E. Renuka, K. Mahesh / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 3, May-Jun 2013, pp.736-740

Statistical property based blind separation algorithm

E. Renuka#1 and K. Mahesh *2
#Research Scholar, Dept. Of Computer Science & Engg., Alagappa Univeristy,Karaikudi, TamilNadu, India.
* Assistant Professor, Dept. Of Computer Science & Engg., Alagappa Univeristy,Karaikudi, TamilNadu, India..

Abstract
A signal that is captured is usually a mix up of two or more original signals and so there is a need to separate them from original sources. It is effective, if this work is done automatically. The process of separating a set of signals from a set of mixed signals without any additional information is known as 'blind source separation'. This work focuses on an image of reflection. Here, we made use of an algorithm developed by Gai et al., This algorithm assumes that the mixtures are linear, with unknown linear mixing coefficients and unknown motions of sources in each image and it is based on the statistics of natural images. Besides the separation of the original sources, the method can automatically identify the number of original images and it has good results even in under-determined cases, where mixtures are fewer than layers

Keywords— *Blind Separation, Moving Images, Image properties, natural image.*

I. INTRODUCTION

The reflection and transmission problem is often encountered in images. Each time a reflective surface, like a glass, window, is present in the captured transmitted scene and the illumination conditions favors reflexions, a reflexion of the scene behind the camera is also captured.

Besides the fact that it is an annoying artifact for human visual perception, it may affect computer vision algorithms like segmentation, object recognition. An automatic algorithm that can separate reflexion in image has got its application in tracking, surveillance, 3D reconstruction, noise removal, deblurring, medical image processing, mass photography and so on.

Gai et al presented a method in which they assume that the input mixtures are a linear combination of the original layers, layers that may have relative motion with respect to each other.
The algorithm first searches for the corresponding motion function by maximizing the correlation of layer gradients. When the correct motion is found, the relation between corresponding layers in different mixtures becomes linear and the correlation is maximized.

In the same time, the linear coefficient gives the mixing coefficient, thus the mixing parameters are found. The reconstruction is done by optimizing a loss function which tends to agree with the mixing model and the extracted layer gradients, estimated using the linear dependence between layers.

The results are impressive and this method seems to be a state-of-the-art for the transmission-reflection separation problem when at least two mixtures are available.

Blind source separation represents the separation of a set of signals from a set of mixed signals, with as little information about the source signals or about the mixing process as possible.
BSS methods rely on the assumption that the source signals do not correlate with each other; for example, the source signals are usually assumed to be statistically independent or decorrelated.

Besides this, other assumptions are made on the set of mixed signals, regarding their independence, normality or complexity with respect to the source signals: the mixtures cannot be independent, since they depend on the same source signals, the histogram of each source signal is more non-gaussian than that of any mixture signal, who tends to have a gaussian histogram (based on Central Limit Theorem), the complexity of a mixture signal is higher than that of the simplest source signal in the mixture.

The basic strategy used for separating the source signals is as follows: if source signals have a property X and the mixture signals don't, then given a set of mixed signals, the goal is to extract signals with as much X as possible; the extracted signals are the required source signals (X can be replaced by any of the properties above)

II. RELATED WORKS

A lot of researchers have focused on solving the problem of transmission and reflexion and a lot of methods have been proposed. Some authors, like James Miskin [2], tried to apply popular methods for blind source separation, like Independent Component Analysis or Variational Bayesian methods.

For a good separation, the number of input mixtures has to be greater or equal to the number of source images and prior assumptions regarding the mixing process are at the base of the algorithm; the mixing process is assumed to be a linear combination of the source images, with additive Gaussian noise.

