



An efficient framework for image/video inpainting

Miaohui Wang^{a,*}, Bo Yan^b, King Ngi Ngan^a

^a Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong

^b School of Computer Science, Fudan University, Shanghai, China



ARTICLE INFO

Article history:

Received 11 September 2012

Received in revised form

29 January 2013

Accepted 11 March 2013

Available online 8 April 2013

Keywords:

Image inpainting

Image interpolation

Image restoration

PDE

Diffusion function

ABSTRACT

Image inpainting has been widely applied to many applications, such as restoring corrupted old photos, erasing video logos, concealing errors in a digital video processing system, and so on. However, traditional geometric inpainting methods suffer low efficiency. To tackle this problem, this paper addresses an efficient transform based framework for geometric methods. Given an image, we firstly decompose it, then separately perform restoration process and finally employ Laplacian diffusion function to hold local texture coherence. Experimental results show that the proposed method not only speeds up and enhances the performances of geometric methods, but also obtains a better restoration results compared with the traditional texture and hybrid methods.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Digital image inpainting, also called *image interpolation* [1–3], *image completion* [4–6] or *image restoration* [7,8], aims at removing corrupted areas but avoiding being noticed obviously by observers, which is proposed by Bertalmio et al. [9–11]. Image inpainting is commonly used to remove video logos [12,13], restore occlusion in video sequences [14–16], edit digital photos [17–19] and compress images [20–22]. In the past decade, many inpainting methods have been widely developed by mathematicians or computer scientists. Generally speaking, inpainting methods fall into two categories: geometric methods (non-texture) and texture based methods.

Geometric methods restore the damaged image by diffusing boundary information to the missing area along the isophote lines (level lines). In [23,24], Masnou et al. restored the occlusion area along level lines. Chan et al. [25–27] developed a total variational (TV) model to restore

the missing parts and later proposed a curvature driven diffusion scheme [28] by considering Euler's Elastica. By contrast, some methods [9,29–31] directly employed partial differential equations (PDE) to restore natural image geometric structures. For instance, Bertalmio et al. [9] adopted Laplace formula, and [29] adopted Navier–Stokes formula. Geometric methods restore the damaged areas from the boundary to center which indicate that only boundary pixels can be used. The constraints of geometric methods are: (1) smooth restoration results, (2) poor performance at the thick area, and (3) extensive computational time.

Texture based methods begin with exploring the self-coherence features and then copy the best texture patch from restoration dictionary to the missing area. Criminisi et al. [32] proposed exemplar based method to restore the missing parts by the best patch. Exemplar based methods employed texture synthesis [33] to integrate the patch with the boundaries. In order to obtain the best patch, sparse representation methods [34–38] were proposed to utilize linear combination of blocks from an image dictionary to produce the patch. Under the assumption of containing repetitive textures, exemplar based methods may create abnormal structures that are hard to understand, because one order greedy approach is used in

* Corresponding author. Tel.: +852 5495 7535.

E-mail addresses: mhwang@ee.cuhk.edu.hk (M. Wang), byan@fudan.edu.cn (B. Yan), knngan@ee.cuhk.edu.hk (K. Ngi Ngan).

restoration and the filling orders greatly bear the restoring results when the missing area is thick. Based on spatial-temporal consistency, Wexler et al. [16] combined geometric and exemplar method to find the best patch. Bugeau [39] proposed a complex energy function to combine PDE and exemplar based texture synthesis together, where they stated that the hybrid energy function works poor when missing area is thick or on a singular location.

Besides the above two main categories, there are some other ponderable methods. For example, in [40], Telea et al. proposed a fast marching method to restore the damaged image; In [41], Sun et al. proposed an interesting method combining user and synthesis technique together to restore isolated parts. In [42], Bornemann et al. proposed a fast hybrid variational based model. Some methods [8,43] restored the given image in wavelet domain. However there are many constraints on restoration in wavelet domain: (1) boundary pixels of corrupted area bring new errors due to subtraction operations in producing subbands; (2) the correlations among pixels have been greatly changed.

In this paper, we propose an efficient hybrid inpainting method for geometric methods. To avoid constraints of geometric methods, the proposed transform based framework restores the given image in the new domain. The proposed framework, to the best of our knowledge, is novel and the performance is impressive. Experimental results show that it not only greatly speeds up traditional geometric methods, but also improves the PSNR by up to 5.10 dB. It should be noted that the proposed framework is not limited to the geometric methods but can also be applied to the texture and hybrid methods. Experimental results show that the proposed framework is able to

produce more reasonable results than texture and hybrid methods.

The rest of this paper is organized as follows. The theory foundation of the proposed framework is given in Section 2. Implementation and optimization details are discussed in Section 3. The effectiveness of our proposed model is evaluated in Section 4. Concluding remarks are given in Section 5.

2. Theory foundation of the proposed framework

Restoration procedure can be well demonstrated by the Bayesian model [44]. In image inpainting, we aim at restoring the original image M by maximizing the following posterior probability formula:

$$P(M|M_0, \Omega) = \frac{P(M_0|M, \Omega)P(M|\Omega)}{P(M_0|\Omega)}, \quad (1)$$

where M_0 and Ω are the given image and the missing data areas, respectively.

When implementing logarithmic likelihood function on both side of Eq. (1), the target becomes

$$E(M|M_0, \Omega) = E(M_0|M, \Omega) + E(M|\Omega) - E(M_0|\Omega) \quad (2)$$

where $E(x)$ is a minus logarithmic likelihood function.

In a communication channel, since the quality degradation is independent of image content itself, it indicates $P(M|\Omega) = P(M)$. Once M_0 and Ω are given, $E(M_0|\Omega)$ is a constant and the minimization problem of Eq. (2) can be rewritten as

$$\arg \min : E(M|M_0, \Omega) \propto E(M_0|M, \Omega) + E(M), \quad (3)$$

where the specific notations of $E(M)$ and $E(M_0|M, \Omega)$ for PDE based methods can be calculated by solving Eq. (4).

