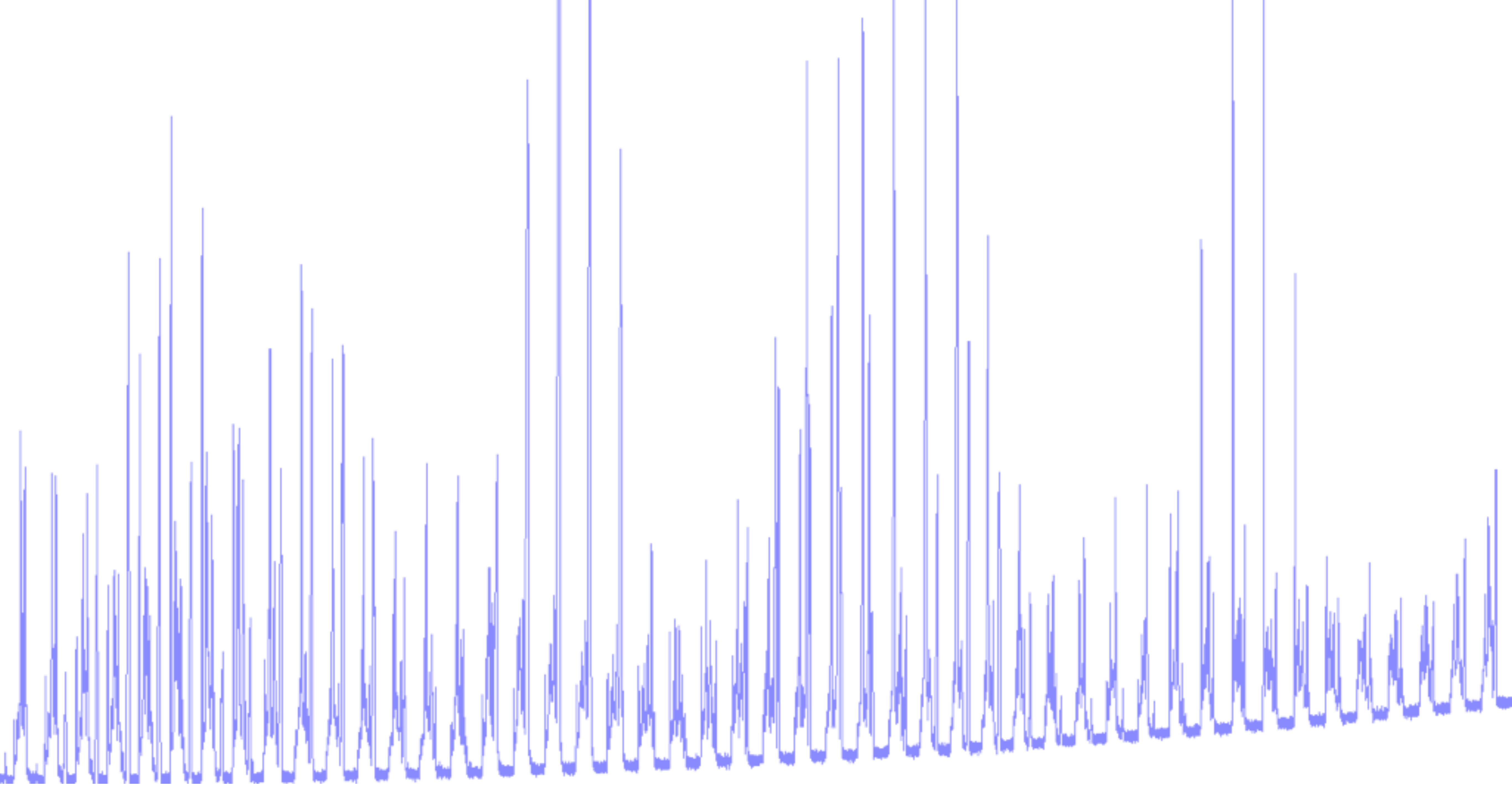


Inertial Sensing

EE382V Activity Sensing and Recognition

UT Austin • Dept. Electrical and Computer Engineering • Fall 2016

Today

Data Collection Assignment

Working with Inertial Sensor Data

- De-noising and Smoothing
- Frames and feature extraction

Papers

- Panel of Experts

BAND ON THE RUN —

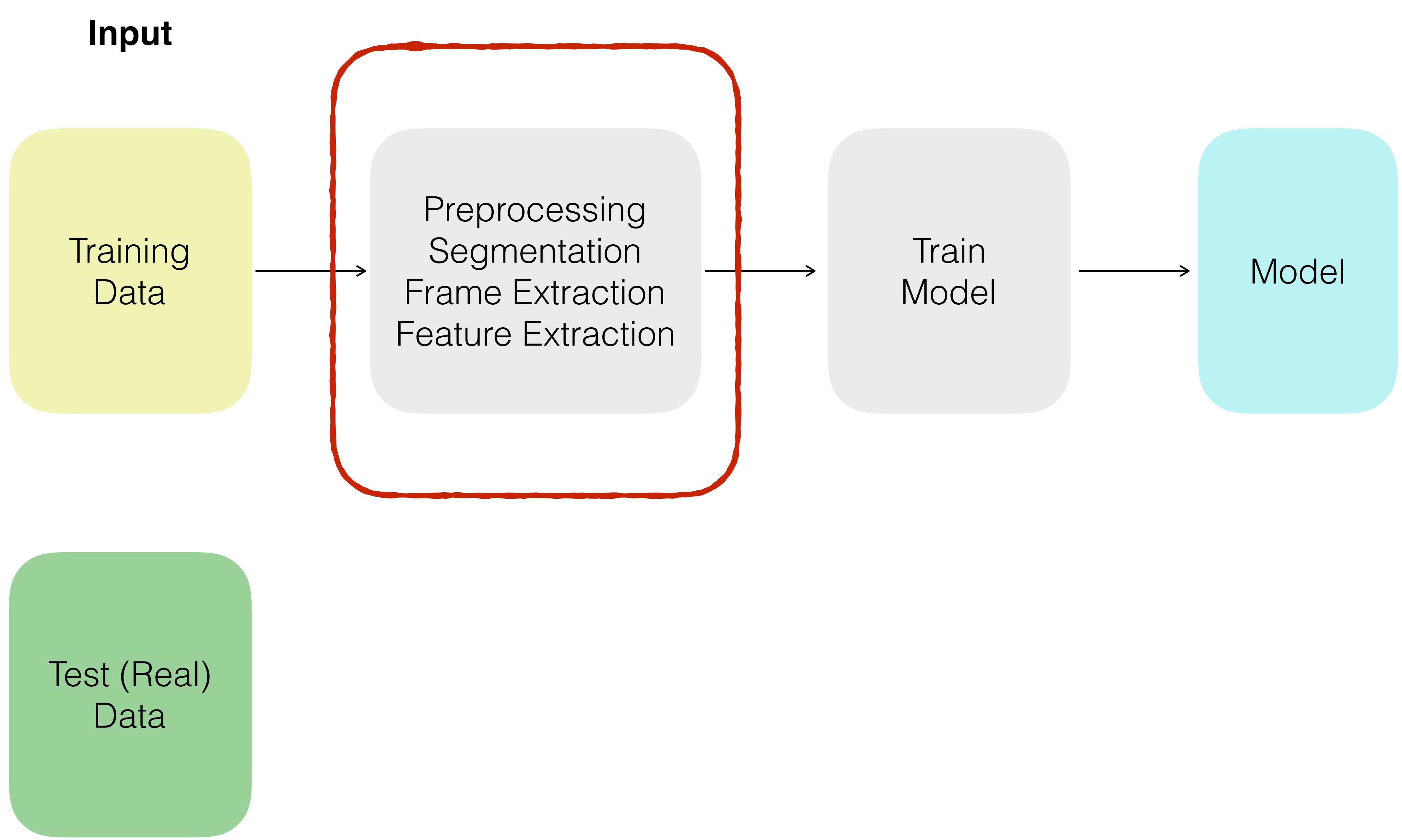
Microsoft disbands the Band: fitness device pulled from stores, no Band 3 this year

Microsoft says it's still exploring wearables space, but has stopped making wearables.

