Remote Diagnosis of Parkinson's Disease from Finger-tapping Videos

A Graph Signal Processing Approach

Raiyan Abdul Baten
Electrical and Computer Engineering
University of Rochester
rbaten@ur.rochester.edu

Pooja Bansal
Simon School of Business
University of Rochester
pbansal@ur.rochester.edu

October 15, 2018

Overview

Motivation

Previous work

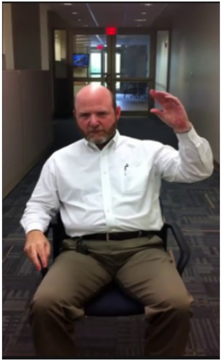
Methodology

Results

Future Directions

Motivation

Non-clinically-verified techniques

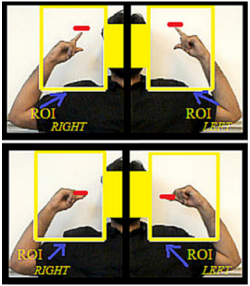


Bigger picture goal – Build an Objective and Ubiquitous Solution

- **Project goal:** Estimating finger tapping frequency from videos using Graph Signal Processing

https://medium.com/parkinsons-uk/taking-part-to-improve-how-parkinsons-symptoms-are-measured-ea04e910c62d
www.youtube.com/watch?v=C1euxtf_YeU

Previous Work



Graph Image Processing has been used for

- de-noising
- compression
- segmentation

among other applications

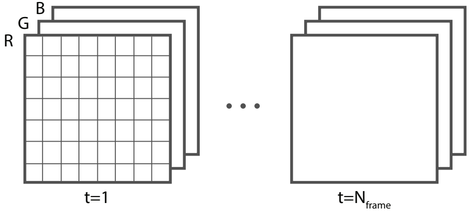
http://www.es.mdh.se/pdf_publications/3417.pdf
https://support.solidangle.com/display/A5AFMUG/Motion+Blur

Dataset

- Recorded 11 videos ourselves
- A dataset is being collected by a collaboration project between URMC and ROC-HCI, but the process is delayed by a scarcity of study subjects

Video data

- A video file can be represented as a tensor $T_{x,y,c,t} \in \mathbb{R}^4$
- x and y represent pixel coordinates, $c \in \{\text{red}, \text{green}, \text{blue}\}$ represents color channels and t represents frame index
- A ten-second 1080p video recorded at 30 fps has 1,866,240,000 entries in T



Preprocessing contd.

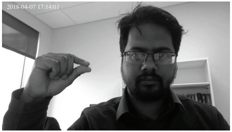
RGB to Gray-scale

- Merge RGB color channels to create gray-scale frame $\mathbf{I}(t) \in \mathbb{R}^2$ as

$$\mathbf{I}(t) := T'_{:,:,t} = \sum_c w_c T_{:,:,c,t}$$

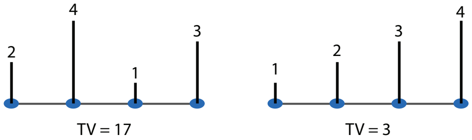
Down-sampling

- Resize each frame $\mathbf{I}(t)$ to a height of 480px using Bi-cubic Interpolation, keeping aspect ratio intact



Total Variation Analysis

- Intuition: Pixel coordinates through which fingers cross are more non-smooth than others
- Total Variation (TV) captures smoothness efficiently

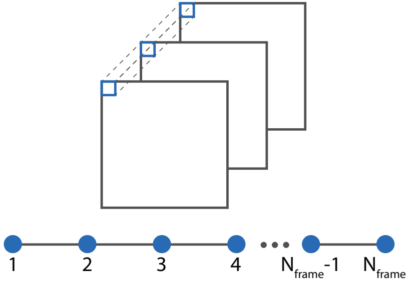


- TV for an undirected graph with
  - Adjacency matrix $\mathbf{A} \in \{0, 1\}^{N_v \times N_v}$
  - Graph Laplacian $\mathbf{L} := diag(\mathbf{A}\mathbf{1}_{N_v}) - \mathbf{A}$
  - associated graph signal vector $\mathbf{f} \in \mathbb{R}^{N_v}$

  is given by

$$\text{TV}(\mathbf{f}) := \mathbf{f}^T \mathbf{L} \mathbf{f} = \sum_{i,j=1, i>j}^{N_v} A_{ij}(f_i - f_j)^2$$

Total Variation Analysis contd.

