*Intelligent Systems Conference 2018*
*6-7 September 2018 | London, UK*

# FMVR:Shall I Get My Video Back? Feature Matching based Video Reconstruction

Ekta Gujral
UC Riverside
egujr001@ucr.edu

***Abstract*—Video Reconstruction from static images has shown to benefit from using multi-aspect feature-based matching e.g. in form of dynamic systems to observe the behavior of the system over a certain period. An orthogonal line of work, broadly known as nearest neighbor, approaches the problem by extracting features from images. In this paper, we introduce FMVR, a novel Image matching based Video-Reconstruction. To the best of our knowledge, it is the 1st approach to incorporate multi-aspect information and feature-based matching, while being able to recover the video. We extensively evaluate FMVR's performance in comparison to state-of-the-art approaches across publicly available six short and two long videos datasets and demonstrate that FMVR, through combining feature matching along with multi-aspect behavior, outperforms the baselines in terms of reconstruction error and time.**

## I. Introduction

Feature detection and matching are important components of various computer vision applications; thus, they have received a significant attention in the last decades and they are being applied broadly in many applications like Image representation and retrieval [7], 3D scene reconstruction [10], motion tracking , and robot localization [1], all depend on the presence of stable and representative features matching in the image. Thus, detecting and extracting the image features are vital steps for video reconstruction applications.

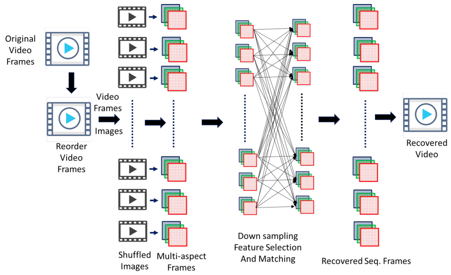

**Fig. 1: Image Matching-based Video Reconstruction**

In this moving and evolving world, it is difficult for researchers to capture the motions that occurs at smalled scale. It is required for sensory network to collect images as fast as possible to capture the system dynamics. But capturing images fast requires lot of energy. For example, in case of protein images, due to high energy demand, few samples destroyed in the process. This type of situation leads to restrict the researchers to capture any information on dynamics of protein. If we have enough sample images then the problem can be considered as a video reconstruction. To establish correspondences among a set of images to construct video, where feature correspondences between two or more image-frames are needed, it is necessary to recognize a set of salient points in each images.

Traditionally, research has focused on shape matching where it focuses on local extrema [15], [8] in a series of Difference of Gaussian (DoG) functions for extracting robust features. Another line of work is proposed by Abbas et al [12] namely ISOMAP which is a graph-based approach that reconstructs the sequence through a nearest neighbors algorithm after reducing the dimensionality of system. In [6],author used SIFT algorithm for remote sensing to match features between images or to localize and recognize objects. We show a snapshot in figure 1 of our algorithm namely FMVR for reconstruction of video. FMVR uses image feature-based matching to find and for computing distinctive invariant local features between frames and uses efficient sorting for reordering them based on the calculated match metrics. Our contributions include:

- **Novel Approach**: We introduce FMVR, a feature matching-based video reconstruction algorithm. Under the hood of FMVR runs our proposed algorithm for feature extraction & matching and reordering of frames.Our parallel reordering algorithm is divided into small blocks to generalize the solution for complex video reconstruction.
- **Experimental evaluation**: We conduct extensive experiments in order to evaluate FMVR's performance in comparison to state-of-the-art methods that either use compressive sensing, or scale-invariant feature transform information. In Section IV we demonstrate that FMVR perform similar or better as compared to other baselines.

The rest of the paper is organized as follows. In Section II, we formally introduce our problem and section III, outlined our proposed method FMVR. Further, in section IV we present our experimental evaluation and Section VI we conclude with a few remarks for future extensions of this work.

## II. Problem Formulation

**A. Preliminary Definitions:** Table I contains the symbols used throughout the paper.

| Symbols | Definition |
|---|---|
| $\mathbf{X}, \mathbf{x}, x$ | Matrix, Column vector, Scalar |
| $\mathbb{R}$ | Set of Real Numbers |
| $f$ | frames |

**TABLE I: Table of symbols and their description**

**B. Problem Definition:** Consider an inverse problems in which one seeks to recover an video $\mathbf{V} \in \mathbb{R}^{m \times n}$ from a collection of measurements $\mathbf{I} \in \mathbb{R}^{m \times n}$ where R is collection of images. While the proposed approach is more general, for ease of discussion and interpretation, we will focus on the case in which the measured data are related to the true image via $V = IH$ where $\mathbf{H} \in \mathbb{R}^{n \times n}$ is matching feature points of image frames.

