Motion tubes for the representation of images sequences

Matthieu Urvoy, Nathalie Cammas, Stéphane Pateux, Olivier Déforges, Marie Babel, Muriel Pressigout

MOTION TUBES FOR THE REPRESENTATION OF IMAGE SEQUENCES

M. Urvoy [a], N. Cammas [a], S. Pateux [a], O. Déforges [b], M. Babel [b], M. Pressigout [b]

[a] Orange Labs, 4 rue du Clos Courtel, 35512 Cesson-Sévigné, France

[b] IETR UMR CNRS 6164, Image and Remote Sensing Group / INSA of Rennes, 20 av. des Buttes de Coësmes, CS 14315, 35043, Rennes, France

ABSTRACT

In this paper, we introduce a novel way to represent an image sequence, which naturally exhibits the temporal persistence of the textures. Standardized representations have been thoroughly optimized, and getting significant improvements has become more and more difficult. As an alternative, Analysis-Synthesis (AS) coders have focused on the use of texture within a video coder. We introduce here a new AS representation of image sequences that remains close to the classic block-based representation. By tracking textures throughout the sequence, we propose to reconstruct it from a set of moving textures which we call *motion tubes*. A new motion model is then proposed, which allows for motion field continuities and discontinuities, by hybridizing Block Matching and a low-computational mesh-based representation. Finally, we propose a bi-predictional framework for *motion tubes* management.

Index Terms— Image sequences, motion tube, texture tracking, hybrid motion model

1. INTRODUCTION

Block-based representations have ruled the video coding world for a while. Standardized schemes, such as MPEG-x and H.264/AVC, use motion-compensated blocks to predict a frame from previously processed frames. Despite its efficiency, this model has been thoroughly optimized and getting significant improvements has become more and more difficult. Besides, the continuities of the motion field on the blocks boundaries cannot be represented by such a model, and the temporal persistence of the textures cannot be fully exploited.

To overcome those issues, some Analysis / Synthesis (AS) schemes have been proposed: textures are analysed, then image sequences are synthesized from the analysed textures. In [1], motion field continuities are handled by the use of triangular motion-adaptive meshes. However, meshes are not able to efficiently represent the motion field discontinuities, nor able to handle texture occlusions and disocclusions. Besides, it also suffers from the computational complexity of the compensation of meshes. Some other techniques propose the use of texture synthesis. Whilst they are not naturally fit to video coding, requiring strong constraints on textures to be efficient, a few have yet been adapted to video coding [2, 3].

In parallel, other works such as Motion Threads, then Barbell lifting, by defining motion threads along blocks [4] or pixels [5], showed how much important it is to catch the temporal persistence of textures in an image sequence. As for region based coders [6, 7], they aim at tracking given areas of the image sequence throughout a Group Of Picture (GOP), maximizing the use of a given textural information. However, they suffer from the complexity of regions representation and coding. Above all, it is a difficult task to provide an efficient and generic way to segment the regions to be tracked.

This paper proposes an AS representation of an image sequence naturally exhibiting the temporal persistence of both textures and motion, while handling textures occlusions and disocclusions. Our representation employs *motion tubes* as a basic structure. These have already been proposed to assess the quality of an image sequence [8]; yet they were not re-used over several successive frames and would simply aim at robustifying the motion field and improving the temporal consistency of the classification. The proposed motion model hybridizes motion-adaptive meshes with standard block-based representation, which allows for motion continuities and discontinuities, while limiting the complexity of the warping operation.

Section 2 will introduce the concept of *motion tubes*; section 3 will study their motion, and will propose a new motion model. Section 4 will bring about the management process of *motion tubes*. Finally, conclusions and perspectives will be given in section 5.

2. TOWARDS THE NOTION OF *MOTION TUBE*

While looking at a sequence of natural images, one can see that a texture is to be found in many successive frames, whether it has been translated, resized, rotated or warped (see fig. 1). Few are the representations that naturally exhibit this temporal persistence, none of which benefiting from the ease of the block-based representation.

