Clustering Document Fragments using Background Color and Texture Information

Sukalpa Chanda[a], Katrin Franke[a] and Umapada Pal[b]

[a]Department of Computer Science and Media Technology, Gjøvik University College,
Gjøvik-2815, Norway;
[b]Computer Vision and Pattern Recognition Unit,Indian Statistical Institute, Kolkata-700108,
India;

ABSTRACT

Forensic analysis of questioned documents sometimes can be extensively data intensive. A forensic expert might need to analyze a heap of document fragments and in such cases to ensure reliability he/she should focus only on relevant evidences hidden in those document fragments. Relevant document retrieval needs finding of similar document fragments. One notion of obtaining such similar documents could be by using document fragment's physical characteristics like color, texture, etc. In this article we propose an automatic scheme to retrieve similar document fragments based on visual appearance of document paper and texture. Multispectral color characteristics using biologically inspired color differentiation techniques are implemented here. This is done by projecting document color characteristics to Lab color space. Gabor filter-based texture analysis is used to identify document texture. It is desired that document fragments from same source will have similar color and texture. For clustering similar document fragments of our test dataset we use a Self Organizing Map (SOM) of dimension 5×5, where the document color and texture information are used as features. We obtained an encouraging accuracy of 97.17% from 1063 test images.

Keywords: Torn Documents, Self Organizing Map, Clustering, Forensic document analysis.

1. INTRODUCTION

In a crime scenario, a forensic expert might encounter thousands of questioned document fragments. This situation could be intentionally created by the criminal in order to intrude false and miss-leading evidences. To ensure reliable forensic analysis in such cases we can narrow down the search space for the forensics expert. Till date not much of work has been done for torn document sorting. Ukovich et al.[1] proposed a system for reconstruction of machine shredded documents, there they proposed matching of the remnants on the basis of the visual content of the strips, described by means of automatically extracted numerical features. There are some research articles in the domain of archeological fragment reconstruction. Sagiroglu et al.[2] proposed a system to solve puzzle problems involved in reconstruction of broken ceramic tiles. They used painting and texture synthesis method to predict the information on the object's outward surface. Then they make an affinity measure of corresponding pieces. Leitao et al.[3] proposed a system for reassembly of two-dimensional fragmented objects. A similar work but purely on torn document fragments are reported in[4] and.[5] Both Biswas et al.[5] and Zhu et al.[4] used a contour shape matching approach. Their approach based on matching contour shapes will not always work plausibly to address the problem of sorting similar document fragments in context of questioned document analysis. Sorting in such scenario should be more based on other physical attributes of document fragments, like document fragment colors, background texture, contents in those fragments etc. Justino et al.[6] proposed a system for reconstruction of shredded document. Here the author first applies a polygonal approximation of the boundaries and then extract relevant features of the polygon to execute local reconstruction. A very similar approach is followed by Solana et al.[7] where they also used a polygonal approximation based method.Pimenta et al.[8] proposed a methodology based on dynamic programming and modified version of prim's algorithm. Firstly, they use a polygonal approximation to reduce complexities in boundaries and extract features from them. Later

Further author information: (Send correspondence to Sukalpa Chanda, E-mail: sukalpa@ieee.org, Katrin Franke, E-mail: kyfranke@ieee.org , Umapada Pal, E-mail:umapada@isical.ac.in

Document Recognition and Retrieval XIX, edited by Christian Viard-Gaudin, Richard Zanibbi,
Proc. of SPIE Vol. 8297, 82970U · © 2012 SPIE · CCC code: 0277-786X/12/$18 · doi: 10.1117/12.910567

those features are used to fed to LCS dynamic programming algorithm. The scores yielded by the LCS algorithm are then used into a modified Prim's algorithm to find the best match amongst all pieces. Recently Kleber et al.[9] discussed about methods for reconstruction of both archaeological artifacts as well as torn documents. A recent endeavor in this context is due to Smet et. al.,[10] where for each segmented fragment the author tries to use a chain-code to trace around its contours. Another very recent effort for reconstruction of torn documents based on paper type analysis and classification of text into printed and handwritten text is proposed by Diem et al.[11] We propose a system to compress the search space for a forensic expert while dealing with huge piles of document fragments, using document's color and background texture information. The "Lab color space" known to mimic human vision perception and Gabor filter-based features for texture analysis are used for this purpose. The rest of the article is organized as follows: In Section 2 we narrate the methodology and also provide detail description of our features. A description of the SOM clustering algorithm is given in Section 3. In Section 4 we give a short description of our experimental setup as well as description about our datasets. Results and discussions on our experiments are provided in Section 5, followed by conclusion in Section 6.