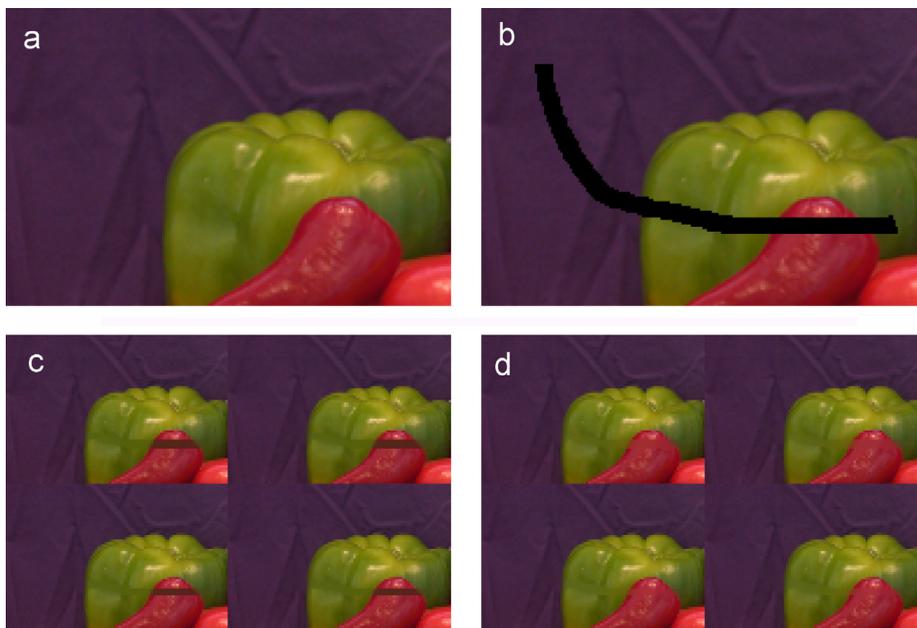


Fig. 1. Error analysis for image inpainting. (c) represents the restoration result $T(M)$ which is restored firstly by the TV method and then transformed in the T domain for comparison. (d) shows the restoration result of $T(M_0)$ which is firstly transformed in the T domain and then restored by the TV method. Both of them consume about 24 s (the same 5000 iterations) [25]: (a) ground truth; (b) scratched M_0 ; (c) $T(M)$ 32.01 dB; and (d) $T(M_0)$ 41.59 dB.

In practice, geometric method [45] is trying to minimize the following equation:

$$\text{arg min} : \int_{\Omega} \sqrt{M_x^2 + M_y^2} dx dy + \frac{1}{2\sigma^2|\Omega^c \setminus \Omega|} \int_{\Omega^c \setminus \Omega} (M - M_0)^2 dx dy, \quad (4)$$

where $\partial\Omega$ is the boundary of the Ω which is assumed to be smooth. Since the above formula is a classical total variational model [46], a PDE is solved to find the result of Eq. (4).

Our goal is to construct a framework to change the initial condition of $E(M|M_0, \Omega)$ so as to improve traditional methods efficiently to solve the above minimization problem. In order to efficiently solve Eq. (4), we restore M in the T domain and numerous experimental results verify our strategy. Formula (5) is the foundation of the proposed transform based framework:

$$E(\tilde{M}|\tilde{M}_0, \tilde{\Omega}) = T(E(M|M_0, \Omega)). \quad (5)$$

The various resolutions of video sequences greatly motivate us, where different devices require various resolutions of the same video sequence for displaying, such as CIF and QCIF formats. The original M_0 is decomposed into four resembling parts as shown in Fig. 1(c). Here the T transform is defined by the following Eq. (6) and the details are illustrated in Table 1:

$$T : M_0 \rightarrow \tilde{M}_0 = \begin{bmatrix} \tilde{M}_0^1 & \tilde{M}_0^2 \\ \tilde{M}_0^3 & \tilde{M}_0^4 \end{bmatrix}, \quad (6)$$

where \tilde{M}_0 and \tilde{M}_0^k ($k = 1, 2, 3, 4$), are the new components in T domain.

After incorporating T transform, Eq. (3) can be rewritten as

$$\text{arg min} : E(\tilde{M}|\tilde{M}_0, \tilde{\Omega}) \propto E(\tilde{M}_0|\tilde{M}, \tilde{\Omega}) + E(\tilde{M}). \quad (7)$$

In the following parts, we will demonstrate the advantages of $E(\tilde{M}|\tilde{M}_0, \tilde{\Omega})$ for restoration. In general, the proposed T transform maintains three merits: holding local coherence, shrinking the thick area and increasing the length of $\partial\tilde{\Omega}$. It is these three properties together that constitute the main theory foundations of the proposed framework.

Table 1
The CST algorithm.

Chain rule based T Transform algorithm	
STransform ($M_{(i)}$)	
BEGIN	
$\tilde{M}_{(i+1)}^1 = \{(x, y) (x, y) = (2x, 2y) \in \tilde{M}_{(i)}\}$	
$\tilde{M}_{(i+1)}^2 = \{(x, y) (x, y) = (2x, 2y-1) \in \tilde{M}_{(i)}\}$	
$\tilde{M}_{(i+1)}^3 = \{(x, y) (x, y) = (2x-1, 2y) \in \tilde{M}_{(i)}\}$	
$\tilde{M}_{(i+1)}^4 = \{(x, y) (x, y) = (2x-1, 2y-1) \in \tilde{M}_{(i)}\}$	
END	

2.1. Holding local coherence

Fig. 1(c) shows how T transform keeps local texture coherence. Pixels on $\partial\Omega$ of M_0 and on $T(\partial\Omega)$ of \tilde{M}_0 have similar neighboring circumstance. This property guarantees that the restoration results in the T domain are similar as that in the given domain, which keeps the traditional inpainting methods performing well in the T domain.

2.2. Shrinking thick area

Inscribed Circle of the Hole (ICH) is the biggest inscribed circle in the Ω areas, and employed in [47]. Generally speaking, the restoration performances are determined by the radius of the *Inscribed Circle of the Hole* (R_{ICH}) as shown in Fig. 2.

$$R_{ICH} = \text{Max}\{R_{\partial\Omega^1}, R_{\partial\Omega^2}, \dots, R_{\partial\Omega^N}\}. \quad (8)$$

The R_{ICH} in \tilde{M}_0^i ($i = 1, 2, 3, 4$) is a quarter of the R_{ICH} in M_0 which is crucial for inpainting methods. Bugeau et al. [39] stated that one of constraints in their method is the restoration of thick area, although they have constructed a hybrid framework for digital image inpainting. Therefore, restoration of thick area is still a challenge task for state-of-the-art inpainting methods. If R_{ICH} decreases, the accumulated error to center pixels also decreases. In that situation, texture methods produce less abnormal repetitive results, and geometric methods obtain more information from boundary pixels for one iteration.

