PortaNum : a Technical Aid for Distance Vision

J. Colineau

Paper ID ***

Abstract. PortaNum is a software-based solution for helping visually impaired people to follow presentations, lessons, or meetings. This software processes in real-time the video signal provided by a local or a network camera, and displays it in a mode compatible with the visual limitations of the user, on the laptop screen. Considerations on the user constraints as well as on algorithmic complexity and efficiency are presented. Field experiments are reported.. . .

1 Introduction

PortaNum is a software-based solution [1] aimed at helping low-sighted people in professionnal or educational situation to watch distant documents. Like a video enlarger, it consists in a video camera, processing unit to transform the image appearence, and a display unit to present the results in a way compatible with the view limitations of the user. The difference here is that these functions are based on off-the-shelf consumer products, allowing to offer a solution which is sub-optimal compared to dedicated products, but with an unbeatable cost figure. The image capture is achieved by a video camera, or a surveillance camera with a good optical zoom and autofocus. The other functions are offered by a simple laptop. In a multi-user configuration, the image acquisition is better achieved by one or several IP-cameras, while the signal diffusion is performed through WiFi links to the different users. In this case, the only equipment needed by the visually impaired user is the laptop that he or she uses already the most often on a daily basis. The PortaNum software, achieving the image capture, transformation and display, is then the only specific piece to install.

2 User requirements

As defined above, we use standard hardware equipment. The computer is a standard configuration, operated under Windows, without extreme computing power or display. The laptop will preferably be the one already used by the person. In the case of a new acquisition, the equipment choice will be dictated mainly by the robustness, autonomy, and the usability by a low sighted person. Whilst sophisticated equipment allows to acquire images in excellent conditions, we will rather use consumer sources, and compensate for the lack of performance by processing.

The software design will take into account the need of processing efficiency, and discard complex algorithms, which would be uncompatible with common computer configurations. This aspect will be discussed later.

The software ergonomy will be suited to impaired user : compatibility with Windows accessibility parameters, extensive use of shortcuts, audio feedback. Moreover, the tuning of image processing algorithms will be extremely reduced, if not fully automatic. The presentation, text size, colours, will be parametrable.

3 Capture

A large number of video capture devices exist, and most of them are accessible at low cost. From the very cheap but quality-limited webcams to the high resolution DV-Camcorders, through specialised high resolution digital cameras, or the motorised IP cameras, the performances as well as the use constraints are very different.

For stand-alone solutions, the main parameter is the viewing angle. Assuming a good quality optics, the resolution, limited by the pixel size, is :

$$x = p * d/f, \tag{1}$$

and the field of view is given by :

$$w = n_w * p * d/f, \tag{2}$$

For a webcam, with a fixed focal length f optics, and a pixel count of n_w in image width, once the minimum detail size x is settled, the distance d as well as the board size w are fixed. This means that this solution cannot be used in situations where the distance to the board can change, and the details dimensions (letter size, ...) can vary. On the contrary, a zoom optics allows to change -often in a large range : - to 10 times- the focal length f. This allows to accomodate easily various situations, including a range of user distances, board width, character or line size. Digital zoom associated with fixed focal length optics and high resolution sensor is an intermediate solution, as the maximal user distance is still given by equ (1), and the minimal depends on the zoomed image resolution. In the best case, one can achieve at most a zoom ratio of 2 to 2.5.

This explains also why a fied camera installation is the most suitable, as the distance, optics view angle, and focus adjustment are settled once. In the case of a large board, a solution can be to install two or more cameras to cover the complete field.

4 Processing

4.1 Purpose

The purpose of image processing in our application is not to deliver high quality images from limited capture devices. It is to facilitate to the low sighted people access to the information represented in the picture. In some cases (printed documents for instance) this means restore an image of the same quality as the original. In other cases, like on hand written on blackboards, this means to enhance contrast. In many cases the raw content is ”‘bimodal”’ information (for instance white written words on a green background), and the tempation is to deliver a bilevel image thanks to preprocessing followed by segmentation (or thresholding). It is not necessarily the best strategy. The user is low sighted, but he or she is able, at least as well as a computer program, to separate written words from background, or to guess badly formed letters. We think that it is important to keep as much relevant information as possible, and to only alleviate the task of trying to perceive non-relevant details (like dots, or traces from badly swept board). At least, one should allow the user to choose between deeply processed images, and only enhanced and swept ones.

