International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:4, 2008

MJPEG Real-Time Transmission in Industrial Environments Using a CBR Channel

J. Silvestre, L. Almeida, R. Marau, P. Pedreiras

***Abstract*— Currently, there are many local area industrial networks that can give guaranteed bandwidth to synchronous traffic, particularly providing CBR channels (Constant Bit Rate), which allow improved bandwidth management. Some of such networks operate over Ethernet, delivering channels with enough capacity, specially with compressors, to integrate multimedia traffic in industrial monitoring and image processing applications with many sources. In these industrial environments where a low latency is an essential requirement, JPEG is an adequate compressing technique but it generates VBR traffic (Variable Bit Rate). Transmitting VBR traffic in CBR channels is inefficient and current solutions to this problem significantly increase the latency or further degrade the quality. In this paper an $R(q)$ model is used which allows on-line calculation of the JPEG quantification factor. We obtained increased quality, a lower requirement for the CBR channel with reduced number of discarded frames along with better use of the channel bandwidth.**

***Keywords*— Industrial Networks, Multimedia.**

I. Introduction

SINCE the mid-nineties multimedia applications have evolved and grown in all environments due to advances in the different areas used by this technology: computer networks, codification algorithms, processing power, etc. The development of different compression standards such as, for example, JPEG (baseline, progressive, hierarchical and lossless modes of operation) [1], JPEG2000 [2], MPEG-2 [3], H.263 [4] and MPEG-4 [5], allowed coping with the requirements of different kinds of applications and reached, today, high compression rates with high quality and robustness.

The general algorithm of the compressors is presented in fig. 1. Compressors can be classified into two main types: still image compressors, such as JPEG and JPEG-2000 which use algorithms that exploit the spatial redundancy that exists in the images. On the other hand, video compressors exploit the temporal redundancy that exists in sequential images acquired with video frequency, typically between 24 and 30 images per second, reaching higher compression rates for the same quality. The more important factors in this process are the coding bit rate (R), and the quality obtained (D: Distortion), both of which depend on the quantification factor (q) used.

One important property of this type of traffic, with impact both on transmission and storage, is its variability in terms of load, thus falling in the Variable Bit Rate (VBR) category. However, most communication channels in real time industrial systems with resource reservation are of the Constant Bit Rate (CBR) type.

J. Silvestre is with the Computer Engineering Departament. Technical University of Valencia, Spain e-mail: jsilves@disca.upv.es
L.Almeida, R. Marau and P. Pedreiras are with University of Aveiro, IEETA, Aveiro, Portugal, e-mail: {lda,pedreiras,marau}@det.ua.pt

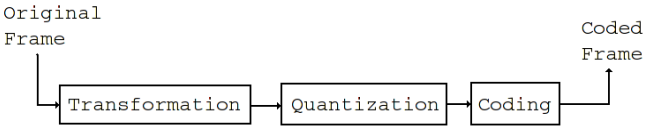


Fig. 1. Generic Codification Process

Although there are compressor algorithms with CBR support (see table I), the use of JPEG baseline is generally only found in digital cameras and monitoring systems (Lumera, Cast, Axis, etc.) mainly because of their low latency and low cost to implement. However they exhibit VBR behavior, which makes it difficult to reserve resources for image storage and transmission. This behavior is due in part to the use of the quantizer scale q on a frame by frame basis, instead of by macroblocks (usually 8x8 pixels).

To correct this behavior there are different algorithms, all of which introduce a latency delay or quality degradation. The smoothing video algorithms use memory buffers between the producer and the consumer to smooth out the bit rate variations [7]. With higher buffer capacities, and therefore greater delays, there is a higher probability of sending CBR traffic. This analysis is generally done off-line for the transmission of stored video, or through the buffer storage of a number of images before their transmission. Other algorithms are based on the q search in function of the R available in the CBR channel. However, these are iterative algorithms that increase the latency and in the best case, when faced with changes in the structure of the scenario[8], do not reach value q until the 3^{rd} iteration.

