EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Blotch and scratch removal in archived film using a semi-transparent corruption model and a ground-truth generation technique

Mohamed A Elgharib[1*], François Pitié[2] and Anil Kokaram[3]

Abstract

This paper has two main contributions. The first is a Bayesian framework for removing two common types of degradations on video known as blotches and line scratches. Most removal techniques assume complete obliteration of the original data at the corrupted sites. This often leads to the introduction of restoration artifacts during removal. Our framework is based on modeling corruption as a semi-transparent layer. This model was introduced earlier by Ahmed et al. (ICIP 2009) for the problem of blotch removal. We show much more blotch removal results than the previous work, and we extend the semi-transparent corruption model to the problem of line removal. The second contribution of this paper is an automated technique for ground-truth generation from infrared scans of corruptions. Previous ground-truth generation efforts require manually inpainting the corrupted regions. The restoration results are evaluated by comparing the reconstructed data against the ground-truth estimates. Comparisons with current blotch and line removal techniques show that the proposed corruption removal framework produces better removal and generates less restoration artifacts.

Keywords: Blotch; line scratch; transparency; Bayesian matting; graph-cut; infrared; ground-truth

1 Introduction

The last century has witnessed an explosion in the amount of video data stored with holders such as British Broadcasting Company, Institut National Audiovisuel (INA, Paris, France), and Raotelevisao Portuguesa, Lisbon, Portugal. Beyond the cultural heritage these data represent, their value is increased by their commercial re-exploitation through digital visual media. This requires the archived data to meet a high level of visual quality. This, however, is usually not the case as the data are often visually degraded during their long-term storage under bad physical and climate conditions. This was a source of concern in the broadcasting community during the 1980s [1], which led to the emergence of a new market for the automatic restoration of digital visual data.

There are several forms of impairments on digital visual data [2]. Perhaps the most common impairments are 'blotches' and 'line scratches'. These impairments mainly rise due to dirt particles adhering to the film material or due to film abrasion. Blotches appear as temporally impulsive dark and bright spots which are distributed randomly over an image sequence (shown in green in Figure 1). Line scratches appear as near vertical lines of high contrast that propagate through time (shown in blue in Figure 1). Removing these impairments usually implies some kind of detection/correction process. Ideally, corrupted sites can be estimated using infrared scans (IR) of medium (see Figure 1 right) [3]. Infrared is transmitted through film but stopped by dust. As a result, the obtained scan is bright in original data regions and dark in dirt, ie., corrupted regions. A simple threshold operation is, therefore, sufficient to segment the corrupted sites from the rest of the image. However, much archived material is no longer available with IR scans; hence, automated digital detection algorithms remain important.

Automatic restoration algorithms for archived visual media have been studied since 1985 [1]. Several authors have since then proposed numerous detection/reconstruction techniques for blotches [4-8]. Most of these techniques detect blotches by looking for temporal

*Correspondence: gharib@bu.edu
[1] Boston University, Boston, MA, USA
Full list of author information is available at the end of the article

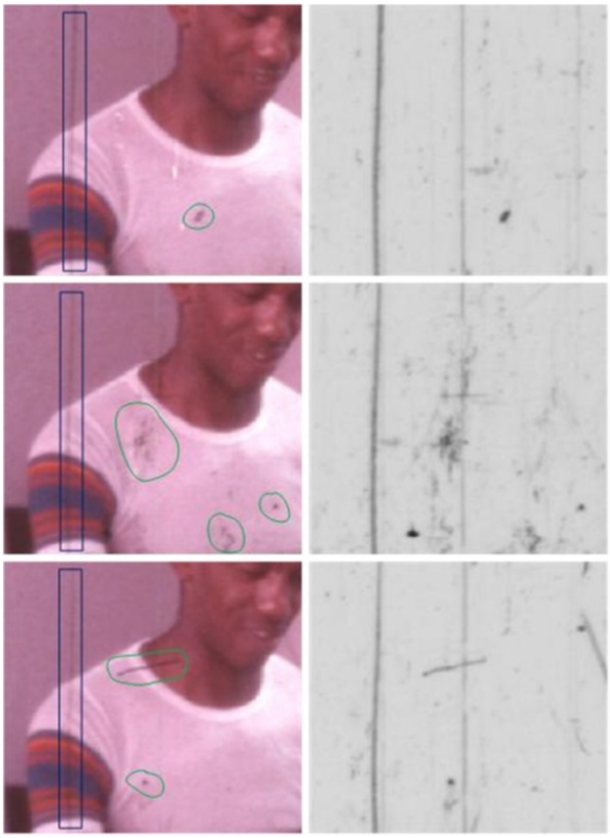

Figure 1 Three consecutive frames (from top) and their corresponding infrared scans on the right. Blotches (shown in green) are temporally discontinues, while line scratches (shown in blue) propagate through frames.

discontinuities between the motion compensated frames at some point in their procedure. As a result, many clean regions often get classified as corrupted, where motion estimation fails. Line scratch detection is considerably more difficult due to their temporally consistent nature. Early scratch detection systems tended to use information from the current frame only [9-12], but the most successful technique uses detection over multiple frames [13]. However, it is still possible to confuse line scratches with true vertical lines in an image sequence, and again undegraded material is then passed for correction.

Any subsequent image correction technique must use nearby clean spatial and temporal information. A ‘good’ correction technique has two main features: (1) correcting only the corrupted portion of the data and (2) resynthesizing the original data correctly regardless of motion and texture complexity. Most of the existing reconstruction techniques assume complete loss of the original information at the detected sites. Hence, they must interpolate quite large areas; because of the shortcomings of the detection step, they tend to attempt to correct a substantial amount of clean data. The key point, though, is that false alarms in many of these techniques tend to occur where the motion and texture information is complex, especially where pathological motion occurs [14]. So in cases where there is motion blur due to fast moving objects, or self occlusion, or a vertical stripe moving behind a line scratch, there are a high degree of false alarms. These areas are precisely the areas where any image interpolator will fail in a video sequence. The fact that existing algorithms find it hard to recover from poor detection remains an obstacle to completely automated handling of these artifacts, and existing industrial software (www.thefoundry.co.uk) use a variety of *ad-hoc* processes to reduce artifact detection to a very conservative process.

However, a quick glance at Figure 1 shows that blotches and lines often do not completely obliterate the underlying area and are often not opaque at all. If we could build a model that would capture that observation, then it might be possible to improve detection precision and increase robustness to poor motion and texture. In a sense, this is related to estimating the level of corruption at a site as a continuous variable between 0 (no corruption) and 1 (totally lost). In this paper, we define the opacity of a blotch in frame n at site $\mathbf{x}$ as $\alpha_n(\mathbf{x})$ and propose a new linear model of corruption as follows:

$$G_n(\mathbf{x}) = \alpha_n(\mathbf{x})F(\mathbf{x}) + (1 - \alpha_n(\mathbf{x}))I_n(\mathbf{x}) \tag{1}$$

Here, $G_n(\mathbf{x})$ is the observed corrupted intensity at a pixel site $\mathbf{x}$ in frame n. I_n is the clean original intensity at that site, and F is a constant intensity representing the underlying blotch color. In this paper, we will use $F = 0$ to model black or rather dark blotches/lines, and $F = 255$ will model brighter/sparkle. Hence, the model implies that the observed data is a linear mixture between the clean original data and a corruption color F. For color images, this model is applied for each RGB channel separately. Most techniques, however, model the corruption as either a completely opaque layer ($\alpha = 1$) or as a purely transparent one ($\alpha = 0$ for clean regions).

This model is clearly related to the image layer model used in the vast quantity of matting work starting with the seminal paper of Chuang in 2001 [15-20]. In matte extraction, an image is assumed to arise as the result of a linear mixture of foreground F and background elements. In our situation, one of the layers is known, and our problem is to estimate α and I to achieve restoration.

Note that the goal of this paper is to address the issue of *removal* of the blotch or line once it has been detected. We do not address the issue of detection, even though the process of defect matte extraction discussed here can be thought of as a detection refinement step. In this paper, we present a unified Bayesian framework for blotch and line scratch removal using a semi-transparent corruption model. This model was briefly introduced in [21] for the problem of blotch removal. Aspects of novelty in this paper are as follows:

(1) More results for blotch removal over [21]. This includes discussing system implementation/parameters in more detail and processing much more data.
(2) A new technique for line scratch removal based on recursive filtering. This technique is not available in [21].
(3) A new analysis of IR scans to create accurate ground truth for these kinds of studies. These analysis are not available in [21].

In the next section, we present an overview of the current state of the art blotch and line removal techniques. We then go on to propose the Bayesian framework and its solution. The results section also then contains new work in the exploration of IR film scans and discusses results in terms of real and artificial data.

2 Review of blotch and line removal algorithms

Previous approaches for removing blotches and line scratches have largely fallen into two categories. The first models the corruption as an opaque layer which obliterates the underlying original data completely [2,22-26]. The second category models the corruption as a semi-transparent layer, assuming the survival of some original data in the corrupted sites [27-31]. Both categories require that the sites of corruption be detected first. Several automatic detectors have been introduced previously [5-8,32], but in this paper, we are concerned with removal, not detection. Detection will, therefore, not be discussed in detail; comments on detection relevant for our purposes are left until later in this section.

2.1 Opaque corruption removal

Define the intensity of the pixel at site $\mathbf{x}$ in the nth frame of the observed corrupted sequence as $G_n(\mathbf{x})$, and in the clean original sequence as $I_n(\mathbf{x})$. Then, many of the ideas in this class can be unified under the following corruption model:

$$G_n(\mathbf{x}) = \begin{cases} I_n(\mathbf{x}) & \text{When } b_n(\mathbf{x}) = 0 \\ C_n(\mathbf{x}) & \text{When } b_n(\mathbf{x}) = 1 \end{cases}$$

Here, $C_n(\cdot)$ is the intensity of the corrupting blotch (or dirt) or line scratch in the nth frame, while $b_n(\cdot)$ is a binary indicator that is 1 when a site is corrupted and 0 otherwise. In a sequence of frames, the typical assumption (usually true) is that a site is not corrupted at the same location in consecutive frames.

Storey [1] in 1984 was the first to use this idea for dirt detection in archive film. Given that C was completely different from the surrounding image material, it was sensible to configure the b field by simple thresholding of interframe differences without motion compensation. Hence, data which cannot be found in previous and next frames are detected as corrupted sites. The corrupted sites were then treated with a three-tap non-motion compensated median filter which removed C as outliers in the underlying observed sequence. Since 1992, Kokaram et al. [2,6,25,26,33] has gone on to introduce motion into the degradation model and employ new spatio-temporal median filters to remove the degradation; eventually, [6] proposed a variety of Bayesian techniques for the joint solution of motion and degradation. The Bayesian ideas are based on modeling the original data and indicator fields (I, b) as Markov random fields (MRFs) and incorporate 3D autoregressive models for the underlying image data. The first technique incorporating those ideas is JOMBADI [6]. JONDI was then proposed in [2]. It builds on JOMBADI and proposes more complete models that allow for occlusion simultaneously. In that body of work, reconstruction of the missing data is always by interpolating the missing sites from surrounding frames in space and time. Many of these ideas can be found in commercial software for blotch treatment in archive film today (www.snellgroup.com, www.thefoundry.co.uk, and www.thepixelfarm.co.uk).

Since 1998, a variety of authors have observed that missing data reconstruction, using motion compensated data, relies heavily on the accuracy of the motion information. When the observed motion is *pathological*, e.g. very fast, containing motion blur, or periodic elements, the motion estimator often fails and the reconstruction process introduces more defects than it removes. This is a problem particularly with the movement of people and clothing. Bornard [14], Corrigan et al. [34], Rares et al. [35], Roosmalen [36], and Kent et al. [37] introduced several mechanisms for dealing with such *pathological motion* effects. Roosmalen [36] concentrated on detecting failure in the motion estimator he used (based on phase correlation) by simply turning off the blotch remover when the displaced frame difference (DFD) was too high over many consecutive frames. Kent et al. [37] simply turned off the blotch remover in any moving foreground region detected by a crude image object segmentation process based on motion. The idea here was that motion estimation typically fails in foreground regions (if it fails at all); so, this yielded a very conservative process. Bornard [14] and Corrigan et al. [34] pushed Roosmalen's ideas into a Bayesian framework, eventually incorporating MRF priors for blotch smoothness in time that allowed the blotch remover to implicitly disable itself when a blotch was being detected in more than one frame consecutively. Rares et al. [35] used machine-learning-type ideas to detect picture material which was difficult for motion estimation, again turning off the process in difficult areas.