4.2 Image types

In the case of distance viewing, one can consider different image contents and situations: the first is handwritten boards (white, black, green). The second case is projected material (video-projector presentations, retro-projectors), whose content can be much more varied : printed texts, graphs, images, and even animations or videos.

The case of proximity viewing is represented by text and images from books or newspapers, as well as handwritten letters.

These contents are generally colour images, and the preservation of colour (or, at least colour changes) is necessary as an element of information.

4.3 Constraints

The main constraint is the processing time. It is necessary to achieve a near real-time processing chain, including capture, image processing and image display. Fortunately, it is not necessary to look for a full fps rate, as (apart from a very small number of cases) the movement restitution is not relevant. What is important is to deliver, with as little latency as possible, the changes on the board.

In the specific case of use as a low-cost video-enlarger, which is not the design purpose of our system, the frame rate and lag become major requirements, as the user needs to be able to look through a large document while controlling his search at the screen.

In both cases, we will then pay attention to the algorithm complexity and choose implementations efficient enough in terms of calculation time, even at the cost of approximations or sub-optimality.

4.4 Processing categories

The simplest type of operation is a transformation applied at the pixel level independently of the spatial environment. This is the case of histogram transformations: these operations are very efficient as, once the image histogram is calculated, and the transformation law deduced, the operation can be realised from a look-up table technique, transforming complicated operations into reading a value in a table and replacing it for each image pixel. This applies to every brightness and contrast manipulations, even with complicated laws like gamma correction, log compression, or histogram equalization.

A second type of operation is convolution-type filtering. This means that for each pixel m operations will be performed, m being the size of the convolution filter. A typical use of these operations is sharpness enhancement. In the case of large size filters (used in low frequency processing), one should prefer the recursive implementations, allowing to reduce the processing to a very small number of operations. However, in the case of images, one should use bidirectionnal or zigzag processing. This has been used in algorithms used to separate background (supposed to be represented by low frequencies in the image) from writing.

A third level of complexity is the case of recursive algorithms at the image level. These algorithms often lead to excellent results in difficult tasks, but cannot be used as is in our case, because of the large amount of calculations for each of the incoming images. Considering that, in most cases, the image content does not vary so much from image to image, one can apply these algorithms on several incoming images, this means, perform one step of recursion on each of the successive images. This allow, in some cases an efficient implementation of calculation demanding algorithms.

4.5 Automatization

All the algorithms used are basically automatic. They do not need any adjustment on a parameter. Most of them have one degree of adjustment for perfect results, but the default value is chosen to give an acceptable result in a vast majority of cases.

5 Algorithm examples

5.1 Automatic processing

This is the preferred, and also the simplest processing. It achieves optimal contrast adjustment. The default processing is histogram stretching. The algorithm computes the image histogram, discards 5 percent of the highest and lowest values, calculates a LUT which optimises the image dynamic range. The LUT applied to the image performs the suited contrast stretching. This simple processing is the one chosen in a vast majority of cases, as this is the one which guarantees the best presentation of the image content to the user. A proposed alternative is histogram equalization, but in most cases the result is not satisfactory, as the images are often bimodal (text on a background), and there is no reason to equalize the histogram of such images.

Another variant is called ”‘dark scene”’ processing. This is basically the same algorithm, but the linear correspondance law is replaced by a gamma law, enhancing the dark areas of the scene, and compressing the white ones. This is also a fast algorithm, as the non-linear law is pre-calculated.