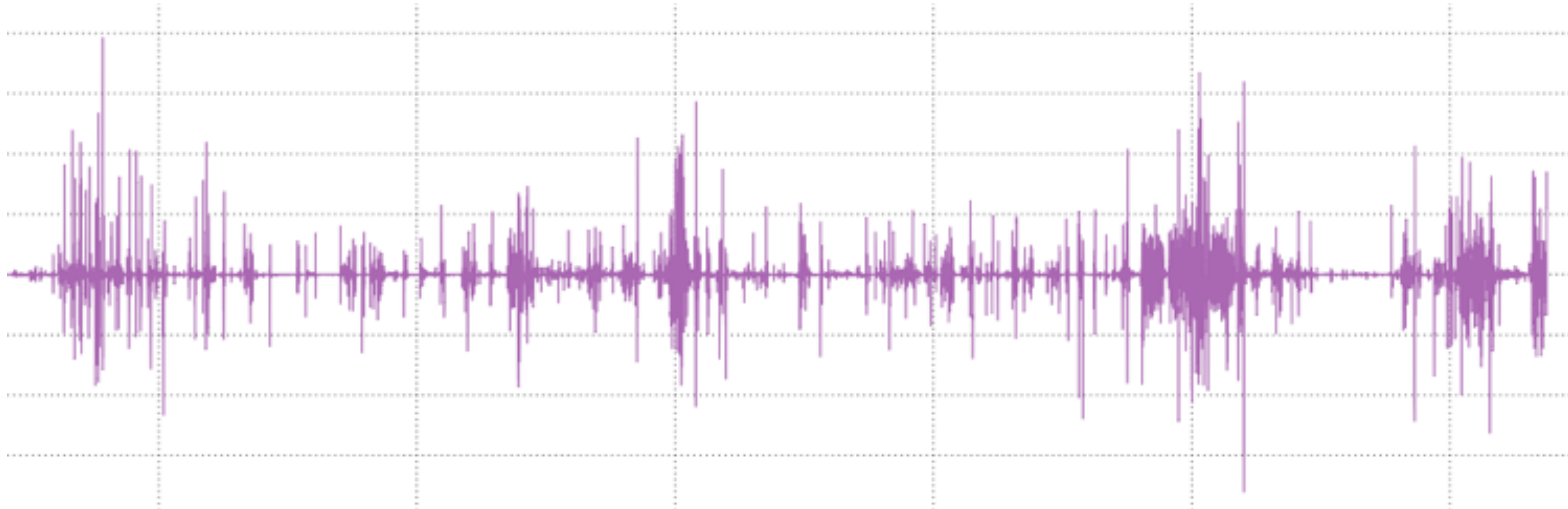
PETER BRIGHT - 10/3/2016, 5:54 PM

entina Palladino

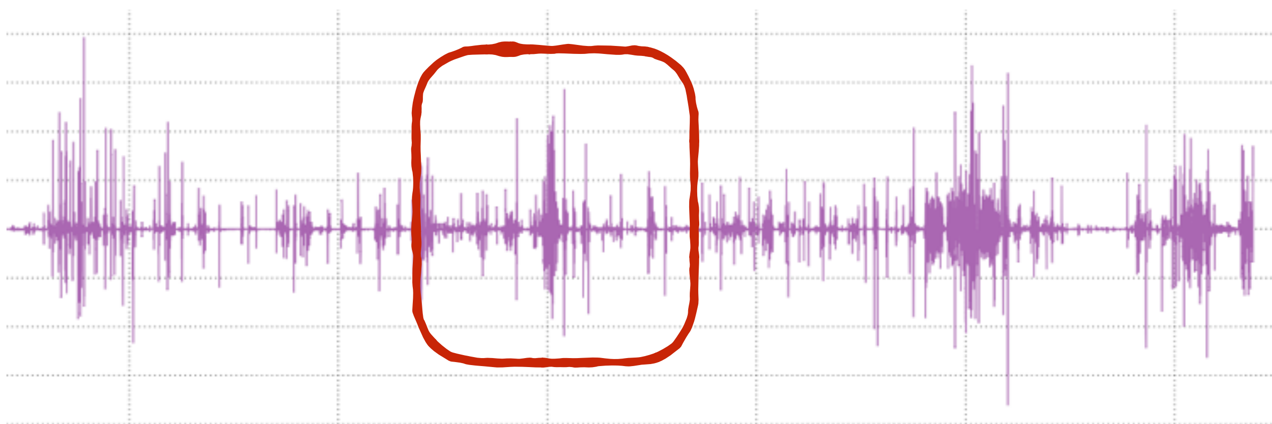
Activity Recognition Pipeline



Inertial Sensor Signal



Classification

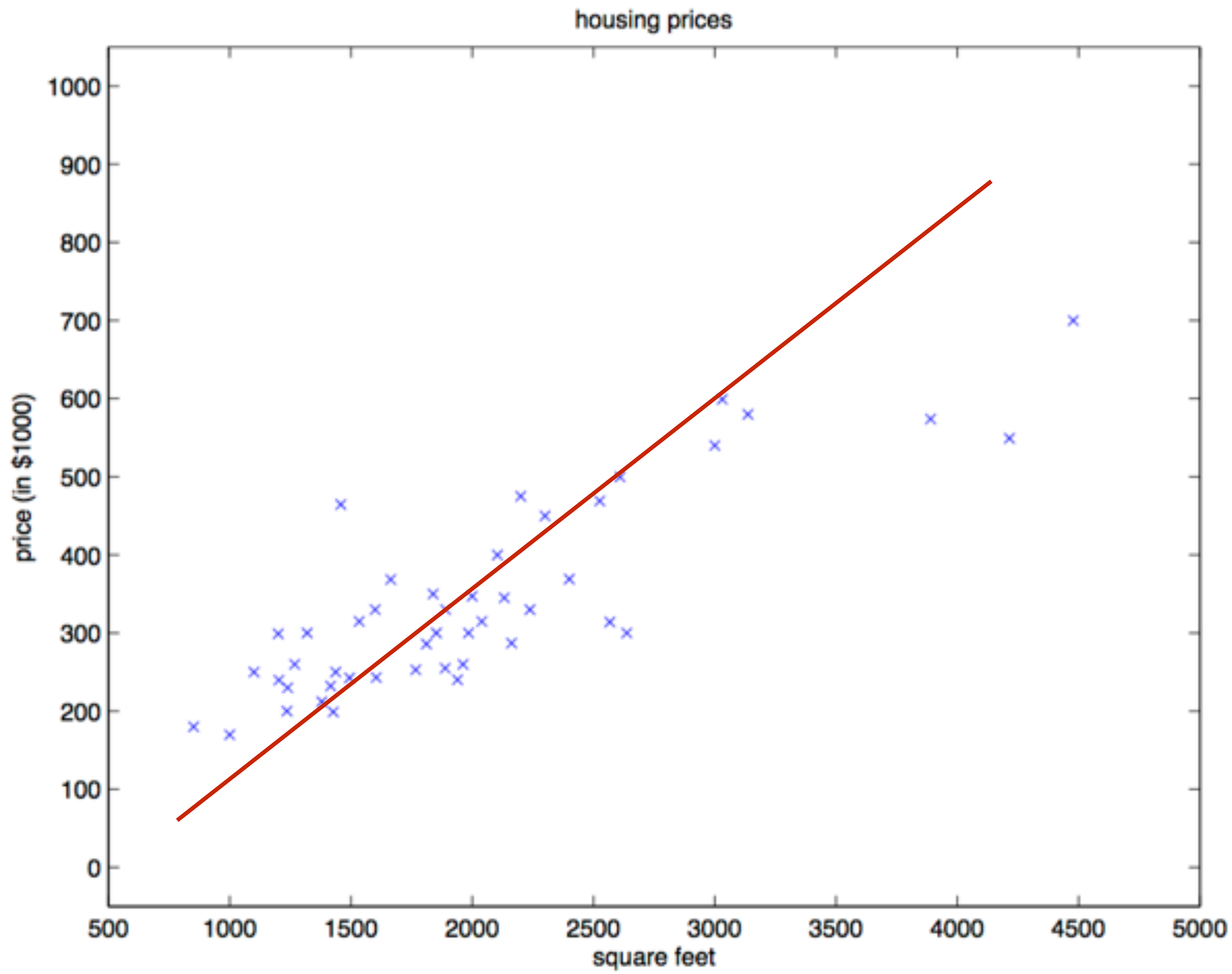


Activity A

Activity B

Machine Learning to the Rescue!