As far as picture reconstruction in the missing region is concerned, Kokaram [24], Kent et al. [37], and Corrigan et al. [34] all relied on motion-compensated picture interpolation of some kind. These were all based on some model of the underlying image sequence. Kokaram employed a variety of 2D and 3D autoregressive models to achieve interpolation for missing data, again based on a Bayesian framework [24]. Spatial picture interpolation is particularly difficult, and that becomes very important when the defect exists in the same place in consecutive frames, i.e., line scratches. Spatial picture interpolation using filters or simple linear models tends to yield a blurred reconstruction in general. However, the remarkable results of Efros and Freeman [38] and Efros and Leung [39] in non-parametric texture synthesis clearly can be adapted for blotch and defect removal in general. Bornard [14] was the first to extend those ideas for spatio-temporal interpolation, and he generated very convincing results for both line scratches and blotches. The idea was to *inpaint* the missing patches using similarity searches not only in the current frame but also in previous and next frames.

Unlike spatial techniques, spatio-temporal techniques cannot be applied directly to line removal due to the temporal consistency of the corruption. Attempts in doing so are based on extending image inpainting and texture synthesis techniques to use clean information from nearby frames [23,40]. Forbin et al. [23] employed the directionality inpainting of Criminisi et al. [22] for line removal. They noticed that scratches often have a very thin width, so a concentric filling would be enough rather than the more complicated filling order adopted by Criminisi et al. [22]. These modifications had the effect of reducing some visual block artifacts generated by [22].

Regardless of the nature of the reconstruction process, temporal consistency in the interpolation is very hard to achieve, especially in the case of line scratches. The most frequent complaint is that line scratch removers leave a *shadow* of a line behind. Even with blotch removal, once the attempt it made to reconstruct the corruption sites completely, there are always going to be knock-on effects. But close examination of blotches and lines shows that they often do not completely remove the underlying data; so, a softer approach is justified.

2.2 Semi-transparent corruption removal

Since 2003, researchers working on degraded photographs (stills) noticed that a blotch can be modeled as a linear mixture of original data and some corruption layer. Stanco et al.'s work in [30,31,41] was the first to model the corruption as some linear function of the observed data, but it was Crawford et al. [29] who introduced an explicit mixture model that was the same as Equation 1. The corruption was, therefore, modeled as a semi-transparent layer with non-binary opacity values: it is the matting equation. Their work focused on the removal of blotches on photographs due to moisture. Removal is addressed in the HSV color space. Chroma components are restored using a simple texture synthesis technique [20]. The luminance channel is first split into an over-complete wavelet representation. The details band is modeled as a linear mixture between dirt and the original data using the matting equation [29]. Similarly to the MRF work in [6], the unknown parameters are assumed to vary smoothly within local patches. This is done by modeling them as MRFs with Gibbs energies, and a solution is formulated within a Bayesian solution. The original information I and the mixing parameter α are then estimated using iterative conditional modes [42]. Having calculated the mixing parameters, the wavelet details are either attenuated in case of spurious edges or left unchanged in case of perfect semi-transparency ($\alpha = 0$). Restoration results show good recovery of the original information; however, their technique assumes that the processed data have very simple texture and fail when blotches cause significant image loss. In addition, estimating α in the luminance space does not fully exploit all the color channels.

However, that work was not fully exploited in an image sequence context as yet. In a sense though, the work of Crawford et al. is related to the work of Hisho et al. in the late 1990s [43,44]. Their idea was to introduce a non-binary index for measuring the level of corruption in the blotch removal problem for image sequences. Unlike opaque corruptions, where a site is assigned a binary corruption value (1 for corrupted, 0 for clean), Hisho et al. [43] proposed the assignment of a non-binary value instead. This value is a function of the temporal discontinuity between the examined site and the nearby frames. The function is learned through a training process [44]. Having calculated the corruption level for each site, the interpolated value is then set to an intermediate value between the original image brightness and the output of an arbitrary spatio-temporal filter, depending on the corruption level. This is a simple process attempting to capture the true nature of the problem which should arise from consideration of Equation 1 as the degradation model.

Few automatic techniques for removing line scratches using a semi-transparent corruption model have been proposed [27,28]. Bruni et al. [28] developed a technique for removing such defects from gray scale still images. Their idea was to reduce the defect intensity until it is no longer visible. The corruption is modeled as a linear mixture between the line profile and the original data. A sinc^2 function is used to model the scratch horizontal profile, and removal is performed in the wavelet domain. The scratch is removed after estimating the attenuation factor which leads to minimal corruption visibility. This factor is calculated using Weber's Law. Results show good scratch removal without changing its surrounding and with no smoothing of texture. This technique was later extended to the removal of color scratches in [27].

2.3 Motivation

Opaque blotch removal techniques in general often tend to correct clean regions which are incorrectly classified as corrupted. The effect is the introduction of unnecessary clearly visible visual artifacts. On the other hand, current semi-transparent removal techniques fail in removing highly opaque corruptions (Saito's and Crawford's techniques [29,44]), can only handle very simple texture (Stanco's and Crawford's [29,31]), or require a detailed model of the corruption profile (Bruni's techniques [27,28]). Furthermore, none of the current semi-transparent removal techniques incorporate useful temporal information. In a sense, there is no corruption removal technique adopting a semi-transparent model which can handle restoration in color image sequences, having complicated texture and undergoing fast motion. This paper presents a technique designed to meet these goals. The proposed approach follows the same corruption model of Crawford; however, we apply the matting equation to all the RGB channels in order to exploit all color information for better reconstruction. We address the problem from a corruption matte extraction point of view and propose a solution which builds on 'Bayesian matting' [15]. The algorithm follows the footsteps of JOMBADI [6] and Crawford [29]. The novelty, however, is in the specification of spatial and temporal priors which can handle complicated texture and motion and the generation of the MAP solution using graph-cuts.

2.4 A note on detection

We assume that detection of the blotched regions has already taken place. This may become available from IR scans of the film material, or more likely from a simple blotch detection process. We employ the simplest blotch detection process here, SDIp [6]. It detects a blotch when the motion-compensated frame difference at a pixel site is large both with the next and previous frames. It is a crude and very simple detector, but because our subsequent process is *soft*, we can recover from false alarms. In a sense, this is just a kick start to the subsequent estimation process. Therefore, given the detection field $b(\mathbf{x})$, our problem is to reconstruct each small collection of pixels, where $b(\mathbf{x}) = 1$, i.e., a reconstruct of the data in a missing patch.

3 Bayesian inference for semi-transparent blotch and line removal

A corrupted pel $G(\mathbf{x})$ is modeled as a linear combination between the original data (background layer) $I(\mathbf{x})$ and the corruption field $F(\mathbf{x})$ (foreground layer) according to a blending factor α. This matting model was discussed previously and repeated here for clarity.

$$G(\mathbf{x}) = \alpha(\mathbf{x})F(\mathbf{x}) + (1 - \alpha(\mathbf{x}))I(\mathbf{x}) \tag{2}$$

Here, $\alpha(\mathbf{x})$ is the mixing parameter, where $\alpha = 1$ represents complete obliteration of the underlying data. In this work, dirt is assumed to be the source of the corruption; therefore, F is fixed. An RGB value of $(0, 0, 0)$ or $(255, 255, 255)$ can be used to model dark and bright blotches, respectively. This model is illustrated in Figure 2 (top row) with a synthetic example. Note that we assume

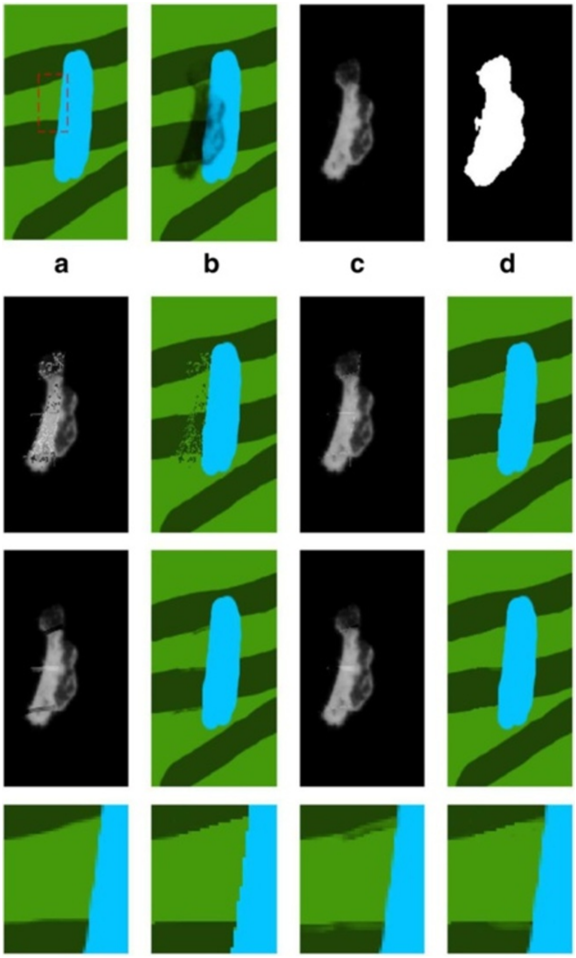

Figure 2 Problem modeling and algorithm overview. First row: Illustrating the proposed corruption model. From left: original image, corrupted image, a synthetic corruption matte α, and the corruption mask b. Second row: matte and restoration using 'Bayesian blotch matting' and 'BTBR-S'. Third row: matte and restoration using BTBR-T and spatio-temporal fusion. Last row: zoomed area of the original image **(a)** (shown in red) and its spatial **(b)**, temporal **(c)**, and spatio-temporal reconstructions **(d)**, respectively. BTBR-S generated sharp reconstruction of the blue edge, while BTBR-T generated poor reconstruction of the green edges. Nevertheless, 'spatio-temporal fusion' (BTBR-F) was able to minimize those two artifacts.

that the detection of the missing patches has already taken place, or at least some kick start is available. As stated above, we use the SDIp here. Hence, small patches of $b(\mathbf{x}) = 1$ have been delineated, and the problem then is to estimate the values of the original data I in these patches. There are, therefore, two unknown parameters at each corrupted pel:α and I; in a sense, the estimation of α amounts to a refinement of the kick start detection field $b(\mathbf{x})$. We will call $b(\cdot)$ as the the corruption mask and that indicates where α, I are subsequently estimated.

As stated previously, this model is related to the matting model of Chuang et al. [15]. Their foreground layer is our corruption layer F, and their background layer is our true original image I. α is the mixing parameter that combines both layers, and it is the *matte* in [15]. While the matting problem requires the solution of α, I and F, we know F. In addition, we have many more temporal priors that can be brought to bear on the problem.

3.1 Bayesian framework

We estimate (α,I) for each corrupted pel from the posterior $P(\alpha, I|F, G_0, ...G_M, \alpha^{\mathcal{N}}, I^{\mathcal{N}})$ (where $\mathbf{x}$ is dropped for clarity). Here, $(I^{\mathcal{N}}, \alpha^{\mathcal{N}})$ are the unknown parameters within a local neighborhood $\mathcal{N}$ from the examined site, while G_n is the nth frame of the sequence containing $M+1$ frames. The posterior is factorized in a Bayesian fashion as follows:

$$P(\alpha, I|F, G_0, ...G_M, \alpha^{\mathcal{N}}, I^{\mathcal{N}}) \propto \\ P(G_{\mathrm{u}}|\alpha, I, F) P(\alpha, I|G_0, ...G_M, \alpha^{\mathcal{N}}, I^{\mathcal{N}}, F) \tag{3}$$

Here, G_{u} denotes the frame under examination. The likelihood $P(G_{\mathrm{u}}|\alpha, I, F)$ forces the estimated (α, I) to reassemble the observed image G_{u} through $\alpha F + (1-\alpha)I$. The prior $P(\alpha, I|G_0, ...G_M, \alpha^{\mathcal{N}}, I^{\mathcal{N}}, F)$ enforces various smoothness constraints in space and time that cause the reconstructed patch to resemble the nearby clean data.

3.2 Bayesian blotch matting

By considering the process as a matting exercise, we derive our first algorithm called *Bayesian blotch matting* (BBM). Following Chuang et al. [15], the clean image prior $P(I|G_n)$ is modeled as a mixture of Gaussians. That mixture is estimated from the nearby clean data in the examined frame. This forces the reconstructed data to be consistent with these regions. Clean samples are collected by extending a circular patch $\mathcal{R}$ from the examined site until a minimum number of uncorrupted pels $\mathcal{M}_{\mathrm{u}}$ (=100 in our case) is included. The patch is then segmented into $\mathcal{M}_{\mathrm{c}}$ color clusters (four in our case) using a color quantization algorithm [45]. This yields a mixture of $\mathcal{M}_{\mathrm{c}} = 4$ Gaussians, each with mean and co-variance $(\bar{I}_j, \mathbf{R}_j)$. This color segmentation step is necessary in order to capture the richness of the examined problem.