Another of the techniques used to adjust the MJPEG stream transmission is to change certain parameters, such as resolution or frame rate (drop or discard frames)[9]. Frame discard has a significant effect on the quality perceived in monitoring applications, particularly when there is a sequence of consecutive frames discarded. If the image has to be processed in the consumer, discarding consecutive frames can

TABLE I
Main Coders Properties [6]

Property	MJPEG	MJPEG2000	MPEG4
Motion Compensation	No	No	Yes
VBR	Yes	Yes	Yes
CBR Support	No	Yes	Yes
Latency	Low	Low to medium	Medium to High
Blocking Artifacts	Yes	No	Yes
Relative Cost	1x	3x	2x

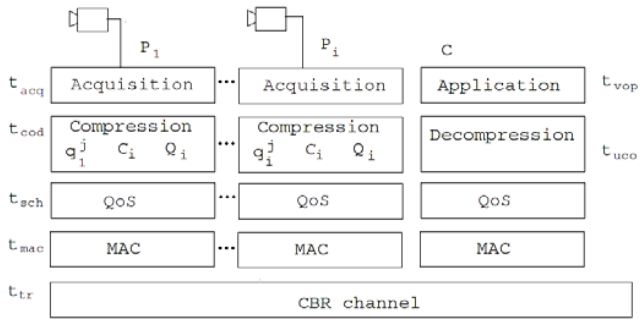


Fig. 2. System Architecture

produce serious errors in the receiver application. Changing the resolution, although acceptable in monitoring applications, is not considered useful where the image can be processed with computer vision algorithms.

II. System Architecture

Our scenario is a generic industrial application (see fig. 2), for monitoring [10] or image processing [11], where sources (P: Producer) send images through a local area network to a destination (C: Consumer). The latency L from the acquisition to the visualisation or image processing from each producer to the consumer is:

$$L = T_{acq} + T_{cod} + T_{mac} + T_{tr} + T_{mac} + T_{uco} + T_{vop} \quad (1)$$

where T_{acq} is the image acquisition time, T_{cod} is the time spent in compression, T_{mac} includes the time spent in the communication stack in each of the sides, including fragmentation and transmission on one side and reception and reassembly on the other, T_{tr} the transmission time, T_{uco} the time used for decompressing the image, and T_{vop} the processing or visualization time in the Consumer application level.

Each Producer i has a stream characterized by each j^{th} frame with size f_i^j compressed with the quantification level q_i^j; the bandwidth assigned R_i; the target quality Q_i; the resolution r_i; the acquisition period T_i and its deadline D_i ($D_i < T_i$). The buffer capacity B_i always allows storage of one image. The bandwidth restriction is

$$\frac{f_i^j}{T_i} < R_i, \forall_{i,j} \quad (2)$$

where f_i^j depends on r_i and on the q_i^j used in the j^{th} frame. Also, it must fulfill

$$L < T_i \quad (3)$$

Between the compressor and the MAC there is a QoS Quality of Service layer that is in charge of implementing the algorithm to adapt the source stream to the R_i available, e.g., discard frames and change q (fig. 2).

III. Scheduling rules in QoS/Scheduler level

A. Progressive compression

Some compressors can codify the information progressively by levels so that the consumer receives the most important (base) information before and the details subsequently. Thus, the more data received the better the final quality, i.e., including more details. Choosing a q that gives an f_i^j adapted to the R_i and T_i, the frames that exceed this value only can transmit R_iT_i bytes, losing in the consumer $f_i^j - R_iT_i$ bytes of the original compressed frame in P. This produces a degradation in the quality perceived but allows the frame transmission although eq. 2 is not fulfilled. In spite of this property, we discard their use since T_{cod} and T_{ucd} in JPEG progressive are much bigger than the values obtained with JPEG baseline, thus increasing the latency so that eq. 3 may not be fulfilled. However, we used this method, that provides better quality, to evaluate the degradation in quality of the algorithms we propose later on.

