High Performance Missing Data Detection and Interpolation for Video Compression and Restoration Applications

Michele Ceccarelli [(1)], Giuliano Laccetti [(2)] , Alfredo Petrosino [(3)]

alfredo@ventotene.dma.unina.it

(1) Università del Sannio, Benevento, ITALY

(2) Università di Napoli "Federico II" Naples, ITALY

(3) National Research Council, Naples, ITALY

Error Concealment in Wireless Transmission

- During wireless transmissions packet loss can occur; some measurements report averages of about 3.6% of packet loss.
- Typical techniques are Forward Error Correction (FEC) and Automatic Retransmission Query (ARQ), which require extra error correction packets to be transmitted.
- Other approaches include the use of available information carried out by surrounding blocks or by nearby motion compensated frames

Digital film restoration

- Digital material is typically degraded due to physical problems in repeated projection or playback or simply the chemical decomposition of the original material.
- Typical problems: noise, dirt and scratches due to dust or abrasion.
- Manual retouching is highly required and automatic batch processing is limited to low degraded video sections (e.g. speckle noise, brightness variation)

Characteristic of Blotchy Noise

- Blotches hardly ever occur at the same spatial location in successive frames;
- The intensity of a blotch is significantly different from its neighbouring uncorrupted intensities;
- Blotches form coherent regions in a frame;
- They might NOT:
  - Be purely black/white
  - Have clear border

Possible causes:

- ❑ Dirt particles covering film
- ❑ Mishandling or aging of film
- ❑ Signal lossing due to transmission

Problems & Challenges

- Huge amount of data
  - Restrict computational complexity
  - Automatic processing preferred
- Motion estimation tricked by :
  - Presence of noise
  - Illumination Change
  - Blurry scene for fast motion
  - …
- Automatic detection not easy
  - Blotchy noise not readily modeled
  - Decision rely on motion compensated results

High performance computing

- A great improvement in this field should be related to the development of high performance software
  - to optimize the response time, too high to adopt the restoration algorithms in interactive modality
    - e.g. recent digital restoration of the movie, named Rory O'More and produced by Sydney Olcott in 1911, has taken 134 hours.
    - usual request is 0.1-0.2 fps
  - to allow the use of highly specialized algorithms in order to ensure (possibly) automatic or semi-automatic restoration of degraded video of great quality.

Blotch detection

Blotch domain

Ω

$$b(x,y,t) = \begin{cases} 1 & \text{if } (x,y,t) \in \Omega \\ 0 & \text{if } (x,y,t) \in D - \Omega \end{cases}$$

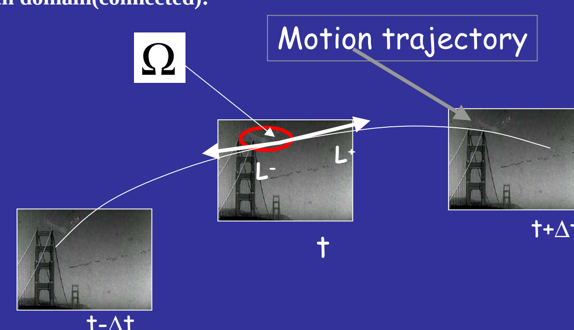
Degraded image

$$Z(x,y,t) = (1-b(x,y,t))\, I(x,y,t) + b(x,y,t)\, c(x,y,t)$$

$\sigma^2(c) < \varepsilon$: the variance of c is small

Blotch detection

Blotch domain(connected):



$$\left|\frac{dI}{dL^+}\right| > k$$

$$\left|\frac{dI}{dL^-}\right| > k$$

For all points (x,y,t) in Ω, the **directional derivatives** of I(x,y,t) computed along the motion trajectory across the two different directions $L^+(t \rightarrow t+\Delta t)$ and $L^-(t \rightarrow t-\Delta t)$ **are large**

Blotch Removal

- Inpainting = **Image Interpolation**.

  (**initially circulated among museum restoration artists; first introduced into I.P. by Sapiro's group [EECS, UMN, 1999]**)

- What makes inpainting difficult is the complexity of images**:**
  - having a large dynamic range of **scales**;
  - intrinsically non-smooth due to **edges and boundaries**;
  - the missing domains can have **complicated topology**;
  - direct classical interpolation tools perform less ideally:
    - polynomials (Lagrange, Hermite, splines);
    - linear filtering (Fourier, wavelets, linear (heat) diffusion);
    - radially symmetric functions (as in spatial statistics).