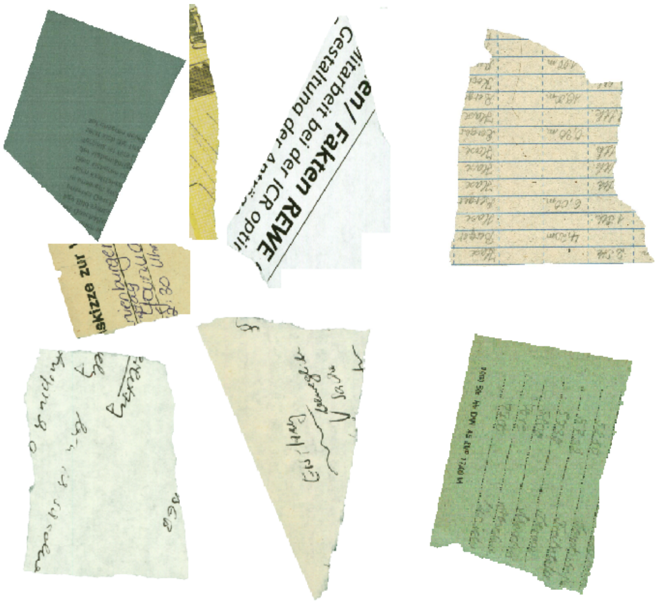

Figure 1. Example of torn documents in different shapes, colors and textures.

2. METHODOLOGY AND FEATURE EXTRACTION

Color and texture of a document fragment is determined by processing all background pixels. Here by background pixels we mean to say the pixels which are not representing any text/data contents. For each document fragment we compute an average RGB value. Color information from three channels (R, G and B) of all background pixels in a document are added with respect to three separate channels, then we compute the average RGB value for a document by dividing those summed up values by the total number of background pixels. As a result for each document fragment we get average RGB color information. Since we cannot compute the Lab color model directly from the RGB color space, we transformed our RGB color space values to corresponding XYZ color space values. Then using a final color transformation technique we transformed those XYZ color space values to Lab color space values. For each document fragment we computed the corresponding Lab color space value of that document fragment using the technique mentioned above. It is worthy to mention that we did some initial experiments for clusterng document fragments by just using RGB values of the document.But the results were not encouraging. Hence we looked for other color space transformation.

For texture information we used a non-overlapping sliding window to glide over the whole document fragment image. At initial stage we tried with different dimensions for the sliding window, and noted that window size of 20 × 20 pixels gave us best optimized results in terms of speed and accuracy. So we carried out all further experiments with the window size of 20 × 20 pixels. If there is no foreground pixel present within the window frame then we call it a "valid window" frame. If there is a foreground pixel then we don't consider that region for texture analysis. We compute the Gabor filter features of the image region which lies beneath the sliding window for all "valid window". So using a window of size 20 × 20 we obtained 400 dimensional Gabor filter features. We sum up the Gabor filter based feature values found within each valid window frame. We note the total number of valid window frame from which we have extracted Gabor filter features. Like color, here also we divide the accumulated Gabor filter based feature values by the total number of valid windows found in the document fragment. This gives us an average Gabor filter based feature value for the whole document image background. We concatenate 3 dimensional Lab color space feature with this 400 dimensional average Gabor filter feature for a document fragment. This gives us a feature vector of dimension 403 for each document fragment. So each node in our SOM lattice hold's a weight vector of dimension 403. A basic flowchart of the scheme is given in figure 2.