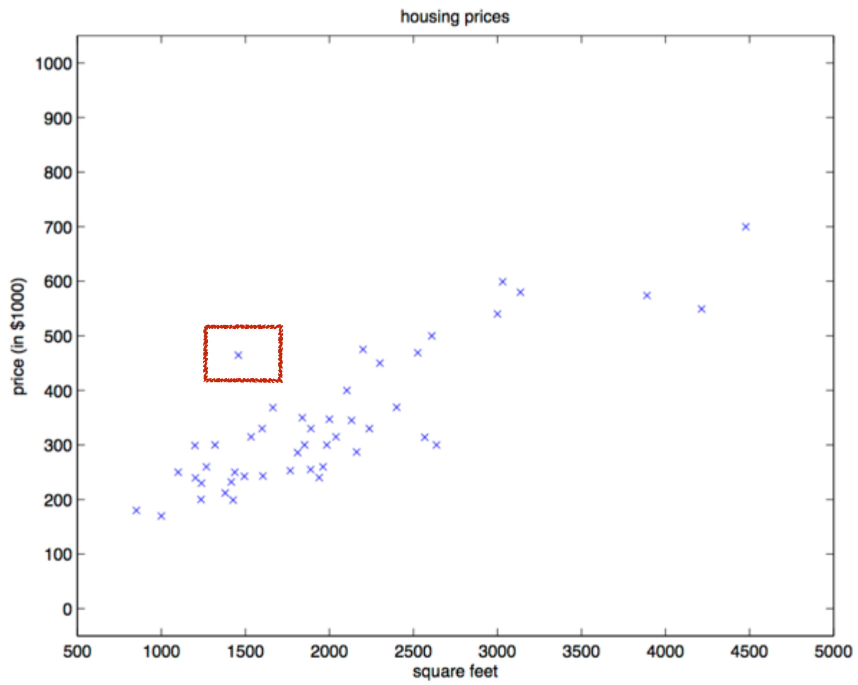
ML Supervised Learning



Our goal: learn a function $h : X \rightarrow Y$ that approximates f

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

ML Supervised Learning

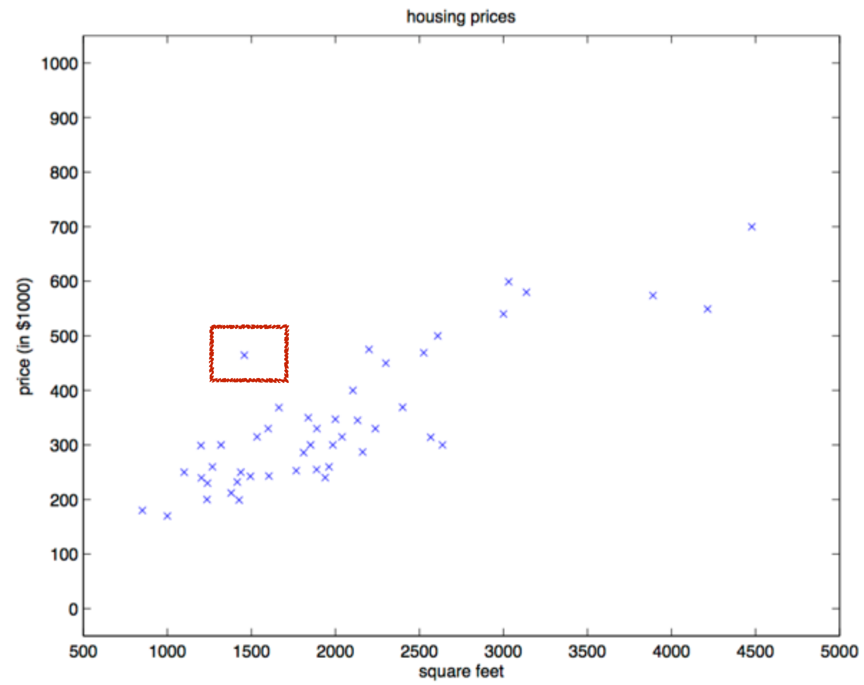


X	Y
0.34	0.76
0.75	0.54
0.02	0.23
…	…

Instance as unit of analysis

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

ML Supervised Learning

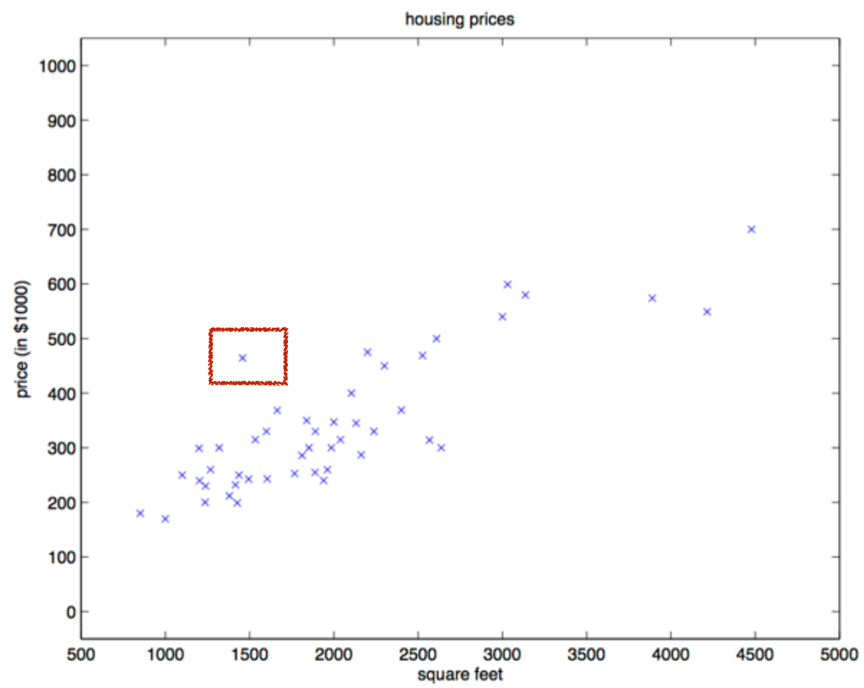


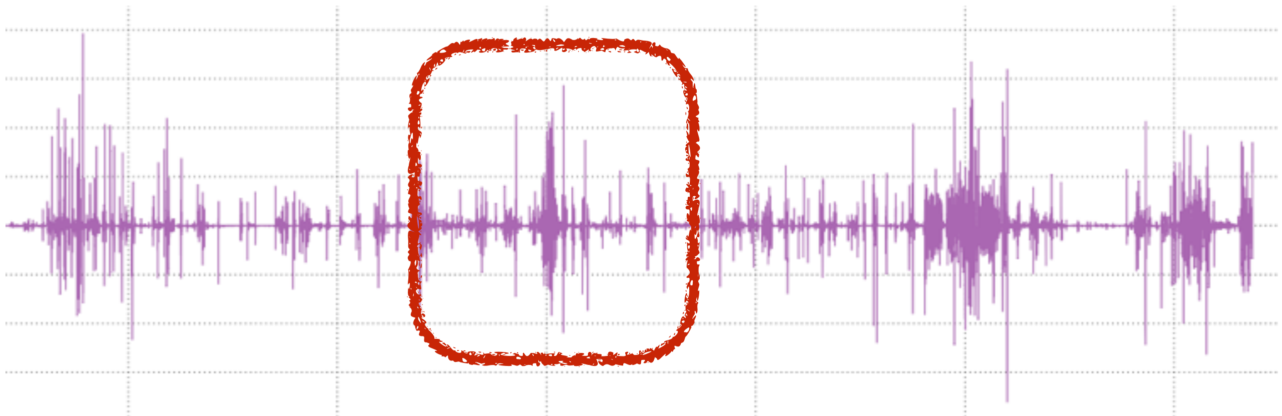
X	Y
0.34	0.76
0.75	0.54
0.02	0.23
…	…

Instances <X,Y>

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

ML Supervised Learning





Instance

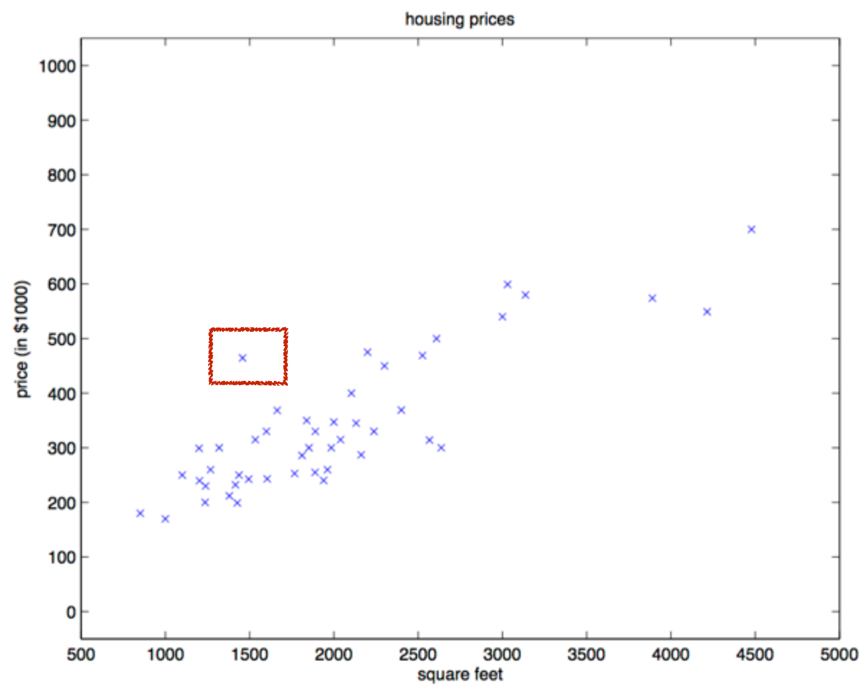
<1002,489>

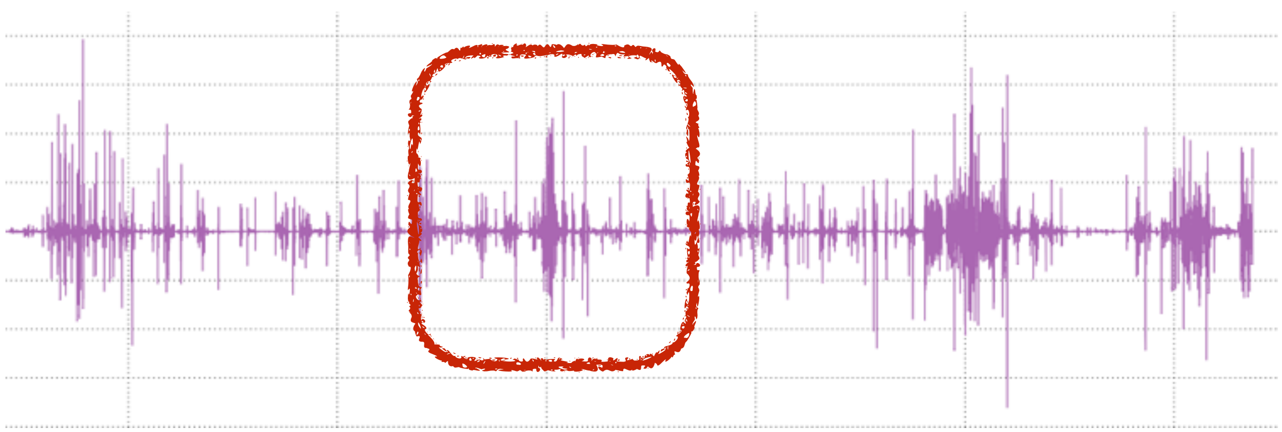
Instance

<?, Activity A>

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

ML Supervised Learning





Instance

<1002,489>

Instance

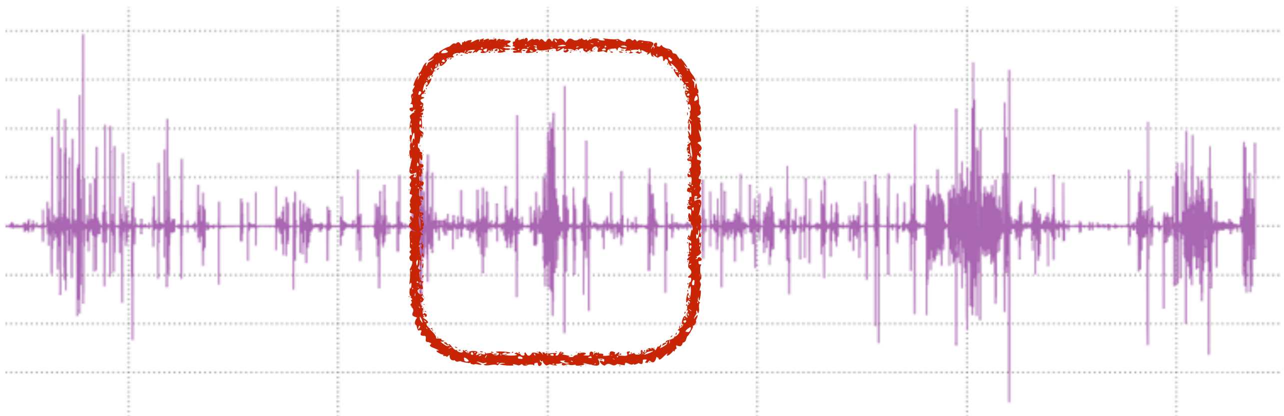
<[45,56,23,…,56], Activity A>

Not a good representation

Why not?