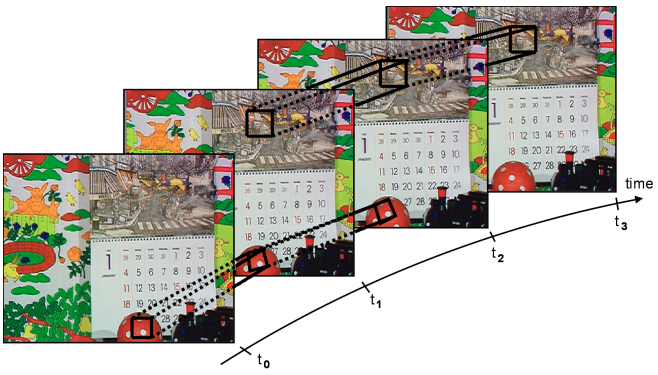


Fig. 1. A *motion tube*: temporal persistence of a given texture

As a preliminary step towards a coding scheme based on the use of such moving textures, this article proposes to track blocks of textures over a period of time, until these textures disappear. An image sequence is then represented by the set of all moving textures that fully reconstruct the sequence. Rather than using a frame as a basic element of an image sequence, a *motion tube*, structure embedding textural information along with its tracking data, will be used. It aims at modelling a spatio-temporal portion of the sequence.

Let $\mathbf{T}$ be a *motion tube*. It starts at an instant t_s and ends at an instant t_e: it has its own lifespan. In figure 1, the tube on the ball is created at $t_s = t_0$ and destroyed at $t_e = t_2$. $\mathbf{T}$ is characterized by its texture information, which can be refined at any instant, to cope with resolution losses and illumination changes. In this preliminary work, we assume that both previous phenomenons won't happen ; they will be later studied. A *motion tube* aims at maximizing the re-use factor of a given texture throughout an image sequence, optimizing its temporal persistence.

Let Ω_t be the support of $\mathbf{T}$ in the image I_t at time instant t, $t_s \leq t \leq t_e$. $\mathbf{T}$ is a $2D + t$ volume whose sections are $\{\Omega_t | t_s \leq t \leq t_e\}$. A set of warping operators $W = \left\{w_{t_i \to t_j} | t_i, t_j \in [t_s, t_e]\right\}$ are provided with $\mathbf{T}$, such that:

$$\Omega_{t_j} = w_{t_i \to t_j}\left(\Omega_{t_i}\right) \tag{1}$$

Warping operators can be composed, and:

$$w_{t_k \to t_j} \circ w_{t_i \to t_k} = w_{t_i \to t_j} \tag{2}$$

Finally, a motion tube is coded by its temporal information (t_s, t_e), its textural information, its texture updates, and its warping operators $w_{t_i \to t_j}$.

A sequence will then be represented by a set of *motion tubes*. Let $\mathcal{L}_t = \{\mathbf{T}_i\}, i \in [1, N]$ be a set of N tubes that exist at instant t. $\mathcal{L}_t$ can be updated according to any received control data: update of the motion information of a tube, removal of an existing tube from $\mathcal{L}_t$, or addition of a new tube to the set. Once $\mathcal{L}_t$ has been updated, $\overline{I}_t$ is then synthesized by rendering all the tubes from $\mathcal{L}_t$, such that:

$$\overline{I}_t = \bigcup_{i=1}^{N} \mathcal{R}\left(\mathbf{T}_i, w_{t_{ref} \to t}\right) \tag{3}$$

where $\overline{I}_t$ is the reconstruction of I_t, and t_{ref} the time instant when $\mathbf{T}_i$ has been initialized. $\mathcal{R}(\mathbf{T})$ operator aims at rendering $\mathbf{T}$ using appropriate ponderations. $\bigcup$ operator will combine all the tubes from $\mathcal{L}_t$ and might also reconstruct some of the unpredicted areas.

