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
Rochester, NY 14627
rbaten@ur.rochester.edu

Pooja Bansal
Simon School of Business
University of Rochester
Rochester, NY 14627
pbansal@ur.rochester.edu

1 Introduction

Parkinson's disease (PD) is a progressive neuro-degenerative disease characterized by a complex motor disorder known as Parkinsonism. The disorder is manifested primarily by resting tremor, rigidity and postural abnormalities [1]. In clinical settings, a naked-eye evaluation of the Rapid Finger-tapping Test (RFT) is one of the most widely adopted approaches for gaining a coarse understanding of PD's symptom severeness. In this test, a subject is asked to tap his/her index finger and thumb consistently in a rapid succession [2]. The frequency, amplitude, rhythm and fatigue of the dominant finger movements act as indicators of the subject's brain motor functions, and therefore of their likelihood of having PD [3].

Over the years, researchers have attempted to move beyond naked-eye evaluations of RFT towards more quantitative approaches, although to date none of those could be developed into a clinical-grade tool [1]. These efforts include using a touch sensor [4], attaching retro-reflective markers to subjects' fingers before video recording finger-tapping activities [5], using electrocardiographic apparatus [6], infrared cameras [7] or smart device sensors [8]. One of the studies used a video processing framework to extract finger-tapping information without using any markers [9]. Their data was recorded in a standardized clinical setting which ensured a noise-free plain background and a constant distance from camera. In order to make automated assessment more accessible for general people, one approach could be to allow subjects to record finger-tapping videos from their homes using webcams, and provide them feedback. However, in the wild implementation, standard clinical settings may not be there, and various obstacles may be encountered. For example, there can be random background movements, image qualities can be different, distance from camera can vary video to video and even within a video, to name a few.

The bigger picture goal is to build a solution that can be both objective and ubiquitous. Ubiquitous in a way that people can record videos at the comfort of their homes; and objective in a way that computers can do a reasonable estimate as to whether doctor consultation needs to be sought or not. A part of this is our project goal, where we aim to explore how Graph Signal Processing (GSP) can help extract finger-tapping information from home-recorded videos. In particular, the primary goal of this project is to extract the frequency of finger-tapping using GSP.

2 Literature Review

Recent developments in GSP have led to applications in fields where conventional signal processing has been used for many years, such as image and video processing [10]. An image can be modeled as a set of nodes denoting pixels, lying on a line or grid graph of connecting edges [11]. The edge weights can be suitably chosen for the requirements of specific applications. For example, assigning smaller edge weights to pixels outside of a contour can help capture geometric structures in images,

so as not to blur them during filtering [12]. In fact, filtering images to reduce noise and improve image quality has been one of the most popular uses of GSP in this domain [13, 14]. Another popular application has been the compression of image and video data [15, 16]. Closer to the problem at hand, a graph-based framework has been employed for video object segmentation and extraction by Fan et al. [17]. In this work, a video is modeled in a 7D feature space that spans color, motion, time and location. Two pixel nodes are connected by an edge based on certain similarity criteria of these features, and then analyzed spatio-temporally to segment and extract objects.

Convolutional Neural Networks (CNN) and other Deep Learning frameworks developed in recent years allow tracking body movements in videos [18, 19, 20]. While such systems are very promising, the technology is not there yet to track the fingers precisely in each frame, especially when the finger movements are too fast, due to motion blur [21], foreground aperture and similar reasons [22]. In this project we are interested in a GSP-based solution to estimating finger-tapping frequencies from videos, and these insights from previous research add value to understanding and solving the challenges better.

3 Dataset

We recorded 11 videos of finger tapping ourselves using a webcam, which we used to develop and test our methodology. The videos are around 10 seconds long and are recorded at 30 fps.

4 Methodology

We use MATLAB for all the analyses described below.

4.1 Preprocessing the videos

A video can be represented as a 4D Tensor $\mathbf{T} \in \mathbb{R}^4$, where the entry $T_{x,y,c,t}$ denotes the pixel at coordinates (x, y), in color channel $c \in \{\text{Red}, \text{Green}, \text{Blue}\}$, at time $t \in \{1, ..., N_{\text{frame}}\}$. Here N_{frame} is the number of frames in the video. This representation is shown in Figure 1. A typical ten-second 1080p video recorded at 30 fps has 1,866,240,000 entries in $\mathbf{T}$, making pre-processing necessary to decrease the number of entries to work with.