Given observation noise $\sim \mathcal{N}(0, \sigma_e^2)$ in the compositing/observation model (see Equation 2), the ML estimate w.r.t., the jth color cluster, is then expressed as follows:

$$\begin{aligned} &P(G|\alpha_j, I_j, F) \\ &\propto \exp -\left(\frac{\|G - \alpha_j F - (1-\alpha_j) I_j\|^2}{2\sigma_e^2} \right) \cdot P(I_j|G_n) \\ &\propto \exp -\left(\frac{\|G - \alpha_j F - (1-\alpha_j) I_j\|^2}{2\sigma_e^2} \right. \\ &\qquad \left. + \frac{1}{2}(I_j - \bar{I}_j)^T \mathbf{R}_j^{-1} (I_j - \bar{I}_j) \right) \end{aligned} \tag{4}$$

We use $\sigma_e^2 = 1$ for simplicity. The likelihood ensures that the color of the data obscured by the blotch resembles the color of the surrounding patch. The last term in the expression constraining the image data to obey a particular Gaussian color is in fact a prior on that color. But because our color prior is collected from regions spatially close to the currently observed data G, one can think of the prior as being data-driven. Hence, we lump the two terms together in this likelihood expression.

3.2.1 ML estimation

Given four Gaussians in the mixture model, attempting to solve for I, α using all four at once would lead in a sense to an *average* color constraint. Instead, we follow Chuang et al. [15] and choose to solve for I, α using each color Gaussian separately. This then yields four candidate solutions of (I_j, α_j), and the candidate that maximizes the likelihood in Equation 4 is selected as the solution of the examined pel.

The ML estimate is calculated by maximizing Equation 4. We generate an estimate for I_j given α_j by taking logarithms of Equation 4, differentiating w.r.t I_j in each color component separately and setting the result to zero. Similarly, given I_j, we estimate α_j by taking logarithms of Equation 4, differentiating w.r.t α_j and setting the result to zero. Given I_j, G, F are all three-color vectors, we end up with four equations as follows (see Appendix section for derivation):

$$\left[\sigma_e^2 \mathbf{R}_{I_j}^{-1} + O_{d_3}(1-\alpha_j)^2\right] I_j = \left[\sigma_e^2 \mathbf{R}_{I_j}^{-1} \bar{I}_j + (1-\alpha_j) G\right] \tag{5}$$

$$\alpha_j = \frac{(G - I_j)^T (F - I_j)}{||F - I_j||^2} \tag{6}$$

Here, O_{d_3} is the 3 × 3 identity matrix. The (α_j, I_j) pair for the color cluster j is calculated by iterating between Equations 5 and 6 using the mean of the previously calculated opacity values (along the scan) as an initial α estimate. For each color cluster j, we use its mean and covariance in the optimization process by substituting their values in $(\bar{I}_j, \mathbf{R}_j)$ in Equation 5. Performing this optimization for each color cluster produces a set of four (α, B) candidates for each examined site. The candidate producing the highest likelihood as calculated by Equation 4 is then selected as the correct solution for the examined pel.

Figure 2 (second row, left) shows the generated corruption matte and the correction of Figure 2b using this approach. Here, Figure 2d shows the used corruption mask. Furthermore, the effect of different $(\mathcal{M}_{\mathrm{u}}, \mathcal{M}_{\mathrm{c}})$ configurations on the generated results is shown in Figure 3. As illustrated, BBM usually produces noisy results. This is because the maximum likelihood estimate does not guarantee the selection of the correct original color. This problem is solved by imposing spatial smoothness on the generated reconstruction as discussed next.

3.3 Spatial priors: Bayesian transparent blotch remover-spatial

We now modify Bayesian blotch matting by imposing spatial smoothness on α, I. We call the *Bayesian transparent blotch remover-spatial* algorithm (BTBR-S). We use eight connected MRFs in the usual way as follows:

$$P(\alpha|\alpha^{\mathcal{N}}) \propto \exp - \left(\beta_a \sum\nolimits_{k \in \mathcal{N}} \lambda_k |\alpha - \alpha_k|^2\right) \qquad (7)$$

$$P(I|I^{\mathcal{N}}) \propto \exp - \left(\beta_b \sum\nolimits_{k \in \mathcal{N}} \lambda_k ||I - I_k||^2\right) \qquad (8)$$

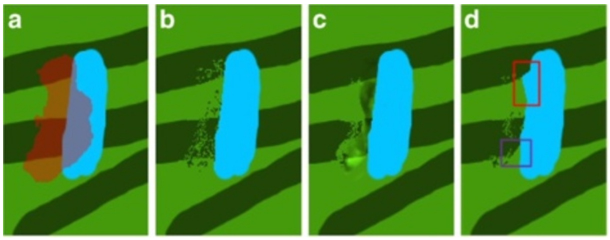

Figure 3 Clean data capturing parameters. Corruption mask in red **(a)**, BBM reconstruction with $(\boldsymbol{M}_{\mathbf{u}}, \boldsymbol{M}_{\mathbf{c}})$ = (100,4) **(b)**, (100,2) **(c)** and (20,4) **(d)**, respectively. In comparison with **(b)**: **(c)** small $\mathcal{M}_{\mathrm{c}}$ will fail to capture texture richness. The result is an averaging-like effect in the generated reconstruction. **(d)** Small $\mathcal{M}_{\mathrm{u}}$ will fail in locating the correct original color if a large portion of this color is obscured. In the red rectangle shown, the light blue leaked into the light green as most of the light green is obscured by the corruption and so was not located due to the small $\mathcal{M}_{\mathrm{c}}$ value. Similar effect is shown by the purple rectangle. The light green leaked into the dark green as most of the dark green is obscured and so was not located due to the small $\mathcal{M}_{\mathrm{c}}$ value.

Here, (β_a, β_b) are weights which configure the importance of each energy term, while λ are parameters representing the level of correlation between adjacent pels in the original image. We use the crude assumption that $\lambda = 1$ at all sites. Even though crude, the results (see Section 4.2) show accurate reconstruction of the original data even at textured regions.

3.3.1 MAP estimation

The posterior is optimized over two stages as follows:

(1) For each segmented color cluster, its corresponding (α, I) estimate is calculated by iterating between Equation 5 and Equation 6 using 0.5 as an initial opacity value. This iterative process is performed until the absolute difference between the current likelihood and the previous likelihood is small enough. A value of 0.1 is used in this work, and the likelihood here is the term shown by Equation 4. Performing this iterative process for each pel will generate a set of possible (α, I) candidates for each site.

(2) The correct (α, I) candidate for each site is selected by finding the MAP estimate. This is done by choosing between two candidates at a time using QPBO graph-cut [46,47]. The wining candidate is then processed with the next solution candidate. This process is iterated over all the remaining candidates until all candidates are examined. This iterative optimization scheme is commonly used in computer vision applications, and it is known as the 'fusion move' [48]. It is also a generalization of the commonly cited 'expansion move' technique [49]. For our problem, we define an order at which candidates are visited for examination. We call this the 'fusion order'. Our fusion order first examines the candidate generating the highest likelihood, followed by the candidate generating the second highest likelihood, then by the candidate generating the third highest likelihood, and so forth until all candidates (four in our case) are examined. Figure 4 shows this iterative optimization scheme with the fusion order that we used. Experiments show that the final reconstruction is hardly affected by any fusion order as long as all solution candidates are examined. It also shows that results often converge after examining the last solution candidate (the fourth one in our case); hence, there is no need to re-examine the candidates again. Please refer to Section 4.7 for more detail.

Figure 5a, b shows the generated corruption matte and the restoration of Figure 2b using this approach (BTBR-S) with $(\beta_a, \beta_b) = (20, 0)$. As shown, an emphasis on the opacity smoothness could lead to matte oversmoothness.

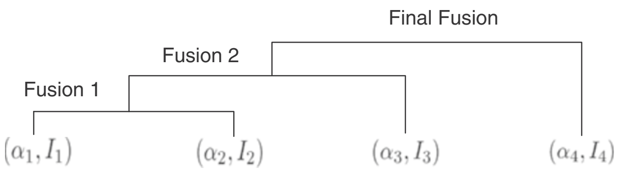

Figure 4 The process of selecting an (α, I) value form a set of possible candidates. Two candidates are processed at a time via QPBO graph-cut, and the candidate optimizing the MAP solution is selected. This process is carried iteratively over the corrupted sites and is referred to as fusion move [48]. Our technique first examines the candidate generating the highest likelihood, followed by the candidate generating the second highest likelihood, then by the candidate generating the third highest likelihood and so forth until all candidates (four in our case) are examined.

The visual impact could be severe reconstruction errors as shown in Figure 5b. Furthermore, Figure 5c shows the generated background reconstruction of Figure 2b with $(\beta_a, \beta_b) = (0, 1)$. As shown, an emphasis on the background smoothness could cause regions to bleed into its neighbors. To overcome the reconstruction errors due to inaccurate settings of these parameters, two different configurations of (β_a, β_b) are used, being $(0.01, 0)$ and $(0, 1)$. The first configuration imposes spatial smoothness on the generated mattes, while the other emphasizes background smoothness. The examined frame is divided into 16 × 16 blocks, and blocks with very low texture are assigned the configuration of $(\beta_a, \beta_b) = (0, 1)$; the rest are assigned $(0.01, 0)$. This turns off the background smoothness at textured regions to prevent possible bleeding of the regions into their neighbors. Bayesian blotch matting reconstruction is filtered by a 5 × 5 median filter. This generates a rough estimate of the underlying texture (see Figure 5d). Image gradients are then calculated using 'Roberts' edge detector where a pel is flagged as edge if its edge value exceeds 3 gray scale levels. The texture complexity of each block is then evaluated by calculating the L_0 norm of this edge map. A block is treated as textured if its L_0 norm value exceeds a threshold value of $\mathcal{T}$. This process is performed on a gray scale version I_G of the examined colored frame I. I_G is calculated using $I_G = \frac{I_r + I_g + I_b}{3}$, where $I_{r,g,b}$ are the red, green, and blue components of the examined frame I. High value of $\mathcal{T}$ may flag textured regions as untextured. This may cause reconstruction oversmoothness in textured regions as a high value of background smoothness will be assigned to these regions. To avoid this problem, we use a small value $\mathcal{T} = 2$. This value is fixed.

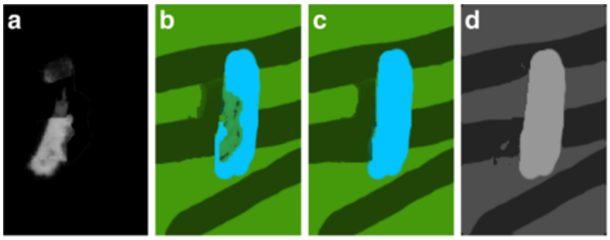

Figure 5 Reconstruction smoothness parameters. (a) Generated matte and **(b)** background restoration of Figure 2b using BTBR-S with $(\beta_a, \beta_b) = (20, 0)$. **(c)** Background restoration of Figure 2b using BTBR-S but with $(\beta_a, \beta_b) = (0, 1)$. As shown, different configurations of (β_a, β_b) lead to different restorations. **(d)** Median filtered image of the background reconstruction of Bayesian blotch matting I. In here, a 5 × 5 median filter is applied on $I_G = \frac{I_r + I_g + I_b}{3}$. This image is used to infer the texture complexity of the original image.

Figure 2 (second row, right) shows the generated matte and background reconstruction of Figure 2b using BTBR-S. In this example, the value of (β_a, β_b) = $(0.01, 0)$ is used over the whole image. As shown, BTBR-S was able to eliminate the reconstruction noise generated in the Bayesian blotch matting. This is mainly due to the incorporation of spatial smoothness on the generated results.

3.4 Temporal priors: BTBR-T

We can improve the background model $P(I|G_n)$ using information from nearby frames, especially given that the corruption does not occur in the same place in consecutive frames. This algorithm is called BTBR-T.

