A GENERALIZED LAPLACIAN PYRAMID AIMED AT IMAGE COMPRESSION

Costin-Anton Boiangiu [1*]
Marius-Vlad Cotofana [2]
Alexandru Naiman [3]
Cristian Lambru [4]

ABSTRACT

Image resampling is a process that involves a trade-off, between efficiency, smoothness and sharpness. Nevertheless, it is a process that we encounter frequently in our digital world: on the internet, for printing and scanning, in photography and television and even in astronomy, when observing distant galaxies.
Image resampling is a mathematical technique used to create a new image, from the original one, which has a different width and/or height in pixels. Reducing the size of an image is called downsampling, while increasing its size is called upsampling.
Upsampling an image, increases the number of pixels, but details that were not in the original one, cannot be created. As a result, images become blurry the more they get enlarged. When images are downsampled, information in the original images has to be discarded. This makes the new image appear sharper, even though it contains less information. As a downside, if an image is downsampled and upsampled, some of the original details will not get recovered.

KEYWORDS: *image resampling, image compression, Laplacian pyramid, Gaussian pyramid, zig-zag ordering, raster ordering, Morton codes, bzip2, deflate, compression algorithms, information theory, nearest neighbor interpolation, Lanczos filter*.

INTRODUCTION

Humans are surrounded by more and more displays: at home, in their pockets, on the streets, at the office, displays that use images to communicate relevant information to them. Digital imagery is changing the world and to benefit from this technology we must always find new ways to improve the processes on which it relies onto.

The need for image compression. A pyramidal approach.

Image compression is the process of reducing the size of an image file, by decreasing irrelevance and redundancy of data, so it can be transmitted in a more efficient manner.

[1*] corresponding author, Professor PhD Eng., ”Politehnica” University of Bucharest, 060042 Bucharest, Romania, costin.boiangiu@cs.pub.ro
[2] Engineer, ”Politehnica” University of Bucharest, 060042 Bucharest, Romania, marius.coto@gmail.com
[3] Engineer, ”Politehnica” University of Bucharest, 060042 Bucharest, Romania, alexandru.naiman@gmail.com
[4] Engineer, ”Politehnica” University of Bucharest, 060042 Bucharest, Romania, cristi.lambru16@gmail.com

There are two types of image compression algorithms: lossy and lossless. **Lossless** compression is preferred for critical images, like medical imaging, technical drawings or file archival, while **lossy** compression is suitable for applications where minor loss is acceptable, in natural photographs, in order to achieve a substantial reduction in file size.

A pyramidal image compression allows for a sequential resampling of an image [2]. For each level of the generator process, a new image is created, that is smaller in size, but incorporates less of the details in the original image. Using this array of images and based on the acceptable distortion limits imposed by the sensitivity of the human eye, one can easily pick the images that offer the best balance between file size and image quality. The Laplacian pyramid encoding scheme requires relatively simple computations which are mostly local and easy to process in parallel.

Resampling color images is done in a similar way as grayscale ones, each primary color being "treated like three black and white images which are separated from the original image, individually resampled, and finally recombined to create the final image" [3].

Related work

Peter J. Burt and Edward H. Adelson were the first ones to experiment with the Laplacian Pyramid for digital image compression, in their paper entitled: "The Laplacian Pyramid as a Compact Image Code" [1], published in 1983. As they stated: "The Laplacian pyramid is a versatile data structure with many attractive features for image processing" and together with the Gaussian pyramid can successfully be used for progressive image transmissions [2][7][8].

At that time though, image compression and transmission were not important topics of research, so their work did not attract too much attention. In our times, the majority of the people have a TV at home, a desktop computer or a mobile device on which they spend a good part of their daily time. Each one of these devices uses images to transmit information. A big part of the internet is composed of images that get streamed from a server to each client's web browser, and the list of examples can keep on going. Image and video compression is an important element in today's digital processes and it should not be neglected [5][6].

THE CLASSICAL APPROACH: LAPLACIAN PYRAMID

Resampling is no easy task. Although there are a lot of performant algorithms that can be used in the process, finding the best one requires thorough analysis of the image and many trials and comparisons between different resampling methods.

Most resampling techniques work by computing new pixels as a weighted average of the surrounding ones. The weights usually depend on the distance between the analyzed pixels, which can range from just the immediate ones to more advanced methods which examine more of the surrounding pixels, in attempt to produce a better result.