B. Frame Discard

This type of algorithm tries to find the best discard frame selection, usually depending on the cost of discarding each frame. There are algorithms based on the distance between the discarded frames, or in the minimization of the cost function. However, the simplest algorithm, known as *Just-in-time selective frame discard* (JITFD) [12], is the only one applicable in real time systems where $B_i = 1$ frame and $D_i < T_i$. Knowing the q value which gives us a mean $R(q)$ value lower that $R_i - 2\delta$, being δ a variation margin, the system will transmit all the frames that satisfy eq. 2, discarding the others independently of the distance to other discarded frames, or the cost that this discard implies.

C. R Models

The use of $R(q)$ models is one alternative. Through these models we can obtain the q value needed to generate a frame size f given T and R available in each moment. One of the models with better results define $R(q)$ as [13]:

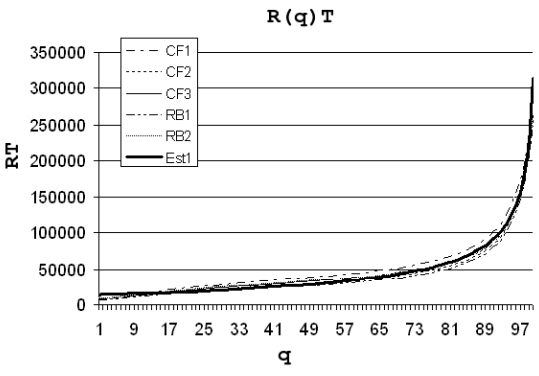
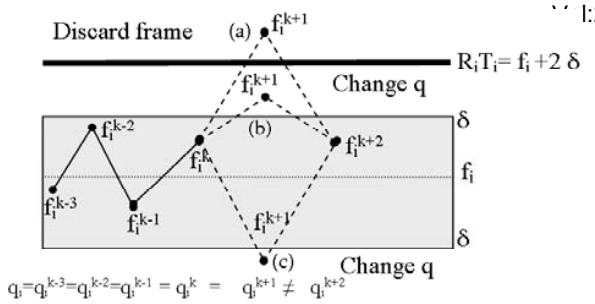


Fig. 3. Estimated and measured values of $R(q)$ scaled with T_i, the frame acquisition period



In the figure we can see how we use the same q until the following cases:
(a) We discard frame k+1 and adjust β and q for frame k+2
(b) f_i^{k+1} is too near of R_iT_i, β must be adjusted to reduce q and avoid future discard frames
(c) f_i^{k+1} is too low, β must be adjusted to increase q and avoid quality reduction in frame k+2

Fig. 4. q change

$$R(q) = \alpha + \frac{\beta}{q^\lambda} \tag{4}$$

where α and β are parameters of a curve where λ regulates the curvature of the same. This model was developed for MPEG, where $q = 1$ is the quantification factor that gives the higher quality and size, and $q = 31$ the lower. As we use JPEG where $q = 1$ gives us the lower quality and size, and $q = 100$ the higher, the model is adjusted using $q' = 101 - q$. In fig. 3 we can see the $R(q)$, scaled with T_i, obtained with the average R in the streams detailed in section IV. Each frame f_i^j has its own model (α_i^j,β_i^j,λ_i^j). There are algorithms to obtain $R(q)$ [14] which obtain better accuracy that eq. 4, but for their calculation they need compression with various q values (from 5 to 8 compressions), increasing the latency.

