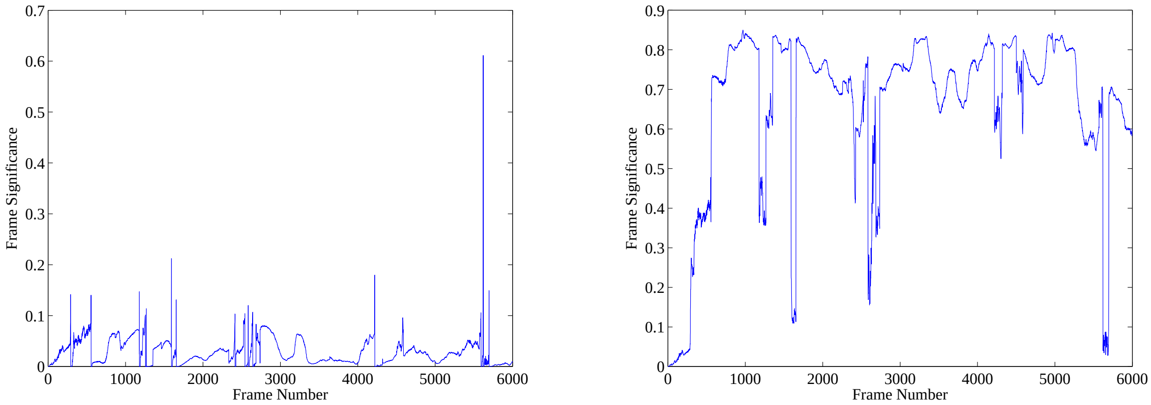

(a)

(b)

Fig. 1. Frame significance for the soccer sequence (a) rankThresh = 0.1, (b) rankThresh = 1

I. Online Algorithm

Before describing the ideas for designing an online algorithm, I need Wael to clarify some points. What is the impact of parameter *significanceThresh*, and could we get rid of it? What does updating the subspace mean and why do we need it? I have run the algorithm on different vidoes for two values of *significanceThresh* namely 0.1 and 1. I used the exact value for the rank, therefore *rankThreshold* is removed from the algorithm. As it is seen from the Figure 1 for the soccer video, using *significanceThresh = 0.1*, would sharply mark the shots boundaries. However looking at the frames extracted by the algorithm (in *pictureGroupStart*), most of them are the blurry ones. Why is it the case, and how do we fix it? I think we need to rewrite the code and simplify it a bit.

A. Algorithm Design

The basic idea is to identify the shots and summarize each according to the given summarization factor, α. The algorithm works in a similar fashion to the previous one. It starts by sliding a window over the seqeunce of frames. A feature matrix is formed and its exact rank is retrieved using SVD. This rank is then used to detect shot boundaries. The algorithm keeps track of all frames in the current shot. These frames are maintained in a max heap (or more efficiently a range tree) data structure using their significance as the search key. As frame significance may change when the window is moving, we use the last updated value for each frame. Once the shot end is detected, the top $N_s \times \alpha$ frames are extracted from the data structure where N_s is the number of frames in the shot.

Such a shot-based has several advantages. A shot is a meaningful building block of the video and it makes more sense for summarization to work on shots. There is some flexibility on which shots to include. We may also break longer shots for efficiency. In addition, it is possible to address user preferences, for example taking more frames from high motion shots and less from others.

Now let us discuss the parameters of the algorithm.

- I have the impression, that a larger window size handles special effects and fadings more efficiently. If this is the case, then even a window of size 2 can be used for videos that have clear cut shots? What is a reasonable value for window size and how do we justify it?
- rankThreshold is removed and its role replaced by the use of exact rank.
- What do we do with significanceThresh? (see Figure 1)
- I have been using only H and S with satsifactory results, any comments on this? I came across one other work that ignores V as well.

B. Quality Metric

In this part, I explain a quality metric that could capture the content of the frames unlike traditional metrics such as PSNR. The basic idea is to project the frames through SVD to a lower dimension subspace before measuring their distance. For each frame we construct the feature vector $x^t = [h_H h_S]^T$. Now we construct the feature matrix as $X^t = [x^t x^{t-1} \cdots x^{t-N+1}]$, where N is the window size, and t refers to the frame at the window start. The feature matrix is factorized using SVD.

$$X^t = U \Sigma V^T$$

We know from SVD, that each column in X^t is mapped to a column in V. In other words, a vector of size $L = h_H + h_S$ is mapped to a vector of size N containing only the most important information. Now we project all the frames along these vectors by constructing the following approximation of X^t:

$$X^t{}_N = \sum_{i=1}^{N} U_i s_i V_i^T$$

A simple metric (e.g. L_2) is then used to find the distance between any two columns in $X^t{}_N$.

Note 1: I am not sure if all these make sense it terms of math or appllication to our summarization framework, but it's an initial idea. Moreover, since it uses similar ideas to the algorithm, we may not want to use it?

Note 2: I belive we need a clear specification of how we define quality. For example, do we care about pixel shift, rotation, change in luminance, zoom, etc. Knowing this could help us define a very simple metric that performs well using traditional concepts such as PSNR, correlation, color histograms, etc.