

Automatic Text Removal and Replacement in Scanned Meteorological Thematic Maps

¹Ahsan Abdullah, ¹Sohyb Sami Abo Alshamat, ²Ahmad Bakhshwain and ³Ahtisham Aslam

¹Department of Information Technology, Faculty of Computing and Information Technology

²Department of Arid Regions Agriculture, Faculty of Meteorology, Environment and Arid Land Agriculture

³Department of Information Systems, Faculty of Computing and Information Technology
King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia

Article history

Received: 23-08-2014

Revised: 18-04-2015

Accepted: 04-06-2015

Corresponding Author:

Ahsan Abdullah
Department of Information
Technology, Faculty of
Computing and Information
Technology, King Abdulaziz
University, Jeddah, Kingdom
of Saudi Arabia
Email: aabdullah1@kau.edu.sa

Abstract: Geo-referenced data is required in many Geographic Information System (GIS) applications and one source of such data being maps. Meteorological maps are color-coded with different regions corresponding to different values of a parameter, parsing the image to convert into data is not very difficult. However, text and different planimetric elements overlaid on the regions in the map makes accurate image to data conversion a challenging problem, because it is almost impossible to exactly replace what was underneath the text or icons i.e., the need for inpainting. In this study, we propose a probabilistic technique that uses the probability of occurrence of colors present in the map along with the occurrence of those colors in the spatial neighborhood of the pixel (corresponding to text) whose color is required to be replaced. We test the limits of our proposed technique using simulated data and compare results with a popular image editing tool using public domain data with promising results.

Keywords: Noise, Image, GIS, Map, Inpainting

Introduction

Capturing features automatically from raster maps is a major challenge in Geographic Information Systems (GIS) for many reasons. First, there are many planimetric elements that overlap each other such as road lines, elevation contour lines, marks or soil features and so on. Second, scanning procedure or image compression may result in poor quality of data that make the recognition more difficult. Last, some old maps may be in black and white (without any other colors) that make the development of an automatic procedure almost impossible.

The problem of image restoration is an active field of research with diverse applications. One application area is image inpainting which consists of the cases where the actual image is covered by text. For such cases, the interest is in “removing” the undesired “overlaid” text and recreating the features of the image, which were concealed from view, because of being under the text. Another application domain of image inpainting being the images having regions with missing data. In such a case, the required image processing task is the filling of missing information based on the available neighboring information.

This in principle is an interpolation problem, where the unseen details of the image are estimated or guessed

based on the information from the observable parts of the image. Research work in this area comprises of special interpolation techniques based on the concepts of Partial Differential Equations (PDEs) such as by Chan *et al.* (1995; Bertalmio *et al.*, 2000). Note that the details that are completely hidden by the text cannot be completely recovered using any statistical or mathematical method. Therefore, the aim of text removal is not to recreate the true and original image; instead to simply create a new image that is a better representation or looks “good” in some sense (as is the case of *PhotoShop*) as compared to the original image. For the category of smooth images i.e., having a water surface, the technique of harmonic inpainting can be used, but harmonic inpainting is not suitable for images consisting of sharp features. For such images, other inpainting techniques are needed.

Another application of inpainting related to GIS is the restoration of historical maps that contain extensive cartographic information, such as network of roads, streams etc., but this information is “locked” inside the map and inaccessible to a Geographic Information System (GIS). Manual map digitization and subsequent “unlocking” requires extensive manual effort and cannot be used for large number of maps. Earlier approaches for automatic map processing generally required expert knowledge in

order to fine-tune parameters of the graphics tools and techniques, therefore they are not readily usable and attractive for non-expert users or for large sets of images.

Historical maps contain specific information that allows reconstruction of paths over time and space for urban area development or land use/cover studies. Usually the approach used for extracting these elements requires user involvement for digitizing, however, this significantly limits its utilization. It is therefore essential to develop automatic methods in order to create reproducible and fast procedures. Capturing features automatically from scanned paper maps is a major challenge in GIS applications for several reasons: (1) overlapping of many planimetric elements (2) scanning issues may result in poor image quality, (3) lack of colors complicating the identification of elements.

More formally, image inpainting reconstructs the marred regions or omitted parts in an image by employing information of neighboring regions. Image inpainting might also be termed as image alteration or manipulation. Ideally, it is desirable to recreate the master image; however, this is entirely not feasible without having the previous historical knowledge about the image i.e., some reference image. This being untrue in case of digital images, where the only image at our disposal and are thus we constrained to filling a hole that covers part of an object. Thus, it is impossible to completely swap that undesired object based on the available material, which in best case is partial. Once we are content with this being the aim of image inpainting, the corresponding image inpainting algorithms reconstruct what is expected to be in that hole, but also create an appealing continuance of data around the hole such that the reconstruction is not noticeable to a casual observer.

Currently image inpainting work is an active area of research in computer graphics. Inpainting has valuable applications in eliminating redundant objects, heritage protection, television and film special effects production etc. In museums, inpainting techniques are used for reconstruction of degraded paintings. Traditionally inpainting is the domain of professional artists and is generally a very time consuming and tedious task. The main goal of inpainting being reconstruction of damaged parts or missing parts of an image. Inpainting techniques have found extensive applications in many domains such as red eye correction, object removal from digital photos/maps, restoration of old films, super resolution, transmission, compression to name a few.

In all applications of image inpainting, the objective is to improve the general quality of an image or remove defects. Image inpainting problem is one of the fundamental image restoration problems. Inpainting also refers to the practice of the artists of refurbishing and repairing paintings as discussed by Nielsen (2003; Bertalmio *et al.*, 2000). Researchers working in different application domains have adopted different names for inpainting such as: Disocclusion, image

interpolation, error concealment and image replacement as described by Chan and Shen (2005) though each of technique carries its own individual characteristics. Image reconstruction problems such as inpainting and deblurring have historically been important image processing activities with numerous real world applications as discussed by Aubert and Kornprobst (2002; Chan *et al.*, 1995).

The paper is organized as follows, in section 2 necessary background of inpainting is provided with discussion of five main inpainting techniques along with formal problem definition, in section 3 related work is discussed and comparison done with the proposed technique, in section 4 i.e., materials and methods the proposed solution is formally presented and briefly explained using examples, section 5 is the results using simulated data, in section 6 the results of simulated as well as real data are discussed and finally section 7 is conclusions.