In figure 2, a sequence is reconstructed using 5 *motion tubes*. At t_0, 4 tubes are initialized. Due to the motion field continuity, they are kept connected until t_1. Later, the motion field discontinuities force them to be set apart. At t_2, the first motion tube is terminated due to its poor prediction. Back at t_1, a fifth tube needs to be added due to the apparition of an unpredicted area. Finally, complex motions appear at t_3: tubes 2 and 4 are warped to fit the motion field.

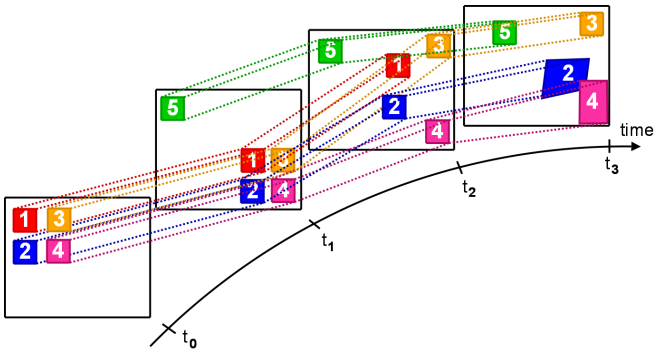


Fig. 2. An image sequence partially reconstructed by a few *motion tubes*

Motion tubes, by their nature, can benefit from the temporal persistence of moving textures: their tracking is simplified. In particular, the motion estimation of a tube, at a given instant, can be guided by its trajectory at previous or next instants. It will tend to reduce the discrepancies of the motion field, and the motion coding cost. It will also maximize the tube's lifespan, due to an enhanced tracking, thus minimizing the amount of textural information to be sent.

Furthermore, *motion tubes* can start and end at any time instant, hence fit appropriately the instants of apparition and disappearance of the tracked textures. An example of temporal management of the tubes is considered in section 4.

Motion tubes offer a very flexible way to represent a sequence, and allow for numerous possibilities. One can mention the possibility of multiple description: several families of tubes are used to represent a sequence. Scalability is also naturally handled by this representation. Spatial and quality scalabilities will be handled by several tubes of different sizes, resolutions or qualities, predicting the same area. Section 4 will illustrate the temporal scalability.

However, several obstacles are encountered while synthesizing a frame. As in Barbell Lifting [5], there will be unconnected and multi-connected areas. Some areas will be unpredicted, while some others will be predicted by more than one tube. The rendering operator $\mathcal{R}(\mathbf{T})$ will have to handle overlapping tubes; while $\bigcup$ operator will have to handle uncovered areas.

Finally, motion tubes can be either dependent or independent from each other, *i.e.* connected or disconnected. Disconnections will ensure the representation of ruptures within the motion field, while connections will keep its continuities. Section 3 proposes a motion model which combines a connected and a disconnected model.

3. ESTIMATION OF THE TRAJECTORY OF A TUBE

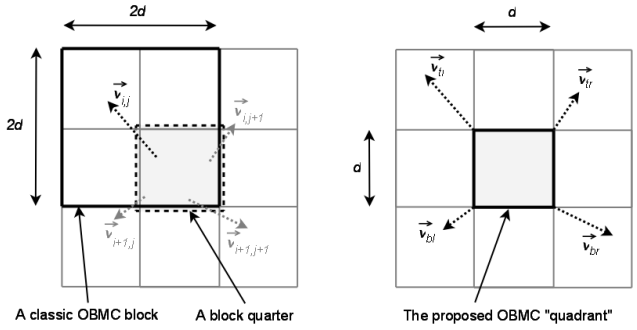
The proposed technique will estimate a tube's motion, sucessively from t_s to t_e. We now focus on the motion estimation between two consecutive instants, a problem which has been deeply studied: numerous warping models have been proposed. Classic Block Matching Algorithms (BMA) have been widely reproached their blocking artefacts, and would often require the use of a deblocking filter [9]. The Overlapped Block Matching Compensation (OBMC) [10] came as a solution, by using overlapping blocks, though it could not model with precision motion field ruptures, nor motion field complex warpings. Besides previous disconnected motion models, Control Grid Interpolation (CGI) [11] has also been proposed, as a connected model, but can only model a continuous motion field. Finally, hybrid models (Switched CGI and Switched OBMC [12]) were proposed to benefit from advantages of the different representations.