In some cases, the results are perceptually correct and satisfy the human visual perception system. However, the algorithms are mathematically and computationally complex and often fail in separating arbitrary reflexion images. Other authors focused on separating reflexions from a single input image. Anat Levin[3] developed a method based on a simple prior knowledge: the correct decomposition should have a small number of edges and corners. The algorithm first searches a database of natural images for possible decompositions and then minimizes a cost function which imposes the constraint on the number of edges and corners to get the best decomposition.
The results show that this simple prior is enough for a good separation. Nevertheless, the algorithm often fails either because the candidate decompositions didn't include the "correct" one or because the optimization process is too difficult and fails in finding the optimum separation.

Some authors developed algorithms for blind image separation based on statistics of natural images. Gai et al presented a method [1] in which they assume that the input mixtures are a linear combination of the original layers, layers that may have relative motion with respect to each other.
The algorithm first searches for the corresponding motion function by maximizing the correlation of layer gradients. When the correct motion is found, the relation between corresponding layers in different mixtures becomes linear and the correlation is maximized.

In the same time, the linear coefficient gives the mixing coefficient, thus the mixing parameters are found. The reconstruction is done by optimizing a loss function which tends to agree with the mixing model and the extracted layer gradients, estimated using the linear dependence between layers.

The results are impressive and this method seems to be a state-of-the-art for the transmission reflection separation problem when at least two mixtures are available.

III. PROPOSED WORK

This work proposes a discrete search for the impulse-shaped objective function in three levels:
1. Alternative optimization: Each time, two parameters from p3 to p8 are selected, while the other four are fixed. Then, the middle level is used to search for the optimal solutions of the selected parameters, which are updated accordingly. The selection order is first p3 and p4, then p5 and p6 and finally p7 and p8.
2. Hierarchical brute force: This level is used to search for the optimal values of the selected parameters. First, there is a need to specify an initial searching interval for each parameter and the corresponding search steps. Then, we test all the discrete values in the intervals to get the optimal solution and use the bottom level to match the optimal translation. After the first search, the intervals and the search steps are halved to refine the result. This refining process is done T times to get a good accuracy. In their implementation, the authors used T = 5.
3. Phase correlation: This level is used to match the optimal translation tp = [p1,p2], given p3 to p8. This is done by using phase correlation, a method used in image processing to estimate relative translations between similar images [3].
This method is based on the Fourier shift theorem1. Given two input images, I1 and I2, we calculate the corresponding 2D Fourier transform: I1 = F {I1} and I2 = F {I2}. Then, we need to calculate the cross-power spectrum:

R = I1I∗ 2 |I1I∗ 2| (1)

where I∗ 2 represents the complex conjugate of I2. The normalized cross-correlation is obtained by applying the inverse Fourier transform: r = F−1(R). The location of the peak in r, (Δx,Δy) = arg maxx,y(r), gives us the relative translation between the two input images.
After applying the inverse Fourier transform, the result is an image that shows the power spectrum; since the translation parameters are given by the location of the maximal value of the image, they will always be positive (pixel locations are from 0 to width / height).

A proper method for the interpretation of the power spectrum to identify negative translations still needs to be found.
The steps are computed iteratively until the objective function does not increase. The intuitive meaning is that first we use scalings, rotations and translations to match significant gradients of a layer and then we refine these parameters to match more and more significant gradients of this layer.
Estimating mixing coefficients

The first step in estimating the mixing coefficients, a2j, is to calculate the scatter plot. For this, we apply a motion candidate found after by optimizing the motion objective function to the first image and we plot all the 2D points (Dk(I1(u(x))),∇kI2(x)), where x denotes the pixel location and k denotes either the vertical or the horizontal gradient (since ∇I(x) = ∂I(x) ∂x2 , ∂I(x) ∂x2 and x = [x1,x2]). Then, we remove all the points near the origin (around a given radius) by setting them to 0.

The second step is to calculate the radian plot by calculating rd = arc tan(y/x) for all points (x,y) in the scatter plot. In the end, we add many −π 2,0, π 2.
The third step is to perform the line clustering algorithm, K-Means, on the radian plot with the number of centers set to our guess of the layer number(˜ g) plus 3 (the extra three centers are for −π 2,0, π 2).

