VIDEO RESTORATION USING MULTICHANNEL-MORPHOLOGICAL COMPONENT ANALYSIS INPAINTING

ADAM S. BACKER, BENJAMIN T. ROLFS, AND FELIX L.Y. YU

December, 2009

Abstract. Morphological component analysis (MCA)[1, 2] is a popular image processing algorithm that extracts degrading patterns or textures from images and simultaneously performs inpainting (estimation of lost pixels). MCA has a wide range of uses, including MRI image enhancement and restoration of old photographs. However, in these authors' opinions, an application that has been widely overlooked is the use of MCA in restoring video footage. We present a novel implementation of an augmented version of MCA known as multichannel morphological component analysis (mMCA), and use it to reconstruct heavily degraded video. Our algorithm examines consecutive frames of video footage in order to identify and remove distortions. Results in this paper indicate that this approach has some significant advantages over the use of conventional MCA on individual video frames.

1. Introduction

Repetitive textures often degrade images, ruining fine detail and obscuring relevant information. Formally, if we represent an image as a vector[1] $\mathbf{X}$, we can decompose it into two or more components, $\mathbf{X} = \sum X_n$. A particularly useful decomposition for image processing applications is into piecewise-smooth (cartoon) and repetitive (texture) components, respectively called X_1 and X_2. In this case, we are interested only in the undistorted cartoon component, and would like to isolate the degrading texture. Figure 1.1 shows a decomposition of an image into cartoon and texture components. We can introduce another complication: the data vector $\mathbf{X}$ may have missing entries. This can occur when the image itself is damaged, or if the system used to capture the image is imperfect. In this case we would like to remove the texture component of an image and simultaneously make the best possible estimate for the missing entries of the cartoon component. While a great deal of previous literature has been devoted to MCA–an algorithm that accomplishes this task for single images–this report focuses on video restoration. We address the following question: given not one vector $\mathbf{X}$, but a sequence of vectors labeled $\mathbf{X}^i$, how is this additional data optimally leveraged to produce accurate estimates of the original, undistorted cartoon components X_1^i? We do this by implementing an extension of MCA called multichannel MCA (mMCA)[3] and applying it to video data.

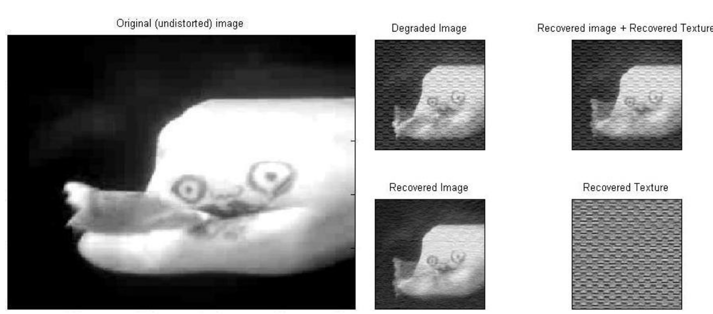

Figure 1.1. MCA, as implemented in the MCALab package[1], used to decompose an image into cartoon and texture components.

[1]Although it is conventional to represent an image as a matrix, for ease of notation we use a vector representation. Reasons for this will be clarified in the theory section.

This paper is organized as follows: In section 2, we outline the theory behind MCA, explaining how it separates morphological components (in this case, cartoon and texture layers), and how it performs inpainting. In this discussion, we describe how the inpainting process is essentially expectation-maximization (EM). In section 3, we outline our approach for restoring video data. In section 4, we present results from a series of experiments to determine the effectiveness of this extension of MCA. We find that when a high percentage (> 95%) of the pixels are missing, our method substantially outperforms merely applying the standard version of MCA to individual video frames. We end with a short conclusion where we consider possible future directions of this research.