In a monitoring or processing image application with fixed cameras it can be assumed that the images acquired have a strong similarity between them. Therefore, it can be assumed that in the stream i $\alpha_i = \alpha_i^1 = \alpha_i^2 = ... = \alpha_i^j$ and $\lambda_i = \lambda_i^1 = \lambda_i^2 = ... = \lambda_i^j$. Giving the q_i value which means that stream i satisfy 2 until frame k, in the frame $k+1$ we can maintain the current use of $q_i^k = q_i$ if the frame size obtained is inside the target size (see fig. 4):

$$R_iT_i - 3\delta <= f_i^{k+1} <= R_iT_i - \delta \tag{5}$$

In the other case, the change in R between two frames k and $k+1$ can be calculated as:

$$\Delta R(q) = (\beta^{k+1} - \beta^k)q^{-\lambda} \tag{6}$$

Then if there is a scene change that produces R changes that do not fulfill eq. 5, we can calculate the new β^{k+1} value as: :

$$\beta^{k+1} = \Delta R(q)q^\lambda + \beta \tag{7}$$

TABLE II
STATISTICAL STREAMS PROPERTIES

CF1	$\hat{f}$	desv	max	min	PSNR	R_i(Mbps)
Q=20	22738.81	812.39	33579	20615	29.88	4.87
Q=40	33670.82	1228.52	50865	30590	32.42	7.22
Q=60	44309,98	1544,47	65924	40506	34.00	9.48
Q=80	66005.12	2115.09	94867	61101	36.36	14.0
CF2	$\hat{f}$	desv	max	min	PSNR	R_i
Q=20	17225.75	673.58	29033	15945	30.68	3.71
Q=40	25708.58	1042.2	44006	23825	33.00	5.56
Q=60	34587.59	1337.26	58140	32264	34.34	7.45
Q=80	53131.3	1834.52	85718	49880	36.29	11.4
CF3	$\hat{f}$	desv	max	min	PSNR	R_i
Q=20	18949.27	830.71	25142	15894	30.69	4.12
Q=40	28490.44	1227.65	38258	24602	33.15	6.20
Q=60	38105,71	1525.72	50807	33657	34.63	8.23
Q=80	57891.62	2033.94	75771	52524	36.69	12.4
RB1	$\hat{f}$	desv	max	min	PSNR	R_i
Q=20	15869.04	195.65	16303	15330	31.63	3.24
Q=40	23437.91	306.66	24083	22735	33.98	4.80
Q=60	31776.88	409.09	32611	30791	35.35	6.52
Q=80	49716.50	575.55	50933	48258	37.29	10.2
RB2	$\hat{f}$	desv	max	min	PSNR	R_i
Q=20	19493.32	569.46	20941	17330	30.54	4.13
Q=40	28555.84	799.56	30681	25572	33.07	6.03
Q=60	37554.01	983.83	40450	34054	34.06	7.90
Q=80	56190.58	1363.56	61264	51962	36.77	11.8

adjusting then q_i^{k+1} to reach the target R_i:

$$q_i^{k+1} = (\frac{R_i - \alpha}{\beta^{k+1}})^{(1/-\lambda)} \tag{8}$$

This new q will be used while eq. 5 is fulfilled in the next frames.

IV. EXPERIMENTS AND RESULTS

For the evaluation of the algorithm proposed in III-C the streams [15] *car factory 1* (CF1), *car factory 2* (CF2), *car factory 3* (CF3), *rubber 1* (RB1), *Rubber 2* (RB2) are used. All of them with r_i= 640x512 pixels, T_i=40msec., and 4500 frames long (3 minutes). Fig 8 and table II show, respectively, some representative frames and the main relevant statistical properties of these streams.

The stream compressed with a fixed q produces a high variability in the frame size f, as can be seen in fig. 5 where $q_i = 80$ is used in all the streams. RB1 stream is the more regular stream from the frame size point of view. RB2 has a similar behavior only in the first minute, changing to a higher variability. CF streams are characterized with a general greater variability, and also with peaks produced by welding operations of the robots.

The JPEG compression and decompression programs were done using Intel IPP 5.0 libraries [16] over Linux with a real-time kernel. The test was done using switched FTT-Ethernet (FTT-SE) [17] which provides CBR channels with controlled R_i to each Producer.

