoint union of all possible states in step 1 and state j in step 2. Thus, we have

$$P_{ij}^{(2)} = P_{i1}P_{1j} + P_{i2}P_{2j} + \ldots + P_{im}P_{mj} \qquad (2)$$

This equation reminds us of a dot product of two vectors. In general, if a Markov chain has m states, then

$$P_{ij}^{(2)} = \sum_{k=1}^{m} P_{ik}P_{kj} \qquad (3)$$

Based on a theorem of the Markov chain, the $_{ij}$th entry $P_{ij}^{(n)}$ of matrix P^n gives the probability that the Markov chain, starting s_i will be in state s_j after n steps.

III. Edge Detection on Coastline Images

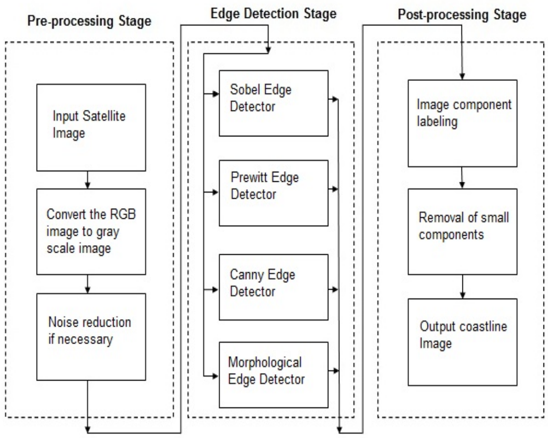


Figure 1: Coastline edge detection process

The implementation of edge detection algorithms on the satellite coastline images has been done using the MATLAB. The well-known edge detection algorithms have been selected and implemented using Image Processing Toolbox functions of the MATLAB [16], which is shown in Figure 1 and Figure 2 respectively. Various coastal satellite images showing different formations and shapes were tested with the edge detection algorithms: Prewitt, Sobel, Canny, and Morphological edge detectors. In [17], various coastal satellite images showing different formations and shapes have been tested with the edge detection algorithms. The experiment is done with 10 different satellite images. However, extra pixels that should not belong to the absolute coastline were still shown in the result images. Therefore a pruning procedure has been applied to the images after applying edge detection algorithms. The procedure of the labeling was applied to generate more continuous coastlines in order to find the 8-connected pixels and mark different objects of the images.

There are three types of connectivity for a pixel: 4-connectivity, 4-diagonal-connectivity and 8-connectivity. 4-connectivity needs the north, south, west and east neighbors of the current pixel to have the same value. For 4-diagonal-connectivity, the pixels in north-east, north-west, south-east and south-west are checked. In 8-connectivity all the adjacent pixels around the current pixel should have the same value. In image processing labeling the image means to scan the image pixel by pixel in order to find the connected pixel regions.

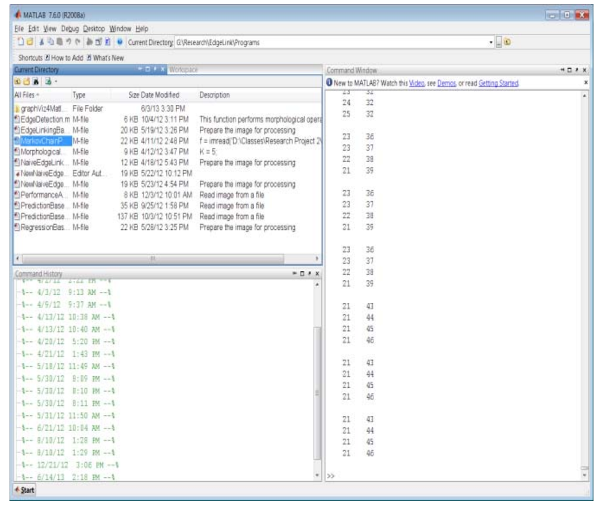


Figure 2: Implementation using MATLAB

The pixels that meet the criteria are labeled with the same number and they demonstrate an object in the image. After labeling, the size of each object is calculated by counting the number of its pixels, but if the connected component has the size less than a threshold it would be deleted. The threshold is determined experimentally and in our implementation we set it to 10

.

Figure 3: A sample coastline image in east USA

The Sobel edge detection algorithm has shown better results than other three detection algorithms, which is given in [17]. Thus, the Sobel algorithm has been chosen to be applied to a sample coastline image in east USA as shown in Figure 3; and the outcome image after applying the Sobel edge detector is shown in Figure 4.