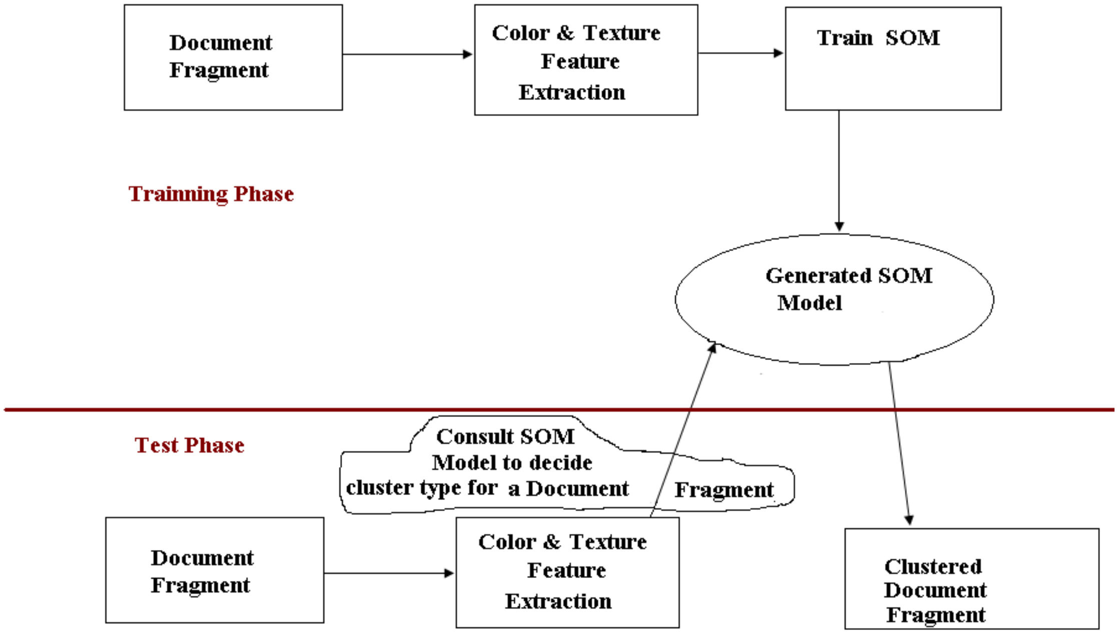
Document Fragment
Color & Texture Feature Extraction
Train SOM
Trainning Phase
Generated SOM Model
Test Phase
Consult SOM Model to decide cluster type for a Document Fragment
Document Fragment
Color & Texture Feature Extraction
Clustered Document Fragment

Figure 2. Flowchart of the basic system.

2.1 (CIELAB) Feature Extraction

CIE L*a*b* (CIELAB) is the most complete color space specified by the International Commission on Illumination (Commission Internationale d'Eclairage, hence its CIE abbreviation). It describes all the colors visible to the human eye and was created to serve as a device independent model to be used as a reference.Hence, to define color similarity we choose to express document color properties in CIELab color space. The intention of "Lab" color spaces is to create a color space which can be computed via simple formulas from the XYZ color space, but is more perceptually uniform than XYZ. The three coordinates of CIELAB represent the lightness of the color (L = 0 yields black and L = 100 indicates diffuse white ;), its position between red/magenta and green (a, negative values indicate green while positive values indicate magenta) and its position between yellow and blue (b, negative values indicate blue and positive values indicate yellow). The nonlinear relations for L, a, and b are intended to mimic the nonlinear response of the eye. The RGB values of each background pixel in a document image were converted to corresponding XYZ color space. For each document image we calculate the average xyz value considering xyz values of all background pixels. Then using the following formula:

$$L* = 116(Y/Yn)^{1/3} - 16, a* = 500[(X/Xn)^{1/3} - (Y/Yn)^{1/3}], b* = 200[(Y/Yn)^{1/3} - (Z/Zn)^{1/3}],$$

the average XYZ color space values for a document are converted to an average Lab color space value. Here in the formula, Xn, Yn, Zn are reference white for D65 standard, and Xn =.9504; Yn =1.0000; Zn =1.0888; X/Y/Z=total value in X/Y/Z normalized by total number of background pixel.