> **Problem Definition. Given** set of images $\mathbf{I}(t_1) \ldots \mathbf{I}(t_N)$, **find** best ordering of the images to reconstruct the video $\mathbf{V}$ in faster and accurate manner.

In the context of feature matching, the first step of any matching system is to detect interest locations in the images and describe them. Once the descriptors are computed, they can be compared to find a relationship between images for performing matching tasks. We combine strongest matching frames together based on the distance between two image multi-aspect frames and merge them to reconstruct the video.

## III. Proposed Method

As we mention in the introduction, there exists a body of work in the literature that is able to efficiently reconstruct the video in the presence of incoming images frames [10], [12], [14]. However, those methods are slow, and eventually are not able to reconstruct the video with large images volume. In this paper we propose FMVR, which takes a different view of the solution. Our algorithm is two step process and includes feature extraction and reordering of frames. Our reordering algorithm is spitted into small chunks to generalize the solution for complex video reconstruction.

**A. STEP 1: Feature Extraction:** The algorithmic framework we propose is shown in Figure 1 and is described below: For reconstruction video from the images, we took original video from Youtube and then shuffled the video frames to use it for feature extraction. Using this randomized video, we extracted the images for each frame i.e. $I(t_1) \ldots I(t_N)$. Each image frame is then converted to tensors or multi-layer matrices. The layers of tensor represent the RGB values of image frame. For example, for $I(t_1)$ frame, $1^{st}$ layer shows the Red component values in the image, $2^{nd}$ layer represents the Green component of image and finally, the $3^{rd}$ layer shows the blue component of image. Then to make the process fast, the tensorized images are down-sampled to remove static background. Finally the features are extracted by approximates Laplacian of Gaussian (LoG) with Box Filter and by using wavelet responses in horizontal and vertical direction as shown in Figure 2 below. For each sub-region of frame, wavelet responses for horizontal and vertical are considered and a vector is formed as $I = (\sum i_x, \sum i_y, \sum |i_x|, \sum |i_y|)$. To improve the feature detection we use sign of Laplacian for underlying interest point. The distance metrics are calculated only from the extracted features which helps to reduce the input data dimensionality. In the matching stage, we only compare features if they have the same type of likelihood. This can be calculated by considering the number of feature matches between frames and the displacement metrics of the matches. Figure 3 shows the outcomes for this step.

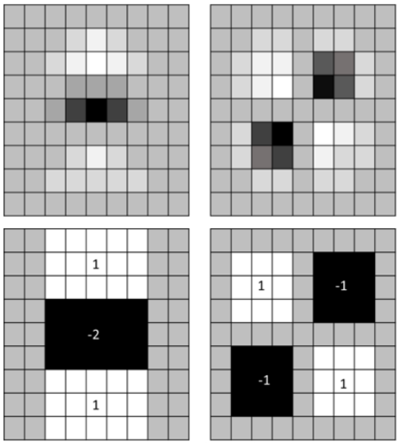

**Fig. 2: Laplacian of Gaussian (LoG) with Box Filter approximation**

**Algorithm 1:** FMVR for Video Reconstruction

```
Input: set of Images $I(t_1) \ldots I(t_N)$.
Output: Reconstructed Video $\hat{V}$.
Create frame groups $f \in I$. $X_F$ is frame matches between frames i and j.
for i=1 . . . N do
    for j=1 . . . N do
        $I_i, I_j \implies$ find strongest match.
        if $(I_i, I_j) \in$ same $f$ then
            $X_F(I_i, I_j) = -1$
        else
            Combine $f_{i,j}$ containing $I_i$ and $I_j$
        end if
    end for
end for
$\hat{V}$ = merge frames with strongest match from $X_F$
return $\hat{V}$
```

**B. STEP 2: Parallel Reordering and Matching:** Given the distance metrics between frames, the reordering algorithms 1 is implemented to reconstruct the video frames in the most likely sequence. It takes into account every possible pairing of frames and sorts by combining the strongest matches until it results in a single continuous sequence. The sorting algorithm is done by looking at all the available frames that can be matched, finding the most powerful match, and then combining the associated sequences.Reordering algorithm takes its time for calculating all the feature matches between each pair of frames which scales as $O(N^2)$. However these calculations can be done in parallel which greatly speeds up the algorithm. So upper bound for this become $O(N^2M^{-1})$ where M is number of parallel execution.