Let $B_{\Omega} = \{\theta(i)|i = 1, 2, 3, \dots, N\}$ be a set containing the boundary pixels of all holes in M_0 , and $\theta(i) = \{(x, y)| (x, y) \in M_{\partial\Omega^i}\}$ be the boundary pixels in the i th hole of Ω . Let ξ_j be the correct propagation rate of the j th step along the normal orientation of level line at pixel P_0 as shown in Fig. 3, and $\varepsilon_i = \{\varepsilon_i^1, \varepsilon_i^2, \dots, \varepsilon_i^k, \dots\}$ be the error information of the i th hole, where ε_i^k is the error information in the k th step along

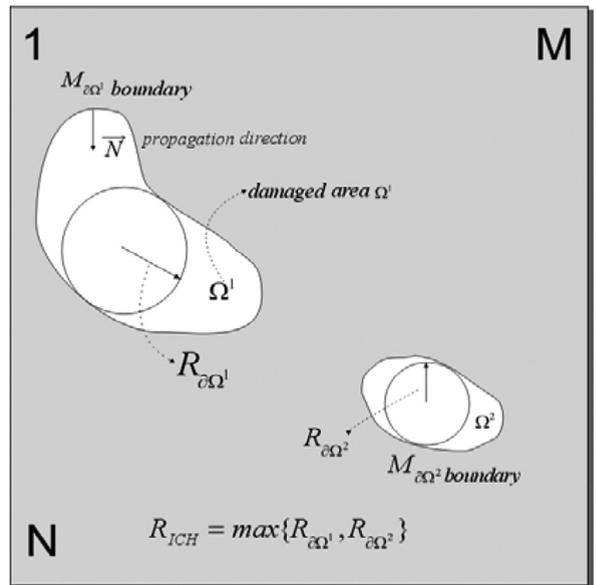


Fig. 2. The definition of the biggest inscribed circle of Ω area. There are two holes with different biggest inscribed circles in the picture with width M and height N .

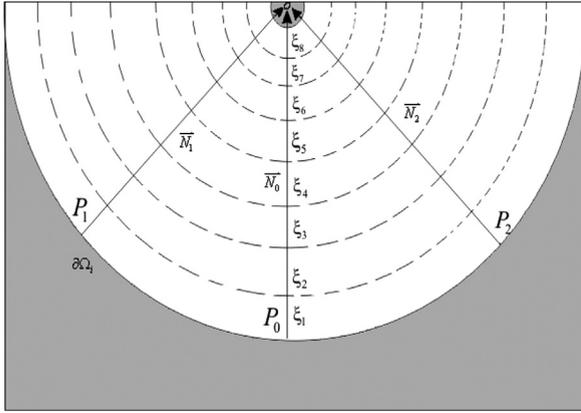


Fig. 3. Error propagation ratio and level lines. \vec{N}_0 , \vec{N}_1 and \vec{N}_2 are the normals of the level line at the pixels P_0 , P_1 and P_2 , respectively. Pixel O is the pixel which needs to be restored, and ξ_j is the correct propagation rate of the j th step along the normal direction of level line.

the normal direction of level line:

$$\varepsilon_i^k = \theta(i) \sum_{j=1}^k \prod_{m=1}^j (1-\xi_1)(1-\xi_2) \cdots (1-\xi_m). \quad (9)$$

Let θ_{error} be the total error information that one restored pixel gets errors from pixels along its level line, and w^i ($i = 1, 2, \dots$) be the weight of boundary pixels in the i th hole. The total error information which is propagated to the restored pixel can be expressed by the following equation:

$$\theta_{error} = \sum_{k=1}^N w^k \varepsilon_i^k = \theta(i) \sum_{k=1}^N w^k \sum_{j=1}^k \prod_{m=1}^j (1-\xi_1)(1-\xi_2) \cdots (1-\xi_m). \quad (10)$$

In the T domain, ε_i^k ($i = 1, 2, \dots, N$) and θ_{error} decrease as R_{TCH} decreases. Experimental results in Section 4 clearly support the inference and show advantages of the proposed framework in solving $E(\tilde{M}|\tilde{M}_0, \tilde{\Omega})$.

Furthermore, error bound, for geometric methods, is presented by Chan et al. [48]

$$err(\Omega) = \sum_i \partial L^i |L_{\Omega}^i|, \quad (11)$$

where $\partial L = L_{\max \partial L_i} - L_{\min \partial L_i}$ is the biggest intensity difference in the i th level line L_i , and L_{Ω}^i is the restoration region of the i th level line.

As proved above, the error bound of \tilde{M}_0^i is smaller than that of M_0 . In other words, the inequality of $err(\Omega) > \max\{err(\tilde{\Omega}^1), err(\tilde{\Omega}^2), err(\tilde{\Omega}^3), err(\tilde{\Omega}^4)\}$ indicates that $E(\tilde{M}|\tilde{M}_0, \tilde{\Omega})$ can obtain a better result.

2.3. Increasing $\partial \tilde{\Omega}$

In the T domain, $\partial \Omega$ is mapped to $\partial \tilde{\Omega}^i$ ($i = 1, 2, 3, 4$), and it can also be proved that the following formula is true.

$$\sum_{i=1}^4 \partial \tilde{\Omega}^i > \partial \Omega. \quad (12)$$

The above inequality is important for geometric methods. Since more boundary pixels can be used to restore the missing parts, Eq. (4) can be solved more quickly than that

of in the given domain. For texture methods, more boundary pixels provide more restoring orders and it indicates that texture methods may produce a better result.

Fig. 1 illustrates that when boundary pixels increase as shown in Fig. 1(d), TV based methods can restore the center pixels better than that of in Fig. 1(c). As the perimeters increase in the T domain, more pixels in Fig. 1 (d) can get their values directly from the boundary pixels. This advantage guarantees \tilde{M} to approach the ground truth and this is why Fig. 1(d) exhibits better visual quality than Fig. 1(c).

3. Implementation and optimization

3.1. Implementation details

The definition of forward T transform is given by Eq. (6), and it can be easily checked that T transform is revertible. In other words, the original image M_0 is able to be completely reconstructed by the inverse transform T^{-1} as defined in Eq. (13):

$$T^{-1} : \begin{bmatrix} \tilde{M}_0^1 & \tilde{M}_0^2 \\ \tilde{M}_0^3 & \tilde{M}_0^4 \end{bmatrix} = \tilde{M}_0 \rightarrow M_0 \quad (13)$$

In the proposed method, chain rule $T \circ T(M_0)$ can be applied to the M_0 if the M_0^i holds the following error bounds as shown in Eq. (14). Finally, the chain rule based T transform is given in Table 1,

$$\left\{ \begin{array}{l} \left| \sum_{(x,y) \in P(x,y)} M_0(x,y) - \sum_{(x_0,y_0) \in P(x_0,y_0)} \tilde{M}_0(x_0,y_0) \right| = \delta_0(x,y) \\ \sum_{(x,y) \in P(x,y)} |M_0(x,y) - \tilde{M}_0(x_0,y_0)| = \delta_1(x,y) \end{array} \right. \quad (14)$$

In Eq. (14), where $n, m = \{-1, 0, 1\}$ and $\delta_0(x, y)$ is the error estimator of patch $P(x, y)$ (3×3 pixels) centered at pixel (x, y) . $\delta_0(x, y), \delta_1(x, y) \in (0, \varepsilon)$, ε is the global error bound, and $\delta = (\sum_{(x,y) \in \partial \Omega} \delta(x, y)) / |\partial \Omega|$ is the term used to measure the feasibility of the T transform.