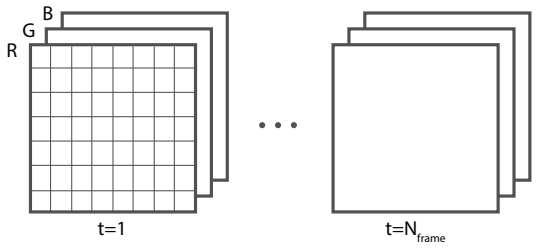


Figure 1: 4D video tensor representation of a video

4.1.1 RGB to Gray-scale

We reduce the dimensionality by converting the RGB image to gray-scale, which is done by taking a weighted average of the three color channels. This gives us $\mathbf{T}' \in \mathbb{R}^3$ generated as

$$T'_{x,y,t} = \sum_c w_c T_{x,y,c,t} \tag{1}$$

where w_c are the weights used for each color channel.

4.1.2 Down-sampling

We then reshape each video to a fixed height of 480 pixel rows using Bi-cubic Interpolation, keeping the aspect ratio intact. Lowering this reshape height further will result in less processing time but

might compromise accuracy. This down-sampling gives us the tensor $\mathbf{T}'' \in \mathbb{R}^3$, which we use for analysis. The number of entries in $\mathbf{T}''$ is an order of magnitude smaller than the original tensor $\mathbf{T}$.

4.2 Total variation based extraction of region of interest

Our working intuition is that the pixel locations which contain finger tapping, a.k.a. the region of interest, will have a temporally non-smooth line-graph, while locations that just hold the background will be temporally smoother. We therefore measure the total variation of gray-scale intensities along the temporal line-graph through each pixel coordinate. For this purpose, we define graph $G(V, E)$ which has a node set $V \in \{1, 2, ..., N_{\text{frame}}\}$ and a uniform-weighted edge set E that connects these nodes temporally. The graph construction is shown in Figure 2.

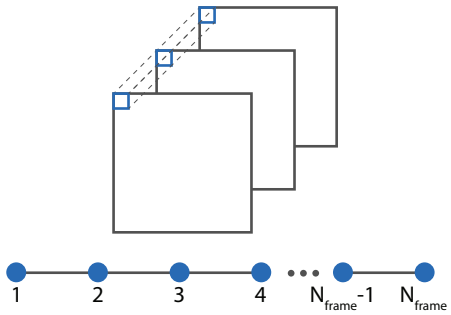


Figure 2: Temporal line graph through each pixel location. There are N_{frame} nodes in the graph, equal to the number of frames.

For graph G, the Adjacency matrix $\mathbf{A} \in \{0, 1\}^{N_{\text{frame}} \times N_{\text{frame}}}$ and the Graph Laplacian matrix $\mathbf{L} := diag(\mathbf{A}\mathbf{1}_{N_{\text{frame}}}) - \mathbf{A}$ are constructed as per standard formulations. Here, G, $\mathbf{A}$ and $\mathbf{L}$ are the same for all pixel locations.

We then construct the signal vectors for each pixel coordinate, $\mathbf{f}(x, y)$, using the gray-scale intensities from $\mathbf{T}''$ as,

$$\mathbf{f}(x, y) := T''_{x,y,:} \tag{2}$$

Given the Graph Laplacian matrix $\mathbf{L}$ and the associated signal vector $\mathbf{f}(x, y)$, we compute the total variation (TV) of the signal in the (x, y) coordinate line-graph as,

$$\text{TV}(\mathbf{f}) := \mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{i,j=1,i>j}^{N_{\text{frame}}} A_{ij}(f_i - f_j)^2 \tag{3}$$

Figure 3a shows an example gray-scale frame. The mesh plot of the total variations for different pixel coordinate's line graphs is shown in Figure 3b. As is evident, the most prominent total variation region corresponds to the finger-tapping pixels. The user moved his head slightly during the course of the video, giving those pixels' total variations a silhouette of the head shape. The top-view heat-map of this mesh plot is shown in Figure 3c, as a gray-scale approximation. Binarizing this gray-scale approximation at 95$^{\text{th}}$ percentile of total variation gives Figure 3d. In summary, this approach helps us label each pixel in the video 1 or 0 based on whether or not it belongs to the most active zone in the video, as measured by at 95$^{\text{th}}$ percentile of temporal line-graph total variation.