2. Theory

The fundamental insight behind MCA with respect to the decomposition problem $\mathbf{X} = \sum X_n$ is that for each layer X_n there often exists a basis $\mathbf{T}_n$ in which the given layer is sparsely represented. These bases are referred to in this paper as 'dictionaries', which is their conventional name in MCA work[1, 2, 3]. For the cartoon/texture decomposition of an image, $\mathbf{X} = X_1 + X_2$ (henceforth, we will only consider this decomposition, although the theory is valid for a general decomposition as well), this gives $\mathbf{X} = \mathbf{T}_1\alpha_1 + \mathbf{T}_2\alpha_2$, with α_i sparse. Furthermore, the dictionaries can be made mutually incoherent, meaning that for instance the cartoon component X_1 is sparsely represented only when transformed by some operator $\mathbf{\Phi}_1$ into the $\mathbf{T}_1$ (cartoon) basis but is not sparse when transformed into the texture basis, and vice versa. In other words, each dictionary is a desirable basis only for its specific component, allowing the MCA algorithm to distinguish between the cartoon and texture components of an image. This formulation leads to the following optimization problem:

$$\{\alpha_1^*, \alpha_2^*\} = \arg\min \|\alpha_1\|_0 + \|\alpha_2\|_0 : \mathbf{X} = \mathbf{T}_1\alpha_1 + \mathbf{T}_2\alpha_2 \qquad (1)$$

Here,$\|\bullet\|_0$ is the l_0 norm, the number of non-zero entries in a vector. Note that due to the structure of the dictionaries, the transform operators $\mathbf{\Phi}_i$, and the nature of the l_0 norm, the optimization problem is more conveniently posed when we represent our image, and each transformed layer α_i, as vectors.

The optimization problem (1) is highly non-convex, and in most cases intractable[2]. Fortunately, it is possible to solve a related, and in many cases equivalent[4] problem resulting from replacing the l_0 norms with l_1 norms. This convexification yields a solvable optimization problem. However, we can only hope that such a decomposition will be sparse if our image $\mathbf{X}$ is indeed composed solely of cartoon and texture components, and thus can actually be represented sparsely in terms of the dictionaries $\mathbf{T}_1$ and $\mathbf{T}_2$. Since this is generally not the case, we can compensate by including a penalized residual term, assumed to be in the form of some noise[2]. This results in the below equation, which is the optimization problem[1] actually solved by MCA:

$$\{\alpha_1^*, \alpha_2^*\} = \arg\min \|\alpha_1\|_1 + \|\alpha_2\|_1 + \lambda \|\mathbf{X} - \mathbf{T}_1\alpha_1 - \mathbf{T}_2\alpha_2\|_2^2 \qquad (2)$$

where the norms have been changed to l_1 norms and the constraint that the transformed components α_k perfectly represent our image has been replaced with a λ-penalized residual term. Solving this optimization problem, with the right choice of dictionaries, yields a decomposition of an image into cartoon and texture components.

The MCA algorithm can be reformulated to solve the inpainting problem. Suppose we have a diagonal 'masking' matrix $\mathbf{M} \in \{\mathbf{0}, \mathbf{1}\}^{NxN}$, whose entries signify whether or not a given pixel in our true image vector $\mathbf{X}$ has been occluded or not. We are only able to observe the masked image $\mathbf{MX}$, but would like to estimate the missing entries as follows[4]:

$$\{\alpha_1^*, \alpha_2^*\} = \arg\min \|\alpha_1\|_1 + \|\alpha_2\|_1 + \lambda \|\mathbf{M}(\mathbf{X} - \mathbf{T}_1\alpha_1 - \mathbf{T}_2\alpha_2)\|_2^2 \qquad (3)$$

which is the optimization problem solved for MCA inpainting. This problem can be motivated from a *maximum a posteriori* (MAP) estimation standpoint by viewing the $\|\mathbf{M}(\mathbf{X} - \mathbf{T}_1\alpha_1 - \mathbf{T}_2\alpha_2)\|_2^2$ term as a log-likelihood, and the cartoon and texture representations $\|\alpha_1\|_1$ and $\|\alpha_2\|_1$ as prior terms that favor sparse (low-norm) solutions. In this way, we can view the true image as $\mathbf{X} = X_1 + X_2$, but degraded by Gaussian noise that leads to the log-likelihood term[2].