After the centers are found, we eliminate the corresponding centers for −π 2,0, π 2 (the ones that are the closest to these points) and remain with the true ˜ g centers, which are estimated radians of the feature lines and a2j = tancentersj.

To find another motion candidate, we now need to eliminate the gradients matched by the current motion. We search through all points in the scatter plot and test whether a point (Dk(I1(u(x))),∇kI2(x)) is near the origin. If it is not near, then the corresponding gradient was matched by the current motion and it has to be eliminated: I set ∇kI2(x) = 0 and ∇kI1(xtrans) = 0, where xtrans = u(x).

The algorithm goes back to optimizing the motion objective function for the remaining gradients and repeats this steps until n motion functions are found.

Extracting layer gradients

The implementation here is straightforward: first calculate the inverse of the motion candidate, then apply it to the second image and calculate the minimal distance from a 2D point(∇kI1(x),Dk(I2(f−1 2j (x)))) to the cluster centers. The point is assigned to the corresponding layer given by the argument of the minimal distance, as described in a previous section.

Besides translations, scalings and rotations, which are easy to invert, the motion function also contains other warping effects. The latter seem to be a combination of simple transformations described by only one parameter

Reconstructing the source layers

The motion transformation is a planar (location to location) transformation, without affecting pixel values, the mixing coefficients linearly change the pixel values and the gradient operator is a combination of vertical and horizontal filters. Thus, all these operations are linear w.r.t. l (the vector containing all pixels of all the reconstructed layers)

min l

J(l) = (ˆ Arl − δ)T(ˆ Arl − δ) + ˆ Erl − τ1, s.t. l ≥ 0 (2)

IV. Observations

1. Texture - In this scenario, how much texture (how many gradients) the original layers have to have for the reconstruction to be perceptually correct is seen. Since the method is mostly based on image gradients for estimating the mixture parameters, if there are few gradients, there might not be enough data for a proper estimation.

Natural images usually are highly textured (i.e., nature scenes), but there are also cases with less texture (i.e., an indoor scene of an empty room with only one table inside). In this case, different motions and mixing coefficients are used to create artificial mixtures.

To get a better view on the amount of texture that the original layers must have for a proper separation, some artificial images are created for testing and the amount of gradients are computed in each image and the corresponding separation error.

The error is not fully reliable, since it highly depends on the amount of texture in the layers. However, my conclusion is that for images with less than 15% gradients each, so very few texture, the reconstruction is not successful, but in all other cases, the reconstruction goes well. This should not be a problem for natural images, since they have a high amount of texture, as can be seen from the results displayed in Appendix B.1. Even in the case of very simple natural images, like the ones in case 6 of an indoor scene with only a table or chairs inside, the amount of gradients is enough for a very good reconstruction.

2. Illumination - this scenario is meant to test the effects of varying illumination. Illumination may vary when the person taking the picture / video cannot control the illumination (i.e., in a museum) or whenever the reflection angles change the distribution between transmitted and reflected lights. To vary the illumination of the mixtures, I played with the mixing coefficients.

For the indoor scenes, which are pretty different in terms of texture, the reconstruction goes well despite the change in illumination. Actually, this variation shouldn't affect the algorithm, since it doesn't take into account intensity values of pixels, but gradients.

Nonetheless, in the extreme case in which one mixture is highly saturated and the other one is underexposed, the reconstruction fails due to the fact that the layers aren't very distinguishable (some texture may vanish because of the saturation and it's hard to recover it from the underexposed mixture).

In the case of natural scenes, which are very similar in terms of texture, the results aren't as good. When one of the mixtures is highly saturated, the reconstruction fails, while in the other cases, the separation is better, but some artifacts (superimposition of layers in the separation) are still present.

The explanation is the same as above: when a mixture is saturated, some texture goes away and we cannot recover the entire layer.

3. Identical motions - There are cases in which both the transmitted and reflected scenes have the same motion and the difference between the mixtures is given only by the mixing coefficients.