Background

The term inpainting is rooted in art renovation, where it is also termed as retouching. Restoration of the primitive artwork started as early as the Renaissance, with the purpose to bring the primitive artwork (mainly paintings) “back to life” as well as to fill the gaps or damaged or missing portions in these paintings. The need to retouch the image in an inconspicuous way, outspread logically from paintings to photography and then to film. However, the purpose remained the same i.e., to undo degeneration caused by e.g., dust spots and scratches in film, or to remove or add entities or objects (e.g., the notorious “airbrushing” of political adversaries during the Stalin era of the former U.S.S.R). In the digital realm, the inpainting problem first got the name of “error masking” in the domain of telecommunications, where the requirement was to fill-in image blocks that went missing while the data was being transmitted. One of the earlier works of automatic inpainting in a typical environment was called “image disocclusion,” since the technique considered the image-gap as an obstructing or occluding object required to be “deleted”, resulting in restoration of the image underneath. Common terms used to symbolize current inpainting algorithms are “image fill-in” and “image completion” algorithms.

Currently different approaches are used for image inpainting that can be classified into five main inpainting classes as follows:

- Texture Synthesis based
- Hybrid
- Exemplar based
- Semi-automatic and Fast
- Partial Differential Equation (PDE) based

In this section we will briefly review the theories, benefits and disadvantages of these five popular techniques.

Texture Synthesis Based Inpainting

These are one of the first breed of image inpainting techniques. These algorithms fill and recreate the missing regions using information of similar neighborhoods of the corrupted pixels. Using an initial seed, the texture generation algorithms subsequently create the new image pixels and attempt to retain the neighboring configuration of the image. Historically, these methods were employed in inpainting to cover and recreate the lost areas by testing/evaluating and then replicating pixels from the neighborhood. For example, the Markov Random Field (MRF) method models the neighboring locations of the pixels and new texture is produced by exploring the texture present and discovering altogether the similar local patterns. The difference between different texture synthesis techniques is how continuousness is maintained between the existing pixels and the synthesized inpainted pixels in the hole.

Thus, the main objective of these techniques is to create texture patterns that are comparable to a given sample texture, such that the synthesized texture keeps the statistical characteristics of the seed-texture and the synthesized pattern does not appear to be just a repeating reorganization of the seed-texture. As opposed to vibrational inpainting i.e., Partial Differential Equation (PDE) inpainting, depending on the choice of user, the content and location of sample selected, texture generation based methods can cover large textured areas inconspicuously.

Observe that in the context of our work, texture-based techniques are not suitable because it is important that the hidden points (i.e., under the text) to have similar agro-meteorological values. Texture generation techniques can be classified into three main classes i.e., (i) pixel-based (ii) Statistical (parametric) and (iii) patch-based (non-parametric). Statistical techniques are inherently expected to produce better results for irregular/stochastic textures, but typically cannot regenerate good results for structured/regular textures. On the contrary, instead of applying filters, the pixel-based techniques “build” on the sample texture pixel-by-pixel at a time and their final results are of better quality as compared to the statistical techniques, but usually cannot generate large regular textures. Lastly, patch-based methods, such as by Efros and Leung (1999) “build” on a sample texture one patch at a time contrary to pixel-by-pixel, thus they quickly generate and provide more credible and consistent textures.

PDE Based Inpainting

The Partial Differential Equation (PDE) based iterative inpainting technique was suggested by Bertalmio *et al.* (2000). PDE-based inpainting uses photometric and geometric information present at the boundary of the blocked area, and applies this information into the region

itself. This is done by transmitting the information in the bearing of least change by means of “isophote lines”. The application is however, restricted by scalability i.e., produces good results for small missing regions, but for large missing regions the algorithm not only slow but the quality of the results also deteriorates. In view of these limitations, Chan and Shen (2001) suggested the Total Variational (TV) inpainting model. The foundation of TV models is the strength of the isophotes and uses anisotropic diffusion and Euler-Lagrange equation. The model produces reasonably good results for small regions and the quality of results do not deteriorates for noise elimination applications. However, the drawback of the proposed model is that it neither synthesizes texture patterns nor connects broken edges. Therefore, Shen and Chan (2001) extended this model to the Curvature Driven Diffusion (CDD) model; CDD covers these aspects and handles the curved structures efficiently.

Telea (2004) proposed a marching PDE method which is not only fast but have a simpler implementation as compared to other PDE-based algorithms. Generally the PDE algorithms are slow and struggle with large damaged regions. PDE-based techniques have number of popular applications in the domains of image restoration, segmentation etc. These algorithms stay focused on the order and structure of the inpainting area, thus producing blurred images. Another drawback of these algorithms is scalability i.e., they are not suitable for reproducing large textured regions.

Exemplar Based Inpainting

The exemplar-based approach has proved to be an effective and important class of inpainting algorithmic techniques. The technique comprises of two steps, step-1 consists of priority assignment and in step-2 the best matching patch is selected. In this approach, parts of the best matching patches are taken from known regions; this matching being quantified based on similarity measures using certain metrics, subsequently the selected regions are pasted inside the target patches of the absent area to complete the task.

Note that exemplar-based inpainting creates the unknown region i.e., the target region in repeating steps, using the highly matching patch in the master region. Following the covering order, the structures in the lost regions are filtered using the neighboring region’s spatial information. This results in an efficient reconstruction of large target regions.

Let the inpainted region be denoted by Ω , source region ($I - \Omega$) denoted by ϕ and boundary of the target region denoted by $\delta\Omega$ as shown in Fig. 1. The algorithm consists of the following four main steps, for details please see Pandya and Pandya (2014):

- Initializing the Target Region
- Computing Filling Priorities

- Searching Example and Compositing
- Updating Image Information

Hybrid Inpainting

There are also hybrid inpainting techniques which are called Image Completion techniques and are used for covering large targets with missing blocks. The hybrid techniques preserve both texture and structure in an aesthetically credible manner by combining the texture generation as well as PDE based inpainting techniques for covering and filling holes Thangamani *et al.* (2011). The main concept of these techniques is that they divide the image into two disjoint parts i.e., texture regions and structure region. The matching separated regions are covered by edge broadcasting techniques and texture creation algorithms. If the region to be filled is not small, these algorithms are computationally expensive. An important inpainting option that is more “natural” is structure completion using segmentation. The technique consists of a two-step approach, in step-1; structure completion is followed by texture generation. Step-1 consists of using the color, geometry and texture information of the input. Subsequently the partitioning regions are inferred using tensor voting to synthesize a complete segmentation for the input. In step-2, tensor voting is repeated for generating color and texture information in each segment.