This MAP estimation structure leads to an EM algorithm that can be used to perform inpainting. In the EM framework, this is done by computing the conditional expectation of the penalized error (log-likelihood) term given the current estimate for full image, and in the M-step optimize each representation vector α_i while holding the others fixed. A sketch of the algorithm for this is as follows[5]:

$$E - Step: \qquad \mathbf{X}_{est} = \mathbf{M}\mathbf{X} + (I - \mathbf{M}) \sum \mathbf{T}_k \alpha_k^{(\mathbf{i})}$$
$$M - Step: \quad \alpha_j^{(\mathbf{i+1})} = \arg\min \sum \left\| \alpha_k^{(i)} \right\|_1 + \lambda \left\| \mathbf{X_{est}} - \sum \mathbf{T}_k \alpha_k^{(\mathbf{i})} \right\|_2^2 \quad for\ j = 1, 2 \qquad (4)$$

To clarify, in the E-step we are using our observed image $\mathbf{MX}$ and our last update for each α_k to estimate the combined texture and cartoon components for our full image, $\mathbf{X}_{est}$. In the M-step, we hold all but one $\alpha_k^{(\mathbf{i})}$ constant, and optimize equaion (4) with respect to that $\alpha_k^{(\mathbf{i})}$ which is not fixed. In each M-step, this update is performed for all k. This iterates for a certain number of steps in order to attain estimates for α_1 and α_2 from which we can reconstruct our full image, thus filling in the missing pixels.

At this point, it merits pointing out that this entire formulation requires access to applicable dictionaries in which our texture and cartoon components can be represented sparsely and with mutual incoherence. Standard dictionary choices are the curvelet[6] basis for the cartoon components and the local-discrete cosine transform (LDCT)[7] basis for the texture components. Fast transform and inverse transform operators exist for both of these bases, allowing projection into each space and reconstruction from coefficients in each basis during the MCA inpainting algorithm. These are the dictionaries we used in our application of MCA.

The final theoretical extension to MCA that must be discussed for our application is that it can be used for inpainting multichannel data[3, 8]. Multichannel morphological component analysis (mMCA) extends mMCA to take into account n m-dimensional observations, for example a color picture or in the case of our application a sequence of consecutive video frames. Analogous to independent component analysis (ICA), mMCA assumes independence between sources of pre-mixed data, and attempts to find an unmixing operator that best separates sources. The difference is that mMCA specifically searches for sparse representations of the morphological layers with respect to mutually incoherent dictionaries. That is, we attempt to find an unmixing operator that maximizes sparsity.

3. Methodology

To solve the minimization problem (3) that defines mMCA, we implement the optimization procedure discussed in [3], which iteratively estimates each α_k and the unmixing operator. Supplementary MATLAB code for our mMCA function, as well as a driver-script and a provisional dataset can be downloaded at `http://www.stanford.edu/~abacker`. Our code performs restoration upon two-frame blocks of video data. For example, to restore frame n, frame n and frame $n+1$ are input. Hence, the final restored video had one fewer frame than the original. Throughout all our experiments, we used curvelet and LDCT dictionaries to separate relevant image content from texture. We calculate such transforms using functions provided in the open-source image-processing suite WaveLab. In all experiments, the penalty parameter λ from (3) was set to the inverse of the estimated standard deviation of the noise in the input images. This estimation was accomplished by band-pass filtering the input image, and then calculating the variance.

To compare the merits of our method versus other approaches, we perform MCA restoration on individual video frames as well. For MCA trials, we use the package MCALab[1], which can be found at `http://www.greyc.ensicaen.fr/~jfadili/`. In order to acquire a suitable video dataset, we captured a three-second video clip using an ordinary camera-phone. Each frame was cropped to 256-by-256 pixels, and color was removed. A synthetic texture was then added to each frame and a random selection of pixels were removed in order to allow inpainting. For inpainting experiments, we had free choice of how to initialize 'missing pixels'. We heuristically found that random initializations worked best for our mMCA implementation, while this practice led to instability and unpredictable results for conventional MCA. The best performance for MCA occurred when missing pixels were initialized to zero.