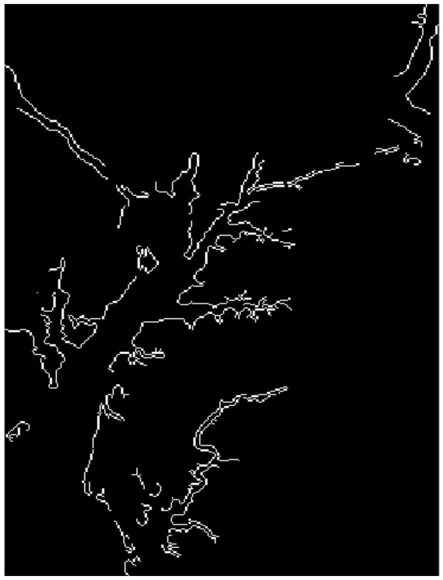
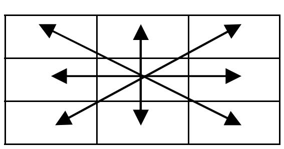


Figure 4: Outcome image after the Sobel edge algorithm applied

IV. LINKING PATTERN ANALYSIS

The source endpoint has pixels connected to it that have the same label and are from the same object as the source endpoint. The line between the source and its connected previous pixel is a two pixel length line and is vertical if the previous pixel is either on the north or south of the source end point. The line is horizontal if the previous pixel is either on the east or west of the source endpoint. The line is a diagonal line with a negative slope (dgn1) if the previous pixel is either on the northeast or southwest of the source endpoint. The line is a diagonal line with a positive slope (dgn2) if the previous pixel is either on the northwest or southeast of the source endpoint.



The direction of this line can be supposed as the current state of the Markov chain and we can find the probability of the next state by the Markov chain prediction method. We can enlarge this window and detect the directions of the line connected to the previous pixel of source endpoint. It is a line of three pixels length and it consists of two lines with different directions such as vertical-vertical, vertical-horizontal, vertical-dgn1, vertical-dgn2 and etc. Using a new scanning window, one can find the probability of the position of the two adjacent pixels exactly after the source endpoint. In the following sections the implementation of each scanning window is discussed.

A. Prediction Based on Two Adjacent Pixels

The Markov property is for the cases that the state of the system at time t+1 depends only on the state of the system at time t. The transition matrix shows us the probability of a specific direction of an edge segment after another one. In order to create the transition matrix, we need to get the states of the edge pixels' directions. The numbers of patterns that occur after a horizontal edge, vertical edge and two types of diagonal edges have to be counted. For example, if a(i,j) is the current pixel, any of the a(i-1,j) or a(i+1,j) can make a vertical edge. Any of the a(i,j-1) or a(i,j+1) can make a horizontal edge. Any of a(i-1,j-1) or a(i+1,j+1) can create a diagonal edge referred as dgn1 edge in this paper and any of a(i-1,j+1) or a(i+1, j-1) can create a different diagonal edge referred as dgn2 edge. Equation 4 shows how we can calculate the probability of being a horizontal edge segment after a vertical edge segment P_{verhor}:

$$P_{verhor} = \frac{\text{the number of horizontel edges after vertical edges}}{\text{the total number of patterns that occur after vertical edges}} \quad (4)$$

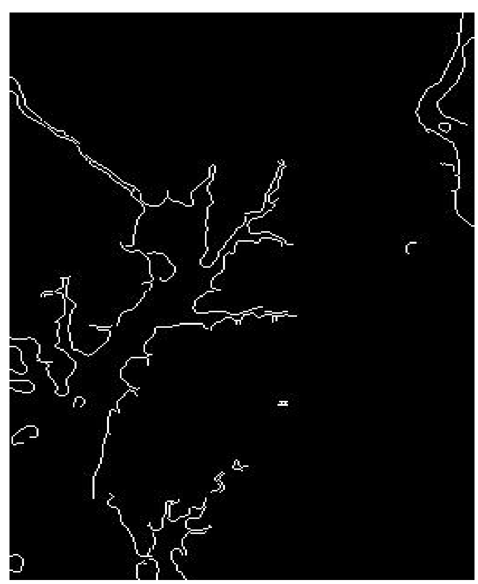


Figure 5: Outcome image after applying Markov Chains to reconnect broken edges

The Figure 5 is one of the images that has analyzed for this edge linking pattern analysis problem using a Markov chain on the Figure 4. The following is the Markov transition matrix generated after processing the image over the MATLAB programs:

$$P = \begin{array}{c} \\ hor \\ ver \\ dgn1 \\ dgn2 \end{array} \begin{array}{c} \begin{array}{cccc} hor & ver & dgn1 & dgn2 \end{array} \\ \begin{bmatrix} 0.6311 & 0.0410 & 0.2295 & 0.0984 \\ 0.0022 & 0.9307 & 0.0671 & 0 \\ 0.2885 & 0.3462 & 0.3462 & 0.0192 \\ 0.3077 & 0.4103 & 0.1026 & 0.1795 \end{bmatrix} \end{array} \qquad (5)$$