We crop the binary connected component from Figure 3d that has the largest number of pixels. The cropped image with the original total variation heat-map is shown in Figure 4a. From the cropped zone (Figure 4a), we consider the pixel locations at 85$^{\text{th}}$ percentile of total variation, as shown in the binary Figure 4b. This helps us get rid of inactive pixel locations while at the same time preserving the locations through which the fingers cross the most. The resulting pixel coordinates from this region of interest (ROI) are collected in the set $S = \{(x, y) \in \text{ROI}\}$.

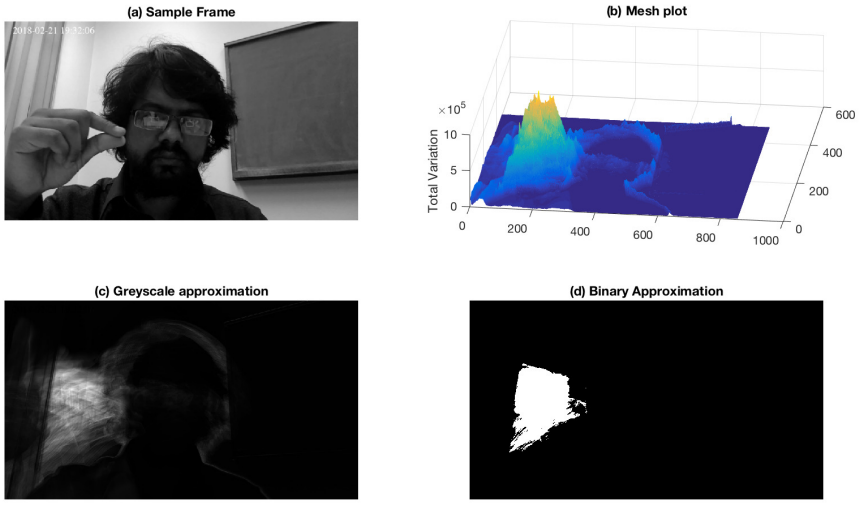


Figure 3: Using total variation to find the most active zone in the video

(a) Gray-scale heatmap of temporal intensity TV in the cropped region of interest

(b) Binary approximation at 85th percentile of TV

Figure 4: Constructing set of ROI pixel locations

4.3 Frequency analysis on finger movement pixels

For frequency analysis, we consider the signals $\mathbf{f}(x, y)$, where $(x, y) \in S$. To illustrate the steps visually, let us consider $\mathbf{f}$ for a single pixel location in S. The location of this sample pixel is marked in Figure 5a. The index finger passes through this pixel twice in every tap. Figure 5b plots the temporal gray-scale intensity of that pixel location. This video has 5 taps, and some repetition information is apparent in the intensity trajectory, although not in a readily usable way. It also displays a DC component. We take the first difference of this signal by subtracting each frame's intensity from its previous frame's intensity to get $\mathbf{f}'(x, y)$ as

$$f'_t(x, y) := T''_{x,y,t} - T''_{x,y,t-1} \tag{4}$$

This is plotted in Figure 5c. This plot shows more intuitive information, with 5 peaks corresponding to 5 taps, and a mean intensity difference around 0. We then run FFT and GFT analysis on this first-difference trajectory $\mathbf{f}'(x, y)$ to extract the tapping frequency. The FFT-generated single-sided amplitude spectrum, $|P1(f)|$, is plotted in Figure 5d. The highest amplitude in the plot corresponds to a frequency of 2.16 Hz (marked with a red dot), which is very close to the manually-calculated ground-truth frequency of 2.01 Hz.