2.2 Gabor Filters

The spectral patterns of document background could be quite different and therefore well suited for texture analysis. Gabor filter-based features are therefore used for this purpose. Gabor filters are capable of representing signals in both frequency and time domain. A two-dimensional Gabor filter in spatial and frequency domain can be defined by the following formula:

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = exp(-\frac{x^{'2} + \gamma^2 y^{'2}}{2\sigma^2}) \cos\left(2\pi \frac{x^{'}}{\lambda} + \psi\right)$$

where

$$\begin{cases} x^{'} = x \cos\theta + y \sin\theta, \\ y^{'} = -x \sin\theta + y \cos\theta, \end{cases}$$

In this equation, λ represents the wavelength of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function. We tried a combination of different values of those parameters during our experiment. We got best optimized results when orientation of Gabor filter was set to $\pi/4$, with spatial frequency set to $\sqrt{2}$, and sigma set to $2*\pi$. The method used in extracting features is as follows: (i) Slide a non-overlapping window of N× N pixels over the background portion of the fragment image. (ii) Compute the corresponding Gabor value within those N × N window. (iii) Encode the Gabor value of each of the pixels in the sliding window as a vector component of our feature vector. (iv) Sum up all feature vector obtained from each N × N window. (v) The feature vector which is formed by adding all feature vector is normalized by dividing its values by the total number of N × N "valid window" frame found in the document fragment.

3. CLUSTER ALGORITHM

Self-organizing maps (SOMs) are a data visualization technique invented by Professor Teuvo Kohonen,[12] and,[13] which reduces the dimensions of data through the use of self-organizing neural networks. The problem that data visualization attempts to solve is that humans simply cannot visualize high dimensional data as it is, so techniques are created to help us understand this high dimensional data. The way SOM go about reducing dimensions is by producing a map of usually 2 dimensions,which plot the similarities of the data by grouping similar data items together. So SOM accomplish two things, they reduce dimensions and display similarities. Kohonen's self-organizing map (SOM) performs a topology-preserving transformation from a higher-dimensional vector space to a lower one, which is usually two-dimensional, and generates a map that can display visually the similarity between vectors. In addition, the units in a SOM can interpolate the intermediate vectors between the input vectors.

The basic algorithm is follows : Training occurs in several steps and over many iteration (learning iterations):
1. We make a 2-d lattice of size n× n.
2. Inside the lattice, each node's weights are initialized randomly.
3. A training sample is chosen at random from the set of training data and presented to the lattice.
4. Every node is examined to calculate which one's weight vector is most similar to the input training sample's weight vector. The winning node is commonly known as the Best Matching Unit (BMU).
5. The radius of the neighborhood of the BMU is now calculated. (i)The radius is initially set to the 'radius' of the lattice, (ii)The radius shrinks each time-step (normally implemented by an exponential decay function). (iii)Any nodes found within this radius are deemed to be inside the BMU's neighborhood.
6. Each neighboring node's (the nodes found in step 4) weights are adjusted to make them more like the input

training sample's input vector. The closer a node is to the BMU, the more its weights get altered.
7. Repeat step 2 for N iterations.

In practice the shape of the neighborhood is generally a Gaussian, we initiated our learning algorithm with initial learning rate initialized to 0.1 and with radius of the Gaussian neighborhood set to 2.5.