The algorithm assumes that relative motions of the layers have to be either identical or different enough, thus the method should give good results.

Even though the mathematical algorithm says that the separation should work, the reconstruction fails in most of the cases. As can be seen, the best results are obtained in the case of the airplanes.

The reason is that the airplanes have a well outlined shape, with strong contour, well delimited from the background, while the other images have more texture which is harder to distinguish with only different mixing coefficients. Here, in the case of identical motions, good results are obtained when the original layers are less textured.

4. Real mixtures - in this scenario, real mixtures are given as input to analyze the performance of an algorithm. Firstly, the mixtures have to be different, either in the motion of the layers or in the mixing coefficients.

A difference in mixing coefficients can be achieved by playing with the reflection angle, which affects the proportion between the reflected and transmitted light.

A difference in layer motion is harder to achieve. For example, if the target object is still (i.e., a painting on a wall) and the person doesn't change his position between snapshots, the relative motion of the layers will be the same, resulting in same layer motions.

This gets easier if one of the objects involved is dynamic (i.e., person in a car or sliding painting).

In any case, only the transmitted scene was recovered, while the reflected layer wasn't. In the paper, the authors present better results in cases of real mixtures and I think the reason is the method of taking the pictures.

Overall, for the majority of my testing cases, the results are perceptually correct and better than the ones obtained with other previous methods. However, the complexity of the quadratic problem makes it impossible for the algorithm to provide an answer in real-time.

V. Simulation Works/Results

We have simulated our system in MATLAB. We implemented and tested with a system configuration on Intel Dual Core processor, Windows XP and using MATLABR2007b. We have used the following modules in our implementation part. The details of each module for this system are as follows:

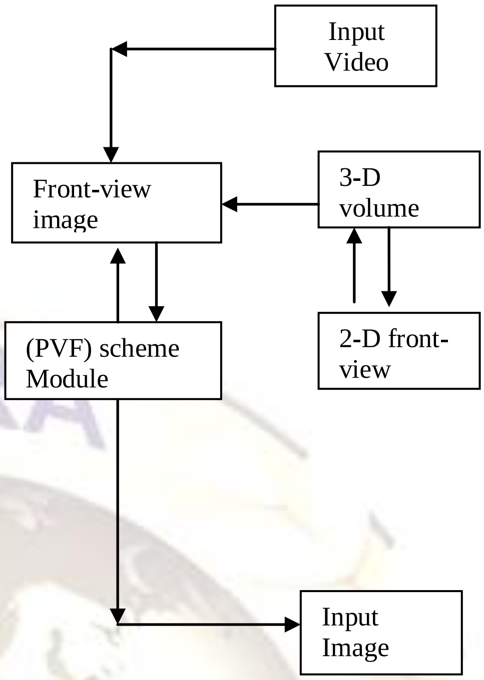

Figure1: System architecture

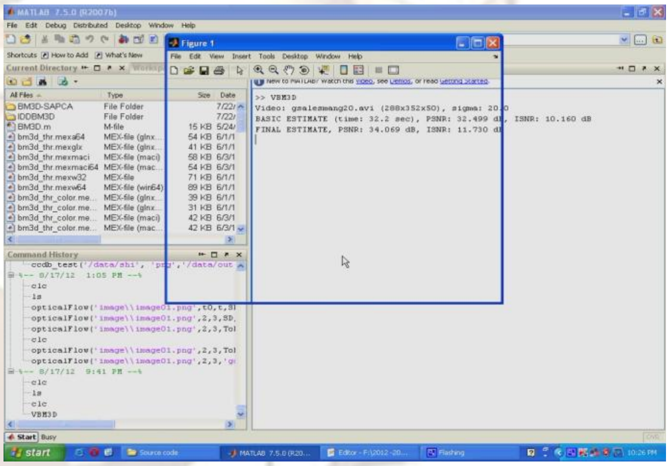

Figure2: Displaying the Video Basic Estimate and Final Estimate values of PSNR and ISNR