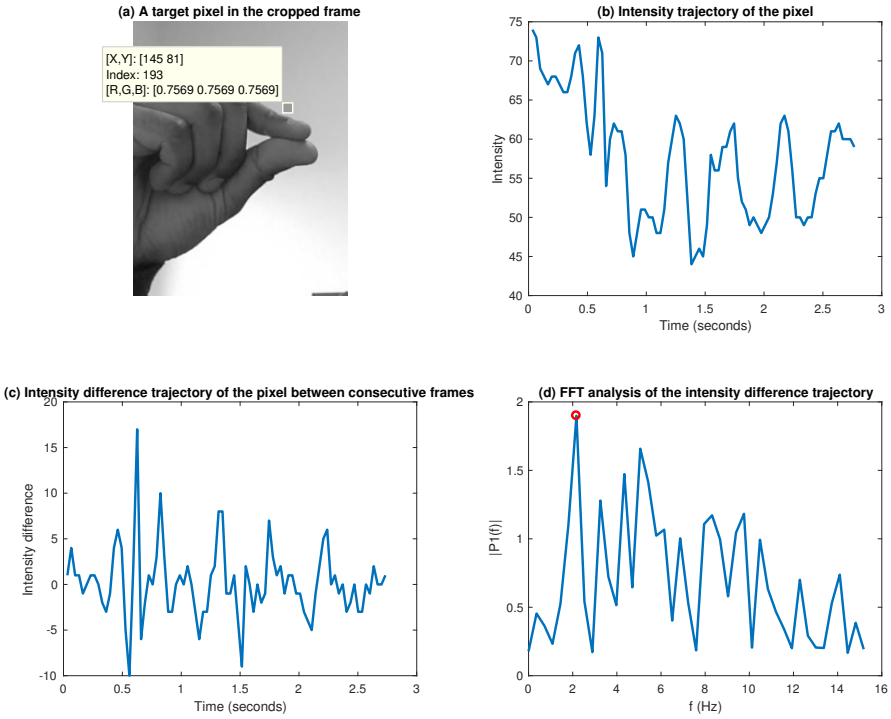


Figure 5: Extracting dominating frequency corresponding to a single pixel location

Alternatively, we can use Graph Fourier Transform (GFT) on $\mathbf{f}'(x, y)$ to extract the dominant frequency. Graph Fourier Transform $\hat{\mathbf{f}}$ of a signal vector $\mathbf{f}$ is defined in terms of the eigenvectors $\mathbf{U} = [\mathbf{u}_0\ \mathbf{u}_1 \dots \mathbf{u}_{N_{\text{frame}}-1}]$ of $\mathbf{L}$, as

$$\hat{\mathbf{f}} := \mathbf{U}^T \mathbf{f}, \tag{5}$$

from which the dominant frequency can be extracted. Although GFT does not give the frequency information in Hz unit, we can construct the frequency axis in Hz using the number of frames and the frame-rate information from the video. An example is given in Figure 6, where the FFT and GFT analysis results are plotted for a signal of 200Hz with superimposed noise. The extracted frequencies are the same for both methods, although there are larger noisy spikes near the dominant frequency in case of GFT. For the videos in our dataset, both methods give the same results.

We repeat the above-mentioned process for extracting the dominant frequency for each of the pixel locations in S. Then we take the mode of all the extracted dominant frequencies, and report it as the finger tapping frequency in the video.

5 Results

Using the methodology above, we are able to estimate the finger-tapping frequencies of the 11 videos in our dataset with an average accuracy of 97.3%. The accuracies are calculated against manually labeled ground-truth frequencies. The results are reported in Table 1.

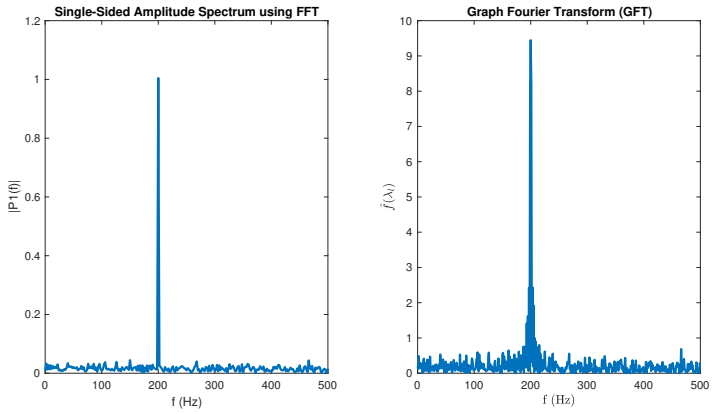


Figure 6: FFT and GFT analysis to extract the dominant frequency from a sinusoid of 200 HZ with superimposed noise.