4. Results

To demonstrate the efficacy of mMCA, we perform two video restoration experiments. In the first experiment, we restore a black-and-white video clip respectively using mMCA and MCA, and make a qualitative comparison of results. In our second experiment, we push these algorithms to their limits by investigating the percentage of pixels that must be supplied to MCA and mMCA in order to yield reasonable restorations. In this experiment, we determine that MCA needs a critical minimum number of input pixels. If fewer pixels are given, MCA produces wildly inaccurate restorations. mMCA, on the other hand, exhibits no threshold of this sort. Results become increasingly inaccurate as the number of input pixels dwindles, however there is no point at which the method catastrophically breaks down – a significant achievement over MCA.

4.1. **Experiment 1.** We include four AVI files to evaluate the respective merits of MCA and mMCA restoration on a short clip of degraded video. The first AVI shows the original video clip, while the second clip shows the image after the addition of a degrading texture, and random removal of thirty percent of the pixels from each frame. The third and fourth videos show the recovered images after applying MCA and mMCA. Qualitatively, both recovery algorithms perform similarly: while it is difficult to distinguish meaningful content before restoration, it becomes obvious after applying MCA and mMCA that the video is of a hand. Furthermore, both algorithms successfully identify and reduce the added texture content. In general, the mMCA algorithm exhibits slightly more blurring. This is due to the fact that it draws upon information from two consecutive frames in order to reconstruct individual images. Some representative frames for both recovery methods are provided in figure 4.1.

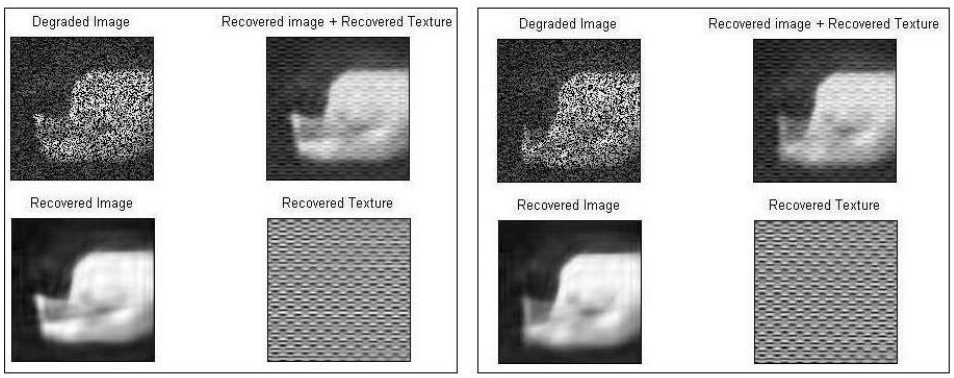

FIGURE 4.1. Video restoration on frames with 40% pixel loss, using MCA (left) and mMCA (right).

4.2. **Experiment 2.** To quantitatively determine how increasingly high percentages of pixel removal impairs MCA and mMCA, we selected two consecutive frames from our video-clip (after the addition of a degrading texture) and repeatedly perform mMCA and MCA on these frames, while removing between ninety-three and ninety-eight percent of the pixels during each run (again, removed pixels were selected randomly). The total error in the separated cartoon component $\|X_1^{true} - X_1^{est}\|_2$ for both MCA and mMCA is plotted in figure 4.2. This plot provides compelling evidence of major failure in the MCA algorithm occurring after about ninety-five percent pixel removal. Around this region, the recovery error of MCA dramatically increases. On the other hand, the error gradually increases for mMCA, but exhibits no abrupt failure. Images produced by the two recovery methods, after removal of 96 percent of the pixels, are shown in figure 4.3. The MCA algorithm retrieves none of the relevant content from the original image. However, mMCA is capable of distinguishing general trends in the data – such as an oval white splotch superimposed upon a darker background. The key reason for the success of mMCA is more subtle than the simple fact that it has access to twice the amount of data. As mentioned previously, we have a number of options in regard to how to initialize the values of the missing pixels. Normally, initializing missing pixels to zero works fine for both mMCA and MCA. However, since sparse vectors can be represented sparsely in the curvelet basis as well, such an initialization strongly biases MCA and mMCA toward extremely sparse recovered images, if an overwhelming number of the missing pixels are initially set to zero. Hence, when