4. EXPERIMENTAL DESIGN AND DATASET DETAILS

It was not known to us that which torn document fragments in our corpus are parts of each other. Due to lack of this ground truth a traditional supervised learning experimental setup cannot be implemented. An alternative to this is an unsupervised learning (clustering) technique, where the learning algorithms automatically evolves groups of samples with similar feature. After executing clustering algorithm, if we could visualize that the formed clusters are visually discriminative; we can conclude that the features used are powerful enough to discriminate different samples. Our dataset consists of 1513 torn documents in total. They were not synthetically generated for our experiment purpose, but those real life torn documents were obtained from a reliable source. Some of the sample images of our torn document fragment are shown in Figure 1. It is evident that the color and texture are quite different from each other in all those samples. We randomly choose 450 document fragments for our tanning set to train our SOM. Using techniques discussed in Section 2.1 and Section 2.2 we extracted features from the background portion of images in the training dataset. The features are sent to a cluster algorithm based on SOM neural network to train our SOM. The rest of the document fragment images in our corpus are used as images for test dataset. We initially prepared SOM of many different dimension (5×5, 10×10, 15×15, etc). Smaller the size of the SOM, lesser the time to compute the model.Though during initial experiments we noted bit more accuracy with bigger dimension maps (10×10, 15×15, etc), we ignored that in order to maintain the trade-off between accuracy and computational cost. During test phase we noticed that many nodes in SOM's of bigger dimension (dimension greater than 5×5) are actually vacant. Most of the document fragments were getting assigned to some selected nodes in the lattice. We realized that in reality there is actually not many clusters of different color and texture in document fragments of our corpus, eventually we stuck to dimension of 5×5, and carried all our further experiments based on this dimension of the SOM map. We intended to analyze the affect of number of learning iterations on rate of accuracy of our SOM model. We tried with different number of learning iterations for the SOM starting from 400 till 1200. We noticed that after a certain number of iterations there is not much of effect in clustering accuracy. Detail results on each of those experiments are provided in Section 5.

5. RESULT AND DISCUSSIONS

Torn document fragments in our corpus consist of different background color. Though not much difference could be perceived by normal human vision in terms of paper texture, yet there were different texture types which got exposed by our clustering algorithm. Our intention was to justify relevance between documents in terms of back ground color and texture. We deployed well known SOM (Self Organizing Map) clustering technique for this purpose. The resultant SOM map generated 25 different document fragment clusters for us. Also we noticed that different color is associated with a different background texture as well. So it can be concluded that both features were capable enough to serve our desired purpose. We experimented with different variants of our SOM model. All those SOM models have a similar training parameter with only difference in terms of the number of learning iterations. We report accuracy of the system on different numbers of learning iterations in Sub-section 5.1.

5.1 Effect of Number of learning Iterations on SOM Accuracy

Here we report the details of our experiment using different number of learning iterations. It can be noted that after a certain number of iterations are performed, there is not much effect of number of learning iterations in the accuracy of the SOM. Here for all experiments the total number of training document fragments was 450 and testing document fragments was 1063. Accuracy with respect to total number of learning iterations is shown in Table 1.

Table 1. PERFORMANCE OF SOM WITH RESPECT TO NUMBER OF LEARNING ITERATIONS

Total number of learning Iterations to generate the SOM model	Percentage of overall accuracy (total number of correctly clustered document fragments out of 1063 test document fragments within bracket)
400	75.25% (800)
600	87.48%(930)
800	93.41%(993)
1000	97.17%(1033)
1200	97.17%(1033)

5.2 SOM Accuracy

Since we were devoid of the ground truth we had to check manually the number of document fragments that were assigned to the wrong node in the SOM lattice. For SOM model with number of learning iterations set to 1000 we obtained an accuracy of 97.17%. A pictorial view of our SOM lattice with representative document fragment in each cluster node is provided in figure 3. Here each cluster node within the lattice is numbered, the first number denotes the row index and the second number denotes the column index of the node within the lattice. For e.g. number "21" denotes a node which is in row 3 and column 2 in the lattice where numbering starts from index 0. In a well trained SOM , nodes close to each other should be more alike than nodes that are far apart in the lattice. From figure 3 it's clearly evident that the SOM we generated fully obeys this characteristic. We can see that the nodes which are close to each other posses document fragment of similar color and texture. For e.g. element in nodes "01" and "02" have very close resemblance in terms of color and texture. They predominantly belong to document fragments with yellowish color. Whereas all pale colored document fragments got assigned to the different nodes in column 1 of the SOM lattice.