Source: (Based in part on) Andrew Ng CS229 Lecture Notes

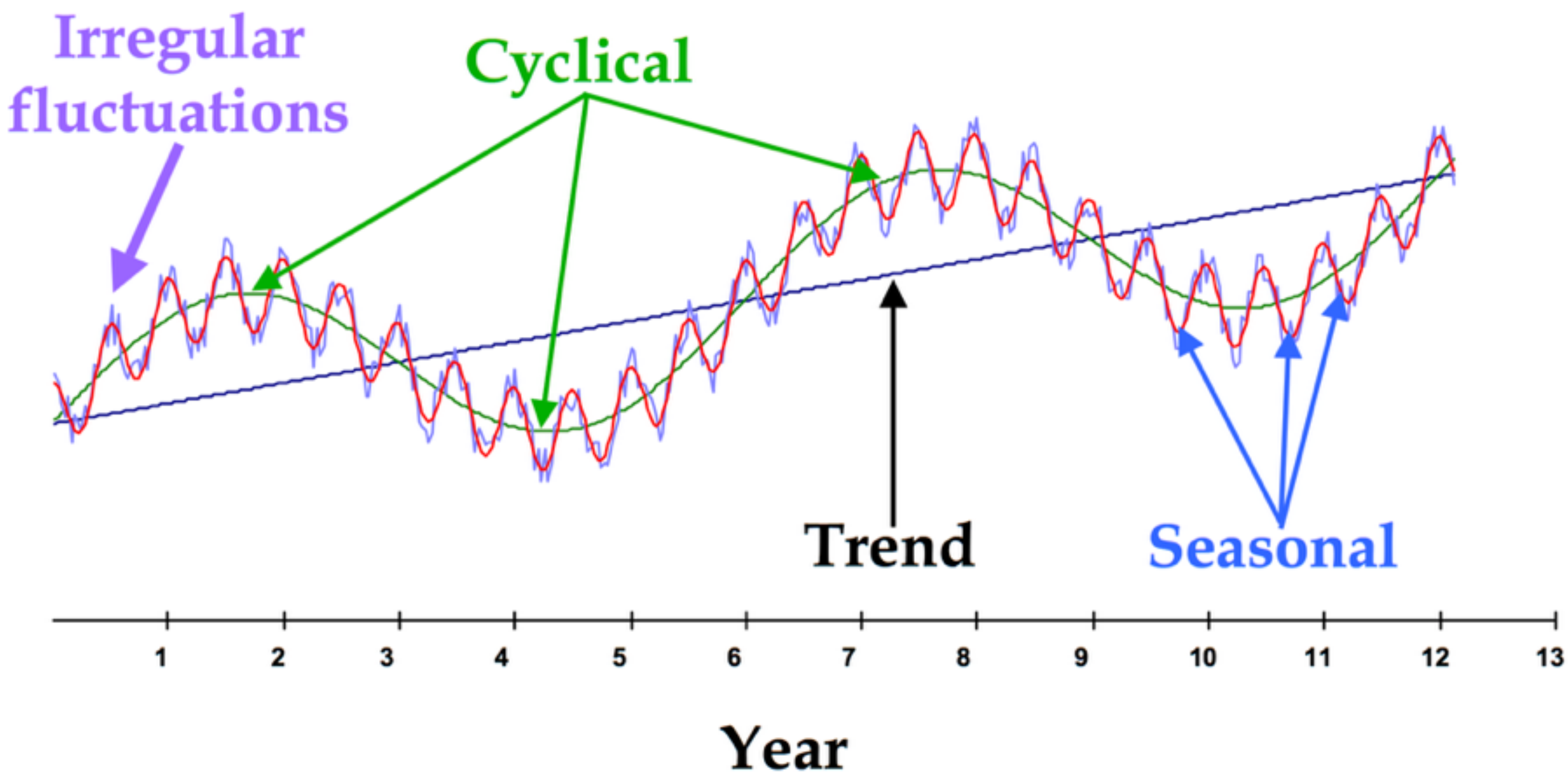
ML Supervised Learning



Need to extract features then…

Are we ready to do that?

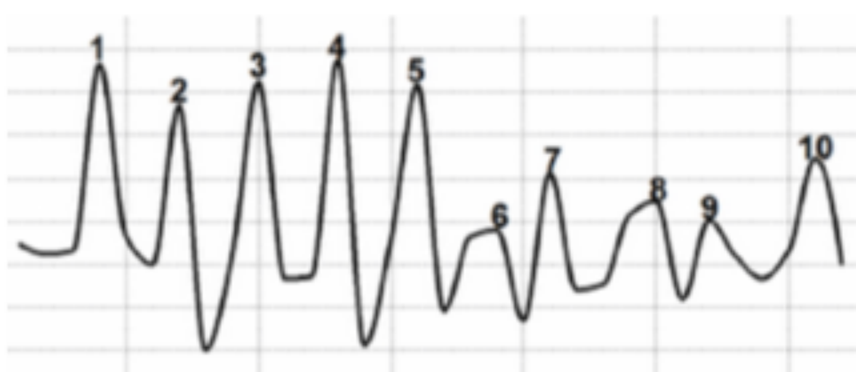
Preprocessing



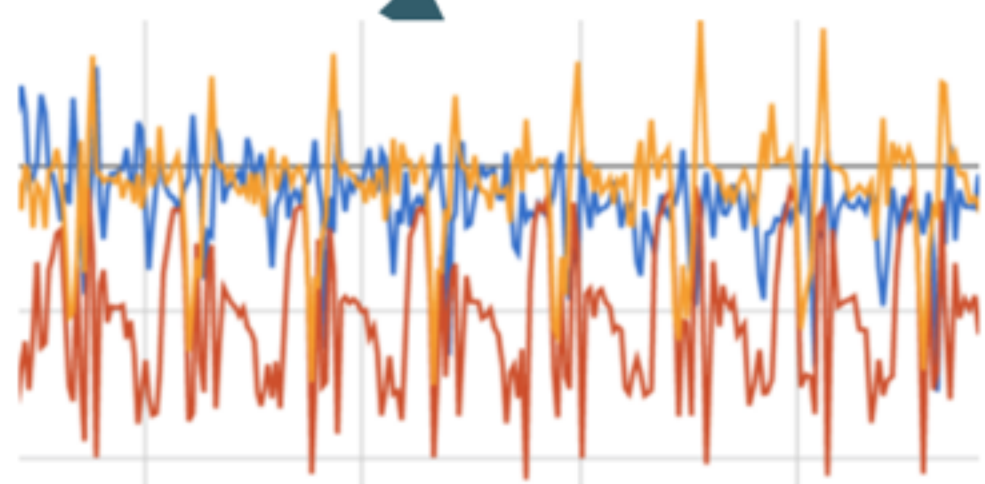
Source: http://www.statoek.wiso.uni-goettingen.de/veranstaltungen/graduateseminar/SmoothingMethods_Narodzonek-Karpowska.pdf

Noise in Sensor Data

Step Detection Algorithm



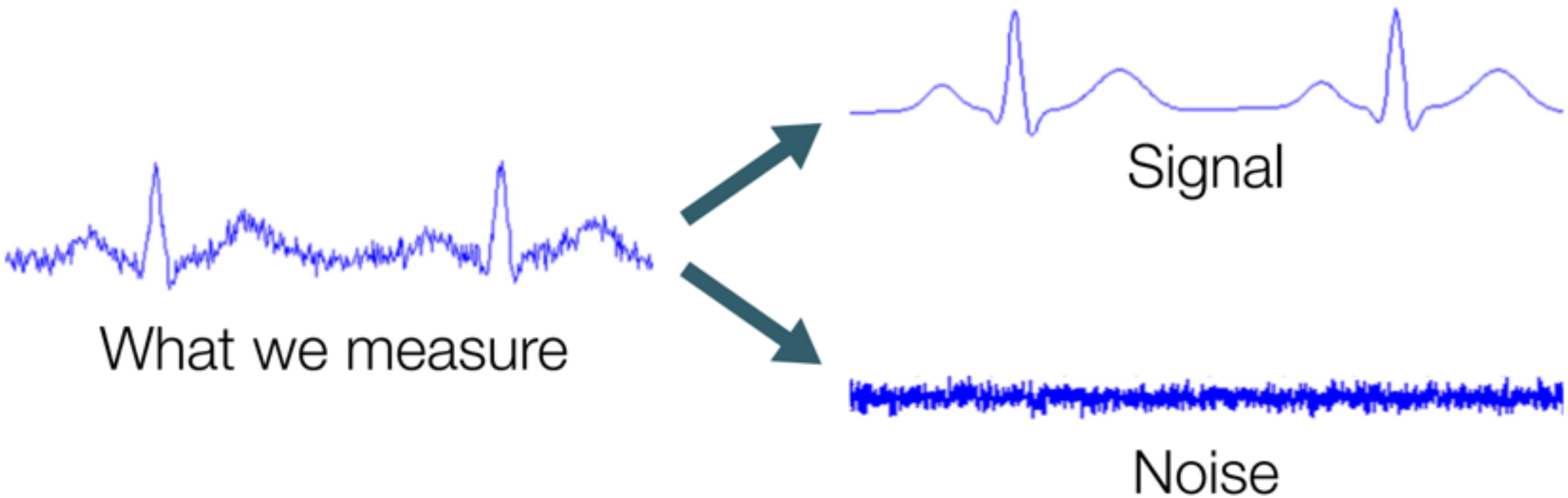
(you expect this)