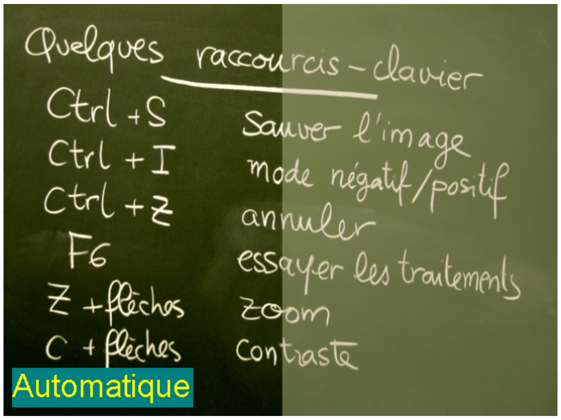


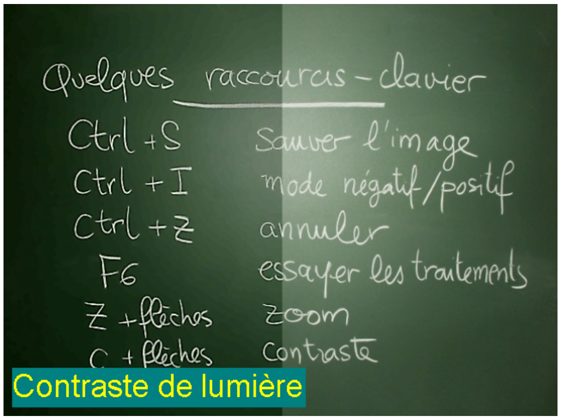
Fig. 1. automatic algorithm (the processed image is on the left side, the native one on the right side

5.2 Non-uniform illumination

In many cases, the scene illumination is not uniform (blackboards, projectors ...). A well-known method for encompassing this is the homomorphic filter. Basically, it consists in considering that the scene illuminance is the product of the incident illumination and the object reflectivity. Taking the logs makes these two contributions additive. One can then separate them by linear filtering. This is generally done in considering that the spatial low frequency content is related to the illumination non-uniformity, whilst the high frequencies are the object details. The illumination signal is attenuated, or even replaced by a constant, while the image content is enhanced. In our case, the image signal is separated into the luminance and chrominance channels. The luminance channel is filtered and enhanced by a non-linear ”‘coring”’ filter, and the two chrominance channel are processed by median filters, before being re-assembled. Alternative algorithms have been implemented, which have higher efficiency, but at the cost of larger artefacts in the image. One is the pure homomorphic filter, implemented with a recursive low-pass filter, in order to have a very fast execution time. The recursive filter works in zig-zag mode, in order to have a more symetrical response. The other is the Contrast Local Adaptive Histogram Equalization algorithm, which is very effective, but also more adapted to natural scenes that to bimodal images. The execution time has been reduced in executing only one step of recursion for each incoming image, and keeping the result as initial value for the next one. This is well adapted to video processing where successive images are generally very much correlated.

5.3 Printed text

This is a simple thresholding on the image. The low frequency content is removed by a bidirectionnal recursive filter. The threshold is settled automatically, a little

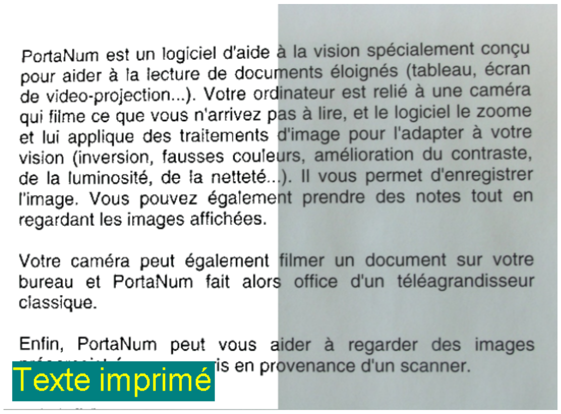


Quelques raccourcis-clavier

Ctrl + S	Sauver l'image
Ctrl + I	mode négatif/positif
Ctrl + Z	annuler
F6	essayer les traitements
Z + flèches	zoom
C + flèches	contraste

Fig. 2. illumination non-uniformity

above (or below if the background is clear) the mean value of the image pixels. The background nature of the image (clear or dark) is detected by comparison between the mean and the median values of the pixels. All these operations are performed on the image histogram, for a faster execution time.



PortaNum est un logiciel d'aide à la vision spécialement conçu pour aider à la lecture de documents éloignés (tableau, écran de video-projection...). Votre ordinateur est relié à une caméra qui filme ce que vous n'arrivez pas à lire, et le logiciel le zoome et lui applique des traitements d'image pour l'adapter à votre vision (inversion, fausses couleurs, amélioration du contraste, de la luminosité, de la netteté...). Il vous permet d'enregistrer l'image. Vous pouvez également prendre des notes tout en regardant les images affichées.