The maximum probability of each pattern in each row can be selected as the best pattern for the next move and they are highlighted in the above matrix. The probability of going from state i to j in n time steps can be found by the transition matrix raised to the power of n. In the edge linking problem n can be the distance between the two endpoints that are going to be linked. For this case the shortest distance is 6 and the transition matrix should be raised from power of 2 to 6. Therefore, we have

$$P^{(2)} = \begin{array}{c} \\ hor \\ ver \\ dgn1 \\ dgn2 \end{array} \begin{array}{c} \begin{array}{cccc} hor & ver & dgn1 & dgn2 \end{array} \\ \begin{bmatrix} 0.4949 & 0.1838 & 0.2371 & 0.0841 \\ 0.0227 & 0.8896 & 0.0862 & 0.0015 \\ 0.2886 & 0.4617 & 0.2112 & 0.0385 \\ 0.2799 & 0.5036 & 0.1521 & 0.0645 \end{bmatrix} \end{array} \qquad (6)$$

$\mathbf{P^{(3)}}$ $\mathbf{P^{(4)}}$ $\mathbf{P^{(5)}}$

$$P^{(6)} = \begin{array}{c} \\ hor \\ ver \\ dgn1 \\ dgn2 \end{array} \begin{array}{c} \begin{array}{cccc} hor & ver & dgn1 & dgn2 \end{array} \\ \begin{bmatrix} 0.1109 & 0.7541 & 0.1189 & 0.0161 \\ 0.1107 & 0.7545 & 0.1188 & 0.0161 \\ 0.1108 & 0.7542 & 0.1188 & 0.0161 \\ 0.1108 & 0.7543 & 0.1188 & 0.0161 \end{bmatrix} \end{array} \qquad (7)$$

Thus, according the transition matrices after the second link we are recommended to choose the vertical pattern for 75 times out of 100 times in reconnecting broken edges; the horizontal one 11 times; diagonal_1 12 times; and diagonal_2 1 or 2 times. This result has been applied to Figure 4 by selecting candidate edge terminators, where the Bresenham's line drawing algorithm applied using the MATLAB programs [10]. Most broken edges in Figure 4 have been reconnected properly as shown in Figure 5.

	hor-hor	hor-ver	hor-dgn1	hor-dgn2	ver-hor	ver-ver	ver-dgn1	ver-dgn2	dgn1-hor	dgn1-ver	dgn1-dgn1	dgn1-dgn2	dgn2-hor	dgn2-ver	dgn2-dgn1	dgn2-dgn2
hor-hor	0.2945	0.019417	0.087379	0.11327	0	0.029126	0.02589	0.022654	0.045307	0.067961	0.042071	0.022654	0.064725	0.064725	0.035599	0.064725
hor-ver	0	0	0	0	0.17949	0.12821	0	0.051282	0.15385	0.025641	0	0.076923	0.25641	0	0.10256	0.025641
hor-dgn1	0.13287	0	0.11888	0.034965	0.048951	0.097902	0.055944	0.062937	0.12587	0.048951	0.1049	0.013986	0.076923	0.006993	0.048951	0.020979
hor-dgn2	0.15315	0	0.054054	0.045045	0.099099	0.10811	0.018018	0.054054	0.11712	0.027027	0.009009	0.045045	0.072072	0.099099	0.018018	0.081081
ver-hor	0.11111	0.11111	0	0	0	0	0	0	0	0	0	0	0.11111	0.22222	0.33333	0.11111
ver-ver	0.0072727	0	0	0.0072727	0	0.62182	0.15273	0	0.025455	0.11636	0.054545	0.014545	0	0	0	0
ver-dgn1	0.086614	0.007874	0.047244	0.023622	0.007874	0.25197	0.1811	0.031496	0.062992	0.03937	0.12598	0.031496	0.015748	0.023622	0.047244	0.015748
ver-dgn2	0.30769	0.11538	0.076923	0.038462	0	0	0	0	0.038462	0.23077	0.038462	0.15385	0	0	0	0
dgn1-hor	0.2459	0.0081967	0.090164	0.065574	0	0.02459	0.016393	0.016393	0.12295	0.065574	0.18033	0.065574	0.016393	0.02459	0.032787	0.02459
dgn1-ver	0.044444	0	0.022222	0.014815	0.0074074	0.32593	0.081481	0.059259	0.066667	0.14815	0.088889	0.0074074	0.051852	0.051852	0.022222	0.0074074
dgn1-dgn1	0.082474	0.015464	0.12371	0.015464	0.0051546	0.07732	0.06701	0.020619	0.06701	0.10825	0.1701	0.051546	0.025773	0.041237	0.1134	0.015464
dgn1-dgn2	0.10294	0.029412	0.044118	0.044118	0.044118	0.014706	0.073529	0.014706	0.11765	0.13235	0.14706	0.17647	0.014706	0.029412	0.014706	0
dgn2-hor	0.29213	0.022472	0.044944	0.14607	0	0.044944	0.011236	0.033708	0.05618	0.067416	0.033708	0.033708	0.05618	0.05618	0.022472	0.078652
dgn2-ver	0.03268	0	0.0065359	0.0065359	0.0065359	0.36601	0	0.11765	0.0065359	0.013072	0	0.013072	0.026144	0.23529	0.039216	0.13072
dgn2-dgn1	0.068966	0.017241	0.068966	0.051724	0.068966	0.068966	0	0.068966	0.034483	0.017241	0.086207	0.051724	0.10345	0.10345	0.13793	0.051724
dgn2-dgn2	0.097403	0.012987	0.0064935	0.045455	0	0.12987	0.038961	0.14935	0.019481	0.032468	0.012987	0.045455	0.064935	0.13636	0.038961	0.16883