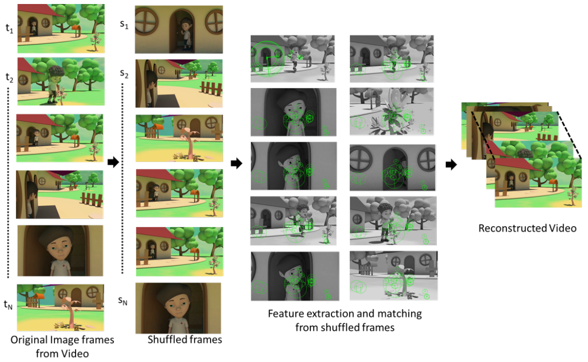
t1
t2
tN
s1
s2
sN
Original Image frames from Video
Shuffled frames
Feature extraction and matching from shuffled frames
Reconstructed Video

**Fig. 3: FMVR response for feature detection and matching process**

## IV. Experimental Evaluation

Matlab implementation and the data we used for experiments are available at link[1].

**A. Data-set description:** In order to truly evaluate the effectiveness of FMVR, we test its performance against six short and two long you tube video datasets. Those datasets are summarized in Table II and are publicly available at http://www.youtube.com.

| Type | Videos | Size (MB) | Length (sec) |
|---|---|---|---|
| Short | At The Opera | 2.304 | 54 |
| | Funny Cat | 2.357 | 52 |
| | Life in 1 Min | 1.538 | 69 |
| | Sunrise | 3.773 | 78 |
| | Talent | 2.592 | 60 |
| | Visual | 2.427 | 58 |
| Long | Suit | 122.574 | 1318 |
| | Piper | 209.487 | 3788 |

**TABLE II:** Table of real datasets analyzed

**B. Baselines:** Here we briefly present the state-of-the-art baselines we used for comparison.
**IsoMap-Nearest Neighbor [12]**: IsoMap-Nearest Neighbor is a graph-based approach that reconstructs the sequence through a nearest neighbors algorithm after reducing the dimensionality of system.
**Random Seed-SIFR [6]**: The SIFT i.e. scale-invariant feature transform algorithm is widely used in computer vision and in remote sensing to match features between images or to localize and recognize objects. The algorithm RS-SIFR select random seed from pool and match with closest pair until all frames are matched.
**CSVideoNet [17]**: The CSVideoNet algorithm directly learns the inverse mapping of compressive sensing and reconstructs the original input in a single forward propagation.

**C. Evaluation Metrics:** In order to obtain an accurate picture of the performance, we evaluate FMVR and the baselines using three criteria: Relative Error, Wall-Clock time and Fitness Score. These measures provide a quantitative way to compare the performance of our method. More Specifically, **Relative or Reconstruction Error** is effectiveness measurement and defined as :

$$RelativeError = \frac{||\mathbf{V}_{original} - \mathbf{V}_{reconstructed}||}{||\mathbf{V}_{original}||}$$

where, the lower the value, the better the approximation.
**CPU time (sec)** indicates how much faster does the reconstruction runs as compared state-of art methods.The average running time denoted by $T_{tot}$ for processing all frames for video, measured in seconds, and is used to validate the time efficiency of an algorithm.
**Fitness Score** indicates the logistic performance and score for frames and can be calculated as

$$S(f_1, f_2) = \frac{1}{1 + e^{(-\frac{1}{2}(|f_1 - f_2| - f_0))}}$$

where, the lower the value, the better the approximation.

**D. Evaluation:**

**1) Comparison with baseline:** For all datasets we compute Reconstruction Error,CPU time (sec) and Fitness Score. The results for the all video data are shown in Table III, the best result is shown in bold. We observe that FMVR performed better than other approaches when applied on ”Sunrise” and ”Talent”. The most interesting comparison, however, is on the large video datasets, since they present more challenging cases than the short video datasets. PIPER is with large size and has high number of variations in adjacent frames.Given that, we found that FMVR achieved the better performance because of its accurate adjacent frame matching capability with no pre-knowledge of the direction. of the video.