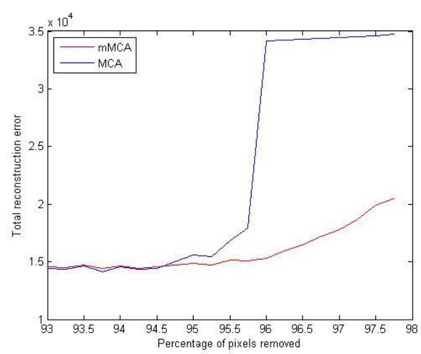

FIGURE 4.2. Error in catoon component for MCA vs. mMCA

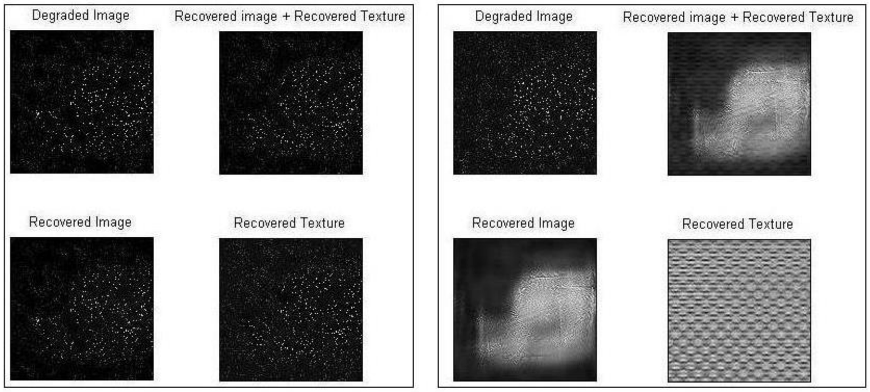

FIGURE 4.3. Video frame restoration for 96% pixel loss using MCA (left) and mMCA (right).

conducting mMCA, it is best to initialize pixels randomly, so that the results are not unduly influenced. Since there is no correlation between the starting values of missing pixels from one frame to the next, a random initialization rarely biases output of mMCA unfavorably. The same trick does not work for MCA, since it has no means of inferring inter-frame correlations (or lack thereof).

5. Conclusions

This report demonstrates a novel application of mMCA to video restoration, which has some advantages over conventional MCA. In particular, we find that performance is enhanced when a very high percentage of pixels have been lost. Under less challenging circumstances, when a larger amount of data is available, both methods produce similar results. Further enhancements to our mMCA algorithm are possible. In the future, we would like to augment our code to consider more than two successive frames. While such an improvement may seem trivial, it encroaches upon dangerous territory, since motion blurring in restored images will increase if the number of input frames to mMCA is large. Hence mMCA could greatly benefit from methods that adaptively determine the optimal number of frames to provide as input to mMCA, depending upon the amount of motion which occurs in those frames.

References

[1] M.J. Fadili, J.-L. Starck, M. Elad and D.L. Donoho. "MCALab: Reproducible Research in Signal and Image Decomposition and Inpainting." *IEEE Computing in Science and Engineering* (2009).

[2] J.-L. Starck, M. Elad, D. Donoho. "Image decomposition via the combination of sparse representations and a variational approach." *IEEE Trans. Image Proces.* **14(**10): 1570-1582 (2005).

[3] J. Bobin, Y. Moudden, J.-L. Starck, M. Elad. "Multichannel Morphological Component Analysis." *Proceedings of Spars05*: 103-106 (2005).

[4] M. Elad, J.-L. Starck, P. Querre, D.L. Donoho. "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)." *J. App. & Comp. Harm. Anal.* **19**: 340-359 (2005).

[5] M.J. Fadili, J.-L. Starck. "EM Algorithm for Sparse Representation-Based Image Inpainting." *Proc. IEEE Int. Conf. Image Processing* **2**: 61-63 (2005).

[6] E.J. Candés, D.L. Donoho. "Curvelets- a surprisingly effective nonadaptive representation for objects with edges." *Curves and Surfaces* (2000).

[7] S.A. Khayam. "The Discrete Cosine Transform (DCT): Theory and Application" (2003).

[8] J. Bobin, Y. Moudden, J. Fadili, J.-L. Starck. "Morphological Diversity and Sparsity for Multichannel Data Restoration." *J. Math. Imaging Vis.* **33**: 149-168 (2009).