Degraded Video Restoration

- The general scheme is:

$$I^{n+1}(\mathbf{x}) \leftarrow I^{n}(\mathbf{x}) + \alpha \Pr(b(\mathbf{x})=1) \cdot CT$$

where *b*(.) is the *blotch mask,* α is a rate parameter and CT a Correction Term.

- Our choices:
  - Blotch Detection
    - Hard thresholding of $\Pr(b(\mathbf{x})=1)$: **Spike Detector Index** (SDI)
  - Blotch Removal
    - Spatio-temporal extension of the reaction–diffusion method for texture disocclusion (parameters α and CT) (Acton *et al.*, IEEE TIP, 2001)

Blotch detection method

- Motion estimation
- Spike detection index (SDI)
  - Initialize $b(\mathbf{x})$ field:

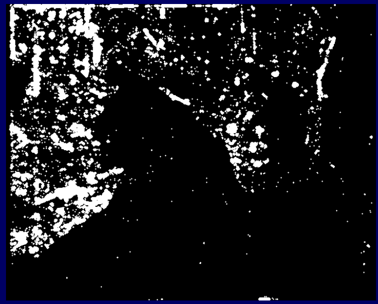
$$b(\mathbf{x}) = \begin{cases} 1 & \text{if}(|\Delta_f| > T_1) \text{ AND } (|\Delta_b| > T_1) \\ & \text{AND } ((sign(\Delta_f) - sign(\Delta_b)) < T_2) \\ 0 & \text{otherwise} \end{cases}$$

  - backward motion compensated frame difference $\Delta_b = I_t(\mathbf{x}) - I_{t-1}(\mathbf{x} + \mathbf{d}_{t,t-1}(\mathbf{x}))$
  - forward motion compensated frame difference $\Delta_f = I_t(\mathbf{x}) - I_{t+1}(\mathbf{x} + \mathbf{d}_{t,t+1}(\mathbf{x}))$

The SDI simply flag a pixel as corrupted when both the forward and backward motion compensated frame differences are higher than some (user selected) thresholds.

Blotch detection method

Original



Blotch mask

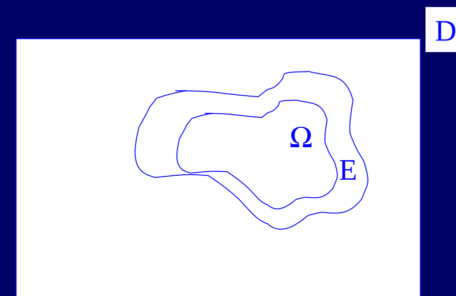
Reaction–diffusion method

$$\frac{\partial I(\mathbf{x})}{\partial t} = \rho_D(\mathbf{x})D(\mathbf{x}) + \rho_R(\mathbf{x})R(\mathbf{x})$$

- The reaction–diffusion method (Acton *et al.*, IEEE TIP, 2001) is able to recreate the graininess and orientation of the original texture.
- It well compares with Level lines methods producing in most cases halved MSE.

- Diffusion and reaction have conflicting objectives.
  - The goal of diffusion is smoothing, while the goal of reaction is pattern formation.

Reaction–diffusion method



Reaction–diffusion method

- The reaction–diffusion mechanism used for texture disocclusion is for a specific image location $\mathbf{x} = (x,y,t)$

$$I^{n+1}(\mathbf{x}) \leftarrow I^n(\mathbf{x}) + \rho_D(\mathbf{x})D(\mathbf{x}) + \rho_R(\mathbf{x})R(\mathbf{x})$$

- Seeding the region with noise identically distributed as the intensities of the surrounding region

$$I^0(\mathbf{x}) = \begin{cases} Z(\mathbf{x}) & \text{if } \mathbf{x} \in D - \Omega \\ R & \text{if } \mathbf{x} \in \Omega \end{cases}$$

- R is a random variable with density $H_E(i)/|E|$ where $H_E(i)$ is the intensity histogram for region E.

Reaction model

- In the reaction process, we encourage formation of patterns of a given granularity and directionality, corresponding to a localized area in the frequency domain covered by a specific Gabor filter **G**

$$R(\mathbf{x}) = \mathbf{G}_\mathbf{x} \otimes [\varphi(\mathbf{G}_\mathbf{x} * I)]$$

- $\mathbf{G}_\mathbf{x}$ is the Gabor filter matched to the dominant component at position **x**.