Votre caméra peut également filmer un document sur votre bureau et PortaNum fait alors office d'un téléagrandisseur classique.

Enfin, PortaNum peut vous aider à regarder des images is en provenance d'un scanner.

Fig. 3. printed text

In the case of manual written text, a dilatation (if the background is dark) or an erosion (in the other case) allows to enlarge lines, for a better legibility.

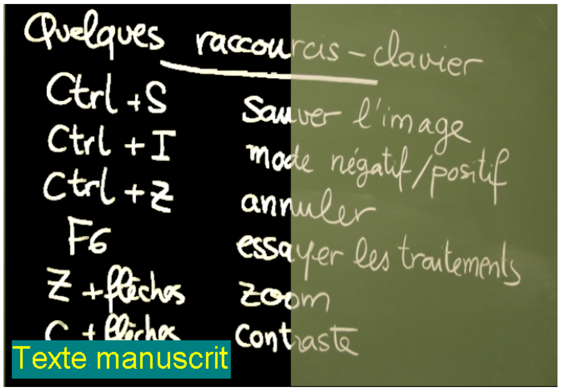


Fig. 4. written text

6 Implementation details

6.1 CV libraries

The simple algorithms have been written from scratch in C language, with the aim to optimize the complexity. For the other, we used the Intel IPP library, a low level image functions library, which offers optimized implementations of base routines. The optimisation comes from the fact that the library recognizes the processor type and architecture at the execution stage, and executes the suited code accordingly. For some algorithms, we also used the OpenCV library [2], which offers high level image processing functions.

6.2 Plugin system

In order to extend easily the palette of processing functions, a plugin mechanism allows to add new dynamic linked libraries. They are incorporated to the software menus, and accessible like native functions, provided they offer legacy entry points and functions.

6.3 Speed factor evaluation

The complexity of an algorithm can be evaluated through the execution time. However, in order to be relatively independant of the image size and the processor clock frequency, one can preferably express this speed factor in terms of clock cycles per image pixel. The execution time of a processing function is given by :

$$t = C * w * h * (1/f_h), \tag{3}$$

where

C is the speed factor, in clock cycles per image pixel (c/p)

w and h are the image width and height

f_h is the processor clock frequency

This allows to evaluate the compatibility of an algorithm with the real-time requirement. For instance, the Portanum ”‘automatic”’ algorithm shows a speed factor of 36 c/p.

A camcorder image (768 x 576) is processed on a battery powered laptop operated under a low consumption profile (clock frequency 0.8 GHz) in $t = 36 * 768 * 576/0.8e9 = 20ms$.

Practically, the speed factor is not so easy to evaluate, and not completely independent of the image size : even for a systematic algorithm, with the same operations on each pixel, there is an overhead in execution, independent of the pixel count. In recursive algorithms, the execution time is often widely dependent of the image content. Finally, the code takes into account the processor architecture and its ability to execute simultaneous operations on data, or prefetch efficiently. Finally, the compilation options, and the optimization choices play a role in the result. Nevertheless, this measure allows comparisons between different algorithms, or implementations, and is sufficient to keep or discard processing functions for this application.

6.4 Speed performances

The speed factor has been evaluated on several computers, with different processor types and clock frequencies: - Intel Pentium 4 at 3000 MHz - AMD Sempron at 2000 MHz - Intel Pentium M at 1600 MHz - Intel Core 2 Duo at 1800 MHz

Table 1. algorithm speed factor (in clock cycles per image pixel)

Processor	Intel	AMD	Intel	Intel
	Pentium 4	Sempron	Pentium M	Core 2 Duo
automatic	61	67	36	30
dark scene	61	67	36	30
light contrast	23	40	7	8
printed text	74	80	43	40
hand written text	95	128	57	47
contour	320	330	190	350
poster	280	220	128	99
PDE restoration	37000	30430	21700	17100

One can see that the processor architecture plays an important role. But, for a given processor, the algorithm speed factor, and thus the execution time varies on a much larger scale, leading to discard some of them.

The five first treatments are the base functions of the algorithm. One will see that they are have low complexity, allowing real-time processing of video signals.

The other have increased complexity, in particular the recursive restoration algorithm whose implementation is too complex for the application.