Table 1: Results of finger-tapping frequency estimation

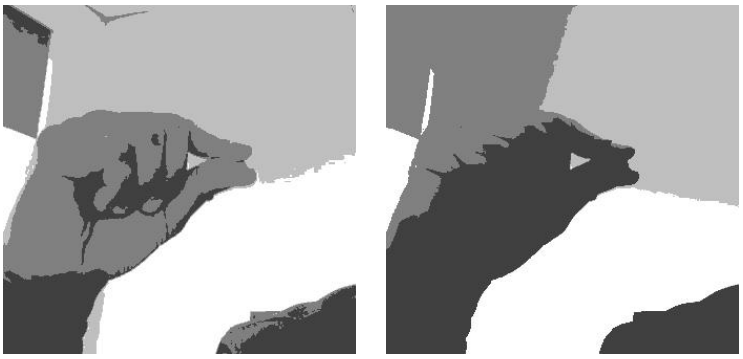
Video #	Original frequency (Hz)	Estimated frequency (Hz)	Accuracy
1	5.06	5.46	92.09%
2	4.36	4.49	97.01%
3	3.87	3.98	97.16%
4	2.01	2.16	92.54%
5	3.06	3.10	98.69%
6	2.271	2.278	99.69%
7	2.29	2.35	97.38%
8	3.01	2.98	99.00%
9	1.645	1.65	99.70%
10	2.481	2.495	99.44%
11	4.395	4.497	97.68%

6 Discussion

While the accuracies in Table 1 are extremely impressive, they are nonetheless biased by the fact that in all of the videos in the dataset, the most active zones indeed corresponded to finger-tapping coordinates. This might not be true in real settings, where there can be significant background movements that constitute the highest total variation region. On the bright side, using total variation makes use of motion blur in a positive way. The subtle changes in pixel intensities due to motion blur add up in TV, which makes estimating finger-tapping frequency feasible without tracking the fingers themselves. Furthermore, the proposed method is immune to skin color, image quality and distance from camera, since variations in these will not go against our assumptions. Using GFT and FFT gave the same results in our dataset, although the results were slightly cleaner in FFT. On the flip side, computing GFT requires only one matrix multiplication, which might make it a preferable choice for deploying in GPU-based parallel processing systems.

7 Challenges, Incomplete Work and Future Steps

We further attempted to explore the problem from a spatial-graph perspective. We connected pixels in each frame with edges in 3D and 5D feature spaces: the 3D space being made of just the three color channels, and the 5D space including the x-y coordinates along with the color channels. We ran k-means clustering of all the pixels in each frame with $k = 4$, separately for the the two feature spaces. This brought similar pixels in the feature vector in the same cluster, allowing us to connect them with edge weights higher than with those in other clusters.



(a) Clustering using 3D feature vector (b) Clustering using 5D feature vector

Figure 7: K-means clustering for connecting similar pixel nodes with edges. Here, $k = 4$ has been used, and the four color shades in the figures correspond to the four clusters.

Sample results of the pixel clustering using 3D and 5D feature vectors are shown in Figures 7a and 7b respectively. The four gray-scale shades in the figures correspond to the $k = 4$ clusters generated by k-means clustering algorithm. It is important to note that the pixel nodes are connected in 3D and 5D spaces, not in the spatial 2D space in which they are shown in the figures. That is why many pixels are seen detached in the figures despite belonging to the same cluster. In Figure 7a, we can see that most of the hand pixels indeed belong to the same cluster. The 5D feature space, shown in Figure 7b, does not do as good a job.

However, we could not combine the information in spatial and temporals graphs in a meaningful way. This remains a part of our future work.

In the era of big-data and deep-learning, perhaps simply feeding a deep neural network with a lot of labeled samples will give exciting results. Generating a huge dataset for this purpose will be challenging, and remains a future work. Eventually, we would like to implement all of these tests on a dataset from real patients.