Fast and Semi-Automatic Inpainting

These popular techniques require user assistance in the form of guidelines to help in structure and complete their task. The method by Xu and Jian (2010) proposed an inpainting technique with structure propagation. In the first step of the two-step proposed technique, critical missing information is manually pinpointed in the hole by outlining entity borders from the identified regions to the unidentified regions. This is followed by a patch-based texture generation that synthesizes the texture to fill the hole. The holes in the image are then synthesized along the outline traced by the user by framing the inpainting problem as a global optimization problem with several consistency and structural limits. In case of a single outline traced, an optimal answer can be obtained by using simple dynamic programming. However, for multiple traces, the optimization is NP-complete thus, a significantly harder problem, therefore, approximate answers are proposed based on belief dissemination. Based on the coverage of the inpainting area, the techniques described here can consume anywhere from minutes to hours to finish, resulting in their impractical use for online near real-time applications.

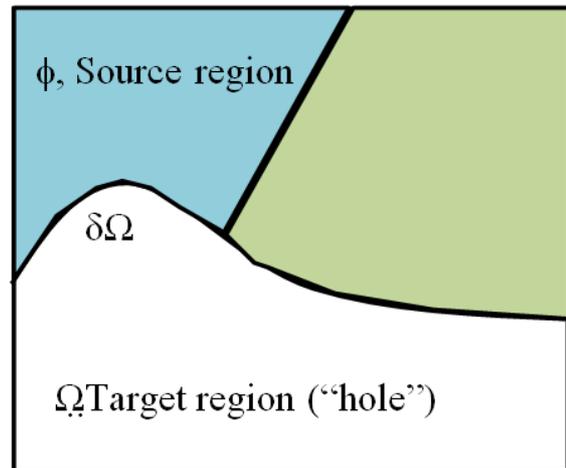


Fig. 1. Terminologies used in inpainting

Problem Formulation

Let $u: \phi \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ be an original image describing a real scene (the unknown on the domain Ω) and let u_0 be the observed image of the same scene (i.e., the data or the degradation of u). The domain Ω is a subset of ϕ where ϕ is the domain of the image. We suppose that $u_0 \in L^\infty(\phi)$. Recovering u from u_0 knowing a given model of degradation is a typical example of an inverse problem. This inpainting problem is from a mathematical point of view, an interpolation problem and can be stated as follows:

For a given domain ϕ , a rectangular image u_0 identified outside hole Ω , with the objective being to find an image u , an inpainting of u i.e., u_0 , such that u matches u_0 outside Ω and has “meaningful” matter inside Ω . Where Ω stands for the region to be inpainted, $\delta\Omega$ for its boundary and no assumption on the topology of Ω is being made. Figure 1 illustrates the problem domain and the inpainting domain, also note that the observed or measured portion u_0 of u can be either noisy or hazy or damaged.

So the goal of inpainting is to retrieve u for the entire image domain ϕ as authentically and as likely using the available data u_0 on $\phi \setminus \Omega$ and answer the following question: How “meaningful” the inpainted image should be?

Related Work

Automatic information extraction techniques are an active field of research and lately several such techniques have been reported for isolating information from hard-copy maps. These techniques can be broadly divided into three categories i.e., (i) text separation (alphanumeric characters), line extraction (contours) (ii) milestones e.g., applied computing, machine learning AI for area/pattern recognition (polygons), structural pattern

analysis (symbols) and (iii) map comprehension and GIS analysis along with text/graphics processing and partitioning of map labels. In the subsequent sections, a brief overview of the associated work is presented covering some of the more popular methods.

Significant work has been reported targeting both semi-automatic as well as automatic information mining from historic paper-based hard-copy maps. The existing work covers automatic data capture and pattern recognition methods for an assortment of maps across the world. For example, Lichtner (1985) suggested pattern recognition by raster to vector transformation. Deseilligny *et al.* (1998) performed text recognition and extracted information from the French national topographic maps and Li (1999) used the US Geological Survey (USGS) maps for similar work. Guerin and Deseilligny (1995) proposed automated digitization and interpretation of scanned maps using spatial associations so as to completely extract and make a list of important geographical objects to improve GIS analysis. Leyk *et al.* (2006) extracted population demographics from the Kenyan legacy African paper maps, in Europe, Kerle and de Leeuw (2009) performed forest cover extraction from historical Swiss national topographic map series.

Traditionally capturing cartographic objects from old maps is based on user intervention (for digitizing) and suffers from scalability. Several researchers attempted to develop automated data capture techniques in order to establish reproducible procedures, such as by Leyk *et al.* (2006; Mahmoud, 2012). However, most of these methods are tailor-made for only certain type of map and generally not customizable.

Several researchers have proposed automatic methods to capture geographical objects in scanned thematic maps such as Lichtner (1985) and Ansoult *et al.* (1990). Many examples of maps have been investigated for feature recognition such as the topographic maps of the United States Geological Survey by Li (1999), the military maps of the Polish Geographic Institute by Iwanowski and Kozak (2012) and the Swiss National historical map by Leyk *et al.* (2006). Generally automatic extraction procedures consist in three steps: (1) a clean-up step, (2) a feature recognition/extraction step and (3) a post-processing step. The first step is needed to make the feature extraction easier while the last step is carried out to improve the extraction results. Steps 2 and 3 can be viewed as a single integrated process according to some authors, such as Wise (1999).

Text extraction is one of the most complicated tasks to automate in old maps. Text and graphics are often overlapping that makes separation of the elements difficult, Cao and Tan (2002) have proposed a method for text recognition in maps. Based on the observation that character strokes generally consist of short segments

as compared to cartographic elements, Levachkine *et al.* (2002) used a combination of line extension and line width to separate elements and improve text extraction. Text can be also detected using color attributes when the color of text characters differs from other objects. For example Centeno (1998) used a Karhunen Loeve color-space conversion to isolate and capture text. Myers *et al.* (1996) proposed a character recognition based method that uses a verification-based approach to detect text without requiring pre-segmentation graphical entities.