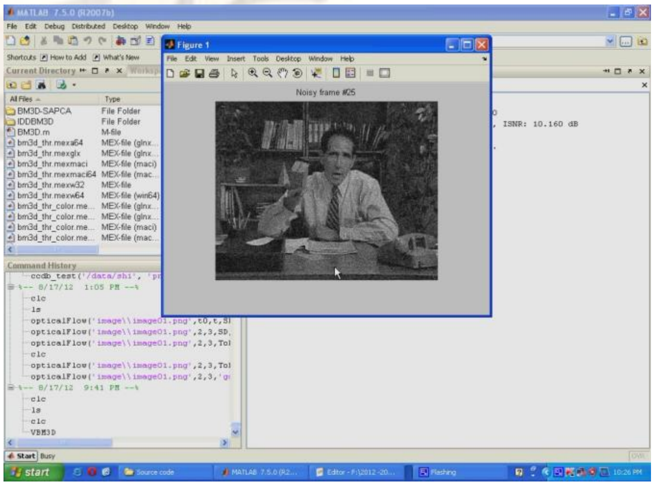

Figure3: Noisy Frame

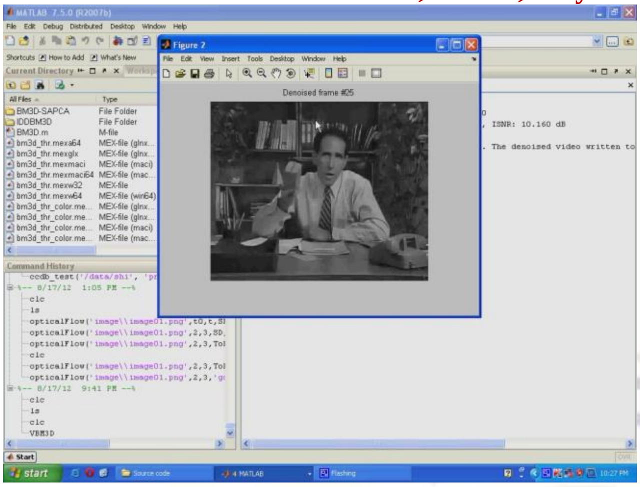

Figure4: Denoised Frame

Figure5: Resultant Video

VI. Conclusion & Future Work

The method of Gai et al gives very good results. It is based on natural image statistics and its intuitive meaning is easy to follow and understand. The mathematical derivations of the algorithm are simple and natural; everything is justified based on the statistics that they have presented. However, when trying to implement the method from scratch, some problems may appear

The algorithm can be further improved in the optimization part, since it's not real time. The time needed for the separation depends also on the size and type of the input mixtures (more time consuming for color images) and on the amount of texture in the layers.

Another drawback of this method is that the input mixtures have to be different in either the relative motion of the layers or the mixing coefficients (reflection angle).

References

[1] Blind separation of superimposed moving images, https://sites.google.com/site/gaikungk/spbsm.

[2] Mosek toolbox, https://www.mosek.com/.

[3] Phase correlation, http://en.wikipedia.org/wiki/Phase_correlation.

[4] Y. Weiss A. Levin, A. Zomet, Separation reections from a single image using local features, IEEE Conf. Computer Vision and Pattern Recognition (2004).

[5] Z. Shi K. Gai and C. Zhang, Blindly separating mixtures of multiple layers with spatial shifts, Proc. Conf. Computer Vision and Pattern Recognition (2008).

[6] Zhenwei Shi Kun Gai and Changshui Zhang, Blind separation of superimposed mov- ing images using image statistics, IEEE Transactions on Pattern Analysis and Ma- chine Intelligence 99 (2011).

[7] Harri Lappalainen and James W. Miskin, Ensemble learning, 2000.

[8] James Miskin, Ensemble learning for independent component analysis, 2000.

[9] James Miskin and David J.C. MacKay, Ensemble learning for blind image separation and deconvolution, 2000.