8 Conclusion

In the application of estimating finger-tapping frequency, we observed that Graph Fourier Transform approach provides a reasonable alternative to Fast Fourier Transform. Also, the overall analysis around total variation has utilized motion blur as a blessing, and provided us with encouraging results. Our project sheds light on the values GSP can add to this particular research scope. When combined meaningfully with the strengths of other tools like CNN, the overall architecture might lead to robust results.

References

[1] Chrystalina A Antoniades and Roger A Barker. The search for biomarkers in parkinson's disease: a critical review. *Expert Review of Neurotherapeutics*, 8(12):1841–1852, 2008.

[2] Charles E Collyer, Hilary A Broadbent, and Russell M Church. Preferred rates of repetitive tapping and categorical time production. *Perception & Psychophysics*, 55(4):443–453, 1994.

[3] Ichiro Shimoyama, T Ninchoji, and K Uemura. The finger-tapping test. *Arch Neurol*, 47:681–684, 1990.

[4] Ryuhei Okuno, Masaru Yokoe, Kenichi Fukawa, Saburo Sakoda, and Kenzo Akazawa. Measurement system of finger-tapping contact force for quantitative diagnosis of parkinson's disease. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1354–1357. IEEE, 2007.

[5] Ákos Jobbágy, Péter Harcos, Robert Karoly, and Gábor Fazekas. Analysis of finger-tapping movement. *Journal of Neuroscience Methods*, 141(1):29–39, 2005.

[6] Ichiro Shimoyama, Kaoru Hinokuma, Toshiaki Ninchoji, and Kenichi Uemura. Microcomputer analysis of finger tapping as a measure of cerebellar dysfunction. *Neurologia Medico-chirurgica*, 23(6):437–440, 1983.

[7] Rocco Agostino, Antonio Curra, Morena Giovannelli, Nicola Modugno, Mario Manfredi, and Alfredo Berardelli. Impairment of individual finger movements in parkinson's disease. *Movement Disorders*, 18(5):560–565, 2003.

[8] Andong Zhan, Srihari Mohan, Christopher Tarolli, Ruth B Schneider, Jamie L Adams, Saloni Sharma, Molly J Elson, Kelsey L Spear, Alistair M Glidden, Max A Little, et al. Using smartphones and machine learning to quantify parkinson disease severity: The mobile parkinson disease score. *JAMA Neurology*, 2018.

[9] Taha Khan, Dag Nyholm, Jerker Westin, and Mark Dougherty. A computer vision framework for finger-tapping evaluation in parkinson's disease. *Artificial Intelligence in Medicine*, 60(1):27–40, 2014.

[10] Gene Cheung, Enrico Magli, Yuichi Tanaka, and Michael Ng. Graph spectral image processing. *arXiv preprint arXiv:1801.04749*, 2018.

[11] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing. *arXiv preprint arXiv:1712.00468*, 2017.

[12] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, 2013.

[13] Jiahao Pang, Gene Cheung, Antonio Ortega, and Oscar C Au. Optimal graph laplacian regularization for natural image denoising. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 2294–2298. IEEE, 2015.

[14] Dong Tian, Hassan Mansour, Andrew Knyazev, and Anthony Vetro. Chebyshev and conjugate gradient filters for graph image denoising. In *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014.

[15] Wei Hu, Gene Cheung, Antonio Ortega, and Oscar C Au. Multiresolution graph fourier transform for compression of piecewise smooth images. *IEEE Transactions on Image Processing*, 24(1):419–433, 2015.

[16] Hilmi E Egilmez, Amir Said, Yung-Hsuan Chao, and Antonio Ortega. Graph-based transforms for inter predicted video coding. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3992–3996. IEEE, 2015.

[17] Lei Fan and Alexander C Loui. A graph-based framework for video object segmentation and extraction in feature space. In *Multimedia (ISM), 2015 IEEE International Symposium on*, pages 266–271. IEEE, 2015.

[18] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[19] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[20] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[21] Rafael C Gonzalez and Richard E Woods. Digital image processing, 2012.

[22] Ester Martínez-Martín and Ángel P del Pobil. Motion detection in static backgrounds. In *Robust Motion Detection in Real-Life Scenarios*, pages 5–42. Springer, 2012.