(you get this)

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

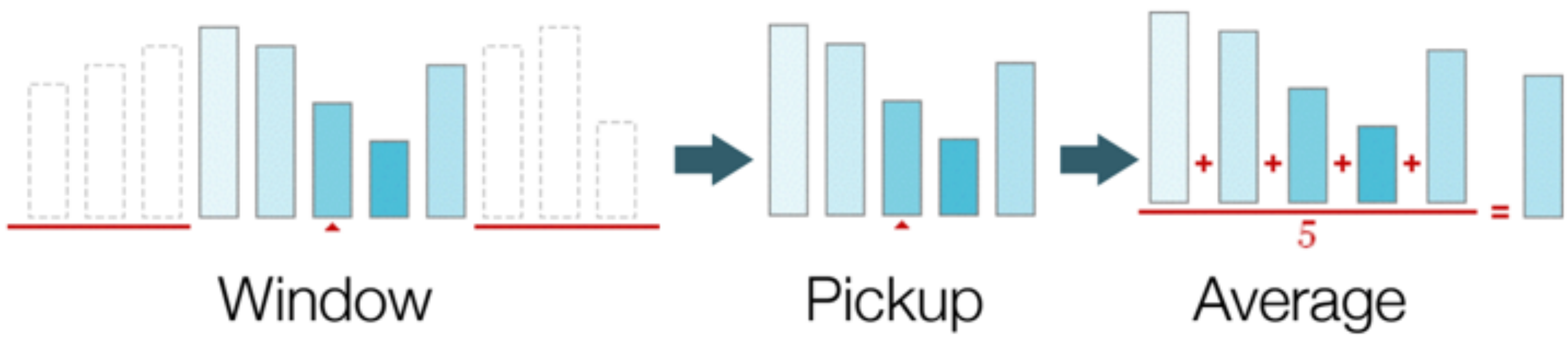
Signal and Noise



We need to remove noise while retaining the signal

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

Moving Average Filter



- Steps in a moving average filter
  - Place a window over samples
  - Pickup samples
  - Output average of window elements

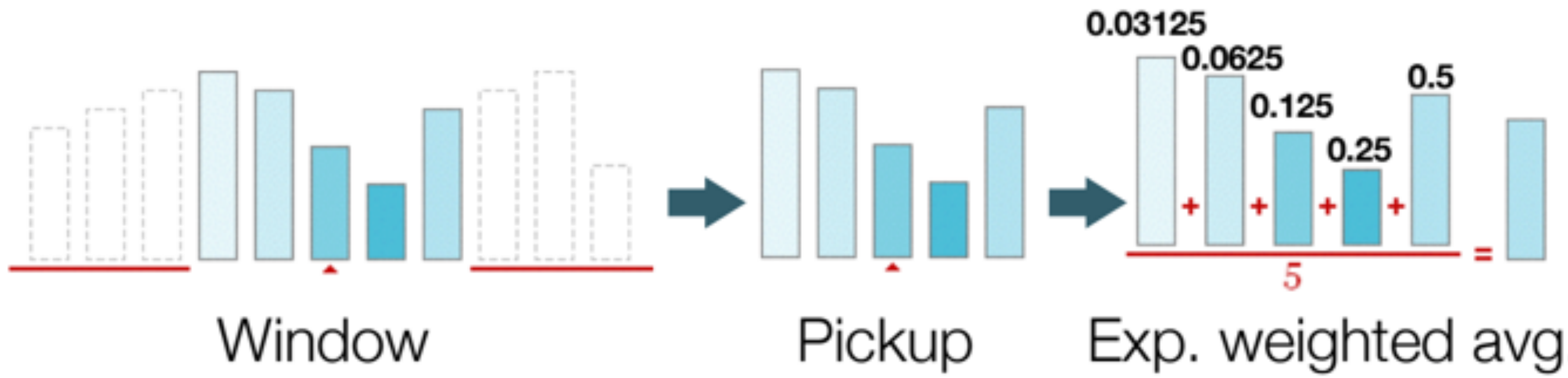
Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

Moving Average Filter

- Moving average smooth signal but…causes delayed peaks

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

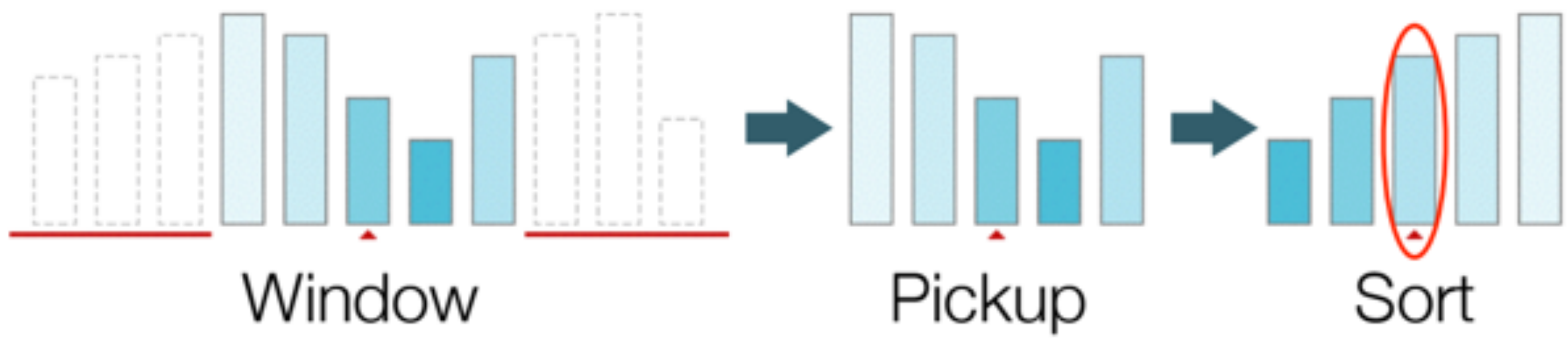
Exponentially Weighted Moving Average



- Steps in a exponential moving average filter
  - Place a window over samples
  - Pickup samples
  - Output exponentially weighted average of elements

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

Median Filter

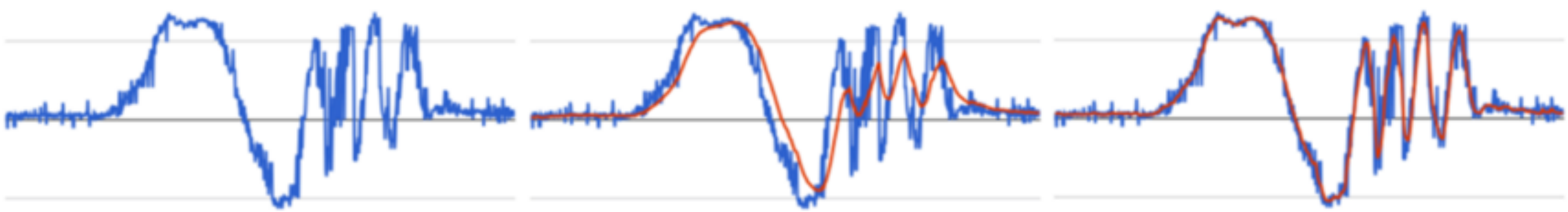


Median Filtering

- Commonly used to remove spikes, and salt-and-pepper noise.
- Preserves true edges in the signal.