Herrault *et al.* (2013) developed an automatic procedure to capture features in old maps. The three-step procedure uses unsupervised classification (K-means algorithm) and color image segmentation to extract forest structures from French historical maps. The procedure starts by first cleaning the maps and eliminating elevation contour lines with filtering techniques. This is followed by color space transformation from RGB to L*a*b color space to improve evenness of the image. Lastly, a post-processing step uses morphological operations and contextual rules to clean the features.

Text in a typical image carries rich high-level semantic information about a scene, this information can be used to assist in diverse applications, such as image indexing and search, image understanding, geo-location or navigation and human computer interaction, to name a few. However, currently most of the text detection and recognition systems are conceived for near-horizontal or horizontal texts, thus limiting their application domain. With the increasingly popular computing-on-the-go devices, detecting texts of arbitrary orientations for images captured by devices such as camera phones and tablets that are under less ordered conditions has become increasingly important and at the same time is a challenging task. Chiang *et al.* (2013) proposed an algorithm to detect text of random orientations in typical images. The algorithm is based on a two-level classification arrangement using two sets of features and is specially designed for capturing both essential and orientation-invariant characteristics of text. For better evaluation and comparison of the proposed method with other existing algorithms, the researchers also generated an extensive and challenging dataset. The synthetic data set included different types of texts in diverse real-world scenarios and environments. The researchers also suggested a novel and suitable evaluation protocol, for benchmarking those algorithms that are designed for texts of varying orientations.

The proposed technique is not geography dependent and requires little or no user input for working, is not dependent on the character strokes for text recognition as text is recognized based on range of color. The technique is simple yet effective as will be demonstrated in sections 5 and 6 using simulated and real data.

Materials and Methods

Our proposed methodology consists of three main steps as follows:

- Noise detection: Noise or text detection is based on the observation that text can be detected using color attributes when the color of text characters differs from other objects, this being the Karhunen Loeve approach as adopted by Centeno (1998)
- Noise removal: Once the noise is detected, it is straightforward to remove all the pixels corresponding to noise, this being an intermediate step
- Noise replacement: This being the challenging step and consists of replacing the noise with the most probable data values. A natural way to deal with this kind of problem is to use probabilities (see, for instance Bouman and Sauer (1993), Chellappa and Jain (1993), Demoment (1989), Geman and Geman (1984)) i.e., the approach that we have also adopted in our work. Our approach is not only based on the probability of occurrence of the colors in the map but also the spatial neighborhood of the pixel whose color is to be replaced

Noise Detection and Removal

Based on color oriented text removal, there can be number of ways of selecting the color to remove, such as, (i) moving the mouse cursor and picking color of every pixel that comes under the mouse or (ii) defining a rectangular region by pressing and holding the mouse button or (iii) hard-coding a range of typical colors used for replacement of text in a map.

We have not used any of these approaches, as it can be observed that generally text in maps is black with white background (or vice-versa) or black text with background consisting of shades of grey. Thus, if we can identify the color of the pixel to be in the grey range or black then no need to use mouse movement to manually identify text.

Observe that for the RGB color model, for black $R = G = B = 0$ and for white $R = G = B = 255$ and for 50% grey $R = G = B = 128$ i.e., when the value of R,G,B are similar we have shades of grey and in extreme case we have black or white color.

Thus it is easy to identify if the pixel in the map belongs to text or data by using the following relation:

$$Abs(R_r - G_r) + Abs(R_r - B_r) + Abs(B_r - G_r) \quad (1)$$

Let R_r , G_r and B_r be the color of the pixel for which it is required to know if it is text or not, then find

Here *Abs* is absolute value, for grey-noise value Equation 1 will be a small number and for colored value

Equation 1 will be large number. Thus, a threshold can be set for Equation 1 i.e., for which particular value (s), the pixel belongs to text. By experience, the value of 10 was observed to provide good discrimination between text and color background. Note that the approach adopted in Equation 1 is not restricted to grey-noise only, for example, setting a large value for R and then increasing G and B from small values up to 255 provides shades of red “noise” for corresponding colored text identification.

Example-1

Figure 2 in Example-1 shows the application of Equation 1 where the result column is the value of Equation 1, it can be observed that non-grey values don't belong to text.

Noise Replacement

After noise i.e., text identification and removal the next step is noise replacement. Noise replacement is based on the assumption that the meteorological parameter values at a particular location are similar to or spatially influenced by the meteorological values at the neighboring locations. Consider an image I of height H and width W and consisting of multiple colors corresponding to different range of values of a particular meteorological parameter.

Let the Prob (C) be the probability that a pixel is of color C:

$$Prob(C) = \frac{No\ of\ pixels\ of\ C}{H * W}$$

Figure 3 shows the pixel P whose color is to be replaced. We proceed by checking the color of pixels in three directions or neighborhoods and perform replacement based on the collective impact of the color of those pixels i.e., considering the spatial influence of the neighboring parameter values.

Let the coordinates of P i.e., pixel to replace color be x,y then we look for the neighboring colors (corresponding to parameter values) in three directions as follows:

- LEFT i.e., y = constant and x = x-1
- RIGHT i.e., x + 1 and y-1 and x = x + 1
- TOP i.e., x = constant and y = y-1

No.	R?	G?	B?	Result	Color	Text <=40
1	10	20	30	40	Black	Yes
2	30	100	40	140	Green	No
3	250	255	100	310	Yellow	No
4	125	125	100	50	Grey	No
5	200	190	205	30	Grey	Yes