The obscured original data in the current frame are estimated from the previous and the next frames using a simple block matching search with a block size configured to include at least 100 uncorrupted pels. For successful block matching, the block size should encompass texture richness of the examined neighborhood. Hence, the block size is related to M_u and is therefore set to its value. This process is made robust to corruption by weighting out its effect with the opacity values of the maximum likelihood solution of BTBR-S. The result is $\hat{G}_n$, a bi-directional motion compensation of the current frame at the corrupted sites. The background prior $P(I|G_{n-1}, G_{n+1})$ is then calculated using clean samples from a 3 × 3 $\hat{G}_n$ block that is centered at the examined site. The prior is modeled as one multivariate normal distribution $\mathcal{N}(\bar{I}, \mathbf{R}_I)$ having the mean and covariance of the clean samples. This prior forces the reconstructed data to be temporally consistent with the clean data in the nearby frames. Furthermore, the small block size imposes spatial smoothness on the generated parameters as the content of these blocks usually vary smoothly from one site to the other.

3.4.1 MAP estimation

The same optimization scheme of Bayesian blotch matting is used. However, in this case, only one color cluster is used since that information is derived from the temporal prior at each site.

Figure 2 (third row, left) shows the generated corruption matte and the restoration of Figure 2b using this approach (BTBR-T). A slightly shifted image of the clean frame (by 2 pels vertically) is used as an estimate of $\hat{G}_n$. As shown, the new prior $P(I|\hat{G}_n)$ was able to restore a good portion of the original data with no reconstruction noise as in Bayesian blotch matting.

3.5 Spatio-temporal fusion: BTBR-F

The quality of the temporal solution degrades as motion gets more complicated, while the quality of the spatial solution degrades as texture becomes more complex. Figure 2 (last row) outlines this fact by comparing different reconstructions of the red region in Figure 2a. As shown, BTBR-S often leads to sharp edge reconstructions, while BTBR-T often generates errors at regions of high displaced frame difference (around green edges in this example). This calls for the generation of a final solution having minimum spatial and temporal reconstruction errors.

A spatio-temporal solution is generated by fusing the spatial and temporal solutions using the same fusion technique outlined in BTBR-S. The difference here, however, is that there are only two candidates of (α, I) at each site. Reconstruction errors are minimized by imposing spatial smoothness on the generated parameters. This is done by modeling pels as MRFs undergoing the same Gibbs energies defined in Method S (see Equation 7 and Equation 8). Moreover, the solution is biased towards temporal reconstruction at sites undergoing slow motion. This information is formulated in the prior term as follows:

$$P(\alpha, I|G_0, ...G_M, \alpha^{\mathcal{N}}, I^{\mathcal{N}}, F) \propto P(I)P(\alpha|\alpha^{\mathcal{N}})P(I|I^{\mathcal{N}}), \tag{9}$$

where

$$P(I) \propto \begin{cases} \exp-\left((|DFD|-Q)^2/2\right) & \text{for spatial candidate} \\ \exp-\beta_t\left(|DFD|^2/2\right) & \text{for temporal candidate.} \end{cases} \tag{10}$$

Here, $P(I)$ is a prior introduced to bias the solution towards the temporal candidate when there is good motion compensation between frames as measured by the DFD. Q is a constant that can be related to motion complexity and which sets it to 30 grey scale levels. This constant has the effect of favoring the temporal solution at sites of small motion errors where DFD $< \frac{Q}{2}$, while favoring the spatial solution otherwise. In addition, β_t is a parameter to configure the effect of temporal solution on the generated results. β_t is set to 1 for sequences undergoing an affine motion as, in this case, the obscured data can be easily located from nearby frames. However, we manually ignore the temporal solution in frames undergoing pathological motion. This is done by setting β_t to a very large value, $\beta_\text{t} = 10000$. In this case, an accurate value of Q is hard to configure and so the effect of the temporal solution is turned off from the reconstruction process. This stage can be carried as a post-processing step. Alternatively, regions of pathological motion can be automatically detected using Corrigan et al.'s work in [34].

Figure 2 (third row, right) shows the generated corruption matte and the restoration of Figure 2b using this approach with $(\beta_a, \beta_b) = (0.01, 0)$ and $\beta_\text{t} = 1$ for the whole image. As shown in Figure 2 (last row), the spatio-temporal BTBR-F was able to compensate between the generated errors in both the spatial and temporal reconstructions.

3.6 Modifications for line removal: BTLR

The background prior in BTBR-T is based on the assumption that corruption is temporally discontinuous; so, the clean obscured data can be well estimated from nearby frames. This assumption is violated for line removal as lines are temporally continuous events. Instead, for line removal, we can build a new background prior from nearby reconstructed frames, assuming that at some point in the past and future, the line does not appear in the original corrupted frames. A line sequence starting and terminating with two line-free frames (G_0 and G_M) is, therefore, restored temporally over three stages (see Figure 6).

(1) A solution is generated by reconstructing the first corrupted frame using BTBR-T and propagating the reconstruction in the forward time direction. The main difference here is that the reconstruction at $n-1$ is used to estimate $\hat{G}_n$ and that the first corrupted frame is motion-compensated using the line-free frame G_0. This will be referred to as 'recursive forward reconstruction'.
(2) A similar solution is generated but by starting the reconstruction from the last corrupted frame and propagating the solution in the backward direction. The main difference here is that $\hat{G}_n$ is estimated from the reconstruction at time $n+1$ and that the last corrupted frame G_{M-1} is motion-compensated using G_M. The resulting reconstruction will be referred to as 'recursive backward-time reconstruction'.
(3) An overall temporal reconstruction is generated by fusing the forward and the backward reconstructions using QPBO graph-cut as in BTBR-S. In here, the background smoothness is emphasized relative to opacity smoothness as the small line width will prevent background oversmoothness. The resulting reconstruction will be referred to as 'bidirectional-time reconstruction'.

Figure 6 illustrates the proposed temporal line removal algorithm. As shown, each method accumulates

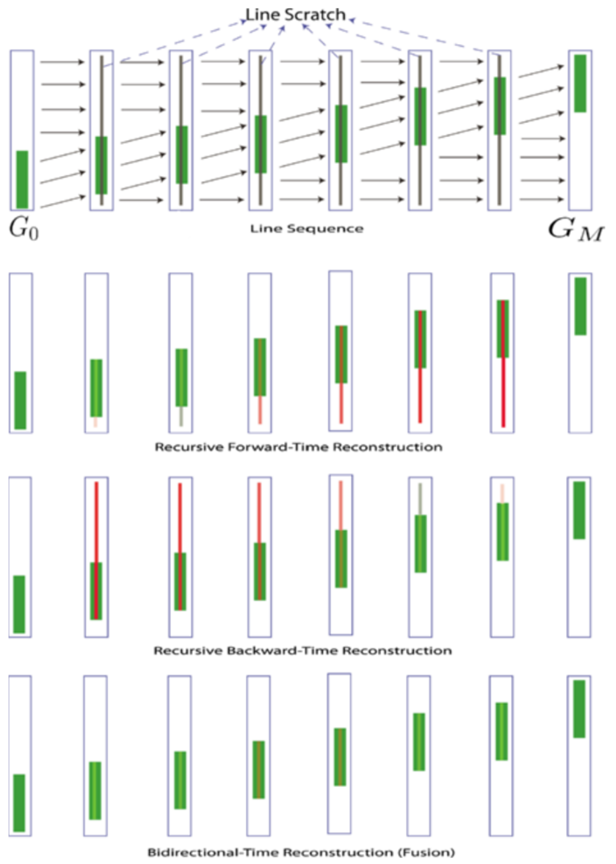
Line Scratch
G_0
Line Sequence
G_M
Recursive Forward-Time Reconstruction
Recursive Backward-Time Reconstruction
Bidirectional-Time Reconstruction (Fusion)

Figure 6 Temporal restoration of a line sequence containing M corrupted frames. A green object is moving through time, and estimated motion vectors are shown in black arrows. Reconstruction errors are represented with different degrees of 'red' ranging from 'light' (for small errors) to 'dark' (for large errors). As shown, bidirectional fusion is designed to minimize reconstruction errors in all frames.

reconstruction errors in the direction of the propagation. However, the bidirectional-time fusion is expected to minimize these errors in all frames. A final spatio-temporal solution can then be generated by fusing the temporal and the spatial solutions using the same framework of Section 3.5. For all examined sequences, we perform temporal propagation over 30 frames. Increasing the number of frames would be expected to improve reconstruction because, in theory, objects would then have more time to move away from the degradation. However, this comes with an increased chance of motion estimation errors. Therefore, this is a downfall which can only be addressed in the field as it works. In fact, as our algorithm operates offline, we can take advantage of as many frames as is available.

4 Results

Image sequences used in this work can be found in www.sigmedia.tv/Misc/TIPS2011. Examples of frames from these sequences are shown in Figures 7 and 8. Four standard definition (720 × 576) sequences are used to evaluate the performance of the blotch removal processes: LabB1, ArtB1, DanceB1 and DanceB2 with 100, 40, 100 and 60 frames, respectively. All sequences undergo fast motion and contain moderate texture. LabB1 is created by corrupting a clean sequence heavily using the relation $G(\mathbf{x}) = (1 - \alpha(\mathbf{x}))I(\mathbf{x})$. Here, α is the dirt opacity obtained from the IR scans. The other three sequences show real archived footage containing blotches. Similarly, four line sequences are used to evaluate the performance of line scratch removal: LabL1, DanceL1, DanceL2 and DanceL3, with 25, 25, 18 and 15 frames, respectively. Again, these show fast motion and contain moderate texture. LabL1 shows synthetic line scratches created in the same way as LabB1, and the others contain real line scratches. In all experiments, the value of β_t is set to 1.

There are a large variety of blotch remover processes that have been proposed; hence, we compare our results with the JONDI estimator of Kokaram [2], since that is the most general of the frameworks proposed in the past and is the basis of many commercial blotch removal processes. For line removal, we compare the results against JOMBEI which is a spatial version of JOMBADI for line removal [24]. We also compare with one commercially available software suite called Furnace from www.thefoundry.co.uk. To illustrate the importance of the opacity term in generating accurate reconstructions, we compare the results against an implementation of BTBR-F which removes the effect of the opacity term, i.e., α is replaced by a binary index. This can be regarded as an implementation of image inpainting [38,39]. We call this implementation BTBR-AOFF.

Evaluation of blotch removal techniques is traditionally difficult because of the lack of ground truth data. Ground truth is traditionally hard to come by since it can only generated by painstaking hand painting of missing patches. In this case, a model that considers semi-transparency is even more difficult. However, we have acquired IR scans of film material, which yield ground truth on real degraded sequences. These IR scans can also be used to synthetically corrupt known clean sequences. We can therefore measure performance in realistic situations. The next section briefly discusses how ground truth is acquired and then we go on to discuss algorithm performance over a variety of different material.

4.1 Ground truth acquisition

Corrupted sites are detected using a simple threshold operation on the IR scans of that material. Values of 210, 180 and 210 are used for the Art, Dance and Lady-Doll sequences, respectively, where a pel is flagged as

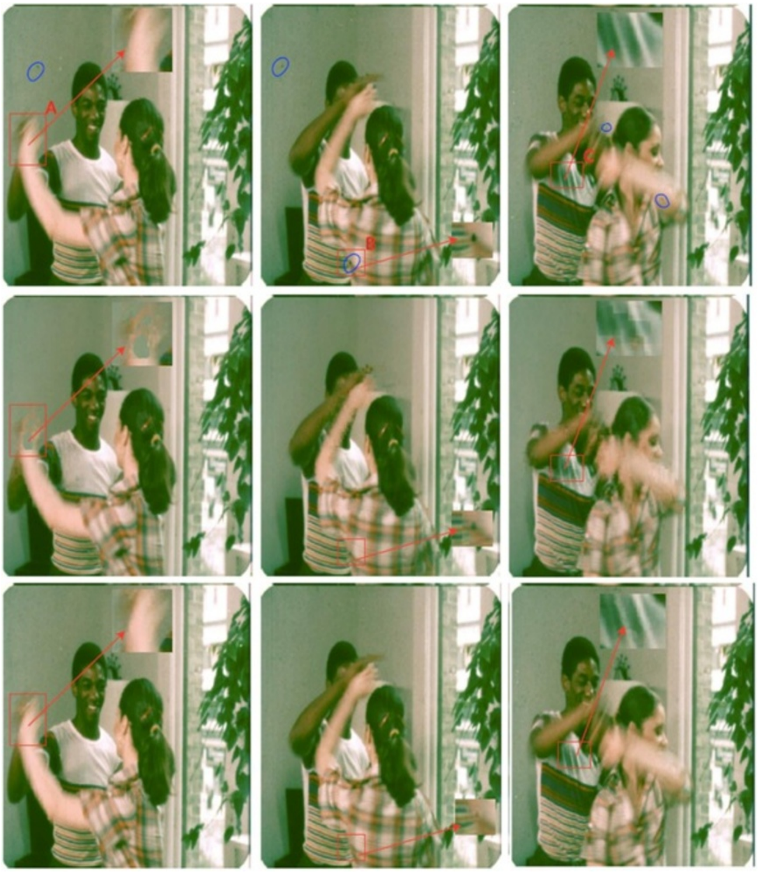

Figure 7 Frames 66, 70 and 77 of DanceB1 (top) with their reconstructions using JONDI (middle) and BTBR-F (bottom). Both systems perform well in removing blotches (shown in blue); however, JONDI could fail in removing blotches that lie near dark regions (box B). In addition, JONDI often misclassifies clean regions as corrupted. This could lead to severe reconstruction errors as shown in boxes A and C. On the other hand, BTBR-F classifies clean regions as uncorrupted, therefore disregard them from reconstruction.

corrupted if it falls below the threshold. This thresholding operation yields the ground truth corruption mask b_g for each sequence. We extract the ground truth opacities α_g for each sequence by relating the actual grayscale value of the IR mask R to the estimated value of alpha under the mask. Estimates for alpha under the mask are generated using BTBR-F. Figure 9 shows the IR/corruption opacity plot for the Dance sequence. IR scans are spread over the range $0 - -1$ for the simplicity of illustration, where 1 denotes a highly corrupted region. Three different fitting functions are superimposed: $y = x^n$, $y = a.x^2+b.x+c$, and $y = \gamma.exp(\lambda x) + k$. The quadratic function gives the best fit for both sequences; hence, we use that to transform IR greyscale values into ground truth opacities.