3.2. Optimization

3.2.1. Pre-processing

Although T transform holds the shape of original object in M_0 , it may cause interlacing problem on $\partial \tilde{\Omega}$ as shown in Fig. 10(d), when the T^{-1} transform is performed on different layers. One way to avoid this problem is to extend the boundary of the original $\partial \Omega$ before the T transform. Considering this kind of preprocessing step, the interlaced problem is greatly removed as shown in Fig. 10(h).

Before introducing the proposed de-interlacing method, some notations are first employed. Let $M_{mask}^i = \{E(j)|j = 1, 2, \dots\}$ represent the mask of \tilde{M}_0 in the i th layer of the j th hole. Let $E_{\partial \Omega^1}^i(j)$ and $E_{\partial \Omega^2}^i(j)$ represent the first and second

Table 2
The workflow of the proposed framework.

DTF based algorithm

Step 1. BEGIN

Step 2. set initial value: $i = 0$;

Step 3. construct a pre-processing picture $M_p^i = M_i$ and calculate δ, ϵ ;

Step 4. $M_p^i(x, y) = M_p^i(x \pm 1, y \pm 1)$, if $M_p^i(x, y) \in E_{\partial\Omega^1}^i(j)$ and $M_p^i(x \pm 1, y \pm 1) \in E_{\partial\Omega^2}^i(j)$ belong to different objects, Formula (15);

Step 5.

$$\begin{cases} \text{if } \delta < \epsilon \\ STtransform(M_i), i = i + 1, Step(3) \\ \text{if } \delta \geq \epsilon \\ maxChain = i, Step(6) \end{cases};$$

Step 6. restore the damaged picture $M_p^i(x, y) =$

$$\begin{cases} \text{if } i = maxChain \\ Candidate Inpainting Algorithms(x) \\ \text{if } i < maxChain \\ Formula (17) Iteration = 10 \end{cases};$$

Step 7. $M_i(x, y) = \begin{cases} M_p^i(x, y) & M_{mask}^i = 1 \\ M_i(x, y) & M_{mask}^i = 0 \end{cases};$

Step 8. reconstruct the picture $M_p^{i-1}(x, y)$;

Step 9. update the argument $i = i - 1$ and perform:

$$\begin{cases} \text{if } i > = 1 \\ Return to Step(7). \\ \text{if } i = 0 \\ Perform Step(7). End-of-DTF. \end{cases};$$

Step 10.END

undamaged boundary pixels in the j th hole, respectively,

$$\begin{cases} E_{\partial\Omega^1}^i(j) = \{(x + \hbar, y + \ell) | \hbar, \ell = 0, \pm 1 \\ \text{and } (x, y) \in \theta(j) \} \\ E_{\partial\Omega^2}^i(j) = \{(x + \hbar, y + \ell) | \hbar, \ell = 0, \pm 2 \text{ and } (x, y) \in \theta(j) \} \end{cases} \quad (15)$$

If $M_{\partial\Omega^1}^i(x, y)$ and $M_{\partial\Omega^2}^i(m, n)$ are neighboring pixels, $M_{\partial\Omega^2}^i(m, n)$ is reset as $M_{\partial\Omega^1}^i(x, y)$ when this pair of pixels satisfy the following conditions:

$$\begin{cases} M_{\partial\Omega^1}^i(x \pm \hbar, y \pm \ell) \neq M_{\partial\Omega^1}^i(m, n) \\ M_{\partial\Omega^1}^i(x \pm \hbar, y \pm \ell) \notin \theta(j) \\ M_{\partial\Omega^1}^i(x, y) \in E_{\partial\Omega^1}^i(j), M_{\partial\Omega^2}^i(m, n) \in E_{\partial\Omega^2}^i(j) \\ (m, n) = (x + \hbar, y + \ell), \hbar, \ell = 0, \pm 1. \end{cases} \quad (16)$$

3.2.2. Post-processing

To guarantee the coherence of boundary pixels at each layer of the T^{-1} transform, Laplacian formula is used to revise the inconsistencies among pixels at each layer during the reverse process as show in Fig. 10(d) and (h):

$$\frac{\partial^2 M_0^i}{\partial x^2} + \frac{\partial^2 M_0^i}{\partial y^2} = 0. \quad (17)$$

Laplacian diffusion function Eq. (17) is employed to maintain neighboring pixels coherence and parameters is chose to avoid resulting in smooth problem. Details, like controlling iterations, have been given in Table 2. In the end of this subsection, the details of the proposed diffusion based T transform framework (DTF) is shown in Table 2.

4. Simulation results

In this section, the effectiveness of the DTF method is evaluated by extensive experimental results, and the objective performance is measured by peak signal to noise ratio (PSNR). The proposed DTF method and other inpainting methods [9,16,25,32,40,42] are tested on a dual-core PC(RAM 3072 M and CPU 2.53 GHz). For simplicity, the item maxChain is manually set to two except for Fig. 4 due to the low image resolutions. The restoration process of DTF takes four steps as shown in Table 2: (1) forward T transform; (2) performing traditional restoration; (3) inverse T transform, and (4) diffusion process. Candidates (x) can be geometric [9,25,40], texture [32], or hybrid methods [16,42].