This paper proposes a competitive motion estimator, which optimizes the use of connected and disconnected motion models. Our estimator takes after SOBMC: it handles BMA and a modified OBMC, which we call Local Adaptive OBMC (LAOBMC). Moreover, the ability to connect or disconnect neighbouring tubes simulates the control points of a SCGI model: along with the LAOBMC, our model tends to behave like a SCGI.

Let's now focus on the proposed OBMC mode. $I_{t_{ref}}$ is split into $2d \times 2d$ overlapping blocks; each of these being half-overlain by its neighbours (see fig. 3(a)). Instead of considering the whole OBMC block, our approach considers its 4 quadrants independently. $I_{t_{ref}}$ is thus split into $d \times d$ blocks, each of these being an OBMC quarter-block overlain by the quadrants of 4 OBMC blocks. A motion tube is then initialized on each of these $d \times d$ blocks. This will simplify the hybridization of the different motion models.

Let $\mathbf{T}_{i,j}$ be a tube whose block coordinates are (i, j) in $I_{t_{ref}}$. Its support $\Omega^{i,j}_{t_{ref}}$ at instant t_{ref} is a square block. At each instant, a motion vector from the set $\Theta^{i,j}(t) = \left\{\vec{v}^{i,j}_{tl}(t), \vec{v}^{i,j}_{tr}(t), \vec{v}^{i,j}_{bl}(t), \vec{v}^{i,j}_{br}(t)\right\}$

is assigned to each of its corners (see fig. 3(b)). Let's now consider motion estimation at instant $t+1$, assuming that the motion of neighbouring tubes $\mathbf{T}_{i,j-1}$, $\mathbf{T}_{i-1,j}$ and $\mathbf{T}_{i-1,j-1}$ has already been estimated at $t+1$. $\mathbf{T}_{i,j}$ will inherit its motion vectors from its respective neighbours $\mathbf{T}_{i-1,j-1}$, $\mathbf{T}_{i-1,j}$ and $\mathbf{T}_{i,j-1}$, such that $\vec{v}_{tl}^{\,i,j}(t+1) = \vec{v}_{br}^{\,i-1,j-1}(t+1)$, $\vec{v}_{tr}^{\,i,j}(t+1) = \vec{v}_{br}^{\,i-1,j}(t+1)$ and $\vec{v}_{bl}^{\,i,j}(t+1) = \vec{v}_{br}^{\,i,j-1}(t+1)$. This will connect neighbouring tubes, each block vertex becoming similar to a CGI control point.



(a) Classic OBMC grid (b) Quadrant-based OBMC grid

Fig. 3. Our quadrant-based approach to OBMC

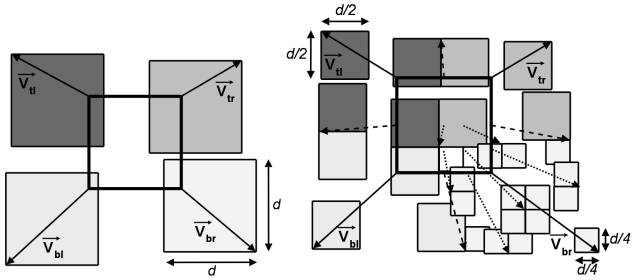
$\vec{v}_{br}^{\,i,j}(t+1)$, is then spatially or temporally predicted, which leaves to the estimation process the search of an optimal $\vec{v}_{br}^{\,i,j}(t+1)$. Compensation is finally done by translating a ponderated version of $\Omega_{t_{ref}}^{i,j}$ in the 4 locations defined by the motion vectors:

$$\Omega_{t+1}^{i,j} = \bigcup_{k=1}^{4} w_k \cdot \vec{t}_{\vec{v}_k^{\,i,j}} \left(\Omega_{t_{ref}}^{i,j} \right) \tag{4}$$

where w_k are appropriate OBM ponderating windows and $\vec{v}_k^{\,i,j} \in \Theta^{i,j}(t+1)$. $\vec{t}_{\vec{v}}$ is the translation operator of vector $\vec{v}$. Contributions for a given pixel are first added up, then normalized. Only one single vector per block needs to be transmitted: $\vec{v}_{br}^{\,i,j}$.