Figure 10 shows some blotches, their IR scans, the calculated dirt opacities using the derived IR/corruption relation, and the corresponding original data reconstruction. The function $y = a.x^2 + b.x + c$ is used as the IR/corruption relation, and reconstruction is achieved by inverting the effect of the dirt using the matting equation directly.

4.1.1 Image reconstruction fidelity

The structure similarity measure (SSIM) [50] is used to measure the reconstruction quality of the examined techniques. This is achieved by comparing the generated reconstruction against a ground truth estimate. In the case of the artificially corrupted sequences, the clean sequence is available. In the case of the real archived sequences, the IR scans are available, and those can be used as discussed above to generate the ground truth reconstruction. Comparison is only performed on sites classified as corrupted during the detection step, since only these sites will be affected by subsequent processing, and this limited measurement would emphasize the difference between different processed.

4.2 Reconstruction quality

Table 1 compares the reconstruction quality for blotch removal with ground truth for all the sequences, artificial (LabB1) and real. SSIM [50] is used here, where an SSIM of 1 means that the sequence is identical to the ground truth reconstruction, while SSIM $= 0$ implies

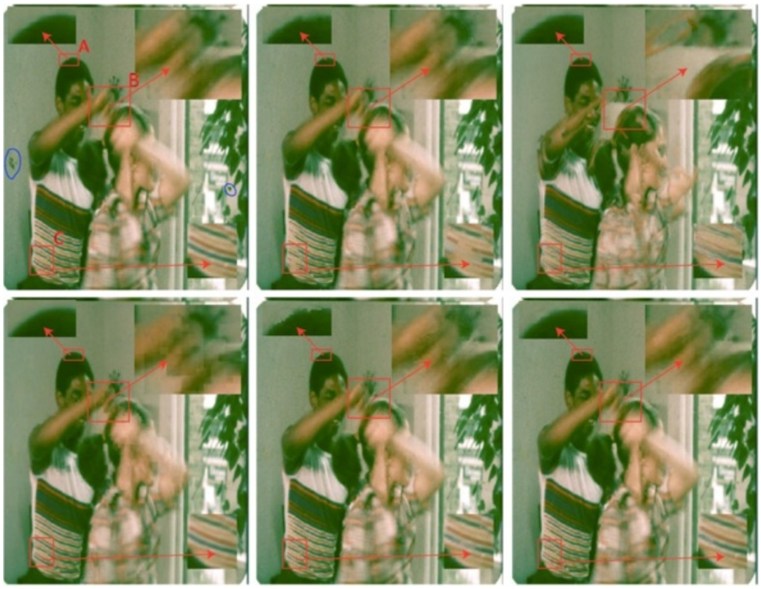

Figure 8 BTBR-AOFF, Furnace, JONDI, BBM, and BTBR-F reconstructions. First row, from left: frame 76 of DanceB1 with its BTBR-AOFF and Furnace reconstructions, respectively. Bottom: reconstructions using (from left) JONDI, BBM, and BTBR-F. As shown, all techniques removed blotches successfully (shown in blue). However, BTBR-F is the best in preserving clean regions (see red boxes). This is mainly due to the incorporation of the opacity term α which calculates the level of corruption accurately and disregards clean regions from the reconstruction process. Full image sequence results are in www.sigmedia.tv/Misc/TIPS2011.

that it is completely different. As can be seen, the average SSIM indicates that BTBR-F and JONDI are the best performing systems. However, the minimum SSIM indicates that JONDI performs worse than BTBR-F during their failure. This is shown in Figure 7. We can see from that the reason JONDI fails is typically because of poor motion information, while BTBR-F is able to recover from this by the opacity term which turns off reconstruction in these regions. Furthermore, JONDI fails to remove opaque blotches that lie near dark regions (see red box B). This makes sense since JONDI at its heart performs a cut-and-paste operation combined with temporal frame averaging in some sense. Hence, the edges of the semi-transparent blotches are often visible after reconstruction. BTBR-F on the other hand models the corruption as a semi-transparent layer, therefore generates an accurate estimate of the regions of corruption. Figure 8 shows the performance of the other techniques. Its worth noting that even though all techniques perform well in removing blotches, they differ in their ability in preserving clean regions. BTBR and BBM preserve clean regions due to the opacity term; however, BTBR generates smoother results due to the incorporation of the background smoothness prior (see red box A). BTBR-AOFF shows that by turning off the opacity term, the system is no longer able to classify clean regions as uncorrupted. This could led to severe reconstruction errors (box C). JONDI generates blocky artifacts; however, it maintains regions of high texture (box C). Last, the Furnace generates severe errors especially in regions of pathological motion (see box B). Similar results are obtained for greyscale sequences, but they are not shown here for the lack of space.

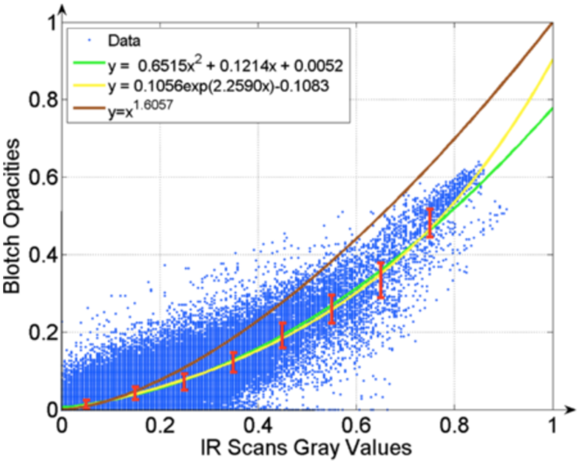

Figure 9 IR values vs. corruption opacities α for the dance sequence. The red bars denote 1 standard deviation of α. Both $y = a.x^2 + b.x + c$ and $y = \gamma.exp(\lambda x) + k$ produce a good fit of the IR/corruption opacity relation. Results for only one sequence are shown here in the interest of brevity.

Figure 11 shows examples where BTBR fails in LabB1. Here, BTBR generates visual artifacts due to background reconstruction oversmoothness. As a result JONDI

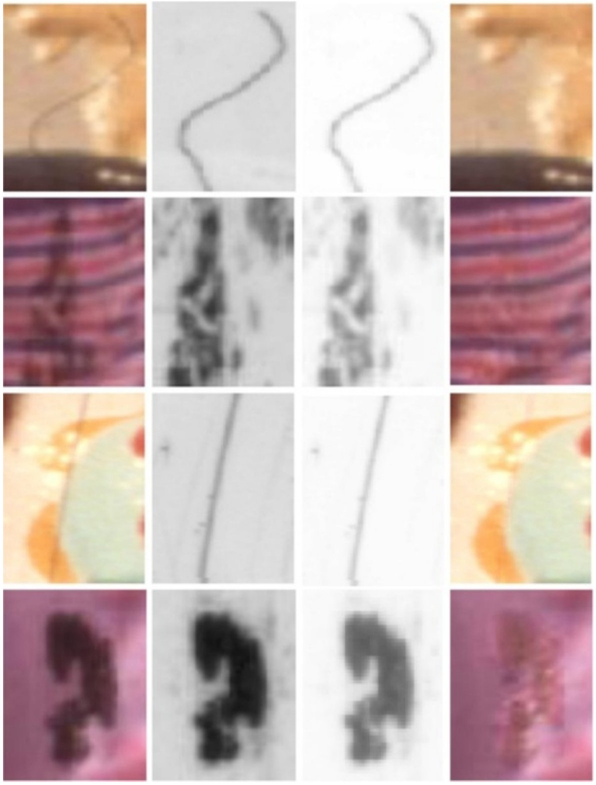

Figure 10 Corrupted image, IR scan, dirt opacity obtained using the derived IR/corruption relation, and corresponding reconstruction. They correspond to the images from left to right. The corruption opacity field is inverted for the simplicity of comparison with the IR. As shown, the reconstruction successfully recovers the underlying original data; however, it fails to remove highly opaque corruptions (last row).

outperformed BTBR in some regions in this sequence (LabB1). This observation is recorded in Table 1 by the slightly better performance of JONDI over BTBR. However, only six of such artifacts were generated, each covering a region of no more than 20 × 20 pels. Hence, both JONDI and BTBR reconstructions still look qualitatively the same in the vast majority of this examined sequence.

Table 2 shows reconstruction SSIM for the line scratch sequences. We can see from the average and minimum SSIM that BTLR is the best performing technique. Figure 12 shows three frames of LabL1 and its BBM and BTLR reconstruction. It is clear that BBM generates noisy results due to the absence of spatial reconstruction smoothness (shown in blue). This noise manifests as flickering during video playback. Figure 13 shows comparison against other techniques. It is evident from the figure that BTLR outperforms all the other techniques. Even though Table 2 shows that JOMBEI and BTLR are of near performance, JOMBEI often blurs the corruption (see blue box A and C). This makes sense as JOMBEI uses spatial filtering in a way. BTBR-AOFF generates artifacts in clean regions due to the absence of the opacity term (box D). Furnace generates incomplete removal, and BBM generates noisy reconstruction as expected (box B). Similar results are obtained for greyscale sequences, but they are not shown here for the interest of brevity.

Table 1 Reconstruction quality against ground truth for the examined blotch sequences

System	LabB1	DanceB1	DanceB2	ArtB1
Furnace				
Average	0.76 ± 0.08	0.83 ± 0.04	0.80 ± 0.06	0.77 ± 0.04
Minimum	0.44	0.64	0.64	0.69
Maximum	0.93	0.98	0.89	0.98
JONDI				
Average	0.98 ± 0.01	0.94 ± 0.01	0.90 ± 0.02	0.98 ± 0.00
Minimum	0.94	0.89	0.84	0.96
Maximum	0.99	0.99	0.94	0.99
BTBR-AOFF				
Average	0.84 ± 0.07	0.90 ± 0.02	0.90 ± 0.04	0.88 ± 0.04
Minimum	0.70	0.84	0.78	0.82
Maximum	0.94	0.94	0.95	0.95
BBM				
Average	0.89 ± 0.03	0.94 ± 0.01	0.94 ± 0.02	0.94 ± 0.01
Minimum	0.82	0.91	0.86	0.93
Maximum	0.95	0.96	0.98	0.97
BTBR-F				
Average	0.97 ± 0.01	0.96 ± 0.01	0.95 ± 0.02	0.98 ± 0.01
Minimum	0.93	0.93	0.89	0.97
Maximum	0.99	0.98	0.98	0.99

SSIM [50] is used here where an SSIM of 1 means that the sequence is identical to the ground truth reconstruction. The average and the minimum SSIM show that BTBR-F is the best performing system.

4.3 Evaluating detection refinement

Recall that SDIp is used as a kick start detection for BTBR. This detection is refined through α estimation.

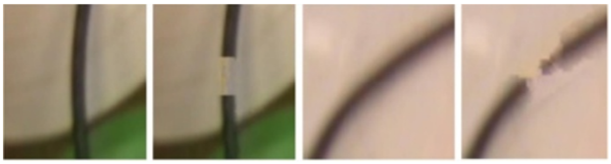

Figure 11 Example of artifacts generated by BTBR. Each example shows the original image on the left and the reconstruction on the right. Such artifacts are generated due to background reconstruction oversmoothness.

