The design and realization of Ascii Art software

PAVEL POKORNÝ, ROMAN ŽÁK, JAROMÍR ŠVEJDA
Department of Computer and Communication Systems
Thomas Bata University in Zlín, Faculty of Applied Informatics
nám. T.G. Masaryka 5555, 760 01 Zlín
CZECH REPUBLIC
pokorny@fai.utb.cz, rzak@fai.utb.cz, svejda@fai.utb.cz http://www.utb.cz

Abstract: This paper describes a design and a realization of the computer software which converts bitmap image to a character sequence. The aim of our development was maximally conservation of the original image information in the created text files. Therefore, program is very complex and has a highly-developed user interface and many adjustable parameters. Due to this fact, it is possible to get many different and interesting text outputs.

Key-Words: Ascii Art, Artwork, Computer Graphics, Image, Picture, Programming, Algorithm

1 Introduction

Ascii Art is composed of two words - Ascii and Art. It is a kind of art which is expressed through the letters of alphabet. The text became popular visual medium in which emphasis is put on its ability to make visualization instead of sense of letters as an information resource. Restricted set of letters or symbols is used to transform proposed objects. The history of Ascii Art dates back to the end of 19 century. French poet and surrealist Guillaume Apollinaire (1880-1918) used the shapes of text as visual information in his poems (Fig.1). The technical devices, which enabled the drawing with letters were invented later; for example, typewriter [1] and especially information technology.

Fig. 1 The poem by Apollinaire

The simple text used to be the most widely used form for visual outputs until the second part of nineteen eighties. At that time, computers had much less computing performance than current computer has; thus, the text processing and outputs were advantageous, because it did not need as much computation efficiency as the graphical output did. The data transfer of text composition is faster than the transfer of common graphical output and it has got also much less memory complexity than the latter has. Graphics became frequently used for the console operating systems where it was used for many operations, e.g. representation of tables or diagrams.

2 The present Ascii Art

Text image outputs are not already utilized so much because of the performance and transfer rate of current computers. However, it is possible to meet them in non-graphical computer terminal or in source codes, where it is used to show the logo of producer or product.

Currently, there is much software which involves the transfer from raster graphics to character sequence; for example, ASCII Art Maker (for raster graphics) [2], ASCII Art Studio (tool for text editing, drawing vector graphics) [3] and ASCII Art Generator (for easy on-line transfer in a web browser) [4].

We attempted to develop own software, which would perform the transfer too, because we wanted to try a variety of transform algorithms with many adjustable parameters. Conception of our software permits us to improve its current functions or extend it by the new ones, e.g. raster GIF animations.

3 Transfer of Algorithms

Before recognizing process, the image has to be separated to smaller parts (samples), which are passed on recognize algorithm. The detection of individual characters is separately performed
for each sample; thus, it is not influenced by previous sample. Among the main demands placing on the creation belongs rate conversation between generate time and quality of achieved results. The efficiency depends on the image size alias its resolution. Further, it depends on maintaining of the same quality through the time-consuming of used algorithm.

3.1 Gradient method

This method transfers a raster image into ASCII graphic. Before characters recognition, the image had to separate into smaller sections (samples).

The detection of the individual characters was processed for each sample separately. Thus, it was not influenced by the previous output. The algorithm used a "gray gradient". This gradient of gray is defined as all the characters are sorted by its dependence on brightness.

This means that the black pixels of sample were counted together. These results were compared with the same character of occupancy black pixels. The procedure was allowed to be similar to counting gray-scale.

In the fact, the principle of the method determined the number of points in the foreground. The algorithm passed through the sample. If it found a foreground character, the counter had incremented by one.

This method was very fast in terms of time-consuming algorithm, but did not evaluate both shape and position of the letter. The effectiveness depends on the size of the image because of its resolution.

3.2 Ratio method

Each point of the area is compared with set of characters placed in the template. If the pixels are identical, the value of counter will increase. Each character has its own degree of similarity with the compared sample. The algorithm looks for maximum agreement. Further, the method is extended by our recognizing algorithm which is called percentage algorithm. Processing is highly time-consuming, because mathematical operations are performed pixel by pixel; furthermore, it is not so accurate in less contrasting patterns.

3.3 Position comparison method

The character area is separated into individual zones. Each zone has different weight based on the numerical rating; thus, each zone contains own loop.

3.4 OCR technology

The text recognition of the image, i.e. Optical Character Recognition technologies (hereinafter called as OCR) is a special case of vectorization. It identifies different shapes in the image and compares them with pattern. In contrast with Ascii Art, OCR program cooperates with dictionary and it can also learn new patterns. Algorithm process only the part of the bitmap, where the text is aggregated. Procedures are based on neural networks, e.g. Hopfield network. OCR algorithms are used to convert scanned documents to the text, recognition of a handwritten text on the tablet or mobile device.