5.3 Error Analysis

Amongst all document fragments in our test dataset, 2.83% (30) document fragments were assigned to wrong clusters by our SOM. We analyzed the errors and found that those errors can be mainly categorized into two different categories. For the first category we identified that those document fragments were very less in number in our corpus, our SOM was not trained using those document fragments during selection of random samples in the training phase. As a result there was no cluster node in our SOM model which could exactly fit those erroneous samples. An example of this type of erroneous document fragment is shown in figure 4; the document fragment in figure 4(a) was assigned to cluster "44". It is true that document fragments in Cluster "44" {an example of a document fragment in cluster "44" is shown in figure 4(b)} is the closest match to the document fragment shown in figure 4(a). Even visually we can see that the document fragment shown in figure 4(b) is of greenish color, also the erroneous test document fragment{figure 4(a)} has a dark greenish background. But after we scrutinized all other document fragments in cluster "44" it was not difficult to notice that this particular test document fragment is an outlier compared to other document fragments in the cluster. The other category of error we noticed is with document fragment of multiple back ground color and texture. For e.g. see the document fragment in figure 4(c). It can be clearly noted that there are two different background color in the document fragment. Since we made an average of the background texture and color for the whole document fragment, the background color and texture in this case might not be well representative for the document fragment, as a result got assigned to a wrong cluster.

6. CONCLUSION

Clustering similar document fragments could be used as a pre-processing step in automated forensic analysis of questioned document fragments. In this article we proposed a method for grouping torn documents based on their physical characteristics like document's background color and texture. We used Lab color space value of those document fragments along with Gabor filter based features for discrimination of color and texture amongst different document fragments. A SOM-based clustering technique is deployed to cluster document fragments based on color and texture information. We obtained an encouraging accuracy of 97.17% on 1063 real document

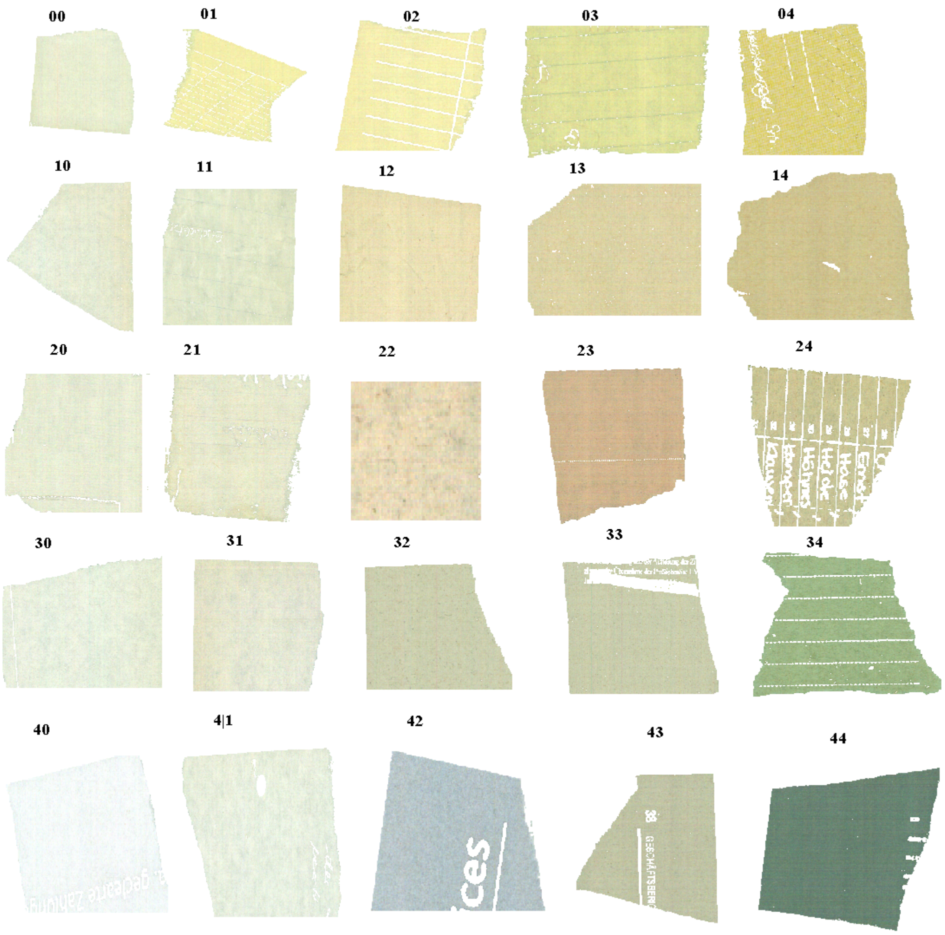

Figure 3. A view of our SOM lattice with representative document fragments in each of its node.