- Given $\mathbf{G}_i(\mathbf{x})$ Gabor filters, i=1,…, n.
- At each pixel, we define the dominant component as the one $\mathbf{G}_i(\mathbf{x})$ that dominates the response of the filter that maximizes the selection criterion

$$\Pi_i(\mathbf{x}) = \frac{|G_i(\mathbf{x})|}{\max_\omega |G_i(\omega)|}$$

- φ is selected as (Zhu and Mumford, IEEE TPAMI 1997)

$$\varphi(\xi) = -\left(1 - \frac{1}{1+(|\xi|/k)^2}\right)$$

Diffusion model

- Since anisotropic diffusion encourages intra-region, not inter-region, smoothing, the texture can be smoothed without eliminating edges.
- A discrete representation of anisotropic diffusion PDE is

$$D(\mathbf{x}) = \sum_{d=1}^{\Gamma} c_d(\mathbf{x}) \nabla I_d(\mathbf{x})$$

– $c_d(\mathbf{x})$ is chosen to be (Perona and Malik, IEEE TPAMI, 1990)

$$c_d(\mathbf{x}) = \exp\left\{-\left[\frac{\nabla I(\mathbf{x})}{k}\right]^2\right\}$$

k being the maximum contrast (intensity difference) within the texture pattern in the surrounding area **E**.

– $\nabla I_d(\mathbf{x})$ is the directional derivative (simple difference) in direction d at location $\mathbf{x}$.

Parameter setting

- The width of *E* is fixed as twice the maximum blotch width, ensuring that the width of the boundary region exceeds one full pattern period.
- The number of iterations was in the range 10-50.

Influence of artifacts on motion estimation

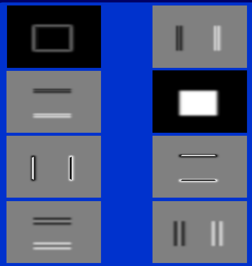
- Estimated motion vectors are less reliable when an image is blotched
- The reference data extracted from the motion-compensated reference frames and used for interpolating the missing data may be erroneous

- A **block matcher** will find the general direction in which data corrupted by blotches move
- **Multiresolution** allows reduction of blotch size, with consequent little influence on the block-matching results at the lower resolution levels
- At the higher resolutions, the blotches cover larger parts of the blocks used for matching, with consequent great influence on the matching results

Approach

- Region-based matching approach, where the best match amounts to maximising a similarity measure between features extracted from two frames
- Each feature is a feature vector extracted using locally tuned filters
- Repeated usage of two given images via the feedback iterative procedure improves the accuracy of optical flow considerably

(Laccetti, Marcellino, Petrosino, Parallel Computing, 2003)



Multiresolution Correlation Feedback Optical flow computation

- A correlation window is formed and centered at each pixel. 5x5 samples of error distribution around (u^n, v^n) can be computed by using the SSD. That is

$$E(u,v) = \sum_{m=-l}^{l} \sum_{n=-l}^{l} (I_t(x+m, y+n) - \Delta_b(x-u+m, y-v+n))^2$$

- 5x5 samples of response distribution can be computed as follows:

$$R_C(u,v) = c^{-kE(u,v)}$$

where k is chosen so as to make the maximum R_c among 25 samples of response distribution be a number close to unity.

Propagation

- The optical flow vector derived at this correlation stage is then calculated as follows, according to the weighted-least-squares estimation

$$u_c^n(x,y) = \frac{\sum_u \sum_v R_c(u,v)u}{\sum_u \sum_v R_c(u,v)} \qquad v_c^n(x,y) = \frac{\sum_u \sum_v R_c(u,v)v}{\sum_u \sum_v R_c(u,v)}$$

- Except in the vicinity of motion boundaries, the motion vectors associated with neighbouring pixels are expected to be similar.
- This constraint can be used to "regularize" the motion field

$$u^{n+1}(x,y) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} w_1(i,j) u_c^n(x+i, y+j)$$

$$v^{n+1}(x,y) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} w_1(i,j) v_c^n(x+i, y+j)$$

Experiment I: *Accuracy*

- Let image velocity **u**=(*u*,*v*) be represented as 3D directional vectors

$$\mathbf{V} = \frac{1}{\sqrt{u^2+v^2+1}}(u,v,1)$$

- The **angular error** between the correct image velocity $\mathbf{V}_c$ and the estimate $\mathbf{V}_e$

$$\psi_E = \arccos(\mathbf{V}_c \cdot \mathbf{V}_e)$$

- Three kinds of image sequences used are utilized here whose ground-truths are known.
- They are *Translating tree 2-D*, *Diverging tree 2-D*, and *Yosemite*
- The first two simulate translational camera motion with respect to a textured planar surface. The Yosemite sequence is a more complex test case; the motion in the upper right is mainly divergent.