The execution time is mostly contributed by the feature match calculations.The figure 4 shows the experimental results for sequential and parallel match.Then the feature matching time reduced approximately 60-70%.

**2) Scaling and Frame Skipping:** We evaluated the performance of FMVR with changing the scale on down sampling of the input images. By reducing the image size that is used for feature extraction, the initial matching runs faster and with a lower memory footprint.The result is shown in Figure 5 for all datasets.

To evaluate the performance of reordering algorithm to handle large frame distance, we evaluate the algorithm by skipping frames in the input images from video before

[1] LinkforCode

| Metric | Method | At The Opera | Funny Cat | Life in 1 Min | Sunrise | Talent | Visual | Suit | Piper |
|---|---|---|---|---|---|---|---|---|---|
| Relative Error[0-1] | ISOMAP-NN | **0.072** | 0.329 | 0.562 | 0.159 | 0.226 | 0.408 | 0.398 | 0.304 |
| | RS-SIFR | 0.234 | 0.302 | 0.468 | 0.123 | 0.178 | 0.341 | 0.356 | 0.302 |
| | CSVideoNet | 0.225 | **0.231** | 0.373 | 0.057 | 0.125 | **0.247** | 0.258 | 0.245 |
| | FMVR | 0.126 | 0.237 | **0.288** | **0.053** | **0.078** | 0.251 | **0.167** | **0.206** |
| Fitness Score [0-1] | ISOMAP-NN | **0.217** | 0.104 | 0.243 | 0.284 | 0.115 | 0.092 | 0.168 | 0.185 |
| | RS-SIFR | 0.239 | 0.106 | 0.229 | **0.224** | 0.11 | 0.086 | 0.161 | 0.172 |
| | CSVideoNet | 0.231 | **0.09** | 0.213 | 0.255 | 0.098 | **0.064** | 0.144 | 0.162 |
| | FMVR | 0.225 | 0.091 | **0.211** | 0.235 | **0.084** | 0.067 | **0.135** | **0.149** |
| CPU Time (s) | ISOMAP-NN | 452.729 | 375.646 | 1013.672 | 1048.696 | 1326.846 | 1466.891 | 3779.122 | 4287.181 |
| | RS-SIFR | 415.197 | 375.294 | 989.429 | 1010.053 | 1322.422 | 1430.874 | 3759.809 | 4271.978 |
| | CSVideoNet | 411.231 | 366.601 | 961.934 | 987.082 | 1289.316 | 1391.843 | 3756.464 | 4261.461 |
| | FMVR | **373.17** | **358.61** | **918.74** | **978.67** | **1267.6** | **1358.97** | **3732.89** | **4231.89** |

**TABLE III: Experimental results for Relative Error, CPU Time (s) and Fitness Score. FMVR mostly outperforms baselines and, in particular, works better in very hard scenarios such as the Suit and Piper video dataset.**

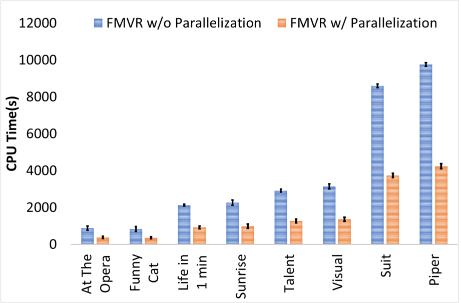

**Fig. 4: FMVR performance (CPU Time(sec)) with and without parallelization**

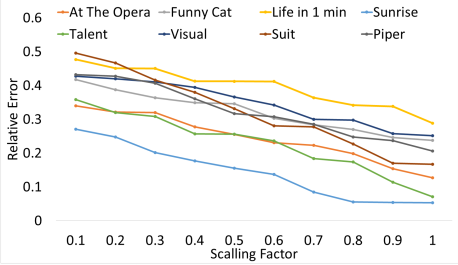

**Fig. 5: FMVR performance (Relative Error [0-1]) with changing input image scales**

shuffling it. The experimental results are shown in Figure 6.