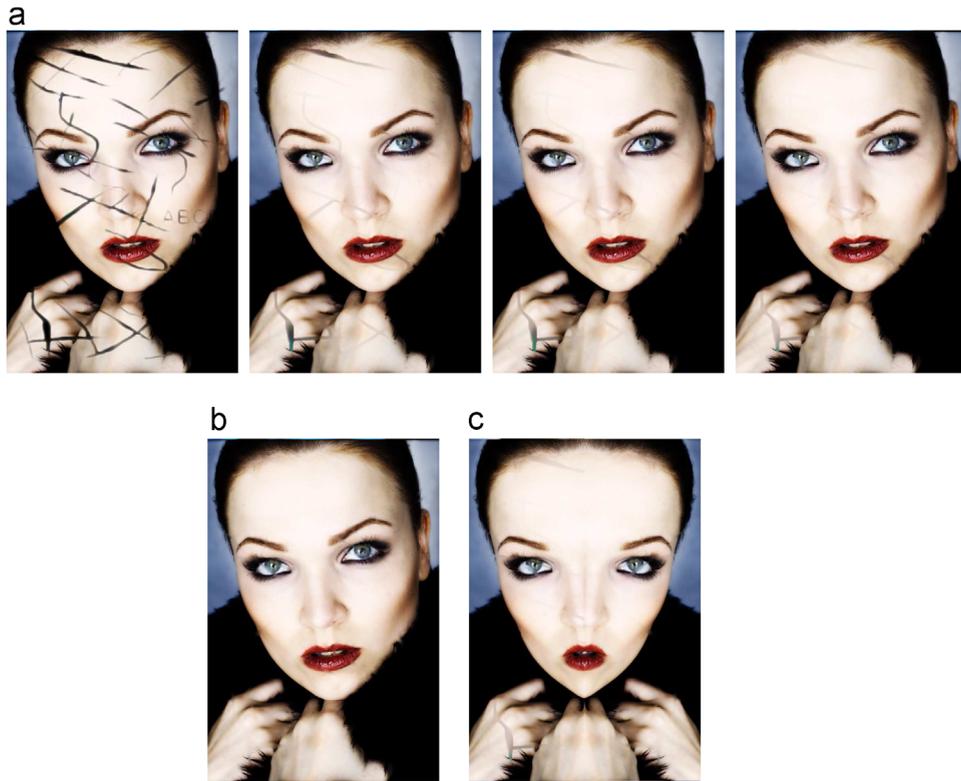


Fig. 4. Results of Bertalmio et al. [9] and DTF based method for Tarja. 4(a) represents the results of Bertalmio et al. [9] with iterations=100, 1000, 1500, 3000 from left to right. The restoration result of DTF method with 30 iterations is shown in (b) ($maxChain=3$). (c) illustrates the comparison result of Bertalmio and DTF model: left is the result of Bertalmio et al. [9] with iteration=3000 (3279 s). Right is the DTF method with iteration=30 (33 s). Note that there still exist clearly scratches on the Tarja's hand: (a) Bertalmio et al. [9], Iteration=100, PSNR=21.78 dB (left), 1000, 30.18 dB, 1500, 32.06 dB, 3000, 35.72 dB (right); (b) DTF Iteration=30, 37.84 dB and (c) Tarja mirror-effect.

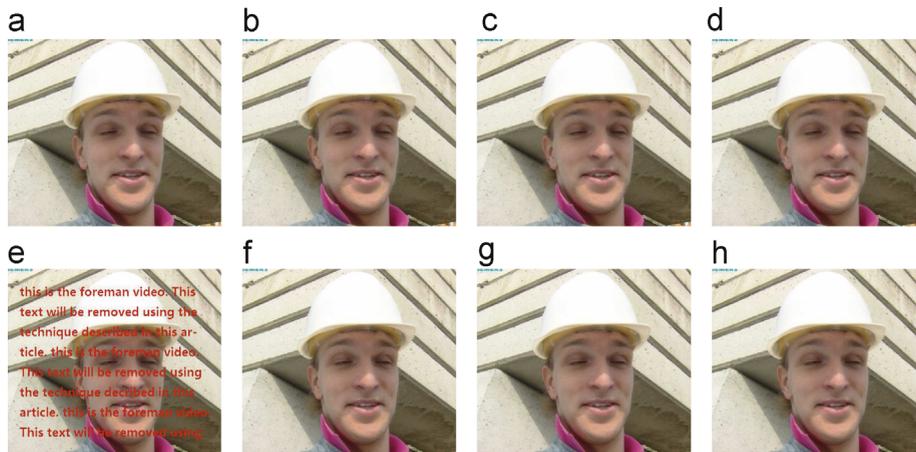


Fig. 5. Results of geometric inpainting methods and DTF method for Foreman video. (b)–(d) show the results of geometric methods [9,25,40], respectively. The DTF based results are shown correspondingly for comparison: (a) original; (b) [9] 41.10 dB; (c) [25] 41.56 dB; (d) [40] 40.86 dB; (e) scratch; (f) DTF 41.47 dB; (g) DTF 42.36 dB; and (h) DTF 41.59 dB.

4.1. DTF for geometric methods

In Fig. 4, it is observed that DTF method remarkably enhances the speed of the classical inpainting method [9],

and it also indicates that DTF method works efficiently under the thick area as shown in Fig. 10(b). Under the same parameters, our DTF method outperforms Bertalmio et al. [9]. It should be noted that, as shown in Fig. 10(c), the

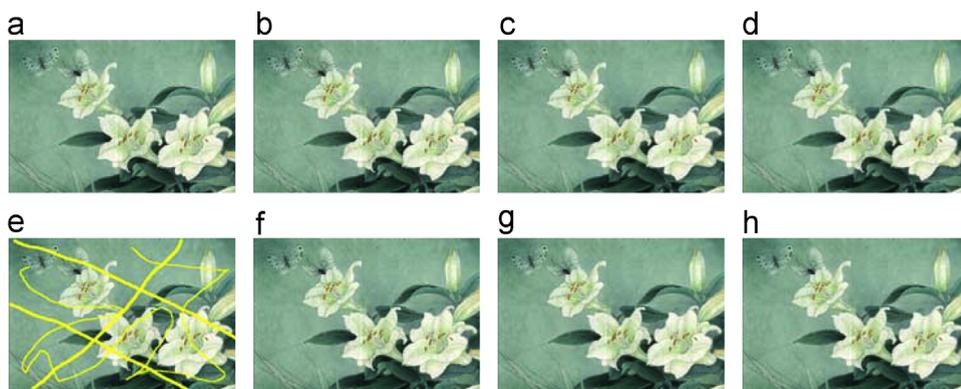


Fig. 6. Results of geometric inpainting methods and DTF method for Peony: (a) original; (b) [9] 30.17 dB; (c) [25] 31.14 dB; (d) [40] 31.20 dB; (e) scratch; (f) DTF 30.85 dB; (g) DTF 31.14 dB; and (h) DTF 31.22 dB.

Table 3

Performances of proposed DTF framework for geometric methods.

Name	Methods	Size	Time saving (s)			PSNR (dB)		
			Ori.	DTF	Δ Gain (%)	Ori.	DTF	Δ Gain
Tarja	[9]	584×864	3279	33	98.99	35.72	37.84	+2.12
	[25]	584×864	1980	61	96.92	37.30	38.18	+0.79
	[40]	584×864	160	29	81.87	37.90	38.10	+0.20
Eye	[9]	200×200	440	11	97.50	19.18	24.28	+5.10
	[25]	200×200	433	16	96.30	25.36	27.46	+2.10
	[40]	200×200	20	5	75.00	27.73	29.37	+1.64
Pepper	[9]	512×384	631	15	97.62	31.41	33.45	+2.04
	[25]	512×384	810	33	95.93	33.04	33.75	+0.71
	[40]	512×384	52	5	90.38	33.52	33.73	+0.21
Peony	[9]	768×512	1580	12	99.24	30.17	30.85	+0.68
	[25]	768×512	1220	25	97.95	31.14	31.14	+0.00
	[40]	768×512	105	5	95.24	31.20	31.22	+0.02
Foreman	[9]	640×560	185	8	95.68	41.10	41.47	+0.37
	[25]	640×560	1170	33	97.18	41.56	42.36	+0.80
	[40]	640×560	120	21	82.50	40.86	41.59	+0.73
Average	Δ Time saving (%)		93.20%					
	Δ PSNR (dB)		+1.17					

DTF method is able to save more time than method [9] and maintain the similar visual quality. For example, the proposed DTF method takes about 33 s, while Bertalmio [9] takes 3279 s nearly 100 times of the DTF method.