The tools that can be used for such a process are variate, many of the existing programming languages implement at least one image processing library. OpenCV is the most popular one, with multi-OS support and interfaces available in C, C++, Java and Python.

The Laplacian pyramid is built using subsequent images that are weighted and rescaled, saving, on each level, the difference between the downscaled image and the one resulted after the upscale process. Each pixel in the new image represents a local average of the corresponding pixel neighborhood on a lower level of the pyramid.

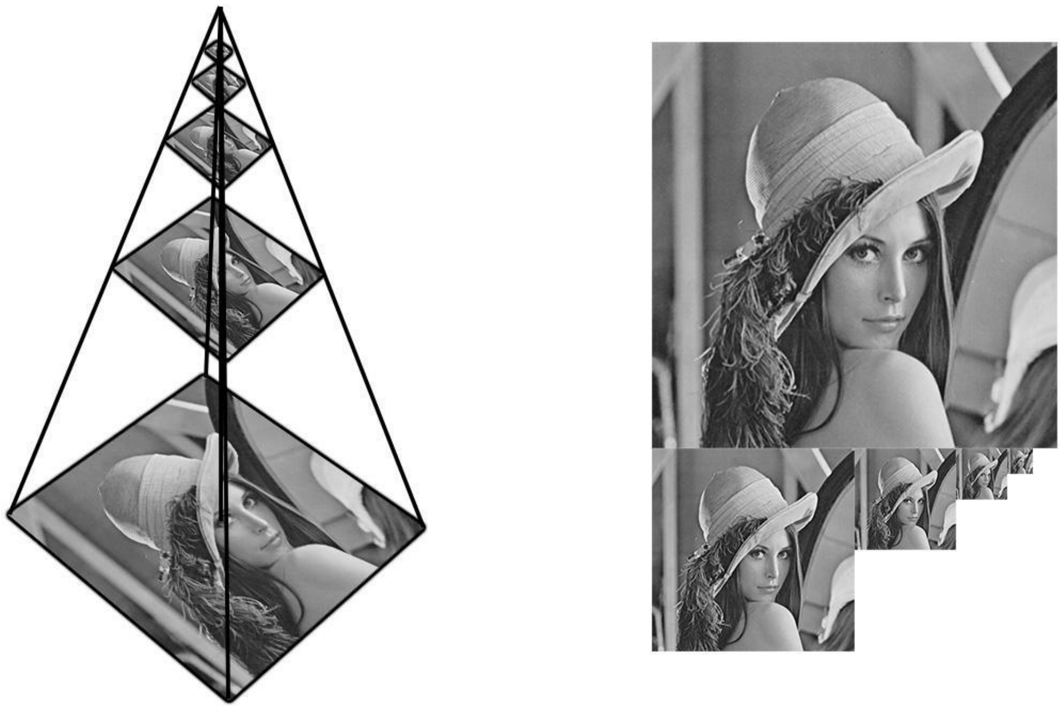

Figure 1. On the left, the pyramid of images and on the right, a size comparison of all levels of the pyramid

Approaches to this problem can be very different, depending on the input image, the scale of the new one, as well as the quality of the final resampled image. One easy, but effective approach uses a minimum pixel value filter for the downsample process which ensures a big compression rate, when used with Huffman coding for compression. This way, the value of the pixels in the new image can be quickly computed, from the minimum value of its neighbors and the small values resulted and it ensures an efficient compression rate. Coupled with this technique, for the upsample process, a box filter can be used, so the total processing time is kept to a minimum. This combination of filters also ensures that the values in the new image are always less than the ones in the original image, so high contrast will not be a problem.

THE PROPOSED APPROACH: GENERALIZED LAPLACIAN PYRAMID

The Laplacian Pyramid of an image is a processing step in getting images ready to be digitally transferred, a process that is very important in the times when everyone is expecting high-speed internet. The proposed generalized form of the pyramid adds a new step to the initial algorithm, by splitting the image on every level into 4 subsequent sections using a cross-cut approach and creating a “generalized pyramid”, using the same construction mechanism for each one of the newly-resulted sections.

The main idea

The main difficulty, in this concept of the “generalized pyramid”, is choosing the point where the image layers will be cross-cut. When cutting position is chosen correctly, the image size can be further reduced, thus the process will become more efficient. As a downside, the extra step of selecting the correct point for the cut will result in more computing time for the possible gains in size reduction.

The key points

Filtered resampling, choosing the best resampling filters

Filters can impact the performance of the resampling process. There are a variety of available filters that can be used, so choosing the right one should be based on the content of the original image and on the previously known statistics of the filter behavior on certain scales. A solid but ineffective practice is to test all filters that are available and choose the one that generates the best results for a particular case.

In our case, we have experimented with 3 filters: Nearest, Cubic, Lanczos, with each of them producing slightly different results, in correspondence with the type of image used and its content.

The resampling scale, choosing the best scale at each level

In most use cases, the resampling scale of the new image is half the width and height of the original one. A smaller scale will generate more intermediary images, while a bigger one can create very few images. The resampling scale should be chosen based on the estimation of the resulted granularity for the intermediary images.

The cross-cut point, “How to choose” heuristics

Choosing the cross-cut pixel is the most important aspect in the algorithm of the generalized Laplacian pyramid. It can determine the efficiency of the process and, when properly chosen, it can guarantee a “fair” processing time for a good quality result.

The biggest influence, in the image segmentation should be imposed by the color variance in the new segments. An easy and safe method of choosing the right segmentation point is to compute the variance of each segment for every possible cross-cut pixel. This action requires significant processing time, but the results can be more effective than the algorithm with no cross-cut point.

Another heuristically method for splitting the original image may be based on the edges that are visible in the image. This method is based on the fact that an edge marks a sharp change in image brightness. Setting the cross-cut pixel on an edge can help separate the two brightness values into different image segments, which in turn can lead to a better compression of the resulting images. The edge detection algorithms available today can run in linear time, so the time cost brought in by this process is not significant.

Choosing the cross-cut pixel based on the estimated size of the resulting images is a method that delivered good results. To assess an image size some may multiply the pixel entropy with the total number of pixels. Since this is a lightweight operation, and because the changes in entropy only need to be recalculated just for the pixels that are included or removed on each new section of the image that is processed, we can do this for every possible piece of the image, consuming little time on decent hardware configurations.

Intra-level scanning patterns: raster, zigzag, Morton codes, etc.

A scanning pattern defines how the pixels of an image get traversed during the image processing phase. Depending on the content of the image, crossing the pixels in a certain pattern can lead to a better compression factor, due to better grouping of similar pixels. Raster and zigzag pattern are easy to use and understand, while Morton codes require more development time to integrate, but generate better results in most of the cases.

Before the final entropy coding, there is an intermediate step that can provide further reduction in the size of the final image. Residual encoding algorithms, such as Run-Length Encoding (RLE) can use data in a more efficient manner, especially in the case of simple graphical images such as icons, line drawings or animations, by encoding repetitive sequences of data into single data and its count.

When to stop the new pyramid level generation?

The process of generating a Laplacian Pyramid could be run until the final image layer is represented by just 1 pixel. But its purpose is to generate images smaller in file size, that can still resemble the original one, and the whole process should be as fast as possible, so it can run on multiple images in a respectable amount of time. As a result, the optimal time to stop the resampling is when no further data reduction in the compression process occurs. This method can generate images with a small storage 'footprint' without sacrificing its content.

THE OBTAINED RESULTS

The file format

Storing the generated image of each level of the pyramid is done using a custom file format. To simplify things, both image data and residue data are stored in the same file, alongside their meta-data. Level-specific information, such as the image width and height, the residue width and height, as well as general information, such as, the number of levels of the pyramid and the number of channels that the images are using, are stored on each step of the process.

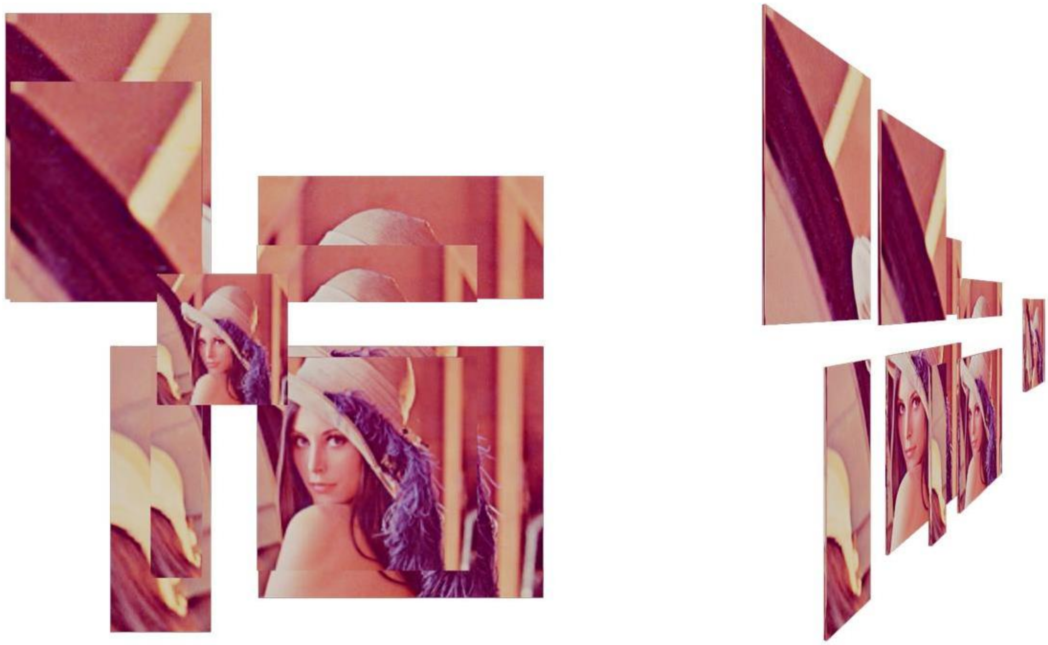

Figure 2. A render of all levels of the generalized pyramid, seen from two different angles

Minimizing the final file size is an important aspect of the transformation process. Compression transforms, such as Burrows-Wheeler used in the bzip2 format can bring down the residue storage space. In our experiments we managed to get a compression factor of 5:1 by using it.

Performance table, discussion

The tests that we have conducted highlight the two compression algorithms that were used: BZ2 and DEFLATE, which combines Huffman algorithm with LZ77 compression. The image used for the tests was the popular "Lenna" portrait, in TIF format, with a resolution of 512x512 pixels and a raw size of 256Kb. For each compression algorithm, 3 types of filters were tested: Nearest, Cubic and Lanczos for both downsampling and upsampling [4].

The results will be analyzed by monitoring the size of the final image (the ‘Compression’ column), the compression factor, which represents the ratio of the final size from the original one, as well as efficiency of each variant that we tested.

In the case of our test image, the BZ2 variant generated the best results, with a compression factor reaching 1.3 and an efficiency level of 0.2 for any of the 3 filters used. The DEFLATE compression algorithm, in comparison, had a significantly worse compression rate, generating negative efficiency, for all 3 filters.

In the case of the Generalized Laplacian pyramid, improvements are easily visible (while testing only the BZ2 compression method): the compression factor is better than the best run of the simple Laplacian pyramid process, while efficiency reaches a value of 0.3. The results of the generalized process can be observed in the table below.

Table 1. Performance of the image compression scheme using the classical Laplacian Pyramid

Compression algorithm	Filter	Number of levels	Compression (bytes)	Compression factor	Efficiency
BZ2	Nearest	2	196383	1.336617732	0.25184
	Cubic	3	207454	1.265287726	0.20966
	Lanczos	3	200918	1.306448402	0.23456
DEFLATE	Nearest	2	250032	1.049821623	0.04745
	Cubic	3	241897	1.085127141	0.07844
	Lanczos	3	248386	1.056778562	0.05372

Table 2. Performance of the image compression scheme using the proposed Generalised Laplacian Pyramid

Compression algorithm	Filter	Number of levels	Compression (bytes)	Compression factor	Efficiency
BZ2	Nearest	2	173627	1.51179828	0.33853
	Cubic	3	179948	1.458693623	0.31445
	Lanczos	3	168740	1.555582553	0.35715

For these tests, the compression times went up, to almost 2 seconds, mainly caused by the overhead of choosing the cross-cut point, while the decompression times remain similar to previous runs. In our tests, the cross-cut point has been chosen using two different approaches: by computing variance and by computing the entropy, both of them providing similar results.

Compared to the original image, compressed with the same algorithm (BZ2) which generated a file of 174169 bytes, the process of the generalized Laplacian Pyramid, managed to reduce the file even more, generating a 168740 bytes file, in the case of using a Lanczos filter. This result translates to a compression factor of 1.5 against a direct compression of the image and an efficiency level of 0.33.