Table 1: $P^{(1)}$ = Initial Transition Matrix with 16 states

$P^{(2)}$ $P^{(3)}$ $P^{(4)}$ $P^{(5)}$

	hor-hor	hor-ver	hor-dgn1	hor-dgn2	ver-hor	ver-ver	ver-dgn1	ver-dgn2	dgn1-hor	dgn1-ver	dgn1-dgn1	dgn1-dgn2	dgn2-hor	dgn2-ver	dgn2-dgn1	dgn2-dgn2
hor-hor	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
hor-ver	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
hor-dgn1	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
hor-dgn2	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
ver-hor	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
ver-ver	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
ver-dgn1	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
ver-dgn2	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn1-hor	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn1-ver	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn1-dgn1	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn1-dgn2	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn2-hor	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn2-ver	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn2-dgn1	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838
dgn2-dgn2	0.11154	0.012964	0.046414	0.038039	0.014661	0.25166	0.075857	0.033572	0.056389	0.085738	0.075739	0.037446	0.037229	0.04879	0.039126	0.034838

Table 2: $P^{(6)}$ = Transition Matrix with 16 states

B. Prediction Based on Three Adjacent Pixels

In this section, we extend the option of using two adjacent pixels to three pixels trying to reconnect more broken pixels than the previous one does. The use of a larger window for the profile and prediction of the coastline image is suggested. The probability of occurring a horizontal-vertical direction after a horizontal-vertical direction and all other 15 possibilities (horizontal-horizontal, horizontal-dgn1, horizontal-dgn2, vertical-vertical, vertical-horizontal, vertical-dgn1, vertical-dgn2, dgn1-vertical, dgn1-horizontal, dgn1-dgn1, dgn1-dgn2, dgn2-vertical, dgn2-horizontal, dgn2-dgn1, dgn2-dgn2) are required. This is going to be a 16 x 16 Markov transition matrix to store the probabilities of all the directions of the edge segments that consist of three adjacent pixels and the directions of the edge segments that happen exactly after them and consist of three adjacent pixels.

The Markov transition probability matrix of the US East Coast image shown in Figure 4 is generated using MATLAB. It is a 16x16 transition matrix to check all the two lines' directions of the edges and the next two lines' directions of the edges that happened exactly after them. The other tables are the results of Transition Matrix where starts from 2 to 6. The maximum value for the probability is highlighted in red in the tables. Testing US East Coast edge result image and other edge result images show only one or two directions are selected after stabilization of the transition matrices and only using these directions we draw a correct line between the endpoints. The results of the transition matrix and its exponentiation to the power of 2 to 6 are shown in Tables 1 and 2 respectively. Although a different result was expected after enlarging the scanning window but the highlighted elements of those tables show that after the third step the same result is repeated. The vertical-vertical link has the maximum probability in the third step after any type of links of three pixels length. This result has been applied to Figure 4 by selecting candidate edge terminators using the MATLAB programs [10]. The outcome image has shown not much improvement when reconnecting the broken edges than the Markov Chain with 2 adjacent pixels as shown in Figure 5.