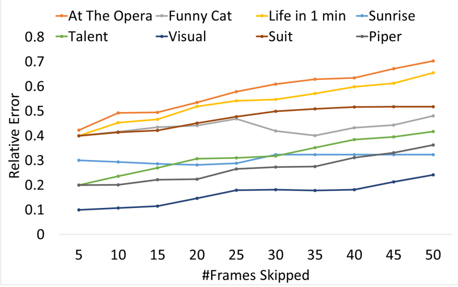

**Fig. 6: FMVR performance with changing skipping input frames**

## V. Related Work

- **Feature Matching approaches** : Reconstructing video from image frames using feature matching technique is not a widely researched. The applications are primarily focused around academic or research communities like protein analysis etc. The ISOMAP technique proposed by Abbas Ourmazd et al. [12] gained popularity for reconstruction of video from images. It is a graph-based approach that reconstructs the sequence of images using a nearest neighbors algorithm after reducing the dimensionality of the system. This essentially helps in the estimation of the geometry by computing a lower dimensional encoding of k nearest neighbors which greatly reduced dataset for analysis. The limitation of this method is that it generally throws away frames when it reconstructs the sequence from the connected sub-graphs. In [6],author used SIFT algorithm for remote sensing to match features between images or to localize and recognize objects. This is widely used in locating objects in videos. Another line of work is proposed by Xu el at. namely CSVideoNet [17] that directly learns the inverse mapping of compressive sensing and reconstructs the original input in a single forward propagation.
- **Compression-based approaches**: Outside of work being done for reconstructing video frames is also used in some video compression algorithms [9], [13], [11]. In these algorithms, frames can be reordered and previous and future frames are predicted using motion estimation [5], [3] algorithms. The motion estimation techniques may prove useful in increasing the algorithm's robustness to noise.
- **Tensor-based approaches**: For a detailed overview of different tensor models we refer the reader to two concise survey paper [4].In [16], algorithm synthesizing avatars from RGB images and presented it with various intermediate steps like body segmentation and dynamic robust data filtering etc. In [2] , author successfully and efficiently able to reconstruct the high-resolution video using image reconstruction.

## VI. Conclusions

This algorithm shows that video frames can be accurately reordered using image feature-based matching. The reordering method provides a more robust ordering that can be greatly accelerated though parallelization. We extensively evaluate FMVR's effectiveness over state-of-the-art approaches in a wide variety of real short and large publicly available You-tube video datasets, demonstrating the merit of leveraging feature-based matching and multi-aspect information towards high quality video reconstruction.

For future work we intend to explore different tensor decomposition models that can also be used for feature matching within FMVR's framework, as well as explore more efficient and scalable implementations for reconstruction of a 3D video.

## References

[1] F. M. Campos, L. Correia, and J. M. Calado. Robot visual localization through local feature fusion: an evaluation of multiple classifiers combination approaches. *Journal of Intelligent & Robotic Systems*, 77(2):377–390, 2015.

[2] R. H. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen. Tight frame: an efficient way for high-resolution image reconstruction. *Applied and Computational Harmonic Analysis*, 17(1):91–115, 2004.

[3] P. E. Eren, M. I. Sezan, and A. M. Tekalp. Robust, object-based high-resolution image reconstruction from low-resolution video. *IEEE Transactions on Image Processing*, 6(10):1446–1451, 1997.

[4] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[5] R. Li, B. Zeng, and M. L. Liou. A new three-step search algorithm for block motion estimation. *IEEE transactions on circuits and systems for video technology*, 4(4):438–442, 1994.

[6] T. Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7(5):10491, 2012.

[7] S. Liu and X. Bai. Discriminative features for image classification and retrieval. *Pattern Recognition Letters*, 33(6):744–751, 2012.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[9] R. F. Marcia and R. M. Willett. Compressive coded aperture video reconstruction. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE, 2008.

[10] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–284, 2007.

[11] I. E. Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[12] P. Schwander, D. Giannakis, C. H. Yoon, and A. Ourmazd. The symmetries of image formation by scattering. ii. applications. *Optics express*, 20(12):12827–12849, 2012.

[13] A. Secker and D. Taubman. Motion-compensated highly scalable video compression using an adaptive 3d wavelet transform based on lifting. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 1029–1032. IEEE, 2001.

[14] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod. Rotation-invariant fast features for large-scale recognition. *SPIE Optical Engineering and Applications*, pages 84991D–84991D, 2012.

[15] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[16] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang. Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(1):36–47, 2008.

[17] K. Xu and F. Ren. Csvideonet: A recurrent convolutional neural network for compressive sensing video reconstruction. *arXiv preprint arXiv:1612.05203*, 2016.