Yet, OBMC cannot catch up with complex warpings. We now introduce the LAOMBC, which tends to simulate a CGI model, and handles such motions. Whenever the 4 motion vectors of the current $d \times d$ block are too much different from each other, the block is split into four $\frac{d}{2} \times \frac{d}{2}$ sub-blocks. The motion vectors of the newly created vertices are derived from the 4 original vectors. The sub-blocks should now undergo a much simpler warping. If not, the above operation is iterated on the sub-blocks whose motion vectors are still too much different from each other. Finally, following (4), each sub-block is motion-compensated in a similar manner. Figure 4 shows the compensation of $\mathbf{T}_{i,j}$ using the classic OBMC representation and the proposed LAOBMC mode. $\Omega_{t_{ref}}^{i,j}$ is represented by the empty thick blocks, while $\Omega_{t_{ref}+1}^{i,j}$ is given by the translated grey blocks.

Despite its interesting features, LAOBMC cannot model the ruptures of a motion field. Hence, a switched model is designed, which hybridizes the previously described LAOBMC model with a classic BMA model. Motion discontinuities are handled by the BMA; the LAOBMC reduces blocking artefacts in motion-continuous areas. Furthermore, the connected / disconnected property of our model acts like the SCGI control points: it can be shown that, for little motion, OBM and CGI are alike. Moreover, the absence of warping operations makes our model less complex than the standard CGI. Table 1 presents both average PSNRs of reconstructed areas and reconstruction percentages obtained for different sequences and different



(a) OBMC mode (b) LAOBMC mode

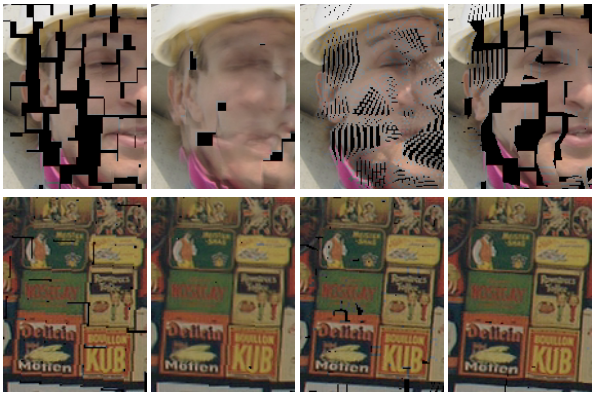
Fig. 4. The two OBM motion modes and their extensions

modes. At reference instant $t_{ref} = t_0$, I_0 is split into 8×8 blocks; a tube is then initialized on each of these blocks and tracked through a 16 frames GOP. Results are then averaged over the GOP.

Motion model	Foreman PSNR Rec. %	Mobile PSNR Rec. %	Crew PSNR Rec. %
BM	30.6 dB 82.8	26.4 dB 93.3	32.9 dB 81.1
OBM	24.1 dB 97.3	22.3 dB 98.7	29.1 dB 97.8
LAOBM	25.3 dB 88.5	19.6 dB 94.2	28.2 dB 86.1
BM/OBM	28.3 dB 91.5	26.2 dB 94.8	32.3 dB 89.8
BM/LAOBM	31.4 dB 83.6	27.2 dB 93.6	33.8 dB 78.6

Table 1. Efficiency of the different motion models on different sequences

OBM and LAOBM are poorly efficient when used alone. Their hybridisation with BM brings significant improvements to the BM model (increased recontruction percentage). Hybridized LAOBM significantly improves both PSNRs and visual quality (see fig. 5).