Fig. 5 depicts the performance of geometric methods and DTF method to restore the Foreman video sequence. It shows that both the DTF and geometric methods obtain the similar visual quality. However, the DTF method saves nearly 90% of time compared with geometric methods [9,25,40].

Fig. 6 depicts the performance of geometric methods and DTF method to restore the Peony picture. Experimental results show that both the geometric and DTF based method efficiently restore the yellow scratches. Obviously, it is a great advantage that our proposed method saves more time than the traditional geometric methods [9,25,40].

In Table 3, we give the overall performance of the DTF method versus the geometric methods. It is observed that

our DTF method improves the PSNR 1.17 dB in average, and meanwhile it saves time by 93.2%.

4.2. DTF for texture methods

In this subsection, we illustrate the results of the DTF method versus texture methods. Fig. 7 shows that the proposed framework improves texture method to restore the missing parts. Fig. 7(b) shows that Criminisi et al. [32] produces irrational white patches on the surface of some steps, while Fig. 7(d) shows that the DTF method restores the texture perfectly.

Fig. 8 shows a girl running on the beach. The exemplar based method [32] is used to remove the girl but it causes explicit ink marks on the missing area as shown in Fig. 8 (b). In Fig. 8(d), the DTF method restores the missing area with the same texture from the background as expected.

In Fig. 9, we show the ability of the DTF method for edge reconstruction and very large object restoration. Fig. 9(b) and

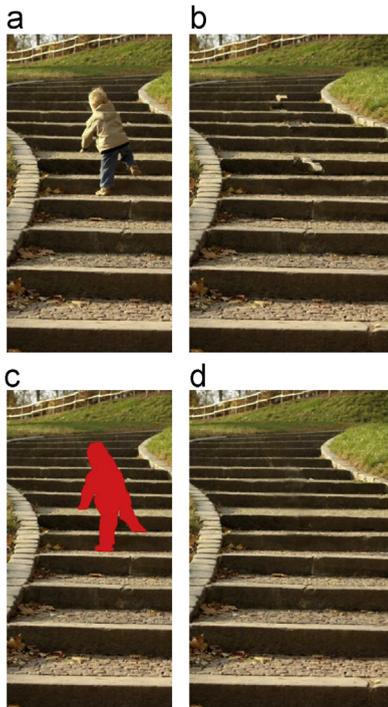


Fig. 7. Results of texture and DTF methods for Stairs picture. Criminisi et al. [32] is shown in (b) and the DTF method is shown in (d): (a) original; (b) [32]; (c) scratch and (d) DTF.

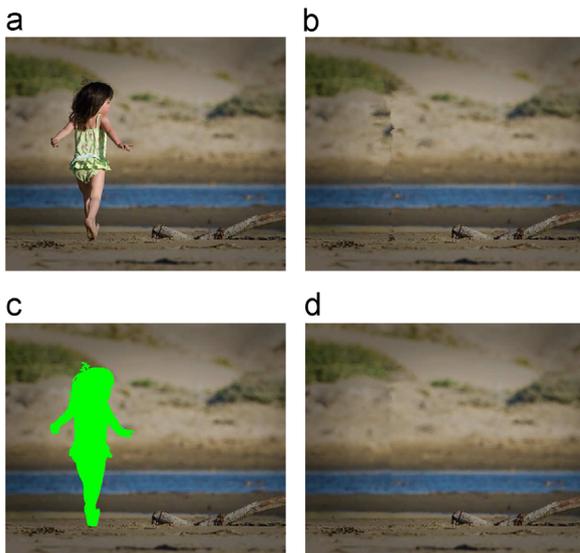


Fig. 8. Results of texture and DTF method for Beach picture. The restoration result of Criminisi et al. [32] is shown in (b) and the comparison result of the DTF method is shown in (d): (a) original; (b) [32]; (c) scratch and (d) DTF.

(d) shows results of the exemplar based method [32] and the DTF, respectively. It is obvious that the proposed DTF method achieves better performances than Criminisi et al. [32].

The runtime of texture methods versus DTF method are summarized as: (1) Fig. 7(b) 31 s and Fig. 7(d) 38 s; (2) Fig. 8

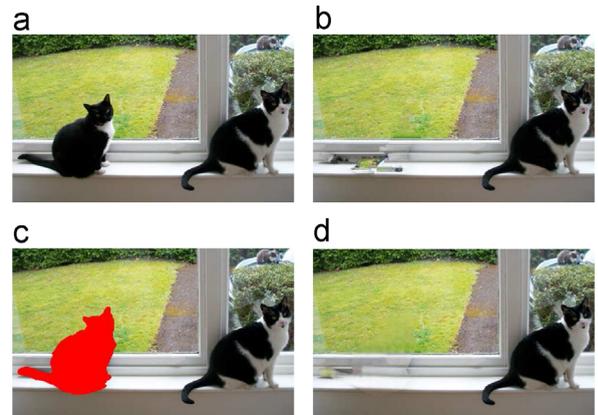


Fig. 9. Results of edge reconstruction and very large object removal. (b) and (d) are the results of [32] and the proposed DTF method, respectively: (a) original; (b) [32]; (c) scratch and (d) DTF.

(b) 41 s and Fig. 8(d) 48 s; (3) Fig. 9(b) 34 s and Fig. 9(d) 59 s. Note that DTF consumes much time because the DTF provides more restoring orders than texture methods.

4.3. DTF for hybrid methods

The performances of DTF method versus hybrid inpainting methods are shown in Fig. 10. Fig. 10(b) shows Wexler et al. [16] with iteration rounds=5, neighborhood radius=3 and levels=3, whereas Fig. 10(f) shows the result of our DTF method with the same parameters. It is obvious that our DTF based method obtains better visual quality with PSNR gain by 3.51 dB than method [16]. As shown in Fig. 10(c), Bornemann et al. [42] performs well, although the pupil has sharp corner.

Fig. 11 shows other results of hybrid inpainting methods versus the DTF method. In Fig. 11(b) and (e), our DTF based method does achieve a better performance with PSNR gain by 2.95 dB than that of Wexler et al. [16]. In Fig. 11(c), Bornemann et al. [42] restores the green pepper in the top-right corner with sickle texture. Comparing with above results, it is not difficult to find that DTF method performs better as shown in Fig. 11(f).