Fig. 2. Identification of text based on grey-noise

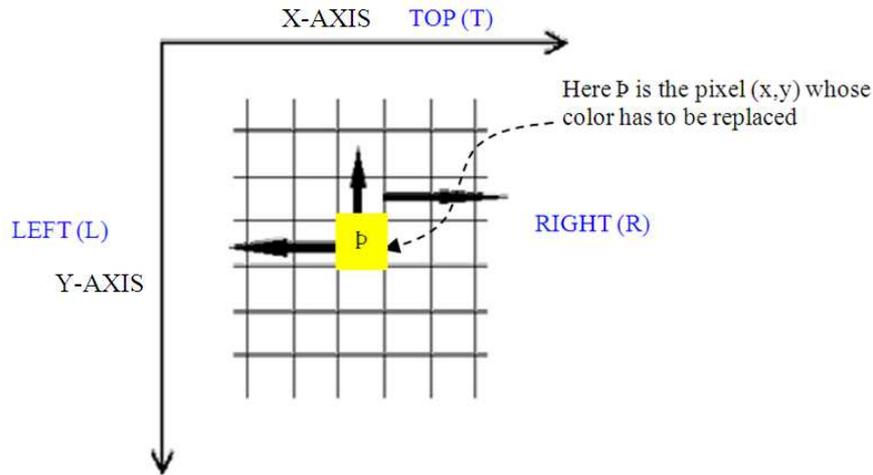


Fig. 3. Neighborhood of the pixel whose color is to be replaced

		R	G	B	
C1 = Blue		0	0	255	$\text{Prob}(C_1) \times W = 2$
C2 = Yellow		255	255	0	$\text{Prob}(C_2) \times W = 2$
C3 = Yellow		255	255	0	$\text{Prob}(C_3) \times H = 4$
Result		191	191	63	Sum 8

Fig. 4. Generating resultant color based on spatial neighborhood

Now the question is, how many pixels to traverse or check in each of the three directions i.e., scope of spatial neighborhood? This traversal is proposed to be as per the probability of occurrence of the color at locations (i) x-1 (ii) y -1 and (iii) x, y-1 or till the color at these locations changes.

The traversal in the three directions to be as follows (Fig. 3):

- LEFT: $\text{Prob}(C_1) \times W$ pixels
- RIGHT: $\text{Prob}(C_2) \times W$ pixels
- TOP: $\text{Prob}(C_3) \times H$ pixels
- Here C_1 is the color of pixel at location x-1, y
- C_2 is the color of pixel at location x, y -1 and
- C_3 is the color of pixel at location x+1, y-1

Let $Mov(D)$ be the number of pixel traversal in direction D i.e., $Mov(LEFT)$ the number of pixels traversal in LEFT direction i.e., $\text{prob}(C_1) \times W$

The replacement color of pixel at location x,y will be floor of the following equation:

$$= \frac{Mov(LEFT) \times C_1 + Mov(RIGHT) \times C_2 + Mov(TOP) \times C_3}{Mov(LEFT) + Mov(RIGHT) + Mov(TOP)} \quad (2)$$

Note that the color replacement will be at the RGB level of pixel i.e., Equation (2) will be applied to each individual R, G and B value of the pixel at location x, y.

Example-2

Consider a 10×10 pixel image with five colors C_1, C_2, \dots, C_5 such that the probability of occurrence of each of the five colors is 20, 20, 40, 15 and 5% respectively. Let the three colors at the starting locations for LEFT, RIGHT and TOP being C_1, C_2 and C_3 respectively be shown in Fig. 4.

Thus Equation 2 is applied to each of the individual RGB values to get the resultant RGB values as follows:

- $R_7 = (2 \times 0 + 255 \times 2 + 255 \times 4) / 8 = \text{floor}[191.25] = 191$
- $G_7 = (2 \times 0 + 255 \times 2 + 255 \times 4) / 8 = \text{floor}[191.25] = 191$
- $B_7 = (2 \times 255 + 0 \times 2 + 0 \times 4) / 8 = \text{floor}[63.75] = 63$

Thus we get the RGB values of the resultant color, which we observe to be close to yellow. Observe that with the change in the number of pixels of a certain color the traversal in the map will be correspondingly different that will subsequently impact the resultant color of the pixel whose color is to be replaced.

Results

In this section, we will perform text removal and replacement under controlled environment, so as to study the limits and strengths of the proposed technique and analyze the results using pixel level comparison. This will be followed by drawing family of curves for text removal and replacement of text of different font sizes. Subsequently in section 6 we will apply the proposed technique on real data and compare the results with the results of using popular image editing tool i.e., *PhotoShop* that uses texture image based in painting.

Simulated Data

For the controlled experiment we consider a checkered yellow-red grid of size 230×230 pixels with overlaid text of 54 font size and the area of the text i.e., "This" being 49×62 pixels, thus the text area being $9,238 \text{ pixel}^2$, the area of checkered box being $52,900 \text{ pixel}^2$ and the area of unit box (red or yellow) being 529 pixel^2 . The ratio of Text: Background being 1:5.72 and the ratio of Text: Unit box being 1: 0.057 as shown in Fig. 5.



Fig. 5. Black text on white background overlaid on checkered box for noise removal-replacement analysis