BM OBM LAOBM BM/LAOBM

Fig. 5. Reconstructions of Foreman's head and of a spinning texture from SpinCalendar using the different motion models

4. OPTIMIZATION OF THE RECONSTRUCTION

In this section, we present a general *motion tubes* framework, where the bi-prediction of *B-tubes* aims at maximizing the reconstruction while minimizing the number of tubes. We saw in section 2 that our two major issues were the overlapping tubes and the uncovered areas. Multi-predicted areas introduce redundancies, while unpredicted areas require the use of complementary information.

The proposed scheme takes after the bi-prediction mechanisms of standardized coders, which have proved to be very effi-

cient. Let's consider the representation of a GOP. A first family $\mathcal{L}_0 = \left\{\mathbf{T}_i^0, i \in [1, N \times M]\right\}$ of *motion tubes* is initialized at t_0. Tubes from $\mathcal{L}_0$ are then tracked until the start of the next GOP (t_{16} in figure 6). Eventually, some areas of I_{16} are uncovered by the tubes. A second family of tubes, $\mathcal{L}_{16} = \left\{\mathbf{T}_i^{16}, i \in [1, N \times M]\right\}$, is initialized at t_{16}, then tracked towards t_0. Most uncovered areas of the successive images are now predicted by the second family, and:

$$\overline{I}_t = \bigcup_{j \in \{0,16\}} \left(\bigcup_{i=1}^{N \times M} \mathcal{R}\left(\mathbf{T}_i^j, w_{t_{ref} \to t}\right) \right) \quad (5)$$

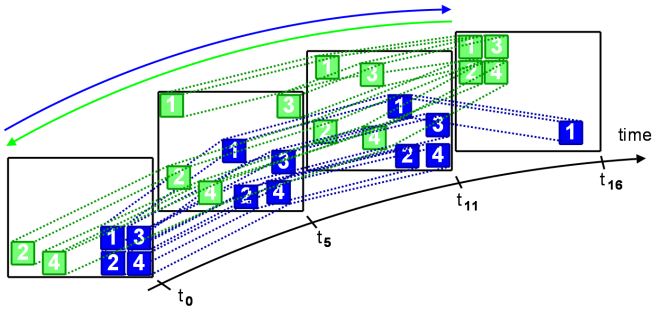


Fig. 6. Motion tubes bi-prediction

However, in order to limit tubes overlapping, redundant tubes are discarded. Remaining unpredicted areas can eventually be reconstructed by inpainting. Figure 6 illustrates the approach on a GOP of 16 images. This way, textures existing at t_0 will be tracked by $\mathcal{L}_0$, while textures existing at t_{16} will be tracked by $\mathcal{L}_{16}$. Table 2 presents the PSNRs obtained on the reconstructed regions along with the reconstruction percentages for different sequences and different families of tubes ($\mathcal{L}_0$ and $\mathcal{L}_{16}$). The hybrid BM/LAOBM motion model is used. It can be seen that the use of both $\mathcal{L}_0$ and $\mathcal{L}_{16}$ significantly improves the PSNRs and the reconstruction percentages. Figure 7 illustrates their impact on the reconstruction of the 8^{th} frame.

Tubes familie(s)	Foreman PSNR Rec. %	Mobile PSNR Rec. %	Crew PSNR Rec. %
$\mathcal{L}_0$	31.4 dB 83.6	27.2 dB 93.6	33.8 dB 78.6
$\mathcal{L}_{16}$	30.7 dB 83.6	26.7 dB 95.6	33.1 dB 76.5
$\mathcal{L}_0 + \mathcal{L}_{16}$	32.1 dB 93.1	28.4 dB 99.4	34.7 dB 93.1

Table 2. Efficiency of *B-tubes* on different sequences : masked PSNR and reconstruction percentage

As further work, the previous principle can be extended to hierarchical B-tubes: the process is iterated on each reference B-frame of a hierarchical GOP. A family of tubes $\mathcal{L}_8 = \left\{\mathbf{T}_i^8, i \in [1, N \times M]\right\}$ is created at t_8, then tracked towards both I_4 instants of reference, *i.e.* t_0 and t_{16}. Again, unneeded tubes are discarded. Finally, the same process is repeated on I_4 and I_{12}, then on I_2, I_6, I_{10} and I_{14}. Temporal scalability is handled by this approach.