The runtime of hybrid methods versus DTF method are summarized as: (1) Fig. 10(b) 121 s and Fig. 10(f) 140 s; (2) Fig. 10(c) 1 s and Fig. 10(g) 1 s; (3) Fig. 11(b) 1350 s and Fig. 11(e) 1564 s; (4) Fig. 11(c) 2 s and Fig. 11(f) 1 s. Note that DTF may consume more time than hybrid methods because more restoring orders are tested in the DTF method.

5. Conclusion and discussion

In this paper, we proposed a hybrid transform based DTF method for image inpainting. The DTF method consists of three components: chain transform part, restoration and diffusion part. Given an image, the DTF method is able to efficiently restore them on the proposed framework. In addition, we gave the theory foundation analysis of DTF method. In order to tackle interlacing problem, we designed boundary strategy and employed diffusion

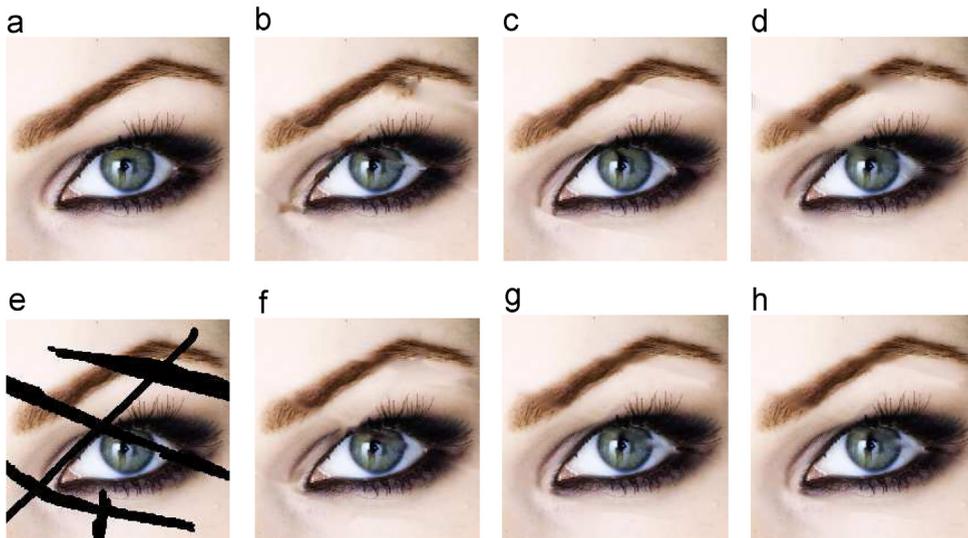


Fig. 10. Results of hybrid inpainting and DTF methods for eye. (b) and (f) illustrate results of Wexler et al. [16] and DTF method for comparison. Bornemann et al. [42] is presented in (c) and the DTF method is shown in (g). (d) and (h) are the results of DTF without/with pre- and post-processing, respectively: (a) original; (b) [16] 25.74 dB; (c) [42] 30.86 dB; (d) 28.36 dB; (e) scratch; (f) DTF 29.25 dB; (g) DTF 31.10 dB; (h) 31.10 dB.

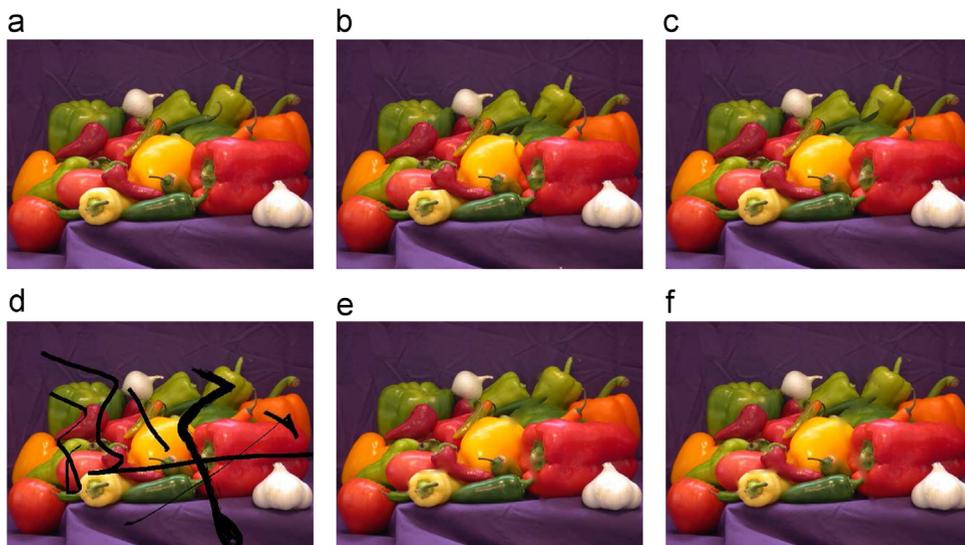


Fig. 11. Results of hybrid inpainting and DTF based method for Pepper. The restoration result of Wexler et al. [16] is shown in (b) and comparison result of our DTF method is shown in (e). Bornemann et al. [42] is shown in 11(f) and our DTF method is shown in (f): (a) original; (b) [16] 30.39 dB; (c) [42] 33.62 dB; (d) scratch; (e) DTF 33.64 dB; and (f) DTF 33.74 dB.

method to refine the results. Finally, we constructed the hybrid framework in a hierarchical way. Experimental results show that the DTF method greatly improves both the efficiency and performance of conventional methods.

Acknowledgments

The authors would like to thank A.C. Telea, A. Criminisi and Folkmar Bornemann for publishing their codes which provide great convenience for our experimental results.

This work is partially supported by NSFC (Grant no. 61073067) and by Hong Kong PGS.

References

- [1] W. Carey, D. Chuang, S. Hemami, Regularity-preserving image interpolation, *IEEE Transactions on Image Processing* 8 (1999) 1293–1297.
- [2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, Filling-in by joint interpolation of vector fields and gray levels, *IEEE Transactions on Image Processing* 10 (2001) 1200–1211.
- [3] J. Hwang, H. Lee, Adaptive image interpolation based on local gradient features, *IEEE Transactions on Signal Processing Letters* 11 (2004) 359–362.
- [4] N. Komodakis, Image completion using global optimization, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 442–452.
- [5] H. Ting, S. Chen, J. Liu, X. Tang, Image inpainting by global structure and texture propagation, in: *Proceedings of the 15th International Conference on Multimedia*, pp. 517–520.