Fig. 3 shows the average $R(q) * T$ curve of the streams used, along with the curve generated with the R model with parameters (-30000, 345000, 0.45). It is assumed that we know $\hat{f}$ and the standard deviation σ, choosing $\delta = \sigma$ and as a target R_i, the values obtained for $q = 60$ in table II. The selection

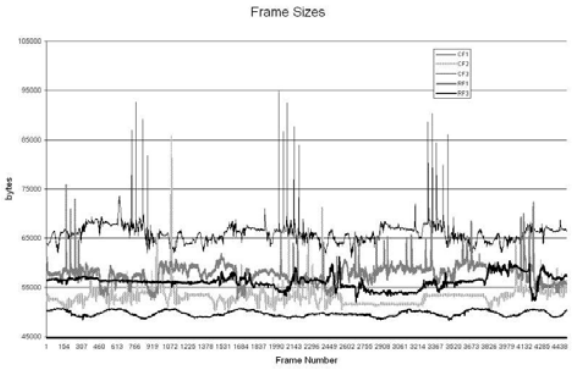


Fig. 5. Size evolution of streams with q=80

TABLE III
JITFD RESULTS

stream	frame discard	PSNR	BW lost	% BW lost
CF1	53	33.63	741kbps	7,8%
CF2	166	33.49	740kbps	9.9%
CF3	96	33.92	796kbps	9.6%
RB1	3	35.33	168kbps	2.5%
RB2	140	33.54	640kbps	8.1%

of a δ could have an important impact on the behavior of the system. A value too high can produce bandwidth waste, a value too low can introduce too many q adjustments and frame discards. The selection of $\delta = \sigma$ has given good results. However, it is necessary to develop a δ selection algorithm, since in many cases we don't know the σ value a priori.

In table III the results obtained in the streams transmission using JITFD can be seen. In all the streams with a medium high variability there is an important number of discarded frames, reducing the quality between 0.4 and 0.8dB, and this quality degradation is proportional to the number of discarded frames. In fig. 6 we can see how the discard of frames 833 and 834 produces a reduction in the quality in the second 33 (average of 31.10dB). In the second 25 the discard of frames from 624 to 643 produces a strong quality reduction due the reception of only 6 frames (average of 9dB). Also, a bad use of the R_i is detected, with a waste between 8-10%. This waste is produced by the frame discard (when $f_i^j > R_i T_i$), but also in the cases of smaller frames, (when $f_i^j < R_i T_i$).

In table IV the results using $R(q)$ and equations 7 and 8 are shown. The quality difference in the medium/high variability streams are -0.2dB, except in RB2 where we obtain +0.5dB. In fig. 6 the R_i and q evolution in seconds 25 and 33 of stream CF1 are shown. In this case $\Delta R(q = 57)$ produces a change to $q^{834} = 33$. As this q is calculated with the information of frame 833 we obtain an R_i lower that $\hat{f}$-3δ, but we recover in only two frames the adecuate q for the target R_i. This q adjustment means no frame loss in second 25, (PSNR=33.04dB) and discards only one in second 33 (PSNR=32.39dB).

Fig. 7 shows, for each stream, the evolution of the average of f using GOPs (Group of Pictures) of one second with the three algorithms. It can be seen how the frame sizes are higher