5. CONCLUSIONS AND PERSPECTIVES

In this paper, we have presented a new framework for the representation of image sequences. By tracking textures throughout the sequence, motion tubes lead to a flexible model, and offer a compact representation of the sequence: they naturally exhibit the textures temporal persistence. Moreover, the proposed motion model allows for both motion continuities and discontinuities to be caught and efficiently represented, maximizing the quality and the reconstruction percentage. However, the major weaknesses of this representation are the overlapping tubes and the uncovered textures: these have been addressed by an efficient management of the tubes based on traditional bi-prediction mechanisms. Still, the selection mechanism deciding whether to keep or not a tube is rather delicate, and calls for further work. Indeed, the need for a way to assess the quality of a tube is vital for the sake of our representation. Used as a new coding mode in an existing coding scheme, an efficient way to compress sequences can be derived from this representation, along with an optimized coding of the tubes information.

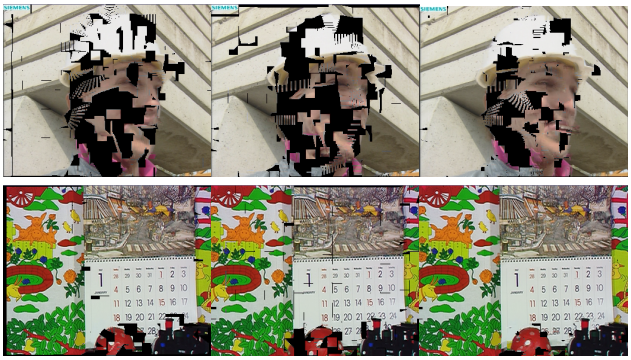


Fig. 7. Reconstruction of I_8 for different sequences and tubes families

6. REFERENCES

[1] N. Cammas and S. Pateux, "Fine grain scalable video coding using 3d wavelets and active meshes," in *SPIE Visual Communications and Image Processing*, Santa Clara, CA, Jan 2003.

[2] P. Ndjiki-Nya *et al.*, "Generic and robust video coding with texture analysis and synthesis," *IEEE ICME*, pp. 1447–1450, July 2007.

[3] C. Zhu *et al.*, "Video coding with spatio-temporal texture synthesis and edge-based inpainting," *IEEE ICME*, pp. 813–816, Apr. 2008.

[4] C. Zhu *et al.*, "Video coding with spatio-temporal texture synthesis," *Proceedings of ICME*, pp. 112–115, July 2007.

[5] R. Xiong *et al.*, "Barbell lifting wavelet transform for highly scalable video coding," in *Proceedings of PCS*, Dec. 2004.

[6] F. Marqués and P. Salembier *et al.*, "A SEgmentation-based coding System Allowing Manipulation of objEcts," in *IEEE ICIP*, Lausanne, Switzerland, Sep. 1996.

[7] P. Salembier, L. Torres, F. Meyer, and C. GU, "Region-based video coding using mathematical morphology," *Proceedings of IEEE*, vol. 83, no. 6, pp. 843–857, June 1995.

[8] S. Péchard *et al.*, "Video Quality Model based on a spatiotemporal features extraction for H.264-coded HDTV sequences," in *Proc. of PCS*, Lisbonne, Portugal, Nov 2007, p. 1087.

[9] G. Raja and M. J. Mirza, "In-loop deblocking filter for H.264/AVC video," in *Proc. of ISCCSP*, Mar. 2006.

[10] M. T. Orchard, "Overlapped Block Motion Compensation: an estimation theoretic approach," *IEEE Transactions on Image Processing*, vol. 3, no. 5, Sep. 1994.

[11] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proceedings of ICSSP*. IEEE, 1991, pp. 2713–2716.

[12] P. Ishwar and P. Moulin, "On spatial adaptation of motion field smoothness in video coding," *IEEE Transactions on circuits and systems for video technology*, vol. 10, no. 6, Sep 2000.