Table 2 Reconstruction quality against ground truth for the examined line sequences

System	LabL1	DanceL1	DanceL2	DanceL3
Furnace				
Average	0.92±0.04	0.85±0.05	0.88±0.03	0.92±0.03
Minimum	0.85	0.75	0.83	0.85
Maximum	0.97	0.92	0.93	0.95
JOMBEI				
Average	0.95±0.01	0.92±0.02	0.91±0.02	0.95±0.02
Minimum	0.93	0.88	0.87	0.89
Maximum	0.97	0.96	0.96	0.96
BTBR-AOFF				
Average	0.94±0.02	0.90±0.03	0.91±0.03	0.92±0.03
Minimum	0.87	0.85	0.88	0.85
Maximum	0.97	0.96	0.96	0.96
BBM				
Average	0.94±0.03	0.91±0.02	0.90±0.02	0.92±0.02
Minimum	0.88	0.85	0.85	0.89
Maximum	0.98	0.84	0.9	0.94
BTLR				
Average	0.96±0.02	0.92±0.02	0.94±0.02	0.95±0.02
Minimum	0.93	0.87	0.90	0.90
Maximum	0.98	0.96	0.96	0.96

SSIM [50] is used here where an SSIM of 1 means that the sequence is identical to the ground truth reconstruction. The average and the minimum SSIM show that BTLR is the best performing technique.

Figure 14 shows ROC plots of BTBR and SDIp on processing DanceB1. ROC is calculated at 8 SDIp thresholds being 5:2.5:22.5 (MATLAB notations). Figure 14 shows that BTBR outperforms SDIp through significant reduction in false detection while nearly maintaining the correct detection rate. For example, in the SDIp threshold of 22.5 (last black marker), BTBR was able to reduce false detection by 0.1 while reducing correct detection by just 0.02. A significant false detection reduction for blotches is 0.1 is a significant cant false detection reduction for blotches.

4.4 Luminance vs. RGB reconstruction accuracy

Two hundred fifty highly textured colored frames of size 576 × 720 pels are corrupted by artificial opacities. Frames are restored in both the luminance and the RGB spaces using BBM. Mean absolute error between the ground truth opacities and the estimated opacities are measured as 0.0548 and 0.0429 for the luminance and RGB reconstruction, respectively. Opacity estimation in RGB space is more accurate than that in the luminance space as RGB reconstruction exploits all color information. To examine the effect of the estimated opacities on background reconstruction, the structure similarity index (SSIM) between the clean sequence and the reconstructed original data was measured and found to be 0.884 and 0.921 for luminance and RGB reconstructions, respectively. Here, SSIM for the RGB reconstruction is carried out in the luminance channel. Results show that the background reconstruction in the RGB space is more accurate than in the luminance space.

4.5 Computational complexity

Estimation of the set of solution candidates for each pel is the most computationally intensive aspect of the algorithm, amounting to about 90% of the execution time for a single frame. The estimation of these candidates is an iterative process which terminates when convergence is reached and so consumes a lot of time. This process is repeated K times, where K is the number of solution candidates per pel. Therefore, for a corruption mask of N pels, the number of operations required to generate the solution candidates is $N \times K \times P$, where P is the average number of iterations to generate one solution candidate. The second most computationally intensive part of the algorithm is the use of QPBO graph-cut for optimizing the MAP function.

The average time for processing one colored standard definition frame with BTBR on a 2.33 GHz Quad Core Processor and with an unoptimized MATLAB code is 25.4 s. This is taken as the average of processing 50 frames with SDIp masks of threshold 7.5 and with 5 solution candidates. QPBO is implemented efficiently by using C++ and by processing only the regions of SDIp masks. The average time to process one colored standard definition frame with JONDI written in an optimized C++ code is around 10 s. Hence, BTBR is nearly as fast as JONDI. Further reduction in computation for BTCR is possible by reducing the size of the corruption mask and reducing the number of solution candidates. By reducing the number of solution candidates from 5 to 3, the the average time to process one frame dropped from 24.5 to 16.7 s. This however has a direct impact on the reconstruction quality especially in textured regions.

4.6 A note on system parameters tuning

All system parameters were fixed by simple tweaking, basically after examining just one frame, except the motion complexity threshold Q of Equation 10 and the amount of background/opacity reconstruction smoothness (see (β_a, β_s) in Equations 7 to 8). Q was fixed after examining seven frames. Here, three candidates of Q were examined being 15, 30 and 50. The value of 15 often pushed the solution towards the spatial candidate, while the value of 50 often pushed the solution towards the temporal candidate. However, it was clear that the value of 30 was able to combine both spatial and temporal solutions in a way that minimizes reconstruction error.

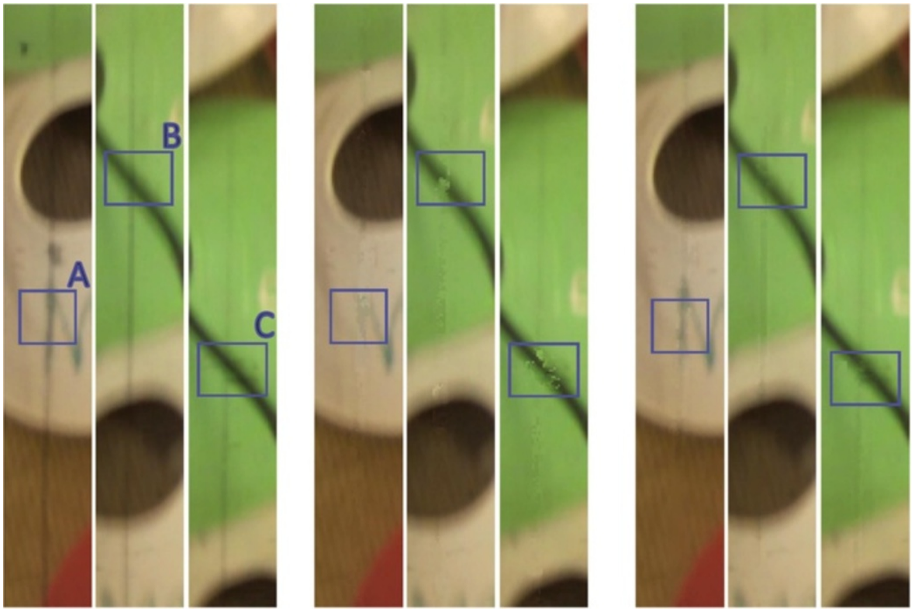

Figure 12 Frames 18, 23 and 28 of LabL1 and their BBM and BTLR reconstructions. They correspond to the images from left to right, respectively. BTLR often generates smoother results than BBM due to the incorporation of a reconstruction smoothness term. The noisy restoration of BBM manifests as flickering during video playback.

We have noticed through experimental observations that different values for the background/opacity reconstruction smoothness often generate different results. Hence, in order to fix the reconstruction smoothness levels, we had to first examine two sequence, LabB1 and ArtB1. The former sequence contains strong texture, while the latter does not. As a result, we generated a solution that chooses between two different smoothness values, one when the examined region contains a strong texture, and the other when it does not. The smoothness values were then fixed through the remaining sequences, and results outperformed existing techniques.

It can be concluded form the above discussion that even though most of the fixed parameters were trained on few frames (one in most cases), they were still able to perform well on all the examined sequences. Hence, we expect the fixed parameter values to perform similarly well on new sequences. We also expect the Q value of 30 to perform well since it showed significant improvement over the other examined values of 15 and 50. Lastly, the fixed values of reconstruction smoothness are also expected to perform well since they generated good results on all the examined sequences (eight in total), even though they were only trained on two sequences. However, a better tuning of the reconstruction smoothness parameters could improve results. This is particularity important in regions where it is hard to calculate the strength of the texture as regions covered by large corruptions.

4.7 A note on the fusion order

Figure 15 shows how the error of the MAP estimate changes as the number of QPBO graph-cut iterations increases. Here, the MAP error is the negative log of Equation 3 divided by the number of pairwise interactions of all the examined pels of the examined sequence. For Figure 15, we examined five different fusion orders. The orders are written between brackets on the top right corners of the figures. The solution candidates are labeled by a number from 1 to 4. Here, label 1 denotes the solution candidate generating the highest likelihood, while label 4 denotes the solution candidate generating the lowest likelihood. Hence, fusion order $(1, 2, 3, 4)$ means that it first examines the candidate generating the highest likelihood, followed by the candidate generating the second highest likelihood, then by the candidate generating the third highest likelihood, and finally followed by the fourth and last candidate. Since four solution candidates exist per pixel, all candidates are examined by the end of the third graph-cut iteration. We kept re-examining the candidates by performing more iterations using the same fusion order so that we can examine the convergence of our approach. Figure 15 shows that all the examined orders eventually converge to the same solution after the third graph-cut iteration. Hence, there is no need to run further iterations since it will mainly just increase the computational load. In addition, Figure 15 shows that different fusion orders eventually lead to the generation of similar results by converging to similar MAP values.

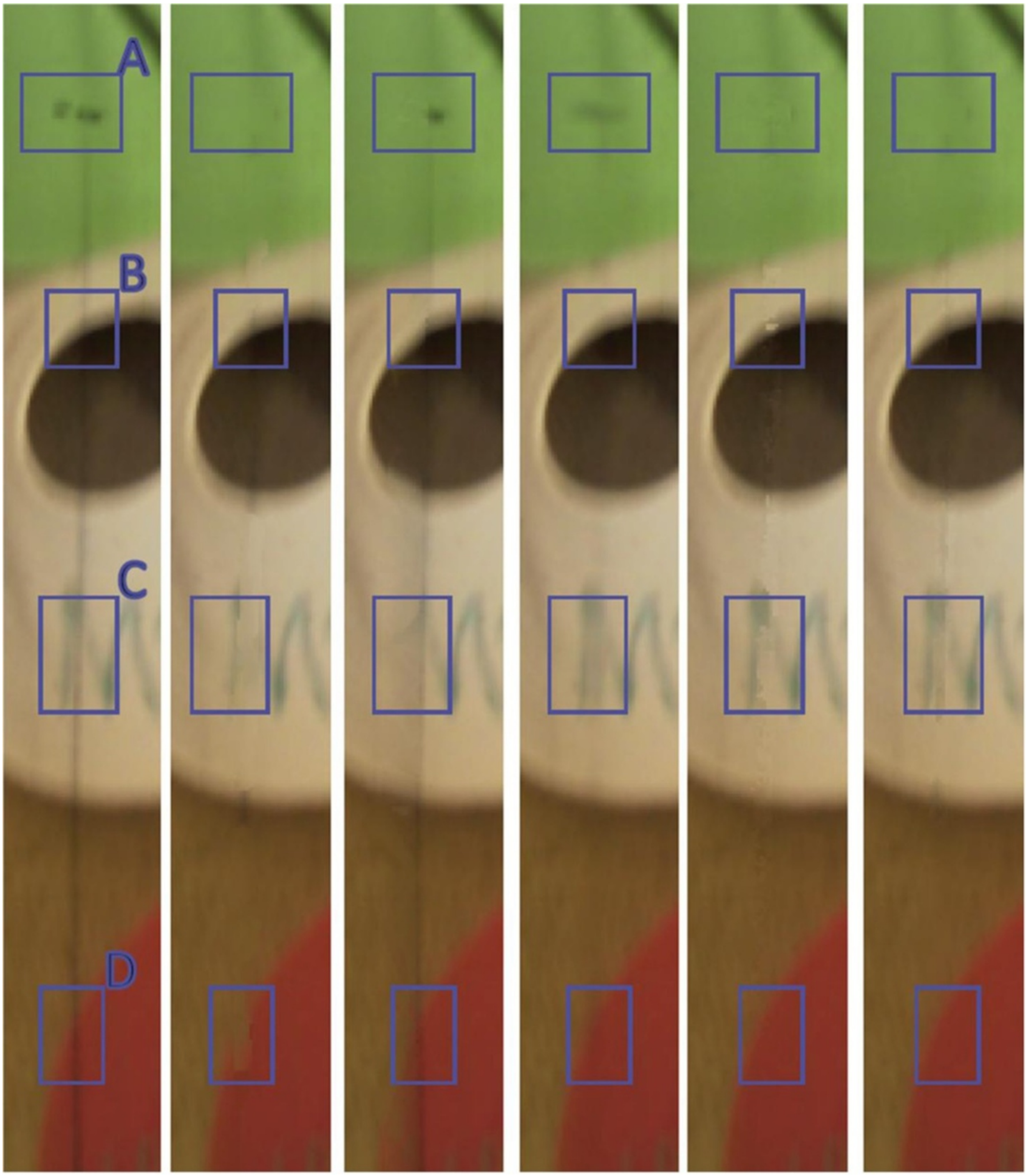

Figure 13 Frame 19 of LabL1 and its restoration using BTBR-AOFF, Furnace, JOMBEI, BBM, and BTLR. The original image is on the far left; the other images correspond to systems mentioned, respectively. As shown BTLR generated the best restoration. JOMBEI often blurs the corruption (box A), BTBR-AOFF corrupts clean regions (boxes C and D), Furnace produces incomplete removal, and BBM generates noisy reconstruction (box B). Full image sequence results are in www.sigmedia.tv/Misc/TIPS2011.