Experiment I: *Accuracy*

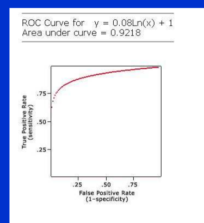
Techniques/Treet		Average error	Standard deviation	Density
Horn & Schunck		2,02°	2,27°	100%
Uras et al.		1,62°	1,52°	100%
Lucas & Kanade		2,65°	3,02°	100%
Fleet		2,56°	3,27°	100%
Nagel		2,44°	3,06°	100%
Anandan (l=3,w = 3)		4,54°	3,10°	100%
Singh (step 2, l=2, w=2)		1,25°	3,29°	100%
Our (l=4,w=3)		**1,45°**	**0,88°**	**100%**

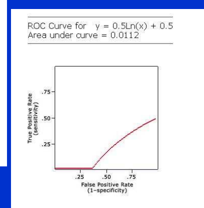
Techniques/Treed		Average error	Standard deviation	Density
Horn & Schunck		2,55°	3,67°	100%
Uras et al.		2,82°	2,73°	100%
Lucas & Kanade		2,76°	3,92°	100%
Fleet		6,54°	4,34°	100%
Nagel		2,94°	3,23°	100%
Anandan (l=3,w = 3)		7,64°	4,96°	100%
Singh (step 2, l=2, w=2)		8,60°	5,60°	100%
Our (l=4,w=3)		**2,45°**	**2,56°**	**100%**

Techniques/Yosemite		Average error	Standard deviation	Density
Horn & Schunck		11,26°	16,41°	100%
Uras et al.		10,44°	15,00°	100%
Lucas & Kanade		12,04°	14,45°	100%
Fleet		13,25°	14,03°	100%
Nagel		11,71°	10,59°	100%
Anandan (l=3,w = 3)		15,84°	13,46°	100%
Singh (step 2, l=2, w=2)		13,16°	12,07°	100%
Our (l=4,w=3)		**11,05°**	**10,50°**	**100%**

Experiment II: *Robustness*

- We compare the SDI detection ability, with different motion estimation techniques, plotting their Receiver Operator Characteristics (ROCs)
- ROC plots the false alarm rate versus the correct detection rate of a detector
- The curves were obtained by letting T_1 vary from 25 to 250, and fixing T_2 to 0.001.
- The ratio of correct detections to false alarms is large
- Four test sequences were used degraded by adding artificial blotches, with fixed image intensity value (missing data)
- The sequences are *Western*, originated by film, *MobCal*, *Manege* and *Tunnel*, recorded by modern cameras.





Example

t-1

t+1

Reference frame

t

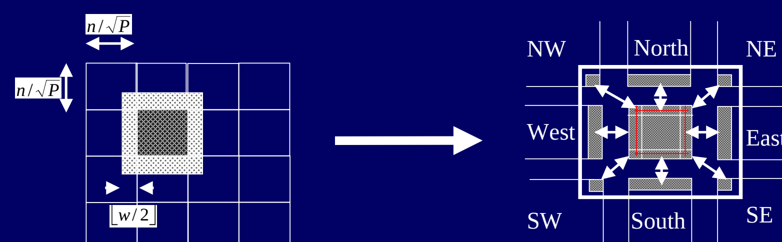
Example

Blotch detection and seeding the region with noise

Restored frame

Parallel algorithm

- Blotch Removal
  - All operations are window-based operations
  - Data parallelism can be naturally exploited



Parallel algorithm

- Blotch Detection
  - SDI computation: all the operations are pixel-based
  - Motion Estimation:
    - Backward and forward motion compensation are computed at each node in an asynchronous and multithreading manner

- Two kinds of motion vectors can be considered:
  - Local motion vectors involving pixels located in a single node
  - Non-local motion vectors involving pixels belonging to more than one node
- Each node is let to run asynchronously and task scheduling based on message arrival is performed

Multi-threading

- The processing at each node is divided into two categories:
  - Processing of local motion vectors can be performed indipendently from other nodes
  - Processing of a part of a non-local motion vector can only performed after performing all local motion vectors

- Multi-threading at user level
  - Whenever data useful for non-local motion vectors thread are exhausted, instead of idling, the node switches to the local motion vector thread.
  - Once new data arrives, the node switches to the global motion vectors thread

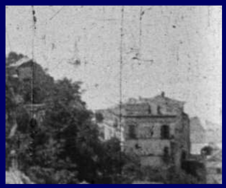
Multi-threading

- Two priority ready queues are adopted:
  - Tasks leading to the computation of non-local motion vectors
  - Tasks leading to the computation of local motion vectors