- [6] N. Komodakis, G. Tziritas, Image completion using efficient belief propagation via priority scheduling and dynamic pruning, *IEEE Transactions on Image Processing* 16 (2007) 2649–2661.
- [7] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, *IEEE Transactions on Image Processing* 17 (2008) 53–69.
- [8] C. Hsieh, Y. Chen, C. Hsu, Fast image restoration method based on the multi-resolution layer, *Tamkang Journal of Science and Engineering* 12 (2009) 439–448.
- [9] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: *Proceedings of Computer Graphics and Interactive Techniques, SIGGRAPH 2000*, pp. 417–424.
- [10] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, *IEEE Transactions on Image Processing* 12 (2003) 882–889.
- [11] S. Rane, G. Sapiro, M. Bertalmio, Structure and texture filling-in of missing image blocks in wireless transmission and compression applications, *IEEE Transactions on Image Processing* 12 (2003) 296–303.
- [12] W. Yan, M. Kankanhalli, Erasing video logos based on image inpainting, in: *IEEE International Conference on Multimedia and Expo (ICME 2002)*, vol. 2, pp. 521–524.
- [13] W. Yan, J. Wang, M. Kankanhalli, Automatic video logo detection and removal, *Multimedia Systems* 10 (2005) 379–391.
- [14] K. Patwardhan, G. Sapiro, M. Bertalmio, Video inpainting of occluding and occluded objects, in: *IEEE International Conference on Image Processing (ICIP 2005)*, vol. 2, pp. 11–69–72.
- [15] K. Patwardhan, G. Sapiro, M. Bertalmio, Video inpainting under constrained camera motion, *IEEE Transactions on Image Processing* 16 (2007) 545.
- [16] Y. Wexler, E. Shechtman, M. Irani, Space-time completion of video, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007) 463–476.
- [17] P. Pérez, M. Gangnet, A. Blake, Patchworks: Example-Based Region Tiling for Image Editing, Technical Report MSR-TR-2004-04, Microsoft Research, Redmond, WA, 2004.
- [18] Y. Pritch, E. Kav-Venaki, S. Peleg, Shift-map image editing, in: *IEEE 12th International Conference on Computer Vision*, pp. 151–158.
- [19] C. Barnes, E. Shechtman, A. Finkelstein, D. Goldman, Patchmatch: a randomized correspondence algorithm for structural image editing, in: *ACM Transactions on Graphics (TOG)*, vol. 28, p. 24.
- [20] C. Wang, X. Sun, F. Wu, H. Xiong, Image compression with structure-aware inpainting, in: *2006 IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, IEEE, p. 4.
- [21] D. Liu, X. Sun, F. Wu, S. Li, Y. Zhang, Image compression with edge-based inpainting, *IEEE Transactions on Circuits and Systems for Video Technology* 17 (2007) 1273–1287.
- [22] Z. Xiong, X. Sun, F. Wu, Block-based image compression with parameter-assistant inpainting, *IEEE Transactions on Image Processing* 19 (2010) 1651–1657.
- [23] S. Masnou, J. Morel, Level lines based disocclusion, in: *IEEE International Conference on Image Processing (ICIP 2005)*, pp. 259–263.
- [24] S. Masnou, Disocclusion: a variational approach using level lines, *IEEE Transactions on Image Processing* 11 (2002) 68–76.
- [25] T. Chan, J. Shen, Mathematical models for local nontexture inpaintings, *SIAM Journal on Applied Mathematics* 62 (2001) 1019–1043.
- [26] T. Chan, J. Shen, L. Vese, Variational PDE models in image processing, *Notices of the American Mathematical Society* 50 (2003) 14–26.
- [27] T. Chan, J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, Society for Industrial Mathematics, 2005.
- [28] T. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions, *Journal of Visual Communication and Image Representation* 12 (2001) 436–449.
- [29] M. Bertalmio, A. Bertozzi, G. Sapiro, et al., Navier–Stokes, fluid dynamics, and image and video inpainting, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1.
- [30] M. Bertalmio, Strong-continuation, contrast-invariant inpainting with a third-order optimal PDE, *IEEE Transactions on Image Processing* 15 (2006) 1934–1938.
- [31] A. Bertozzi, S. Esedoglu, A. Gillette, Inpainting of binary images using the Cahn–Hilliard equation, *IEEE Transactions on Image Processing* 16 (2007) 285–291.
- [32] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, *IEEE Transactions on Image Processing* 13 (2004) 1200–1212.
- [33] A. Efros, T. Leung, Texture synthesis by non-parametric sampling, in: *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV 1999)*, vol. 2, pp. 1033–1038.
- [34] M. Fadili, J. Starck, Em algorithm for sparse representation-based image inpainting, in: *IEEE International Conference on Image Processing*, vol. 2, pp. 11–61.
- [35] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Transactions on Image Processing* 15 (2006) 3736–3745.
- [36] O.G. Guleryuz, Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising. Part II.: Adaptive algorithms, *IEEE Transactions on Image Processing* 15 (2006) 555–571.
- [37] X. Li, Image recovery via hybrid sparse representations: a deterministic annealing approach, *IEEE Journal of Selected Topics in Signal Processing* 5 (2011) 953–962.
- [38] Z. Xu, J. Sun, Image inpainting by patch propagation using patch sparsity, *IEEE Transactions on Image Processing* 19 (2010) 1153–1165.
- [39] A. Bugeau, M. Bertalmio, V. Caselles, G. Sapiro, A comprehensive framework for image inpainting, *IEEE Transactions on Image Processing* 19 (2010) 2634–2645.
- [40] A. Telea, An image inpainting technique based on the fast marching method, *Journal of Graphics Tools* 9 (2004) 23–34.
- [41] J. Sun, L. Yuan, J. Jia, H. yeung Shum, Image completion with structure propagation, *ACM Transactions on Graphics* 24 (2005) 861–868.
- [42] F. Bornemann, T. März, Fast image inpainting based on coherence transport, *Journal of Mathematical Imaging and Vision* 28 (2007) 259–278.
- [43] D. Cho, T. Bui, Image inpainting using wavelet-based inter- and intra-scale dependency, in: *19th International Conference on Pattern Recognition (ICPR 2008)*, pp. 1–4.
- [44] J. Shen, Inpainting and the fundamental problem of image processing, *SIAM News* 36 (2003) 1–4.
- [45] T. Chan, J. Shen, Variational image inpainting, *Communications on Pure and Applied Mathematics* 58 (2005) 579–619.
- [46] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1992) 259–268.
- [47] M. Wang, B. Yan, H. Gharavi, Pyramid model based down-sampling for image inpainting, in: *IEEE International Conference on Image Processing (ICIP 2010)*, pp. 429–432.
- [48] T. Chan, S. Kang, Error analysis for image inpainting, *Journal of Mathematical Imaging and Vision* 26 (2006) 85–103.