5 Conclusion

This paper has presented a new framework for removing dirt and lines from image sequences. It addresses the issue of incomplete blotch removal from image sequences when traditional blotch removers are used. It also allows the removal of semi-transparent damage in general from

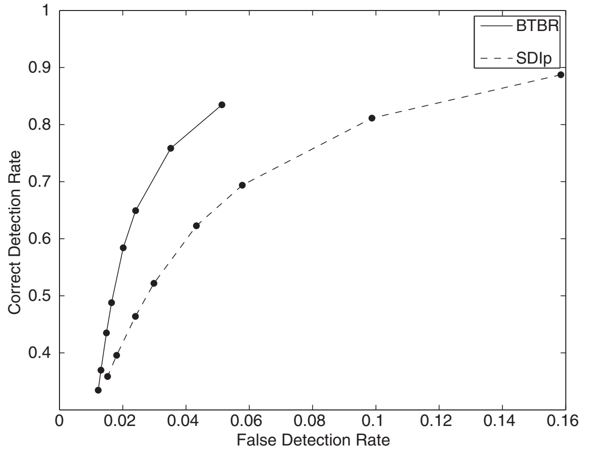

Figure 14 ROC of BTBR and SDIp on processing DanceB1. The black circled markers denote the different SDIp thresholds used in the ROC evaluation.

film sequences. The novelty here is in using a corruption model which explicitly generates a semi-transparent layer. Corruption removal is then addressed as the problem of separating the dirt layer from the original background layer through a variant of the matting problem. A Bayesian framework was presented exploiting both spatial and temporal priors. The algorithm is initialized with rough binary corruption masks which are refined into a non-binary opacity mattes. These mattes estimate the amount of dirt at each pixel and, therefore, disregards clean regions from the correction process.

The second contribution of this paper is in presenting a technique for generating ground truth estimates of the original data. A relation between dirt opacity and its IR scan is derived. This relation is then used to estimate the amount of dirt represented by IR scans through the means of a non-binary opacity matte. Results showed that the original underlying data can be estimated by inverting the effect of the this matte. The estimated data are a near ground-truth estimate of the original data.

We compared the performance of our corruption removal techniques against four blotch removal techniques (Furnace, BTBR-AOFF, JONDI and BBM) and four line removal techniques (Furnace, BTBR-AOFF, JOMEI, and BBM). Reconstruction quality was evaluated against the ground-truth estimates generated from IR scans. Results showed that our techniques generate better reconstruction over all the examined blotch and line removal techniques. This confirms that our dark corruption model is valid enough. In particular, the BTBR algorithms can remove the extremities of blotches very well, in comparison to the cut-and-paste operators used currently. Furthermore, because of the *soft* corruption model, we are able to bring more robustness to the PM problem without actually detecting PM explicitly [34-37]. Of course, a novel practical system would employ a PM detector with our BTBR to yield industrial strength performance.

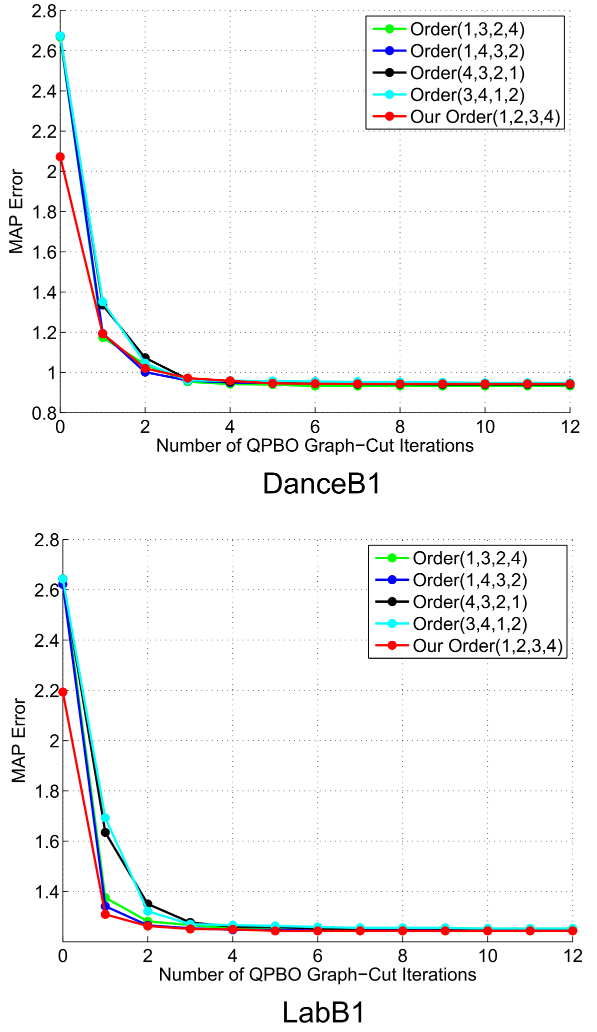

Figure 15 MAP error for DanceB1 and LabB1 as the number of QPBO graph-cut iterations increases. Five different fusion orders are examined. By the end of the third iteration, all solution candidates are examined once. It is clear from the graphs that there is little gain from re-examining the same solution candidates after the third iteration. In addition, all examined fusion orders eventually converge to similar solutions.

A limiting factor of our technique is the inability of estimating the exact required amount of background smoothness for perfect reconstruction. This sometimes has an impact on the reconstruction quality as too much smoothness would cause neighboring regions to interfere with each other, while too little smoothness could lead to incomplete removal. It is clear that we could incorporate our model directly into the detection/reconstruction problem along the lines of JOMBADI [6]. That would imply the estimation of the texture parameters alongside detection and motion information. Although this might seem a daunting task, it provides much potential for future work.

6 Consent

Written informed consent was obtained from the patient for the publication of this report and any accompanying images.

Appendix

Derivation of the maximum likelihood estimate of BBM

Given observation noise $\sim \mathcal{N}(0, \sigma_e^2)$ in the compositing/observation model (see Equation 2), the ML estimate w.r.t. the *j*th color cluster is then expressed as follows:

$$P(G|\alpha_j, I_j, F) \propto \exp - \left(\frac{\| G - \alpha_j F - (1-\alpha_j) I_j \|^2}{2\sigma_e^2} + \frac{1}{2}(I_j - \bar{I}_j)^T \mathbf{R}_j^{-1} (I_j - \bar{I}_j) \right) \quad (11)$$

Here, given that *I*, *G* and *F* are the clean, observed, and dirt layers, respectively. $(\bar{I}_j, \mathbf{R}_j)$ is the mean and covariance of the *j*th background color cluster (see Section 3.2, first two paragraphs for more detail). Calculating the ML estimate is equivalent to minimizing the exponent of Equation 11. Hence, given that I_j, *G*, *F* are all three-color vectors, taking the logarithm of Equation 11 generates

$$\begin{aligned} E(\alpha, I) = {} & \frac{(G_\mathrm{r} - \alpha F_\mathrm{r} - (1-\alpha) I_\mathrm{r})^2}{2\sigma_e^2} \\ & + \frac{(G_\mathrm{g} - \alpha F_\mathrm{g} - (1-\alpha) I_\mathrm{g})^2}{2\sigma_e^2} \\ & + \frac{(G_\mathrm{b} - \alpha F_\mathrm{b} - (1-\alpha) I_\mathrm{b})^2}{2\sigma_e^2} \\ & + \frac{1}{2}\left((I_\mathrm{r} - \overline{I}_\mathrm{r}) z_{11} + (I_\mathrm{g} - \overline{I}_\mathrm{g}) z_{21} + (I_\mathrm{b} - \overline{I}_\mathrm{b}) z_{31} \right) \\ & \times (I_\mathrm{r} - \overline{I}_\mathrm{r}) + \frac{1}{2}\left((I_\mathrm{r} - \overline{I}_\mathrm{r}) z_{12} + (I_\mathrm{g} - \overline{I}_\mathrm{g}) z_{22} \right. \\ & \left. + (I_\mathrm{b} - \overline{I}_\mathrm{b}) z_{32} \right) (I_\mathrm{g} - \overline{I}_\mathrm{g}) + \frac{1}{2}\left((I_\mathrm{r} - \overline{I}_\mathrm{r}) z_{13} \right. \\ & \left. + (I_\mathrm{g} - \overline{I}_\mathrm{g}) z_{23} + (I_\mathrm{b} - \overline{I}_\mathrm{b}) z_{33} \right) (I_\mathrm{b} - \overline{I}_\mathrm{b}) \end{aligned} \quad (12)$$

Here, $(I_\mathrm{r}, I_\mathrm{g}, I_\mathrm{b})$, $(G_\mathrm{r}, G_\mathrm{g}, G_\mathrm{b})$, and $(F_\mathrm{r}, F_\mathrm{g}, F_\mathrm{b})$ denote the RGB components of *I*, *G*, and *F*, respectively. $z_{m,n}$ denotes the *m*-row *n*-column element of $\mathbf{R}^{-1}$. We dropped the *j* cluster index for clarity.

Solving for I

We solve for each color component of *I* separately. To solve for I_r, we calculate the derivative of Equation 12 w.r.t I_r and equate the result to zero as follows:

$$\begin{aligned} \frac{\partial E}{\partial I_\mathrm{r}} = {} & \frac{-2(G_\mathrm{r} - \alpha F_\mathrm{r} - (1-\alpha) I_\mathrm{r})(1-\alpha)}{2\sigma_e^2} + \frac{2 z_{11}(I_\mathrm{r} - \overline{I}_\mathrm{r})}{2} \\ & + \frac{(z_{12} + z_{21})(I_\mathrm{g} - \overline{I}_\mathrm{g})}{2} + \frac{(z_{13} + z_{31})(I_\mathrm{b} - \overline{I}_\mathrm{b})}{2} \end{aligned} \quad (13)$$

Setting $F = 0$ to handle dark blotches and using the fact that $\mathbf{R}^{-1}$ is symmetric, we get

$$\begin{aligned} \frac{\partial E}{\partial I_\mathrm{r}} = {} & (1-\alpha)^2 I_\mathrm{r}/\sigma_e^2 - (1-\alpha) G_\mathrm{r}/\sigma_e^2 + z_{11}(I_\mathrm{r} - \overline{I}_\mathrm{r}) \\ & + z_{12}(I_\mathrm{g} - \overline{I}_\mathrm{g}) + z_{13}(I_\mathrm{b} - \overline{I}_\mathrm{b}) \end{aligned} \quad (14)$$

Equating $\frac{\partial E}{\partial I_\mathrm{r}}$ to 0 generates

$$\begin{aligned} z_{11} I_\mathrm{r} + z_{12} I_\mathrm{g} + z_{13} I_\mathrm{b} + (1-\alpha)^2 I_\mathrm{r}/\sigma_e^2 = {} & z_{11} \overline{I}_\mathrm{r} \\ + z_{12} \overline{I}_\mathrm{g} + z_{13} \overline{I}_\mathrm{b} + (1-\alpha) G_\mathrm{r}/\sigma_e^2 & \end{aligned} \quad (15)$$

I_g and I_b are solved using the same approach by calculating $\frac{\partial E}{\partial I_\mathrm{g}}$ and $\frac{\partial E}{\partial I_\mathrm{b}}$, and equating the result to zero. Performing the same steps of Equations 13 to 15 to I_g and I_b generates

$$\begin{aligned} z_{21} I_\mathrm{r} + z_{22} I_\mathrm{g} + z_{23} I_\mathrm{b} + (1-\alpha)^2 I_\mathrm{g}/\sigma_e^2 = {} & z_{21} \overline{I}_\mathrm{r} \\ + z_{22} \overline{I}_\mathrm{g} + z_{23} \overline{I}_\mathrm{b} + (1-\alpha) G_\mathrm{g}/\sigma_e^2 & \end{aligned} \quad (16)$$

$$\begin{aligned} z_{31} I_\mathrm{r} + z_{32} I_\mathrm{g} + z_{33} I_\mathrm{b} + (1-\alpha)^2 I_\mathrm{b}/\sigma_e^2 = {} & z_{31} \overline{I}_\mathrm{r} \\ + z_{32} \overline{I}_\mathrm{g} + z_{33} \overline{I}_\mathrm{b} + (1-\alpha) G_\mathrm{b}/\sigma_e^2 & \end{aligned} \quad (17)$$

Equations 16 to 17 can also be obtained by direct analogy with Equation 15. We finally group Equations 15 to 17 together in one matrix operation as follows:

$$\left[\sigma_e^2 \mathbf{R}_{I_j}^{-1} + O_{d_3}(1-\alpha_j)^2 \right] I_j = \left[\sigma_e^2 \mathbf{R}_{I_j}^{-1} \bar{I}_j + (1-\alpha_j) G \right] \quad (18)$$

This is the same as Equation 5, where O_{d_3} is the 3 × 3 identity matrix.