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

Median Filter vs. Exponential Smoothing

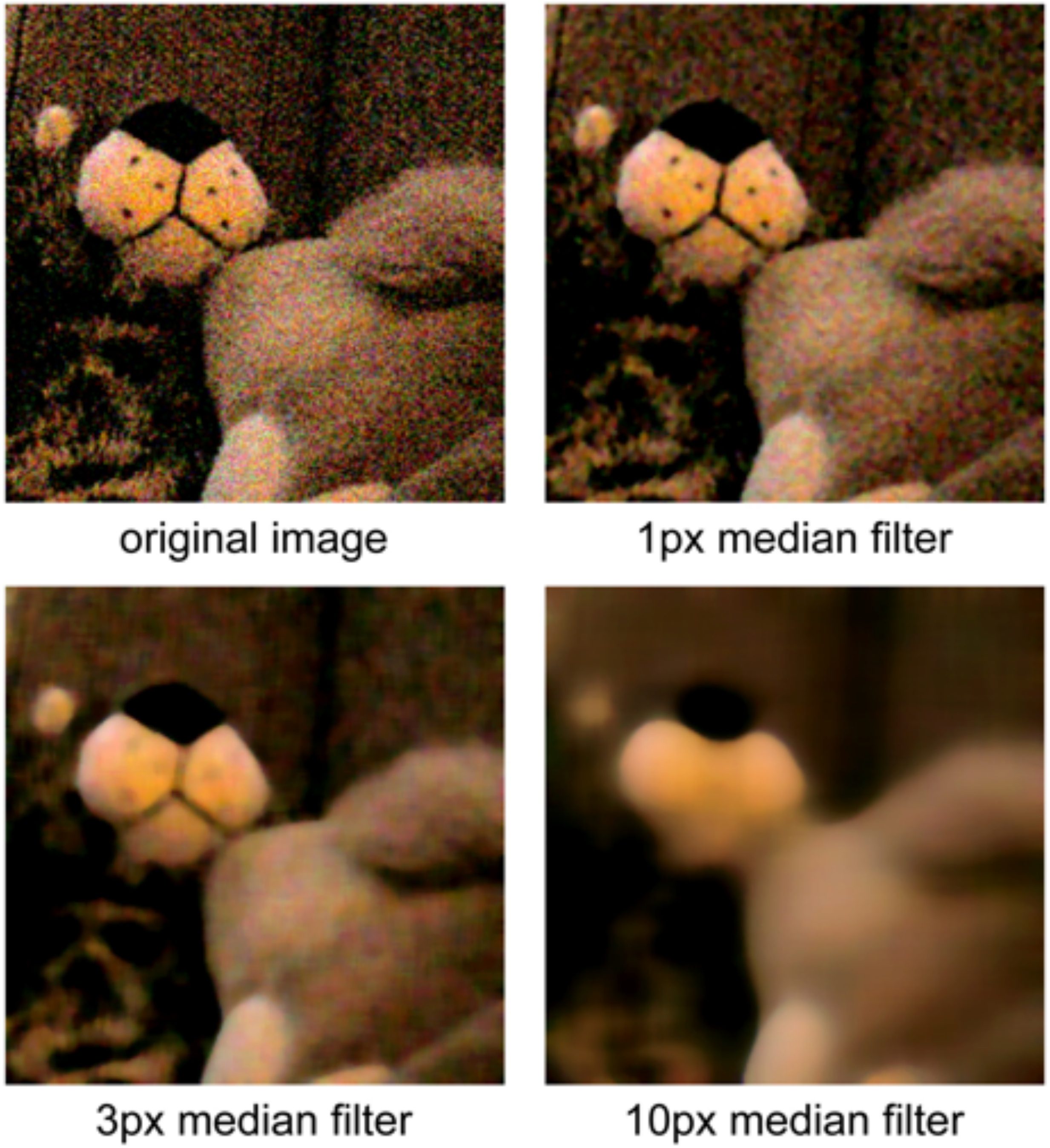


(a) Raw accelerometer readings (b) Exponential smoothing (c) Median filtering

Median filter has no lag, and preserves heights of peaks while removing spike noise.