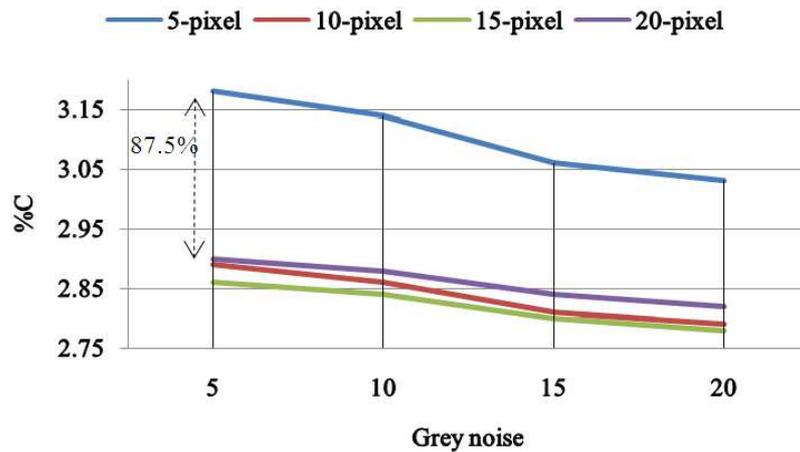


Fig. 6. Family of curves with font size 54 on red-yellow checkered background

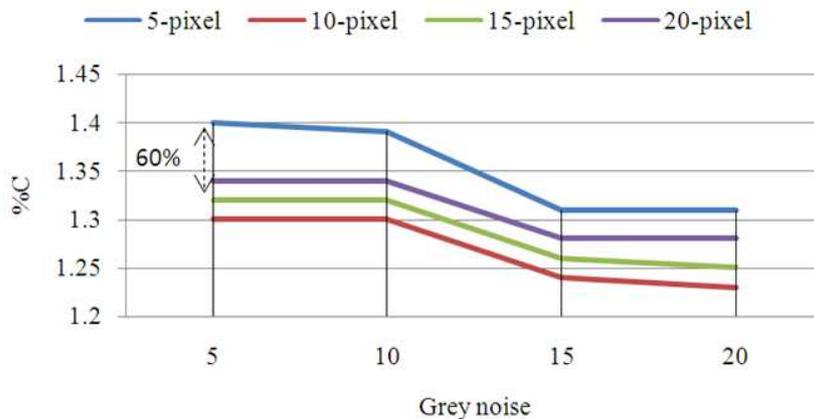


Fig. 7. Family of curves with font size 32 on red-yellow checkered background

The indigenous noise removal-replacement technique was used with four values of grey-noise (Equation 1) i.e., 5, 10, 15 and 20. Since there are equal number of red and yellow pixels, therefore the probability of finding each colored pixel is same i.e., 0.5 thus the traversal in each of the three directions will be same. For traversal in the three directions i.e., LEFT, RIGHT and TOP we force the application to traverse 5, 10, 15 and 20 pixels. To quantify the amount of noise removal-replacement, we use the Image Diff tool (www.perforce.com) that compares the two images i.e., text on checkered background and only checkered background and provides two comparison parameters i.e., % change in the location of pixels (%P) and % change in the color of pixels (%C). Note that the smaller the two values (%P and %C) the better i.e., the cleaned image being closer or similar to the image without text. The corresponding family of curves for font size 54 and 32 are shown in Fig. 6 and 7, respectively. Since the %P values remained almost same for the family of curves, therefore, the corresponding family of curves are not shown here. In Fig. 6 and 7 grey-noise is the RGB value used for identification of noise as discussed in section 4.1.

By comparing Fig. 6 and 7a significant difference can be observed in terms of closeness and compactness of the curves. Therefore, to explore further we draw additional family of curves for the font size that lies between the two fonts (32 and 54), which could be the average i.e., 43. However, when we closely observe the relationship between the font size and the text area covered for the font a different picture emerges as follows:

- The area taken by T of font size 54 being $48 \times 62 = 2976 \text{ pixel}^2$
- The area taken by T of font size 32 being $29 \times 36 = 1044 \text{ pixel}^2$
- The unit font per pixel for 54 size font being $54/2976 = 0.018 \text{ pixels}$
- The unit font per pixel for 32 size font being $32/1044 = 0.030 \text{ pixels}$

It can be observed that the relationship between number of pixels and font size is nonlinear. For the sake of accuracy, we take the average area for T for the two font sizes i.e., 54 and 32 which come to be 2010 pixel^2 . Thus the question is for which font size for T the area is close to 2010 pixel^2 . This turns out to be 45 font sizes instead of 43 (average of 54 and 32). The corresponding family of curves is shown in Fig. 8.

Discussion

Simulated Data

From Fig. 6 to 8 it can be observed that as the font size increases, the difference in the range of %C values

increases. The % difference between the two top values of %C is for the family of curves is shown next to the dashed line from Fig. 6 to 8. Note that since we considered fonts based on pixel^2 , thus the difference are also proportionate i.e., the % difference between 54 and 45 size fonts is close to the average of the differences for the two fonts. For the font size 32 the difference in the range of %C values for the quality of results is similar for all the pixel length traversals (10, 15 and 20-pixels) for L-R-T (LEFT-RIGHT-TOP). The results for the pixel length-5 deteriorate quickly as compared to other pixel length traversals as the font size increases. Best results obtained when the grey noise values are large instead of being small. Thus the results are not good for small traversal length i.e., 5-pixels or large traversal length i.e., 20-pixels, but are good for intermediate values i.e., 10-pixels and 15-pixels. The order of quality of results is almost same i.e., worst for 5-pixels, followed by 20-pixels and then 10 and 15-pixel lengths.

Real Data

Now we apply the proposed technique on real data and compare the results with the results obtained using *PhotoShop*. For this comparison we use the Saudi Arabia rainfall map covering a period of 50 years, the map is available at http://www.nar.ucar.edu/2008/RAL/goal_1/priority_2.php.

PhotoShop is an excellent tool, with powerful features, one of those features being removal and replacement of selected part of image by an option called as content aware which is conceptually similar to texture synthesis based inpainting. *PhotoShop* however has a steep learning curve for which digital artists could be willing to spend the required amount of time, but not a scientist. For example, the text removal process using *PhotoShop* consist of 7 steps (<http://www.wikihow.com/Remove-Text-from-a-Photo-in-Photoshop>) using the content aware approach, however the proposed technique uses a single step for text removal. What content aware feature of *PhotoShop* does is that it replaces the removed content with the neighboring content which gives a nice appealing look in terms of hiding the content removed. The problem with this approach is that when there is high contrast and variation of colors around the content to be replaced (as in a map Fig. 9a), when *PhotoShop* generates the content in the text removed space the replacement results are highlighted instead of getting concealed, as can be observed in Fig. 9b.

The indigenous noise removal technique adopts a different approach as compared to *PhotoShop* for selecting, removing and then replacing text. First the text is identified based on the value of difference in RGB values of a pixel instead of selection using *PhotoShop* eyedropper tool and then the pixels corresponding to text are removed.