V. CONCLUSION

In this paper the Markov Chain has been selected, trying to reconnect broken edges of a coastline images to provide pattern analysis upon the images that taken by the satellite camera in different time periods at the same location. There has been Markov chain research on soil conservation [18], on which the textures of soil profiles are quantified. Hence, the Markov transition matrix of the same image in different time periods can be used to figure out any changes of a coastline numerically. As seen in Figure 4, the Markov chain has reconnected the broken edges well, however it could not reconnect them all properly; so that there should be more study on how to improve the reconnection ratio using Markov Chain. Further study would reveal how may adjacent pixels are adequate when generating a Markov chain for a given coastline image; and provide a better scheme when reconnecting the broken edges using the Markov chain.

This Markov transition matrix of Figure 3 of East coastline image has been used as a profile of the coastline image taken by a satellite, which can be compared with a new Markov transition matrix that is created by a new coastline image taken later at the same GPS location by the same satellite. Thus the proposed technique using the Markov Chain indentifies numerically any changes of the same coastline, which would not be easy to detect otherwise. This research using the Markov chain as temporal profiler for coastline images is expected to enhance its accuracy by extending the current work, building a profile database of Markov chain matrices of coastline images.

REFERENCES

[1] H. Liu and K. C. Jezek, "Automated Extraction of Coastline from Satellite Imagery by Integrating Canny Edge Detection and Locally Adaptive Threshold Methods"; International Journal of Remote Sensing, vol. 25, no. 5, pp. 937-958, 2004.

[2] "Consortium, NASA and Marine Science"; Research Collaboration Report, Wallops Island, VA, 2011.

[3] A. Guariglia, A. Buonamassa, A. Losurdo, R. Saladin, M. L. Trivigno, A. Zaccagnino and A. Colangelo, "A Multisource Approach for Coastline Mapping and Identification of Shoreline Changes"; Annals of Geophysics, vol. 49, no. 1, pp. 295-304, 2006.

[4] J. Lee and I. Jurkevich, "Coastline Detection and Tracing in SAR Images"; IEEE Transactions on Geoscience and Remote Sensing, vol. 28, pp. 662-668, 1990.

[5] D. C. Mason and I. J. Davenport, "Accurate and Efficient Determination of the Shoreline in ERS-1 SAR Images"; IEEE Transactions on Geoscience and Remote Sensing, vol. 34, pp. 1243-1253, 1996.

[6] T. W. Ryan, P. J. Sementilli, P. Yuen and B. R. Hunt, "Extraction of Shoreline Features by Neural Nets and Image Processing"; Photogrammetric Engineering and Remote Sensing, vol. 57, pp. 947-955, 1991.

[7] H.-G. Sohn and K. C. Jezek, "Mapping Ice Sheet Margins from ERS-1 SAR and SPOT Imagery"; International Journal of Remote Sensing, vol. 20, no. 15-16, pp. 3201-3216, 1999.

[8] B. Kaur and A. Garg, "Comparative Study of Different Edge Detection Techniques"; International Journal of Engineering Science and Technology, vol. 3, no. 3, pp. 1927-1935, 2011.

[9] T. Lindberg, Edge Detection, Hazewinkel, Michiel, *Encyclopedia of Mathematics*; Springer, 2001.

[10] J. Bresenham, "A Linear Algorithm for Incremental Digital Display of Circular Arcs"; Communications of the ACM, vol. 20, pp. 100-106, 1977.

[11] O. Ghita and P. F. Whelan, "Computational Approach for Edge Linking"; Electronic Imaging, vol. 11, no. 4, pp. 479-485, 2002.

[12] W.-K. Ching and N. Michael K., *Markov Chains: Models, Algorithms and Applications*; Springer, 2010.

[13] W. R. Gilks, S. Richardson and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*; Chapman and Hall/CRC, 1995.

[14] A. Neyman, "From Markov Chains to Stochastic Games"; Stochastic Games and Applications, Vols. 570 of NATO Science Series C, Mathematical and Physical Sciences, pp. 9-25, 2003.

[15] C. M. Grinstead and J. L. Snell, *Introduction to Probability*; American Mathematical Society, 2003.

[16] R.C. Gonzalez, R.E. Woods and S.L. Eddins, *Digital Image Processing Using MATLAB*; 2nd Edition, Pearson Prentice Hall, 2004.

[17] N. Abolhassani and H. Kimm, "Performance Analysis on Edge Detection Algorithms for Coastline Image Detection"; Proceedings of the 10th IASTED International Conference on Visualization, Vision and Image Processing, Banff, Canada, July 3–5, 2012.

[18] D. Li and M. Shao, “Simulating the vertical transition of soil textural layers in north-western China with a Markov chain model,” Journal of Soil Research 51(3), pp. 182-192, June 2013.