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

Median Filter in 2D



Blurring

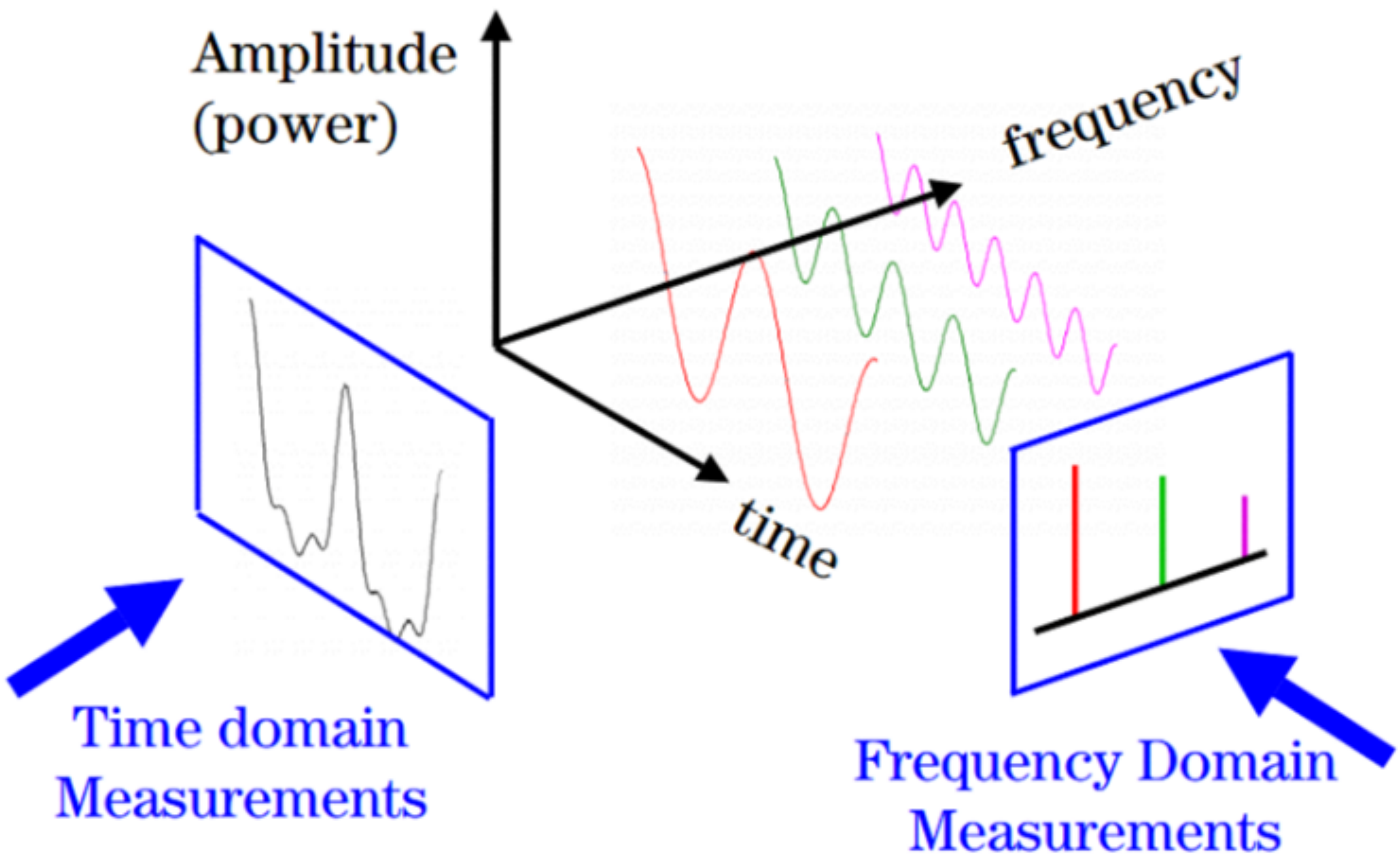
Source: Wikipedia

Median Filter in 2D

Image Reconstruction

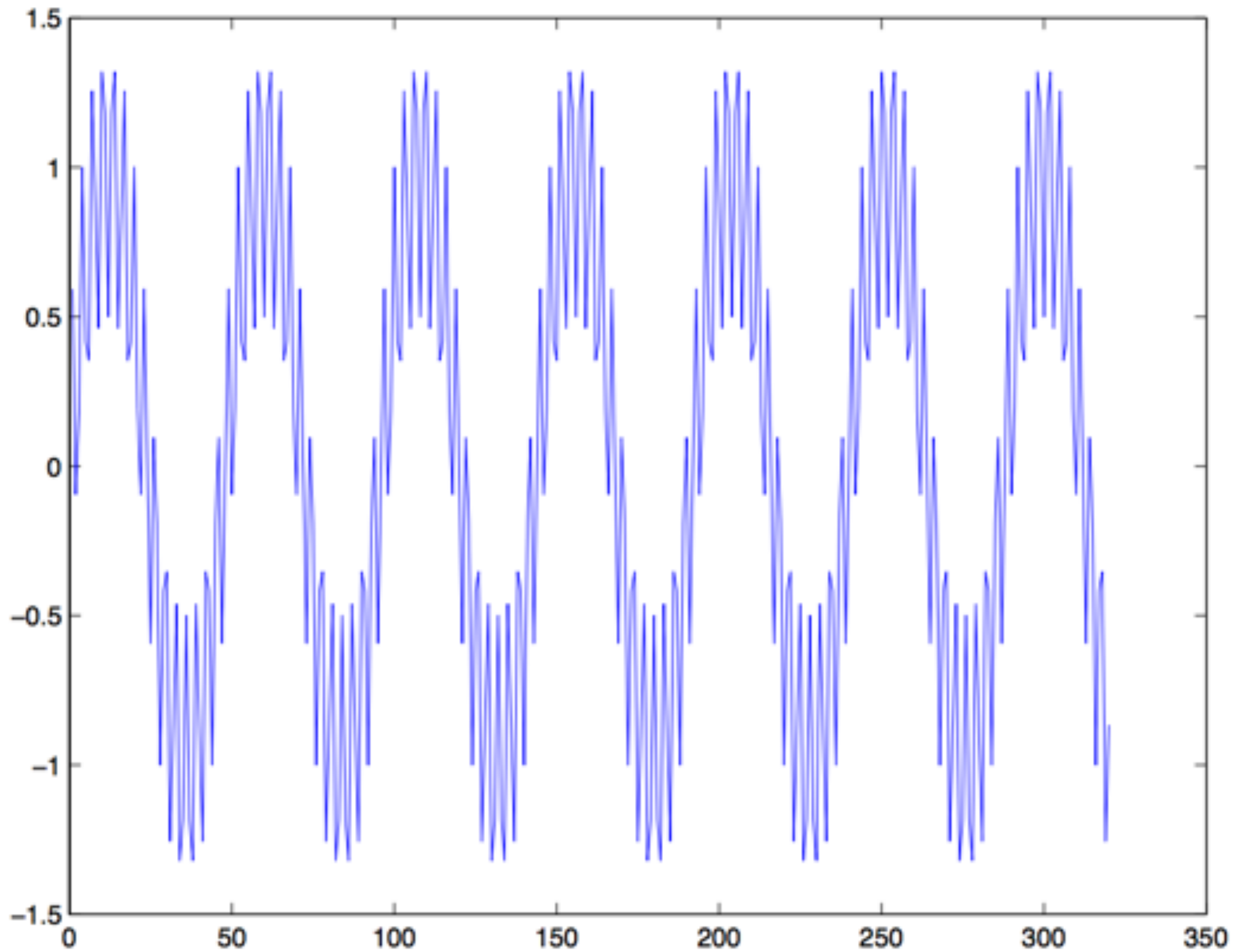
Source: Wikipedia

Frequency Domain Filtering



Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

Frequency Domain Filtering

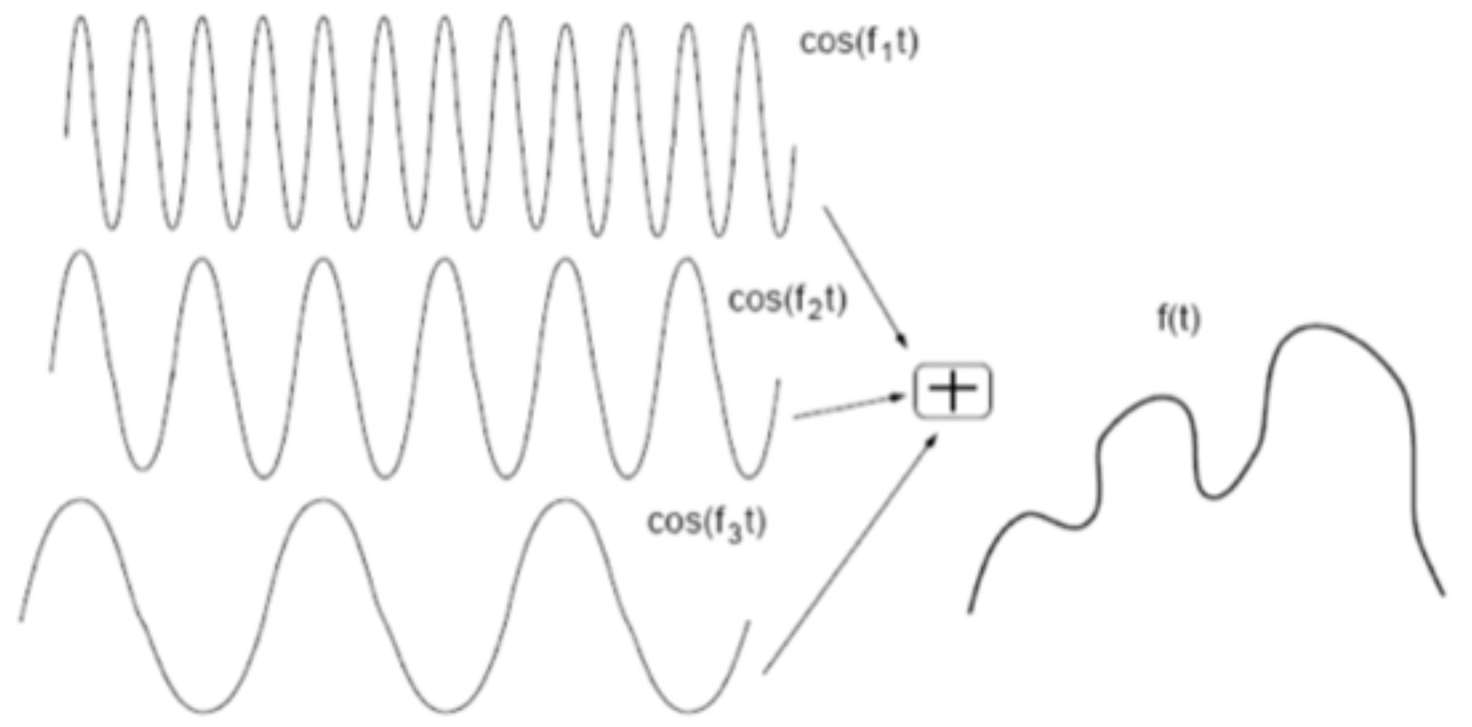


Noisy signal. Bummer!

Source: Prof. Mayank Goel, CMU

Frequency Domain Filtering

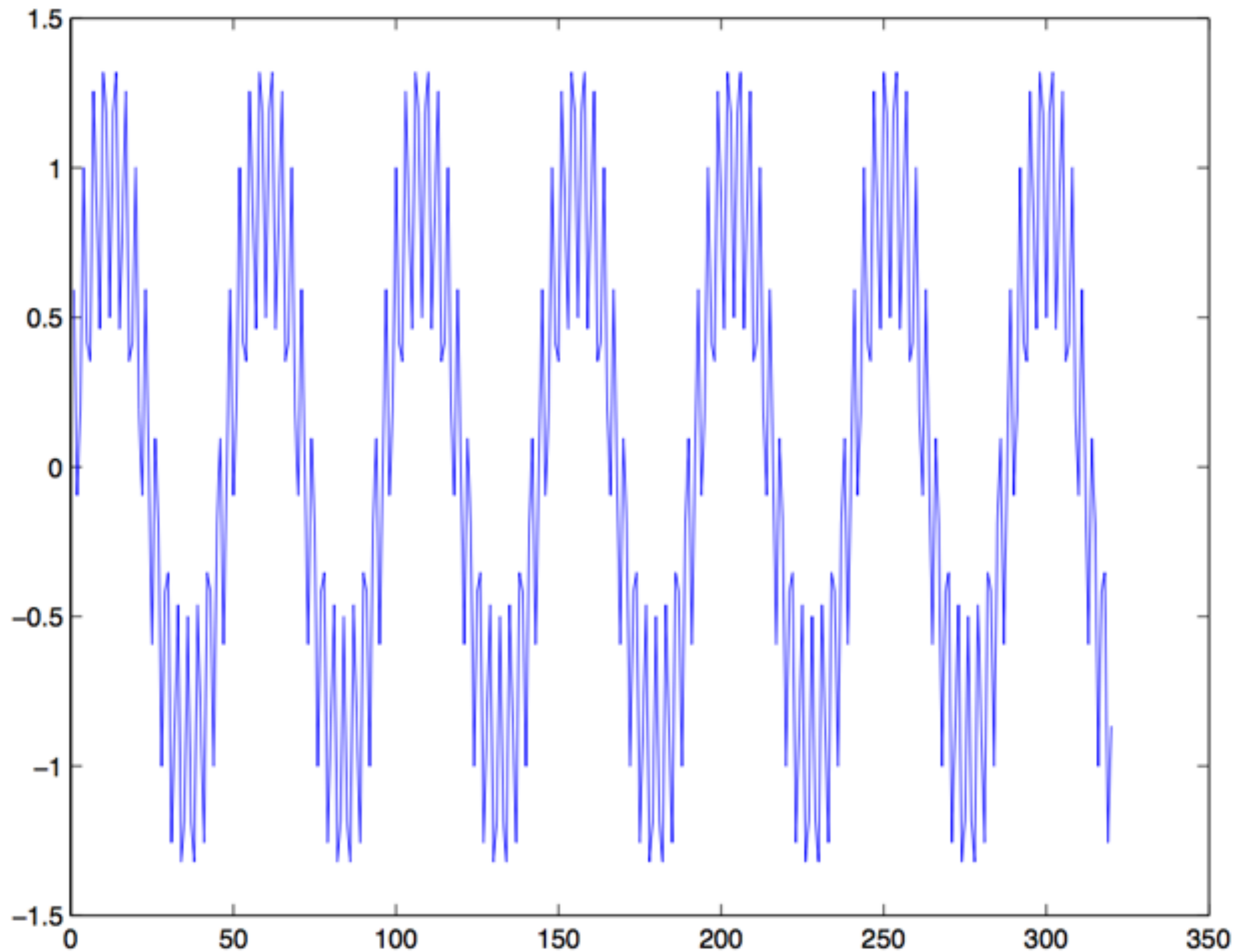
Jean-Baptiste Joseph Fourier



“any periodic waveform can be expressed as the sum of an infinite set of sine waves”

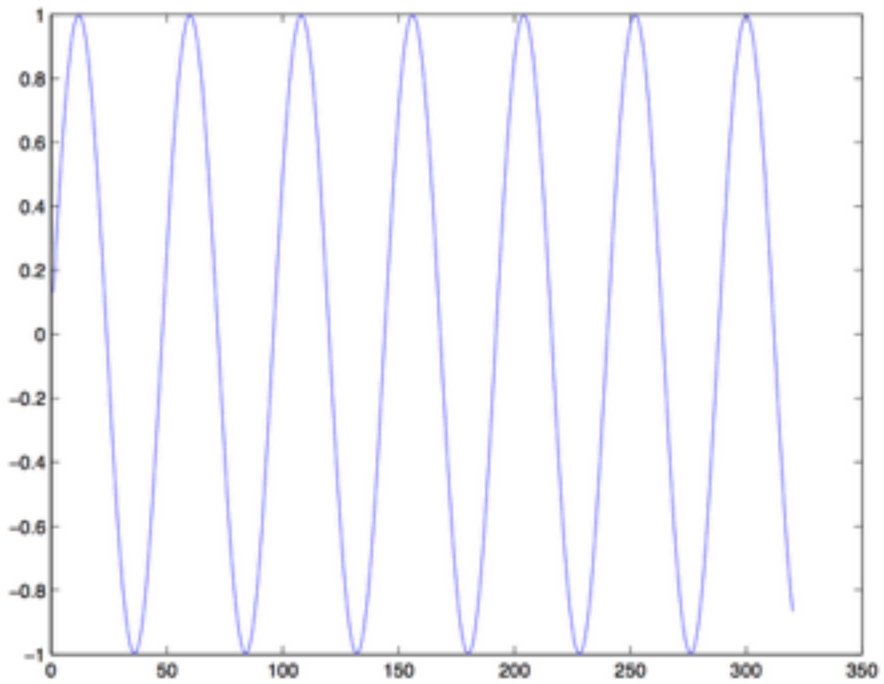
Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst and Wikipedia