3.5 Comparison of methods

The results tested the change of the raster image size depending on time. The original dimensions of the loaded images were 580x580 pixels. Each method of transfer was set with different parameters. The gradient method of transfer was set with these values of parameters: the threshold was set to 30%, the output set was adjusted to all characters except an empty and a full character, and text size was reduced by half. The percentage method was set with these values of parameters: the threshold was set to 35%, the output set was adjusted to all characters with "." (dot) as an empty character, and the text size was reduced by half. The ratio method was set with these values of parameters: the threshold was set to 65%, the output set was adjusted to all characters except a blank character, and the text size was reduced by a quarter.

The results of the transfer from a raster image into ASCII graphic are given in Table 1. As can be seen, each method was compared with others depending on time. The final visual text output was compared in terms of time needed for the transfer. The gradient method was the fastest algorithm. The resulting images could not be compared to see which one of them was better or worse because the result of view is always subjective in art. The form of ASCII art output depends on the granularity of the raster. A defect could be suppressed by coloured
letters and by a unique set of parameters (brightness, contrast, threshold, etc.).

Table 1 - time of transfer

<table>
<thead>
<tr>
<th>Ratio of size [%]</th>
<th>Gradient method [s]</th>
<th>Ratio method [s]</th>
<th>Percentage method [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.016</td>
<td>0.021</td>
<td>0.023</td>
</tr>
<tr>
<td>100</td>
<td>0.031</td>
<td>0.061</td>
<td>0.065</td>
</tr>
<tr>
<td>150</td>
<td>0.078</td>
<td>0.129</td>
<td>0.118</td>
</tr>
<tr>
<td>200</td>
<td>0.140</td>
<td>0.214</td>
<td>0.176</td>
</tr>
</tbody>
</table>

Fig. 3 Gradient method: one of test pictures

4 Implementation of Software

Implementation of the proposed algorithms is performed in C++ with wxWidgets libraries. This traditional language was chosen for its speed, availability and objects features. wxWidgets allows developers to target Microsoft Windows, Linux/Unix with the GTK+ toolkit and MacOS. [5]

4.1 Class model

The internal structure layout of the program includes several sub-objects. By way of illustration, there are shown only the most important items in Fig. 4. According to mentioned diagram, the program actually contains the following class structure.

Main class (class MainFrame): This class launches and initializes the application window for Ascii Art. Further, it is inherited from both MainFrameBase class and Interface class.

Graphical User Interface (class MainFrameBase): Base class which declares every visual and control component of graphical user interface (GUI). It is created in environment of auxiliary software which is called wxFormBuilder (it is actually wxWidgets framework for GUI edit).

Operation (class Interface): The class creates instances of Image class and RDA class. Further, it contains a definition of all essential variables and methods using in the Main class; for example, measure of time, range switch of the text lists, print to status bar.

Thread (class Thread): The class is inherited from wXThread class and it overloads Entry() method for run of the thread which calls transfer methods. Further, it overloads method called OnExit().

Image and Modifications (class Image): It encapsulates all of image manipulation such as sample separation, saving files and loading files, adjustments of brightness and contrast, transfer either to grayscale or to negative, transformation of size. Raster image is stored in variable which is type of wxBitmap. After thresholding pixels, the sequence of pixel values is stored as std::vector.

Recognition algorithm (class RDA): This is the core of the whole application, which includes three recognition method mentioned above (gradient method, ratio method, percentage method). It also contains a function which is intended to sampling letter shapes of Courier New font. The function uses wxFont object.

Fig. 4 Structure of classes

5 Graphical User Interface

In addition to transferring images to text, application allows simple adjustments of image. Further, it provides many adjustable parameters of transfer. The design of graphical user interface is shown in Fig. 5.
5.1 Loading and editing of image

The first step of all operations is load of bitmap image (BMP, GIF, JPG, PCX, PNG, TGA and TIF). During the load operation, image file is converted to a separated matrix of samples with values of pixels. The mean colour is obtained from every sample. Further, it is used to changing the colour of output characters. The user can change other parameters of the input image, i.e. brightness, contrast, size, negative and grayscale. Every change of any parameter is logged into the history of changes.

5.2 The transfer

During the idle time of user, the transfer may be processed on the background. Firstly, the recognition algorithm returns a sequence of final ASCII characters to auxiliary text string. Secondly, the bitmap image is created from text string. Finally, the text output of Ascii Art is printed to the user edit field. The final output is drawn to the screen at once.

5.3 Types of outputs

As soon as the users are satisfied with their work, they may begin with editing of image or they may save the file. It is possible to save the file as a common unformatted text or as a string which is drawn into the new bitmap.

6 Conclusion

This paper evaluates the feasibility of using the transfer for generating text from the image data. The user has the possibility to customize his own image according to his ideas and convert it to Ascii Art. In general, used methods of transfers are illustrated demonstrations to show how closely the computer graphics is linked with mathematics. The findings of this study are restricted to the proportional font. The results indicate, overall, that the final visual text output is compared in terms of time needed in order to process your transfer, and especially compared with the specified original. We would like to point out that the reader has to be necessarily familiar with basic computer graphics concepts and principles of object-oriented programming, without which it would be impossible to imagine the implementation of this complicated task. On the whole, the program will be expanded in future.

References: