INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Signal Processing Analysis applied to Image Phylogeny

A. Melloni, M. Tagliasacchi, S. Tubaro, F. O.Costa, M. Oikawa, Z. Dias, S. Goldenstein and A. Rocha

Technical Report - IC-14 -02 - Relatório Técnico

January - 2014 - Janeiro

Signal Processing Analysis applied to Image Phylogeny

Ambra Melloni[1], Marco Tagliasacchi[1], Stefano Tubaro[1], Filipe de O.Costa[2], Marina Oikawa[2], Zanoni Dias[2], Siome Goldenstein[2] and Anderson Rocha[2]

[1] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano
{ambra.melloni, marco.tagliasacchi, stefano.tubaro}@polimi.it

[2] Instituto de Computação, Universidade Estadual de Campinas
{filipe.costa, oikawa, zanoni, siome, anderson.rocha}@ic.unicamp.br

Abstact. This work is a report about the effect of denoising filters applied to image phylogeny, as conclusion of the collaboration between Politecnico di Milano and UNICAMP, inside the REWIND project. The purpose of image phylogeny is discovering dependencies among a group of images representing similar or equal contents in order to construct a tree describing image relationships. We applied a family of image processing techniques to a set of images, in order to create parental relationship. Operating in the wavelet domain, it is possible to apply denoise filters in the images, separating each image in two contributions: a component part (the denoised image) and a randomness part (the noise). We evaluate the effectiveness of this method for three different database, using subsets of the image processing functions: in this report, we show that geometric transformation imperfections influence the results of denoising algorithm highly, when reconstructing trees for component and randomness parts.

Keywords: image phylogeny tree, signal processing, denoising.

1. Introduction

The purpose of image phylogeny is discovering dependencies among a group of images representing similar or equal contents in order to construct a struct describing image relationships.

We applied a family T of image processing techniques (color adjustment, JPEG compression, and geometric transformations) to a set of images, so that parental relations are created among the images. Those relations are based on the computing of dissimilarity matrix; we have two possible ways:

- Pixel-to-pixel comparison in RGB domain space [1].
- Pixel-to-pixel comparison after signal processing analysis: operating in wavelet domain, we separate denoised part from noise part of an image [3].

We use the dissimilarity matrix as input of oriented kruskal algorithm to perform phylogeny reconstruction [1].

2. Signal processing analysis

We consider the image as the composition of two separable and independent contributes:

- The content component $[I]_C$, describing the content of the real scene.
- The randomness component $[I]_R$, representing the content-independent characteristics of the image, the peculiarities of the process that produced the image.

Separation of image parts is made possible through the wavelet-domain denoising algorithm: the denoised image corresponds to the component contribute$[I]_C$, the noise to the randomness contribute$[I]_R$.

State of the art

De Rosa et al. [2] propose to detect the image dependencies within a set of images relying on the image separable parts (component and randomness).

Image I_B can be obtained approximately by applying some of the available image processing functions to image I_A: color matching, compression matching, image registration.

The first step is the matching of the image content parts for estimating the family of image processing functions. Using those functions, it is possible to transform image I_A to image I_B , computing color matching, compression matching and image registration They compute correlation coefficient between the two images based on their content-independent parts.

Finally, they perform tree reconstruction chosen the direction and correlation value of highest response.

System architecture

The proposed method relays on the works by De Rosa et al., with some differences in the component order of the pipeline and in the evaluated transformations.

Given two images, I_A and I_B, we first perform geometric transformations, color matching and compression matching; we gathered this pipeline from Dias et al. [1]. The denoising process is the last block of our pipeline: we use both component and randomness parts' comparison to build two dissimilarity matrix. Phylogeny reconstruction is performed by oriented kruskal algorithm.

Image registration

We calculate corresponding points between images I_A and I_B using SIFT algorithm. SIFT features are used as anchors for the registration algorithm to increase its accuracy. We can robustly estimate the geometric transformation for image I_A respect to image I_B: resampling (global scaling), affine warping (rotation, translation, off-diagonal correction), cropping. Finally we just apply the estimated transformation to image I_A.

Color matching

In the second block, we transfer the color characteristics of I_B to I_A.

The first step is chrominance matching. Source and target images defined in RGB color space: in this domain, we have channel strong correlation. We need to migrate to the non-correlated $l\alpha\beta$ color space [5]:

- l achromatic channel
- α chromatic yellow–blue opponent channel
- β chromatic red–green opponent channel

We normalize image I_A according to the I_B chromatic channel mean and standard deviation.

We have additional linear and non-linear operations:

- We perform gamma correction in $l\alpha\beta$ color space, estimating the correction in the achromatic channel l. We apply blind gamma estimation [6]: gamma correction introduces specific higher-order correlations in the frequency domain. These correlations can be detected using tools from polyspectral analysis. The amount of gamma correction is then estimated by minimizing these correlations.

- Brightness adjustment: we calculate the mean of each I_B RGB color channel and normalize image I_A color channels using such measure.
- Contrast adjustment is evaluated through histogram matching. For each RGB channel, we calculate the 256-bin histogram and its cumulative distribution function (CDF). Histogram matching can be done through a look-up: a table maps between the pixel value of the source image and the pixel value of the target image that give the equivalent CDF values.

Compression matching

We apply our method only on images stored in JPEG format. We need first to recover I_B quantization table qf_B. Then, we can compresses image I_A according to I_B quantization table.

Denoising

We separate the two image contributes, content component $[I]_C$ and randomness component $[I]_R$ through the wavelet-domain denoising algorithm: the denoised image is $[I]_C$, while the noise is $[I]_R$.

The high-frequency wavelet coefficients of the noisy image are modeled as an additive mixture of a locally stationary i.i.d. signal with zero mean (the noise-free image) and a stationary white Gaussian noise $N(0, \sigma_0^2)$ (the noise component).

We compute five-level wavelet decomposition of the image with 8-tap Daubechies QMF [4]. We obtain vertical, horizontal, and diagonal subbands as $v(i,j)$, $h(i,j)$, $d(i,j)$, where (i,j) runs through an index set J that depends on the decomposition level. E.g., we consider the vertical subband $h(i,j)$.

1) We estimate local variance of noise-free image for each wavelet coefficient. Consider a square neighbourhood N

$$\sigma_W^2(i,j) = \max\left(0, \frac{1}{W^2} \sum_{(i,j)\in N} h^2(i,j) - \sigma_0^2\right), \qquad (i,j) \in J, W \in \{3,5,7\}$$

2) We take the minimum of the three variances as the final estimate:

$$\hat{\sigma}^2(i,j) = \min\left(\sigma_3^2(i,j), \sigma_5^2(i,j), \sigma_7^2(i,j)\right), \qquad (i,j) \in J$$

3) We apply Wiener filter to calculate denoised wavelet coefficients

$$h_{den}(i,j) = h(i,j)\,\frac{\hat{\sigma}^2(i,j)}{\hat{\sigma}^2(i,j) + \sigma_0^2}, \qquad (i,j) \in J$$

4) We repeat Steps 1)–3) for each level and each subband.

Finally, we obtain the denoised image applying inverse wavelet transform to the denoised wavelet coefficients. We subtract the denoised image to the original noise-free image to obtain noise part.

3. Evaluation parameters

At the end of the pipeline, we obtain the final transformed image $\tilde{I}_A \leftrightarrow [[\tilde{I}_A]_C, [\tilde{I}_A]_R]$, given the target image $I_B \leftrightarrow [[I_B]_C, [I_B]_R]$.

We calculate the point-wise dissimilarity between component and randomness parts of both images:

$\text{dissimilarity}([I_B]_C, [\tilde{I}_A]_C)$

$\text{dissimilarity}([I_B]_R, [\tilde{I}_A]_R)$

We can evaluate two different parameters to compute the dissimilarity matrix. If we consider the correlation ρ between the two images, we obtain the dissimilarity matrix as $(1 - \rho)$.

4. Results

We consider three case studies:

- De Rosa *et al.*: set of 10 RGB images. We have two independent natural images as roots, 2 independent trees of 5 images each one. This case study has a limited transformation set: histogram stretch, scaling and rotation, JPEG compression.
- Controlled dataset with geometric transformations (dataset A): set of 90 RGB images: one independent natural image as root used in 9 tree topologies, 10 nodes each one.
- Controlled dataset without geometric transformations (dataset B): set of 250000 RGB images: 50 root images, for each one 10 fixed tree topologies with 10 transformation variations and 50 nodes.

We compare the results from the algorithm by De Rosa *et al.* and the one from the method by Dias *et al.* (pixel-to-pixel comparison in RGB domain space), applied to the controlled datasets, with the results using our denoising pipeline. For performing the tree reconstruction, we use Oriented Kruskal algorithm.

We operate in a controlled environment with complete trees and JPEG compressed images.

Reconstructed trees

For the case study by De Rosa *et al.*, our method performs better results, achieving 100% in half of the evaluated parameters (Table 1).

Our denoised pipeline produce worst results comparing to Dias *et al.* method on dataset A, achieving 10% less performance (Table II).

If we operate on dataset B, without hard geometric transformations (only global scaling), our method outcomes increase (Table III). We perform scaling just comparing image size, without calculating SIFT correspondences.

High frequency filter used for denoising algorithm performs high-resolution images as output: if there are even minimal misalignments after geometric transformation, when comparing pixel-to-pixel denoised images we will have negative results. Images need to be perfect aligned, but rotation and cropping introduce errors. Instead, the dataset from De Rosa *et al.* and dataset B introduce very small geometric transformations: in this case, our method performs better results.

We also inspect Fourier analysis applied to image noise component $[I]_R$: we calculate energy and entropy of DFT spectrum for both magnitude and phase. Values of energy and entropy are very similar for all the images: the pattern noise due to the camera is the same (one-camera images); image randomness parts are highly related because output of image transformation processing.

TABLE I: De Rosa *et al.*

De Rosa *et al.*

	Root	Edges	Leaves	Ancestry
Content	50%	77,8%	100%	66,7%

Denoising

	Root	Edges	Leaves	Ancestry
Component	100%	100%	100%	100%
Randomness	100%	87,50%	100%	72,20%

TABLE II: Dataset A

Dias *et al.*

	Root	Edges	Leaves	Ancestry
Content	100%	91,37%	84,22%	94,56%

Denoising

	Root	Edges	Leaves	Ancestry
Component	88,89%	79,52%	76,86%	84,69%
Randomness	100%	72,87%	74,07%	84,91%

TABLE III: Dataset B

Dias *et al.*

	Root	Edges	Leaves	Ancestry
Content	%	%	%	%

Denoising

	Root	Edges	Leaves	Ancestry
Component	97%	89,62%	83,64%	94,80%
Randomness	100%	90,61%	85,03%	93,11%

5. Conclusions and future work

High frequency filter used for denoising algorithm performs high-resolution images as output: if there are even minimal misalignments after geometric transformation, when comparing pixel-to-pixel denoised images we will have negative results. Images need to be perfect aligned, but rotation and cropping introduce errors.

We created a new test set (dataset B) with limited family of transformations: global scaling, color transfer, compression matching. We perform scaling just comparing image size, without calculating SIFT correspondences.

We used a different dataset to evaluate our method with deep geometric transformations: using the same root images and the same topologies of dataset B, we can evaluate the denoising pipeline introducing also rotation, cropping and off-diagonal correction.

Next step is analyzing the importance of compression process in tree reconstruction and its interaction with color matching. Reconstruct a tree with PNG images (no compression, so lossless), with only color transformation, is a challenging task, because color information is not enough to detect the parental orientation in a relation between two images, once that we obtain the same value for both directions.

It seems that is necessary loss information to right estimate the phylogeny: if we consider color transformation as the only component of image transformation family, we want to introduce some loss in the process. We want to compare good color matching and lossy color matching performance.

6. References

[1] Z. Dias, A. Rocha, and S. Goldenstein, “Image phylogeny by minimal spanning trees”, *IEEE Trans. on Information Forensics and Security*, 7(2): 774-788, April 2012

[2] A. De Rosa, F. Uccheddua, A. Costanzo, A. Piva, and M. Barni, “Exploring image dependencies: A new challenge in image forensics,” in *Media Forensics Security II*, 2010, pp. X1–X12

[3] M. Kıvanç Mıhçak, I. Konitsev, and K. Ramchandran, “Low-complexity image denoising based on statistical modeling of wavelet coefficients”, *IEEE Signal Processing Letters*, 6, 300–303, December 1999

[4] J. Lukáš, J. Fridrich, and M. Goljan, “Digital Camera Identification From Sensor Pattern Noise”, *IEEE Trans. on Information Forensics and Security*, vol. 1, no. 2, June 2006

[5] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications*, 3, 34-41, September 2001

[6] H. Farid, “Blind Inverse Gamma Correction”, *IEEE Transactions on Image Processing*, 10(2): 1428-1433, October 2001