6.5 Complexity and autonomy

The algorithm computation complexity has another consequence: it increases the power consumption, which leads to lower operation time in battery mode. Although the dependance is not linear, and is function of the processor and board architectures, the effect may be non negligible, as shown in the following table, and is a concern in a laptop solution. In table 2, the power consumption is the part due to the image processing algorithm itself, excluding the other functions of the program. This measure was done on a laptop equipped with a Pentium M processor, in battery mode. The camera was a webcam, and the image size 640x480. The two first columns correspond to the usual preference choices in battery mode. The last is done in maximum performance mode, but with a reduced image rate.

Table 2. power consumption on the battery for the different algorithms (in mW @ im/s)

battery mode	max performance	max autonomy	economy mode
automatic	2200 @ 22	750 @ 20	250 @ 5
dark scene	3700 @ 21	830 @ 21	600 @ 5
light contrast	4400 @ 22	1200 @ 18	1150 @ 5
printed text	2200 @ 22	1200 @ 20	450 @ 5
hand written text	2200 @ 22	1100 @ 20	400 @ 5
contour	11700 @ 19	1800 @ 9	750 @ 5

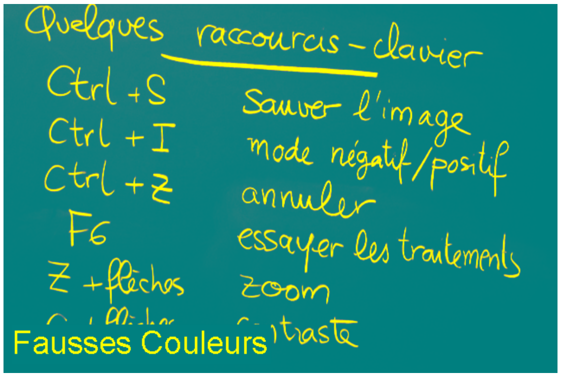
One can see that the relation between the speed and consumption tables is not evident, as a fast execution may be achieved with a large number of gates commuting simultaneously (for instance use of the math coprocessor), while a long execution time can be done with a small power consumption. Moreover, in a processor like the Pentium M, parts of the chip can shut down if not used, even during a single clock cycle. In any case, the power consumption can be reduced by decreasing the frame rate. We have introduced for this reason the ”‘economy mode”’ of the program, which decreases the image rate to 5 im/s, allowing much lower power consumption when autonomy is needed.

7 Visualization

7.1 Modes

The image is presented either in a resizeable window or in full screen. In both cases, in order to get the best quality, the image is interpolated by the software. This time consuming operation can be chosen between three modes: neigborough, linear, cubic interpolation. the preferred mode is linear, which is a good compromise between quality and execution time.

The image representation can be chosen between several modes, as usual for such visual aids: natural colours, false colours, luminance inversion.



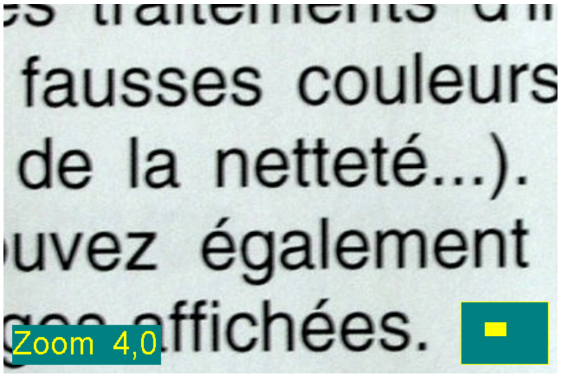
Quelques raccourcis-clavier

Ctrl + S	Sauver l'image
Ctrl + I	mode négatif/positif
Ctrl + Z	annuler
F6	essayer les traitements
Z + flèches	zoom

Fig. 5. false colours

7.2 Zoom and navigation

A visual aid is provided in order to allow the user to figure the presented part on the image in the whole document. The navigation is performed through the use of the mouse or directional keys. Representing a small portion of a large document to low-sighted users remains an ergonomy challenge. We have favoured the interface responsivity, and represented the situation of the viewed part in the whole document.



fausses couleurs
de la netteté...).
uvez également
affichées.