Improvements and optimizations

Used on different images, the results tend to vary, depending on the compression algorithm and resampling filters used. The former has the biggest impact on the size of the final compressed image, generating a factor of just over 0.2. The content of the image is also relevant in the compression process, icons and simple line drawings producing far better results.

An optimized compression process, using the Laplacian pyramid, would do a pre-scan of the original image and decide automatically which the best filters for that specific case are and what type of compression should be used. This improved version of the algorithm would also decide what is the optimal level when the pyramid process should stop, generating smaller images with little or no visible distortion for the human eye.

CONCLUSIONS

The research described by this paper has generated some encouraging results. Using a process that is light on computational resources, the algorithm of building the Generalized Laplacian pyramid can successfully reduce the file size of an image, while creating a range of images, with different resolutions, which allows picking the appropriate one, by comparison with other.

The potential of the presented solution

With formats as 360 degree images, that are ready for virtual reality, getting more popular, the areas where image compression has an important role, get broader and broader. This is also supported by the number of cameras that get embedded in devices around us, and the advancements in image sensors, that take great quality photos, but sacrifice proportional storage space.

The small gains in image compression that our experiments have obtained, can further be improved by testing advanced image filters and other compression algorithms. Automating the whole process of generating the image pyramid, would simplify the steps involved and could make it a feasible solution for even more use cases.

Future work

The next step of this research would be to start using it in real situations of digital image transmissions. Starting with small images in custom web applications and extending it to more popular platforms, can make the process resilient to some types of errors that are not experienced in a research environment. Adding in more options regarding filtering and compression methods, as well as automating the process through machine learning and other advanced techniques could make the Laplacian pyramid-based compression algorithm ready for the technologies of the future.

Another research direction may be toward the usage of super-resolution [9] methods instead of the classical low-pass filter-based approaches. This may introduce perhaps a bigger amplitude in the residue stream which may be compensated by a lower variance, thus leading to even better performance inside the entropy coder executed in the final processing stage.

ACKNOWLEDGEMENTS

The authors would like to thank associate professor Ion Bucur, teaching assistant Mihai Zaharescu and students Silviu Ivascu, Bianca Tudora, Ioana Stavar, Bianca Tudora, Claudiu Baronea, Antonio Gavrila, Adrian Matei, Ana Pellian, Ana Birligea, Adrian Cojocaru, for their great support and assistance with this paper.

REFERENCES

[1] Burt, Peter and Adelson, Ted, "The Laplacian Pyramid as a Compact Image Code", IEEE Transactions on Communications, Volume 31, Issue: 4, pp. 532 – 540, 1983, doi: 10.1109/TCOM.1983.1095851

[2] E.H. Andelson and C.H. Anderson and J.R. Bergen and P.J. Burt and J.M. Ogden. "Pyramid methods in image processing", 1984

[3] Jonathan Sachs, "Image resampling", 2001

[4] K. Sreedhar Reddy, Dr. K. Rama Linga Reddy, "Enlargement of Image Based Upon Interpolation Techniques", International Journal of Advanced Research in Computer and Communication Engineering, Volume 2, Issue 12, 2013

[5] Adrian Enache, Costin-Anton Boiangiu, "A Pyramidal Scheme of Residue Hypercubes for Adaptive Video Streaming", International Journal of Computers and Communications, Vol. 8, pp. 128-133, 2014

[6] Adrian Enache, Costin-Anton Boiangiu, "Adaptive Video Transmission Using Residue Octree Cubes.", Computers, Vol. 3, No. 2, pp. 58-68, 2014

[7] Emil Calofir, Radu Ionut Dan, Vlad Lionte, Ion Bucur, "Image Reconstruction after A Succession of 2:1 Downsampling – Upsampling", Journal of Information Systems & Operations Management (JISOM), the Proceedings of Journal ISOM Vol. 8 No. 2, pp. 252-261, December 2014

[8] Mihai Cristian Tănase, Mihai Zaharescu, Ion Bucur, "2:1 Upsampling-Downsampling Image Reconstruction System", Journal of Information Systems & Operations Management (JISOM), the Proceedings of Journal ISOM Vol. 7 No. 2, pp. 294-299, December 2013

[9] Florin Manaila, Costin-Anton Boiangiu, Ion Bucur – "Super Resolution From Multiple Low Resolution Images", Journal of Information Systems & Operations Management (JISOM), the Proceedings of Journal ISOM, Vol. 8 No. 2, pp. 316-322, December 2014.