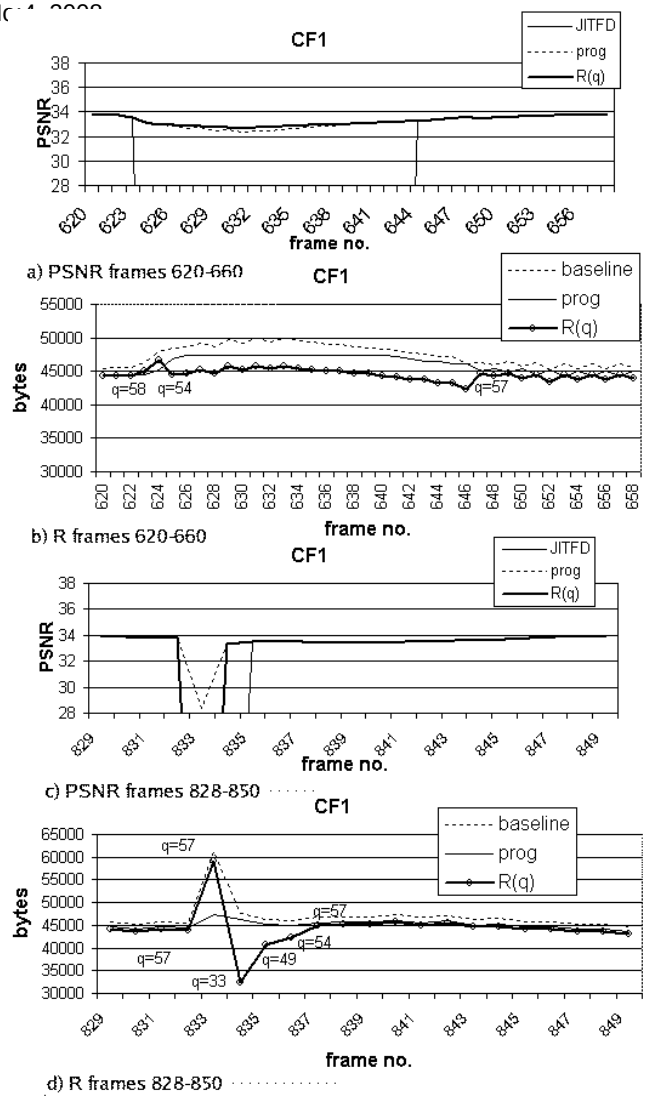


Fig. 6. PSNR and R*T in seconds 25 and 33 on CF1

with $R(q)$ model with respect to JITFD not only when there are discarded frames, but generally due to the use of a better q selection obtained through the model.

The number of discarded frames is reduced significantly. Although we cannot avoid some discarded frames, e.g. due to the important differences between frames when there is welding, in RB3 we practically avoid discarded frames. Also, in all the streams we achieve a better use of the assigne channel bandwidth, reducing its waste to 5-8%.

The values obtained with progressive JPEG are shown in table V. It can be seen how the quality is practically the same as in table II, although there is an increase in the bandwidth waste. This is due to the fact that with the same q, it produces a lower f (3%) with respect to baseline JPEG thus using the channel less efficiently. The quality is however higher because there are no frames discarded, when they are larger they are truncated to the channel width and the receivers uses only that amount of information. When this happens, JPEG progressive

is 100% efficient in using the channel but this happens seldom when compared to the cases in which the frame size is shorter.

V. Conclusion and future work

Industrial multimedia systems require the use of compressors with very low latency but also capable of producing CBR traffic without an increase in the latency. JPEG has a very low latency but it is VBR. The use of high values for R_i (CBR channel width), with respect to $\hat{f}+2\delta$, leads to waste of channel bandwidth, an important resource in industrial real time systems. The use of an R_i closer to $\hat{f}+2\delta$ reduces this bandwidth waste, but generates discarded frames that reduce the quality. To avoid this, there are well know algorithms but these increase significantly the latency. In this paper an algorithm for the generation of MJPEG streams based on $R(q)$ models is shown. Through this it is possible to obtain a good approximation to CBR traffic, reducing the number of discarded frames and increasing the quality in the consumer, in addition to a more efficient use of the synchronous bandwidth assigned to the CBR channel.

The correctness of the algorithm depends on the accuracy of the $R(q)$ parameters, which in this paper adjust correctly in the range $q = [35, 85]$. For this it is necessary to improve the calculation of the $R(q)$ parameters, and a method to obtain this on-line or in the initialization phase.

On the other hand, although the model adjusts well q for the target R_i, this produces a reduction in q and in the quality of the image. If these changes are not punctual, but hold during a period of time, it is neccesary to develop a mechanism to negotiate a new R_i value to restore the initial quality level. These issues will be considered in future work.