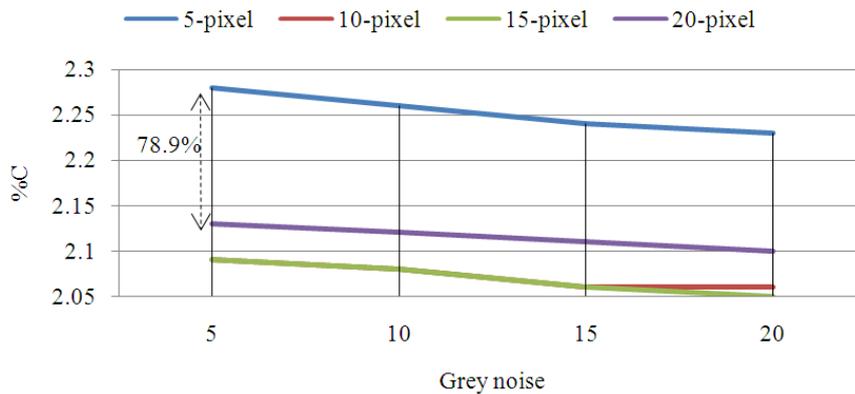


Fig. 8. Family of curves with font size 45 on red-yellow checkered background

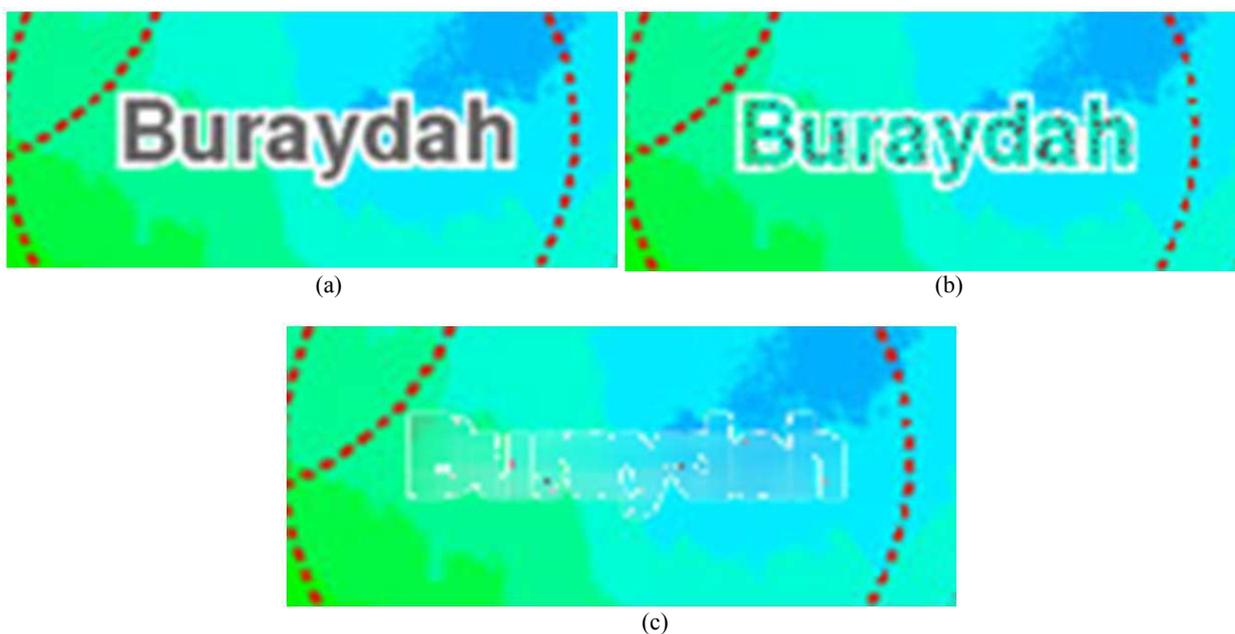


Fig. 9. Comparison of text removal for real data using *PhotoShop* and noise removal tool (a): Sample text in map (b): Text removal using *PhotoShop* (c): Text removal using indigenous technique

Subsequently our probabilistic approach of color replacement is used by averaging the RGB values of pixel colors based on the probability of occurrence of color of neighboring pixels, the results are shown in Fig. 9c, the values of grey scale and LEFT-RIGHT-TOP were guided by the family of curves of Fig. 6 to 7 i.e., grey noise = 20 and L=R=T=10 pixels.

Since we are considering removing text from parameter maps, therefore, it is rather logical to consider the average impact of the parameter values of the spatial neighborhood instead of replacing the former text pixels with parameter values of neighborhood pixels. This not only results in logically sound approach, but as can be observed from Fig. 9c text removal is significantly “better”. The quality of results using indigenous

technique can be seen to be significantly better than the technique based on the popular image editing tool. To quantify noise removal-replacement we use the Image Diff tool pixel-by-pixel comparison. The tool not only visually displays the difference between the images, but also calculates the % change in pixels and color between the images compared.

Figure 9 also shows the quantified difference between the two images i.e., with text present and with text removed-replaced. Observe that the more the difference in % P and % C the better i.e., higher level of text removal. This is indeed true in case of the indigenous Noise Removal technique where the % change in pixels and colors as compared to the given map is higher as compared to *PhotoShop*.

Conclusion

Hard copy printed maps, as well as digital maps are a rich source of information, provided the overlaid text can be removed with minimum loss of information underneath the text i.e., inpainting. There are many good inpainting techniques that produce good results, but are slow with a steep learning curve. In this study, we have proposed a novel inpainting technique that performs overlaid text removal and data replacement using the probability of occurrence of colors in the spatial neighborhood of the pixel (corresponding to text) whose color is required to be replaced. Using real data, we have shown that our proposed technique is fast, without a steep learning curve and produce comparatively better results as compared to the commercial tool, which makes inpainting results look “good” instead of being meteorologically correct.

Acknowledgement

This project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH)-King Abdulaziz City for Science and Technology-the Kingdom of Saudi Arabia-award number 12-AGR2709-3. The authors also, acknowledge with thanks Science and Technology Unit, King Abdulaziz University for technical support.

Author's Contributions

A. Abdullah: Author is the Principal Investigator of the funded project and made major contributions in developing the concept and design, application development, analysis and interpretation of results, considerable contribution in drafting the article with significant intellectual input, critical review for content and guiding the co-authors.

S. Alshamat: Assisted in conducting experiments and compilation of results, also assisted in literature review.

A. Bakhshwain: Assisted in analysis of results, drafting and review of the article.

A. Aslam: Assisted in concept and interpretation of results.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Ansoult, M.M., P. Soille and J. Loodts, 1990. Mathematical morphology: A tool for automated GIS data acquisition from scanned thematic maps. *Photogrammetric Eng. Remote Sensing*, 56: 1263-1271.