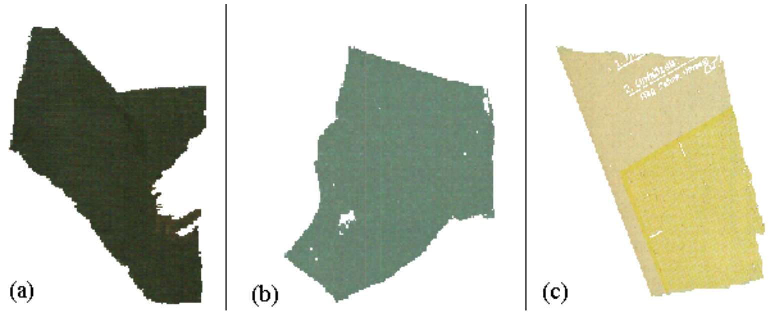

Figure 4. Examples of two erroneous test samples:(a) and (c) which got assigned to wrong cluster nodes in our SOM.

fragments. In future we intend to combine parametric and non-parametric density estimation-based clustering techniques in order to achieve better accuracy over erroneous samples of our test dataset.

References

[1] Ukovich, A. and Ramponi, G., "System architecture for the digital recovery of shredded documents," in [*Image Processing: Algorithms and Systems*], 1–11 (2006).

[2] Sagiroglu, M. S. and Ercil, A., "A texture based matching approach for automated assembly of puzzles," in [*International Conference on Pattern Recognition*], **3**, 1036–1041 (1993).

[3] da Gama Leitao, H. C. and Stolfi, J., "A multiscale method for the reassembly of two-dimensional fragmented objects," *IEEE Trans. Pattern Anal. Mach. Intell.* **24(9)**, 1239–1251 (2002).

[4] Zhu, L., Zhou, Z., and Hu, D., "Globally consistent reconstruction of ripped-up documents," *IEEE Trans. Pattern Anal. Mach. Intell.* **30(1)**, 1–13 (2008).

[5] Biswas, A., Bhowmick, P., and Bhattacharya, B. B., "Reconstruction of torn documents using contour maps," in [*International Conference on Image Processing*], **1898**, 517–520 (2005).

[6] Justino, E., O.S.OLoveria, and Freitas, C., "Reconstructing shredded document through feature matching," *In Forensic Science International Journal* , 140–147 (2006).

[7] Solana, C., Fernandes, L. A. F., Justino, E. J. R., Oliveira, M. M., da Silva, R., andFlavio Bortolozzi, L. S. O., and Crespo, G. J., "Document reconstruction based on feature matching," in [*XVIII Brazilian Symposium on Computer Graphics and Image Processing*], 163–170 (2005).

[8] Pimenta, A., J.R.Justino, E., Oliveira, L. S., and Sabourin, R., "Document reconstruction using dynamic programming," in [*International Conference on Acoustics, Speech, and Signal Processing*], 1393–1396 (2009).

[9] Kleber, F. and Sablatnig, R., "A survey of techniques for document and archaeology artifact reconstruction," in [*International Conference on Document Analysis and Recognition*], 1061–1065 (2009).

[10] Smet, P. D., "Semi-automatic forensic reconstruction of ripped-up documents," in [*International Conference on Document Analysis and Recognition*], 703–707 (2009).

[11] Diem, M., Kleber, F., and Sablatnig, R., "Document analysis applied to fragments: feature set for the reconstruction of torn documents," in [*International Workshop on Document Analysis System*], 393–400 (2010).

[12] Kohonen, T., "The self-organizing map," *Neurocomputing* **21(1-3)**, 1–6 (1998).

[13] Kohonen, T., "Self-organizing maps of massive document collections," in [*International Joint Conference on Neural Network*], **2**, 3–12 (2000).