TABLE V
Progressive JPEG

stream	PSNR	BW lost	% BW lost
CF1	33,97	827kbps	8.7%
CF2	34.36	681kbps	9.1%
CF3	34.63	766kbps	9.3%
RB1	35.35	278kbps	4.2%
RB2	34.67	505kbps	6.3%

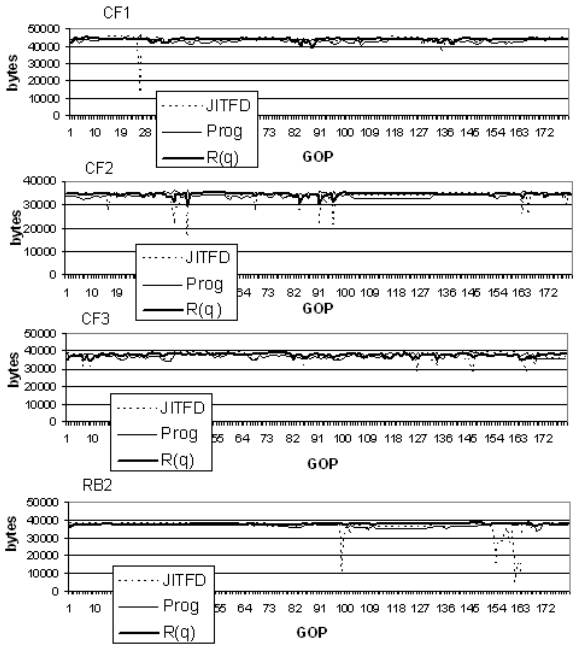


Fig. 7. R*T evolution

References

[1] G.K. Wallace *The JPEG still picture compression standard*, Commun. ACM, Apr. 1991, vol. 34, pp. 30-44.
[2] M.W. Marcellin, M.J. Gormish, A. Bilgin and M.P. Boliek, *An Overview of JPEG-2000*, Proc. DCC 2000. Mar. 2000, pp. 523-541
[3] D. LeGall, *MPEG: A video compression standard for multimedia application*, Commun. ACM ,Apr. 1991, vol. 34, pp. 46-58
[4] *Video Coding for Low Bit Rate Communications*, ITU-T, ITU-T Recommendation H.263, version 1. 1995
[5] T. Sikora, *The MPEG-4 video standard verification model*, IEEE Trans. Circuits Syst. Video Technol ,Feb. 1997, vol. 7, 19-31
[6] B. Jentz *Low-Cost Solutions for Video Compression Systems*, Global Signal Processing Expo, Santa Clara, CA, Oct. 2005
[7] S. Sen, J.L- Rexford, J.K Dey, J.F. Kurose and D. F. Towsley *Online Smoothing of Variable-Bit-Rate Streaming Video*, IEEE Trans. on Multimedia, Mar. 2000, vol. 2, no. 1, pp. 37-48
[8] A. Bruna, S. Smith, F. Vella and F. Naccari, *JPEG Rate Control Algorithm for Multimedia*, IEEE Int. Symp. on Consumer Electronics, Sep. 2004, pp. 114-117

TABLE IV
R(Q) results

stream	frame discard	PSNR	BW lost	% BW lost
CF1	20	33.87	649kbps	6.8%
CF2	21	34.21	583kbps	7.8%
CF3	24	34.45	661kbps	8%
RB1	2	35.33	172kbps	2.6%
RB2	1	34.6	391kbps	4.9%