Solving for α

To solve for α, we calculate the derivative of Equation 12 w.r.t α and equate the result to zero as follows:

$$\begin{aligned} \frac{\partial E}{\partial \alpha} = & -2(G_\mathrm{r} - I_\mathrm{r} - \alpha(F_\mathrm{r} - I_\mathrm{r}))(F_\mathrm{r} - I_\mathrm{r}) \\ & - 2(G_\mathrm{g} - I_\mathrm{g} - \alpha(F_\mathrm{g} - I_\mathrm{g})) \\ & \quad (F_\mathrm{g} - I_\mathrm{g}) - 2(G_\mathrm{b} - I_\mathrm{b} - \alpha(F_\mathrm{b} - I_\mathrm{b})) \\ & \quad (F_\mathrm{b} - I_\mathrm{b}) \end{aligned} \quad (19)$$

Equating $\frac{\partial E}{\partial \alpha}$ to 0 generates

$$\begin{aligned}(G_\mathrm{r}-I_\mathrm{r})(F_\mathrm{r}-I_\mathrm{r})+(G_\mathrm{g}-I_\mathrm{g})(F_\mathrm{g}-I_\mathrm{g})\\ +(G_\mathrm{b}-I_\mathrm{b})(F_\mathrm{b}-I_\mathrm{b})\\ =\alpha\left((F_\mathrm{r}-I_\mathrm{r})^2\right.\\ \left.+(F_\mathrm{g}-I_\mathrm{g})^2+(F_\mathrm{b}-I_\mathrm{b})^2\right)\end{aligned} \tag{20}$$

Writing it in a matrix form, we get

$$\alpha=\frac{(G-I)^T(F-I)}{\|F-I\|^2} \tag{21}$$

This is the same as Equation 6.

Competing interests

The authors declare that they have no competing interests.

Authors' information

The information, research, and publications of MAE are in http://blogs.bu.edu/gharib/. FP is a research fellow at the Sigmedia Lab (www.sigmedia.tv). AK is leading the Sigmedia Group since 1998 (www.sigmedia.tv). He is now a technology Lead at Google Inc.

Acknowledgements

This work has been supported by the Irish Research Council for Science Engineering and Technology's Embark initiative and Science Foundation Ireland (grant no. 08/IN.1/I2112). Some images used are courtesy of INA, Paris.

Author details

[1]Boston University, Boston, MA, USA. [2]Trinity College Dublin, Dublin, Ireland. [3]Google Inc., Mountain View, CA, USA.

Received: 17 July 2012 Accepted: 16 May 2013
Published: 6 June 2013

References

1. R Storey, *Electronic detection and concealment of film dirt*. UK Patent Specification no. 2139039 (1984)
2. A Kokaram, On missing data treatment for degraded video and film archives: a survey and a new Bayesian approach. IEEE Trans. Image Process. (TIPS). **13**(3), 397–415 (2004)
3. AD Edgar, System and method for image recovery. US Patent no. 5,266,805 (1992)
4. V Bruni, A Crawford, A Kokaram, D Vitulano, Semi-transparent blotches removal from sepia images exploiting visibility laws. Signal Image Video P. **7**(1), 11–26 (2013)
5. V Bruni, A Crawford, D Vitulano, F Stanco, Visibility based detection and removal of semi-transparent blotches on archived documents. International Conference on computer vision theory and applications (VISAPP), Setúbal, Portugal 25–28 February 2006
6. AC Kokaram, Motion Picture Restoration (Springer 1998)
7. AC Kokaram, R Morris, W Fitzgerald, P Rayner, Detection of missing data in image sequences. IEEE Trans. Image Process. (TIPS). **4**(11), 1496–1508 (1995)
8. M Nadenau, S Mitra, in *International Workshop on Time-varying Image Processing and Moving Object Recognition*, ed. by V Cappellini. Blotch and scratch detection in image sequences based on rank ordered differences (Elsevier New York, 1997), pp. 27–35
9. V Bruni, D Vitulano, A generalized model for scratch detection. IEEE Trans. Image Process. (TIPS). **13**, 44–50 (2004)
10. A Kokaram, Detection and removal of line scratches in degraded motion picture sequences. Signal Process. VIII. **1**, 5–8 (1996)
11. R Morris, W Fitzgerald, A Kokaram, A sampling based approach to line scratch removal from motion picture frames. Int. Conf. Image Process. (ICIP). **1**, 801–804 (1996)
12. D Tegolo, F Isgro, Scratch detection and removal from static images using simple statistics and genetic algorithms. Int. Conf. Image Process. (ICIP). **1**, 265–268 (2001)
13. L Joyeux, O Buisson, B Besserer, S Boukir, Detection and removal of line scratches in motion picture films. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR). **1**, 548–553 (1999)
14. R Bornard, Probabilistic approaches for the digital restoration of television archives. PhD Thesis, Ecole Centrale Paris, 2002
15. YY Chuang, B Curless, DH Salesin, R Szeliski, A Bayesian approach to digital matting. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR). **2**, 264–271 (2001)
16. A Levin, D Lischinski, Y Weiss, A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Int. (PAMI). **30**(2), 228–242 (2008)
17. M Ruzon, C Tomasi, Alpha estimation in natural images. IEEE Conf. Comput. Vision Pattern Recognit. **1**, 18–25 (2000)
18. AR Smith, JF Blinn, Blue screen matting. Paper presented at the annual conference of Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH), New Orleans, USA, 4–9 August 1996
19. J Sun, J Jia, CK Tang, HY Shum, Poisson matting. ACM SIGGRAPH. **23**, 315–321 (2004)
20. P White, W Collis, S Robinson, A Kokaram, Inference matting. Paper presented at the IEEE European conference on visual media production (CVMP), London, UK, 30 November–1 December 2005
21. MA Ahmed, F Pitié, AC Kokaram, Extraction of non-binary blotch mattes. Paper presented at the international conference on image processing (ICIP), Cairo, Egypt, 7–10 November 2009
22. A Criminisi, P Perez, K Toyama, Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. (TIPS). **13**(9), 1200–1212 (2004)
23. G Forbin, B Besserer, J Boldys, D Tschumperle, Temporal extension to exemplar-based inpainting, applied to scratch correction in damaged image sequences. Paper presented at the international conference on visualization, imaging and image processing (VIIP), Benidorm, Spain, 7–9 September 2005
24. A Kokaram, Removal of line artefacts for digital dissemination of archived film and video. Int. Conf. Multimedia Comput. Syst. (ICMCS). **2**, 245–249 (1999)
25. AC Kokaram, S Godsill, MCMC for joint noise reduction and missing data treatment in degraded video. IEEE Trans. Image Process. (TIPS), Special Issue MCMC. **50**(2), 189–205 (2002)
26. AC Kokaram, PJW Rayner, System for the removal of impulsive noise in image sequences. SPIE Visual Commun. Image Process. **1818**, 322–331 (1992)
27. V Bruni, P Ferrara, D Vitulano, Removal of color scratches from old motion picture films exploiting human perception. EURASIP J. Adv. Signal Process. **2008**, 1–9 (2008)
28. V Bruni, D Vitulano, A Kokaram, Fast removal of line scratches in old movies. Int. Conf. Pattern Recognit. (ICPR). **4**, 827–830 (2004)
29. A Crawford, V Bruni, A Kokaram, D Vitulano, Multi-scale semi-transparent blotch removal on archived photographs using Bayesian matting techniques and visibility laws. Int. Conf. Image Process. (ICIP). **1**, 561–564 (2007)
30. F Stanco, L Tenze, A De Rosa, An improved method for water blotches detection and restoration. Paper presented at the international symposium on signal processing and technology, Rome, Italy 18–21 December 2004
31. F Stanco, L Tenze, G Ramponi, Virtual restoration of vintage photographic prints affected by foxing and water blotches. J. Electron Imaging. **14**(4), 043008 (2005)
32. J Ren, T Vlachos, Efficient detection of temporally impulsive dirt impairments in archived films. Signal Process. **87**(3), 541–551 (2007)
33. AC Kokaram, R Morris, W Fitzgerald, P Rayner, Interpolation of missing data in image sequences. IEEE Trans. Image Process. (TIPS). **4**(11), 1509–1519 (1995)
34. D Corrigan, N Harte, A Kokaram, Pathological motion detection for robust missing data treatment. EURASIP J. Adv. Signal Process. **2008**, 1–16 (2008)
35. A Rares, M Reinders, J Biemond, Complex event classification in degraded image sequences. Int. Conf. Image Process. (ICIP), **1**, 253–256 (2001)
36. V Roosmalen, Restoration of archived film and video. PhD Thesis, Delft University, 1999
37. B Kent, A Kokaram, B Collis, S Robinson, Two layer segmentation for handling pathological motion in degraded post production media. Int. Conf. Image Process. (ICIP), **1**, 299–302 (2004)

38. A Efros, W Freeman, Image quilting for texture synthesis and transfer. Paper presented at the ACM SIGGRAPH , Los Angeles, CA, 12–17 August 2001
39. AA Efros, T Leung, Texture synthesis by non-parametric sampling. Int. Conf. Comput. Vision (ICCV) **2**, 1033–1038 (1999)
40. M Kemal Gullu, O Urhan, S Erturk, Scratch detection via temporal coherency analysis and removal using edge priority based interpolation. Paper presented at the international symposium on circuits and systems (ISCAS), Island of Kos, Greece, 21–24 May 2006
41. F Stanco, G Ramponi, L Tenze, Removal of semi-transparent blotches in old photographic prints. COST 276 workshop, Prague, Czech Republic, 2–3 October 2003
42. J Besag, On the statistical analysis of dirty pictures. J. Roy. St. B. **48**(3), 259–302 (1984)
43. T Hoshi, T Komatsu, T Saito, Film blotch removal with a spatiotemporal fuzzy filter based on local image analysis of anisotropic continuity. Int. Conf. Image Process. (ICIP). **2**, 478–482 (1998)
44. T Saito, T Komatsu, T Ohuchi, T Hoshi, Practical nonlinear filtering for removal of blotches from old film. Int. Conf. Image Process. (ICIP). **3**, 164–168 (1999)
45. M Orchard, C Bouman, Color quantization of images. IEEE Trans. Image Process. (TIPS). **39**(12), 2677–2690 (1991)
46. V Kolmogorov, C Rother, Minimizing nonsubmodular functions with graph cuts-a review. IEEE Trans. Pattern Anal. Mach. Int. (PAMI). **29**(7), 1274–1279 (2007)
47. C Rother, V Kolmogorov, V Lempitsky, M Szummer, Optimizing binary MRFs via extended roof duality. Paper presented at the IEEE conference on computer vision and pattern recognition (CVPR), Minneapolis, Minnesota, 18–23 June 2007
48. V Lempitsky, C Rother, S Roth, A Blake, Fusion moves for random field optimization. IEEE Trans. Pattern Anal. Mach. Int. (PAMI). **32**(8), 1392–1405 (2010)
49. Y Boykov, O Veksler, R Zabih, Fast approximate energy minimization via graph-cuts. IEEE Trans. Pattern Anal. Mach. Int. (PAMI). **23**(11), 1222–1239 (2001)
50. Z Wang, A Bovik, H Sheikh, E Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. (TIPS). **13**(4), 600–612 (2004)

doi:10.1186/1687-5281-2013-33
Cite this article as: Elgharib *et al.*: **Blotch and scratch removal in archived film using a semi-transparent corruption model and a ground-truth generation technique.** *EURASIP Journal on Image and Video Processing* 2013 **2013**:33.