- The algorithm was implemented on a **Beowulf** (employing 30 Pentium nodes) and on the **SP2** (employing 14 RISC6000 nodes)
- The code was written using C and the MPI library (without parallel I/O)
- I/O time for loading and collecting the image data is not considered in the reported times

Test sequences

- Several motion picture sequences from:
  - the website Internet Moving Images Archive: Movie Collection (http://www.archive.org/movies/)
  - Kokaram book
  - kindly provided by Dyte s.r.l. (mutimedia data processing firm), Italy

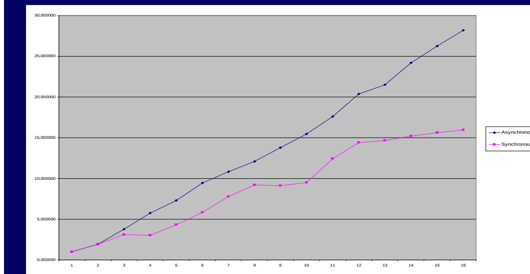


Test sequences

- Almost-static sequence with severe blotchy noise: (view of Golden Gate Bridge)
- Complicated scene with lots of motion and heavy noise (scene of people walking around on Golden Gate Bridge)
- Scene containing some apparent motion and also obvious noise (closing up show of man and a flying flag)
- Scene with lots of motion but less noticeable noise (children playing with swings and running)

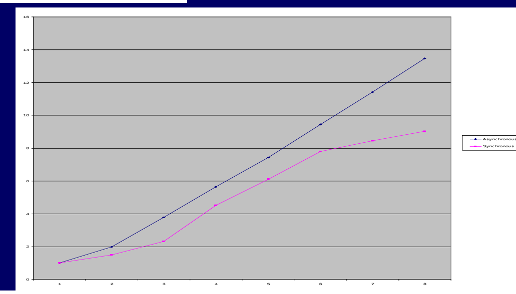



Experiment I: *Speedup Asy. vs Sy.*

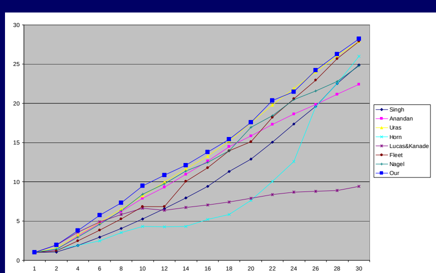


SP2

Beowulf

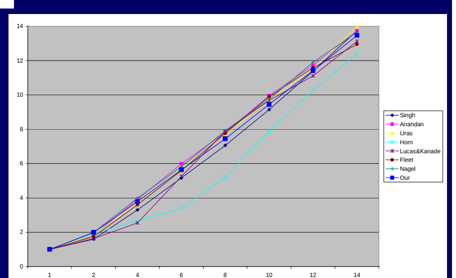


Experiment II: *Speedup*

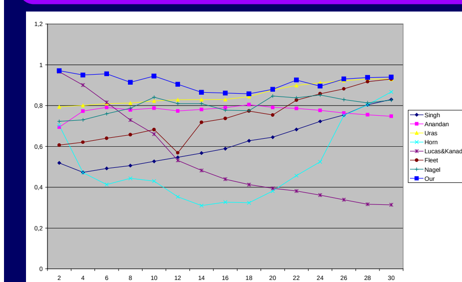


SP2

Beowulf

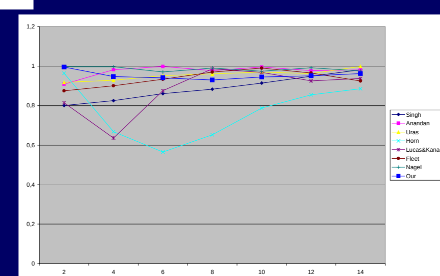


Experiment III: *Efficiency*



SP2

Beowulf



Experimental results

- Given a 256x256 blotched image, blotches can be restored in **2,910140** seconds on a 14-node SP2 and in **1,548352** on a 30-node Beowulf. A serial implementation takes **40,228437** seconds and **45,278277** on a single node of SP2 and Beowulf, respectively.
- For a given 256x256 blotched image, we obtained speed-ups of **13,82354** and **29,24288** using our algorithm on a **14**-node SP2 and a **30**-node Beowulf, respectively.

Some results

....

On going ...

- Stopping procedure based on differences between motion vectors computed between two consecutive levels.
- Detection whether motion exists so as to exclude unnecessary motion estimation.
- Motion may only exist in part of the frame, so the detection could be carried out in a section-wise manner and spatially adaptive to different regions of each frame.

Thanks for the attention ...