Fig. 6. zoom and navigation help

7.3 Performances

On a laptop with a Pentium M and 1.6 GHz clock frequency, all the base processings are applied in real time and the achieved frame rate is 20 im/s, with a full screen display.

8 Network

In despite of the generalization of WiFi for home and public sites, and the use of this means for videos and images transmission over-the air, the transmission of real-time video through wifi is not so easy up-to now [3]. In fact, two constraints add to make this task difficult: the first is the necessity to compress and transmit the images without lag, as the user is present in the conference room, and it is not possible to use buffering techniques to regulate the received streams. The second is that in most of cases, the image will be processed (contrast enhancement, contour extraction or underline, ...), and this added to the nature of images (written or printed text, charts ...) requests a good transmitted material quality.

8.1 Solutions

Several video streaming solutions are available, including open source solutions [4]. The chosen framework is a streaming video solution ([5]) optimised for real-time compression and transmission. The compression schemes which give good efficiency and low artefacts in presence of non-natural images are MPEG4 and WMV, based on wavelet decomposition. With datarates of 400 kb/s, images of size 640x480 can be transmitted at a frame rate of 3 im/s, which is enough for the application. A choice had to be done between unicast and multicast schemes. In the second case, only one flux is transmitted, which is large bandwith reduction in the case of many simultaneous users. Practically, multicast diffusion over WiFi networks is rather difficult because of two main limitations. The first is the low bandwith available, as the wifi network uses the lowest data rate (1.5 Mb/s, even with 802.11b 11Mb/s network). The second is the high channel error rate (typically 1e-3 to 1e-4) which makes difficult the error free transmission of the images. To overcome these limitations, we developed a first version using multicast diffusion, and error protection with a (223,255) Reed-Solomon burst error correcting code, which offered satisfactory performance over a 802.11b network. The present solution is based on 802.11g, and for a moderate number of users (i.e. up to ten), unicast transmission is more flexible as it does not request special protocols or error correction schemes.

8.2 Performances

The following table summarizes the performance achieved and degradations in function of the number of clients.

Table 3. performance of the video diffusion

number of clients	2	4	6	8
server throughput (kB/s)	120	240	360	480
missing images (%)	0.7	2.0	4.6	5.0
arrival time jitter (ms rms)	30	53	86	98

9 Field experimentation

The Portanum solution has been extensively tested and used in different work situations [6], including office meetings or presentations, scientific workshops or conferences, and benefits from several years of field experience. In the education field, the solution has been installed in a primary school for visually impaired children in a network configuration. In high schools, stand-alone and network installations have been used, allowing to gather interesting feedback on the software functions and ergonomy. Downloading and use of this software is free of charge [1].

Presently, several installations are being done in university conference and classrooms. In each case, the user number is 5 to 10 students, and the installation is a wifi network configuration, with 2 or 3 cameras in order to catch the whole board. If necessary, electronic documents projected on video projectors are also retransmitted on the network. One of the biggest challenges to overcome is still the autonomy of the user laptop, which must work on a day basis. This is, depending on the hardware, and particularly the processor architecture, very much related to the algorithm complexity, and the image refresh rate, as each processing operation directly costs in energy on the battery.

10 Conclusion

The Portanum system is essentially an attempt to ease the access to distance visual information, thanks to non-dedicated equipment. Based on ordinary video equipment, preferably pre-installed, and accessible through WiFi network, it consists in an efficient image pre-processor and a viewer part adapted to the low-sighted user needs. We have explained how to use the algorithm approach to facilitate the user perception, without going too far in the complexity, in order to take into account the real-time constraints as well as the system autonomy requirements.

References

1. : Portanum. (http://www.portanum.com)
2. : Opencv. (http://opencvlibrary.sourceforge.net)
3. Majumdar, A., Sachs, D., Kozintsev, I., Ramchandran, K., Yeung, M.: Multicast and unicast real-time video streaming over wireless lans. Circuits and Systems for Video Technology, IEEE Transactions on **12** (2002) 524–534
4. : Videolan vlc. (http://www.videolan.org)
5. : Unreal media server. (http://www.umediaserver.net)
6. Douard, M., Colineau, J., Goasduff, L.: Portanum: a reading aid for a classroom blackboard. AICPS, Proceedings of the 1st French-speaking conference on Mobility and ubiquity computing **64** (2004) 25–28