- Aubert, G. and P. Kornprobst, 2002. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. 2nd Edn., Springer Science and Business Media, New York, ISBN-10: 0387445889, pp: 379.
- Bertalmio, M., G. Sapiro, V. Caselles and C. Ballester, 2000. Image inpainting. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, July 23-28, ACM Press/Addison-Wesley, USA, pp: 417-424. DOI: 10.1145/344779.344972
- Bouman, C. and K. Sauer, 1993. A generalized Gaussian image model for edge-preserving MAP estimation, *IEEE Trans. Image Process.*, 2: 296-310. DOI 10.1109/83.236536
- Cao, R. and C. Tan, 2002. Text/graphics separation in maps, *Proceedings of the 4th International Workshop, GREC 2001 Kingston, Sep. 7-8*, Springer Berlin Heidelberg, Canada, pp: 167-177. DOI: 10.1007/3-540-45868-9_14
- Centeno, J.S., 1998. Segmentation of thematic maps using colour and spatial attributes. *Proceedings of the 2nd International Workshop*, Aug. 22-23, Springer Berlin Heidelberg, France, pp: 221-230. DOI: 10.1007/3-540-64381-8_51
- Chan, T.F. and J. Shen, 2001. Nontexture inpainting by curvature-driven diffusions. *J. Visual Commun. Image Representa.*, 4: 436-449. DOI: 10.1006/jvci.2001.0487
- Chan, T.F. and J. Shen, 2005. Image processing and analysis: Variational, PDE, wavelet and stochastic methods. *SIAM Society Industrial Appl. Math.*, DOI: 10.1137/1.9780898717877
- Chan, T.F., H.M. Zhou and R.H. Chan, 1995. Continuation method for total variation denoising problems. *Proceedings of the Advanced Signal Processing Algorithms*, Jul. 9, SPIE, United States, pp: 314-325. DOI: 10.1117/12.211408
- Chellappa, R. and A. Jain, 1993. *Markov Random Fields: Theory and Application*. 1st Edn., Academic Press, Boston.
- Chiang, Y.Y., S. Leyk and C. Knoblock, 2013. Efficient and robust graphics recognition from historical maps. *Proceedings of the 9th International Workshop, GREC 2011, Seoul, Sep.15-16*, Springer, Korea, pp: 25-35. DOI: 10.1007/978-3-642-36824-0_3
- Demoment, G., 1989. Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Tran. Acoustics Speech Signal Process.*, 37: 2024-2036. DOI: 10.1109/29.45551
- Deseilligny, M.P., R. Mariani, J. Labiche and R. Mullot, 1998. Topographic maps automatic interpretation: Some proposed strategies. *Proceedings of the 2nd International Workshop GREC' 97 Nancy, Aug. 22-23*, Springer, France, pp: 175-193. DOI: 10.1007/3-540-64381-8_48

- Efros, A. and T.K. Leung, 1999. Texture synthesis by non-parametric sampling. Proceedings of the 17th IEEE International Conference on Computer Vision, Sep. 20-27, IEEE Xplore Perss, Kerkyra, pp: 1033-1038. DOI: 10.1109/ICCV.1999.790383
- Geman, S. and D. Geman, 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Tran. Pattern Analysis Machine Intell., 6: 721-741. DOI: 10.1109/TPAMI.1984.4767596
- Guerin, P. and M.P. Deseilligny, 1995. Global analysis of scanned maps for an automatic interpretation. Proceedings of the 5th International Conference on Image Processing and its applications, Jul. 4-6, IET, Edinburgh, UK, pp: 80-84. DOI: 10.1049/cp:19950624
- Herrault, P.A., D. Sheeren, M. Fauvel and M. Paegelow, 2013. Automatic extraction of forests from historical maps based on unsupervised classification in the CIE Lab color space. Springer Int. Publish., pp: 95-112. DOI: 10.1007/978-3-319-00615-4_6
- Iwanowski, M. and J. Kozak, 2012. Automatic detection of forest regions on scanned old maps. Electronic Rev., 88: 249-252.
- Kerle, N. and J. de Leeuw, 2009. Reviving legacy population maps with object-oriented image processing techniques. IEEE Trans. Geosci. Remote Sens., 47: 2392-2402. DOI: 10.1109/TGRS.2008.2010853
- Levachkine, S., A. Velázquez, V. Alexandrov and M. Kharinov, 2002. Semantic analysis and recognition of raster-scanned color cartographic images. Proceedings of the 4th International Workshop, GREC 2001 Kingston, Sep. 7-8, Springer, Canada, pp: 178-189. DOI: 10.1007/3-540-45868-9_15
- Leyk, S., R. Boesch and R. Weibel, 2006. Saliency and semantic processing: Extracting forest cover from historical topographic maps. Pattern Recogni., 39: 953-968 DOI: 10.1016/j.patcog.2005.10.018
- Li, L., 1999. Cooperative text and line-art extraction from a topographic map. Proceedings of the 4th International Conference on Document Analysis and Recognition, Sep. 20-22, IEEE Xplore Perss, Bangalore, pp: 467-470. DOI: 10.1109/ICDAR.1999.791826
- Lichtner, W.P., 1985. Pattern recognition procedures for automatic digitizing of cadastral maps. University of Hanover FGR.
- Mahmoud, M., 2012. Information Extraction from Paper maps using object oriented analysis. MSc Thesis, Ensched, Netherlands.
- Myers, G.K., P.G. Mulgaonkar, C.H. Chen, J.L. De Curtins and E. Chen, 1996. Verification-based approach for automated text and feature extraction. Proceedings of the 1st International Workshop University Park, Aug. 10-11, Springer, USA, pp: 190-203. DOI: 10.1007/3-540-61226-2_16
- Nielsen, M., 2003. Harmonic and TV inpainting, Lecture Notes.
- Pandya, N. and M. Pandya, 2014. An image inpainting using patch-based synthesis Via sparse representation. Inter. J. Sci. Techn. Eng., pp: 11-14.
- Shen, J. and T.F. Chan, 2001. Mathematical Models for Local Nontexture inpaintings. SIAM J. Appl. Math., 62: 1019-1043. DOI: 10.1137/S0036139900368844
- Telea, A., 2004. An image inpainting technique based on the fast marching method. J. Graphics Tools, 9: 23-34. DOI: 10.1080/10867651.2004.10487596
- Thangamani, K., T. Ishikawa, K. Makita and T. Kurata, 2011. Hybrid inpainting algorithm with superpixels and hash table for inpainting the 3D model. Proceedings of the International Symposium on System Integration, Dec. 20-22, IEEE Xplore Perss, Kyoto, pp: 667-672. DOI: 10.1109/SII.2011.6147528
- Wise, S., 1999. Extracting raster GIS data from scanned thematic maps. Trans. GIS, 3: 221-237. DOI: 10.1111/1467-9671.00019
- Xu, Z. and S. Jian, 2010. Image inpainting by patch propagation using patch sparsity. IEEE Trans. Image Process., 19: 1153-1165. DOI: 10.1109/TIP.2010.2042098