[9] Z.L. Zhang, S. Nelakuditi, R. Aggarwal and R. P. Tsang *Efficient Selective Frame Discard Algorithms for Stored Video Delivery across Resource Constrained Networks*, INFOCOM, 1999, pp. 472-479.
[10] V. Sempere, T. Albero and J. Silvestre *Analysis of Communication Alternatives in a Heterogeneous Network for a Supervision and Control System*, Computer Communications, 2006, vol. 29, no. 8, pp. 1133-1145
[11] V. Sempere and J. Silvestre, *Multimedia applications in industrial networks: Integration of image processing in Profibus*, IEEE Trans. on Industrial Electronics, Jun. 2003 vol. 50, no. 3, pp. 440-448
[12] S. Nelakuditi *Localized Approach to Providing Quality-of-Service*, Technical Report 01-059. U. Minnesota, 2001.
[13] W. Ding and B. Liu *Rate control of MPEG video coding and recording by rate-quantization model*, IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 12-20, Feb. 1996
[14] L.J. Lin and A. Ortega *Bit-Rate Control Using Piecewise Approximated Rate-Distortion Characteristics*, IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 446-459, Aug. 1998
[15] J. Silvestre, V. Sempere and T. Albero *Industrial Video Sequence for Network Performance Evaluation*, IEEE Int. Workshop on Factory Communication Systems, Sep. 2004.
[16] S. Taylor *Intel Integrated Performance Primitives. How to optimize Software Applications Using Intel IPP*, Intel Press, 2004
[17] P. Pedreiras, P. Gai, L. Almeida and G.C. Buttazzo *FTT-Ethernet: a flexible real-time communication protocol that supports dynamic Q0S management on Ethernet-based systems*, IEEE Trans. in Industrial Informatics, Aug. 2005, Vol. 3, no. 1, pp. 162-172..

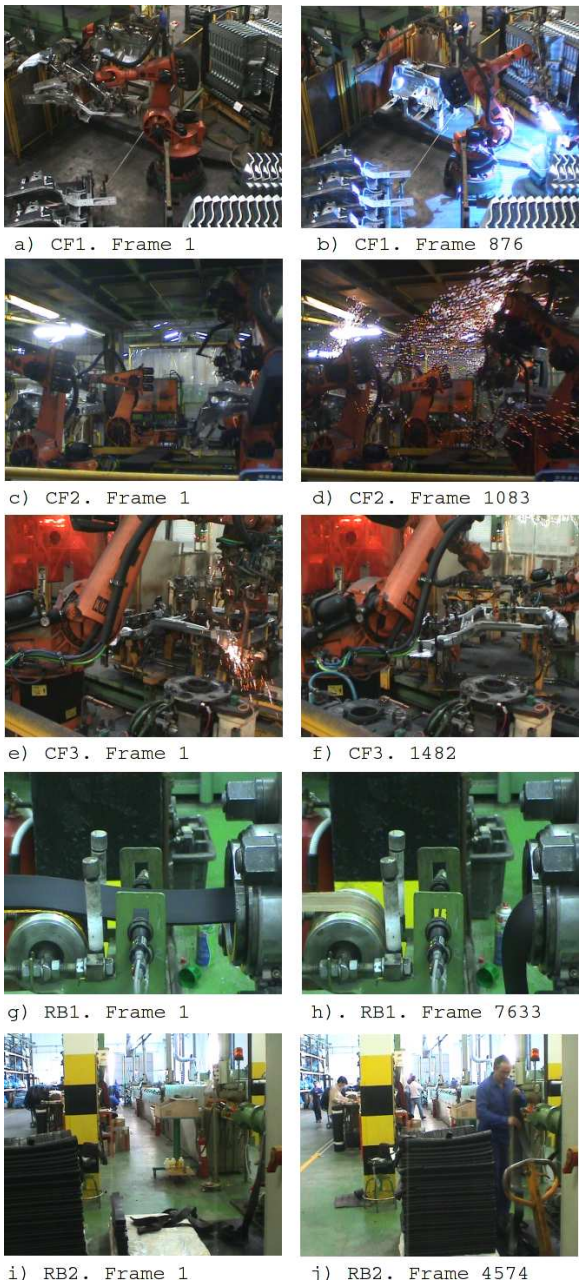


Fig. 8. Streams