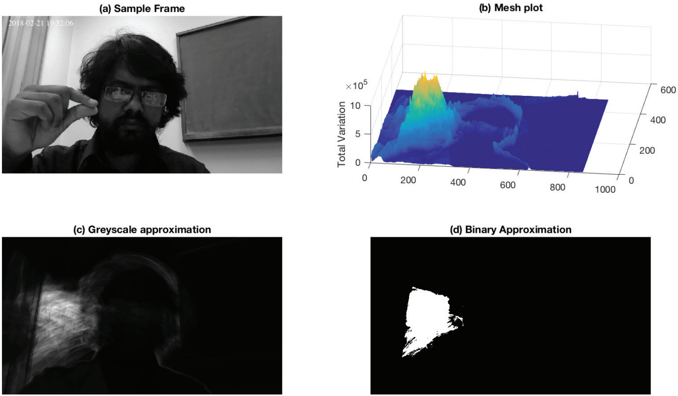


- Define temporal line graph $G(V, E)$ through each pixel location, with $N_v = N_{frame}$ nodes and uniform-weight edges
- $\mathbf{L}$ and its eigenvectors are same for all pixel locations
- Graph signal $\mathbf{f}(x, y)$ contains the greyscale intensities of the nodes in the line graph as

$$\mathbf{f}(x, y) := T'_{x,y,:}$$

ROI Extraction

- Compute TV for each pixel coordinate (x, y)
- Mark coordinates with 98^{th} percentile TV as region of interest (ROI)

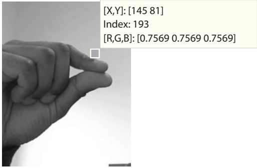


ROI Extraction contd.

- Crop ROI: Makes remaining analyses computationally less costly
- Mark coordinates with 85th percentile TV within this cropped region to include motion blur coordinates
- Frequency analysis to be run only on these coordinates

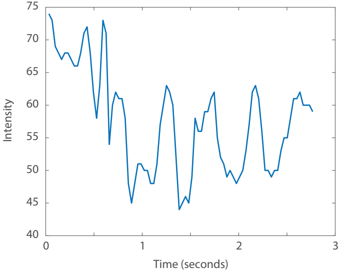
First Difference Trajectory: An Example

(a) A target pixel in the cropped frame

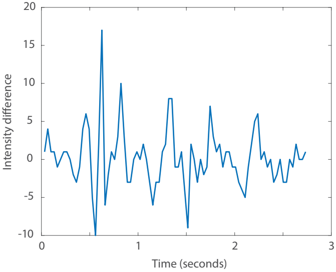


- Example showing progression of $\mathbf{f}(145, 81)$
- First differences computed as $\mathbf{I}'(t) = \mathbf{I}(t) - \mathbf{I}(t-1)$

(b) Intensity trajectory of the coordinate



(c) First difference trajectory

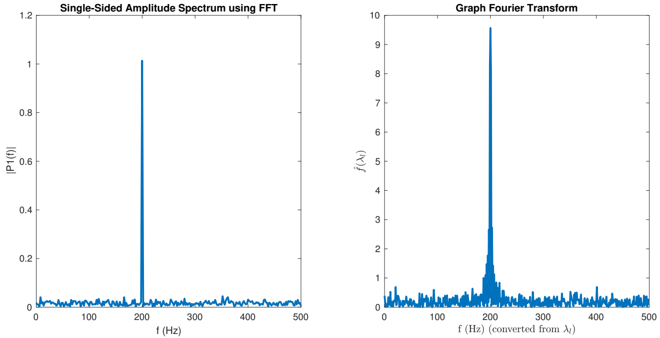


Frequency Analysis

- Graph Fourier Transform $\hat{\mathbf{f}}$ of signal vector $\mathbf{f}$ in terms of eigenvectors $\mathbf{U} = [\mathbf{u_0}\ \mathbf{u_1} \ldots \mathbf{u_{N-1}}]$ of $\mathbf{L}$ is given by:

$$\hat{\mathbf{f}} := \mathbf{U}^{\mathsf{T}}\mathbf{f}$$

- Compute dominant frequency for each selected coordinate's graph
- Report their mode



Results

Video #	Original frequency (Hz)	Estimated frequency (Hz)	Accuracy
1	5.06	5.46	92.09%
2	4.36	4.49	97.01%
3	3.87	3.98	97.16%
4	2.01	2.16	92.54%
5	3.06	3.10	98.69%
6	2.271	2.278	99.69%
7	2.29	2.35	97.38%
8	3.01	2.98	99.00%
9	1.645	1.65	99.70%
10	2.481	2.495	99.44%
11	4.395	4.497	97.68%

Future Directions

- Improve robustness for unsteady hand: Combine with CNN based approaches
- Test on a dataset from real patients

Summary

- TV analysis makes motion blur a blessing
- GFT provides a reasonable alternative to FFT in this application

Thank You!