Frequency Domain Filtering



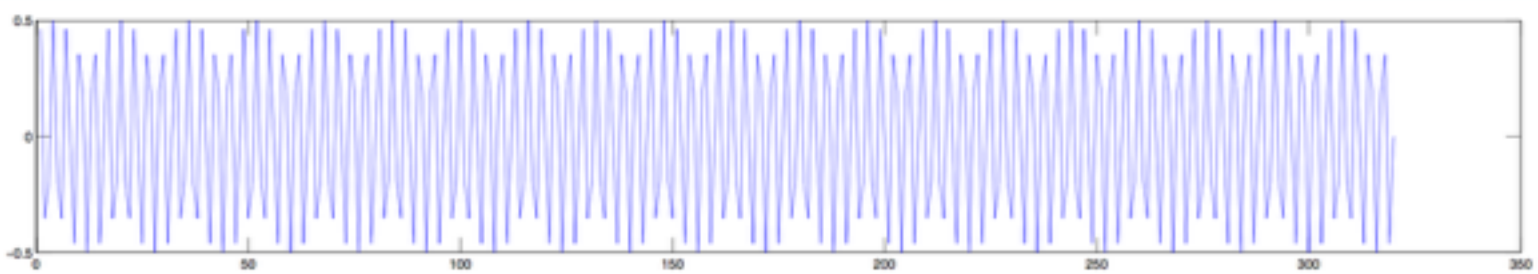
Source: Prof. Mayank Goel, CMU

Frequency Domain Filtering



1kHz

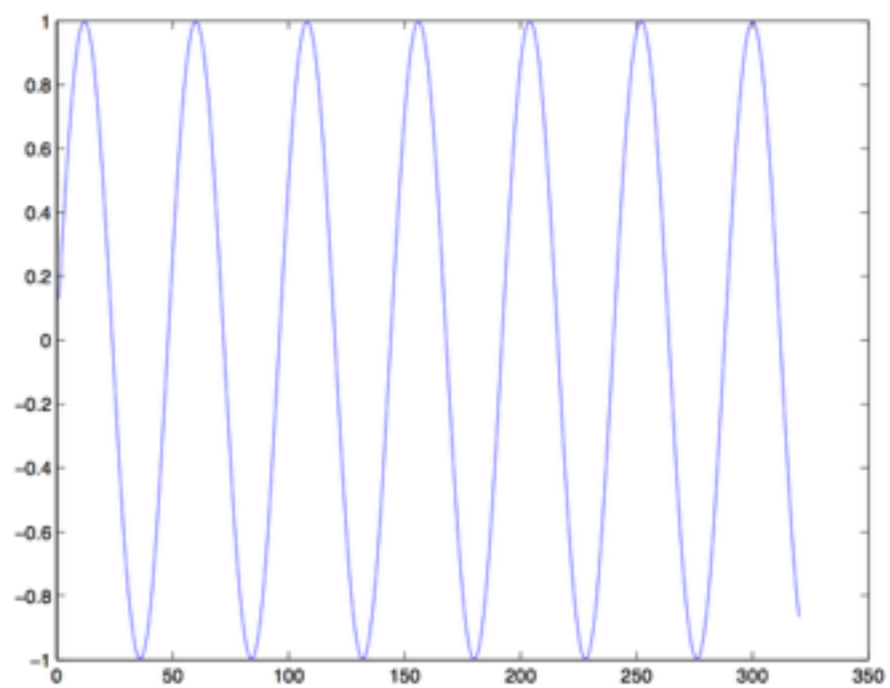
+



15kHz

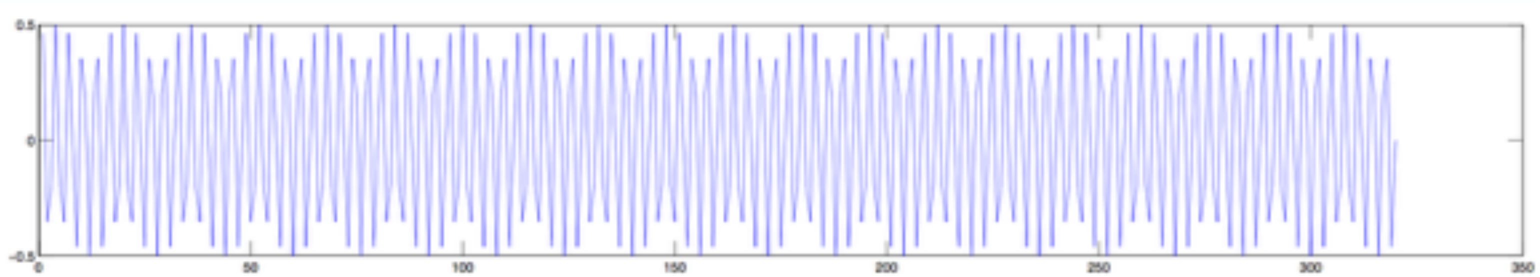
Source: Prof. Mayank Goel, CMU

Frequency Domain Filtering

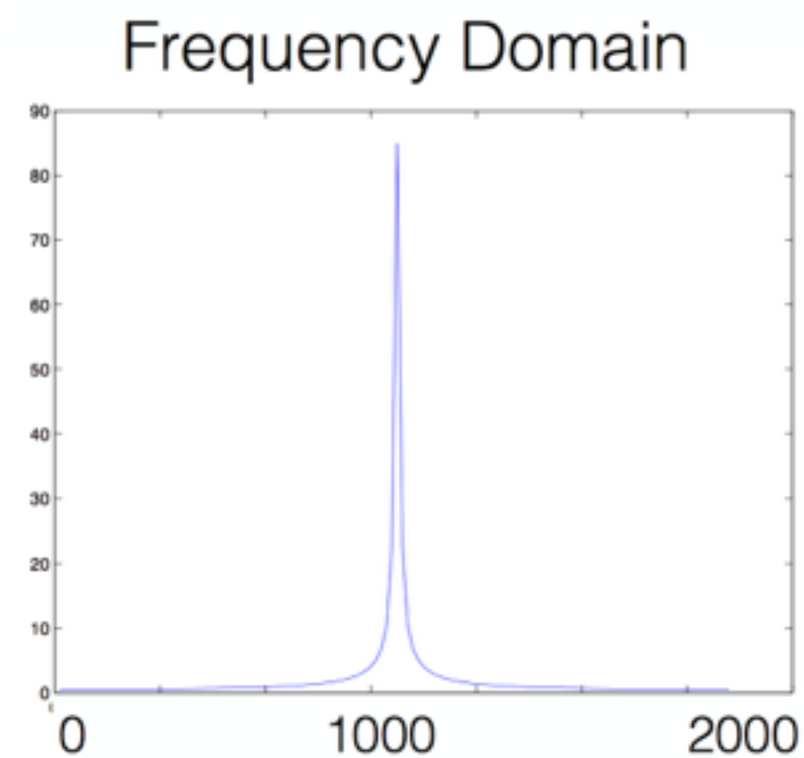


1kHz

+

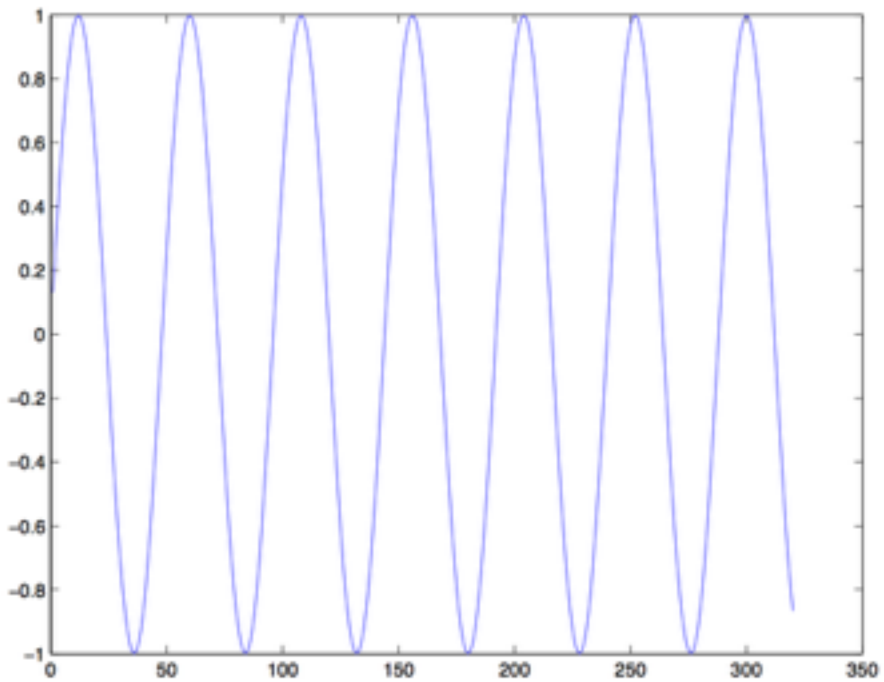


15kHz



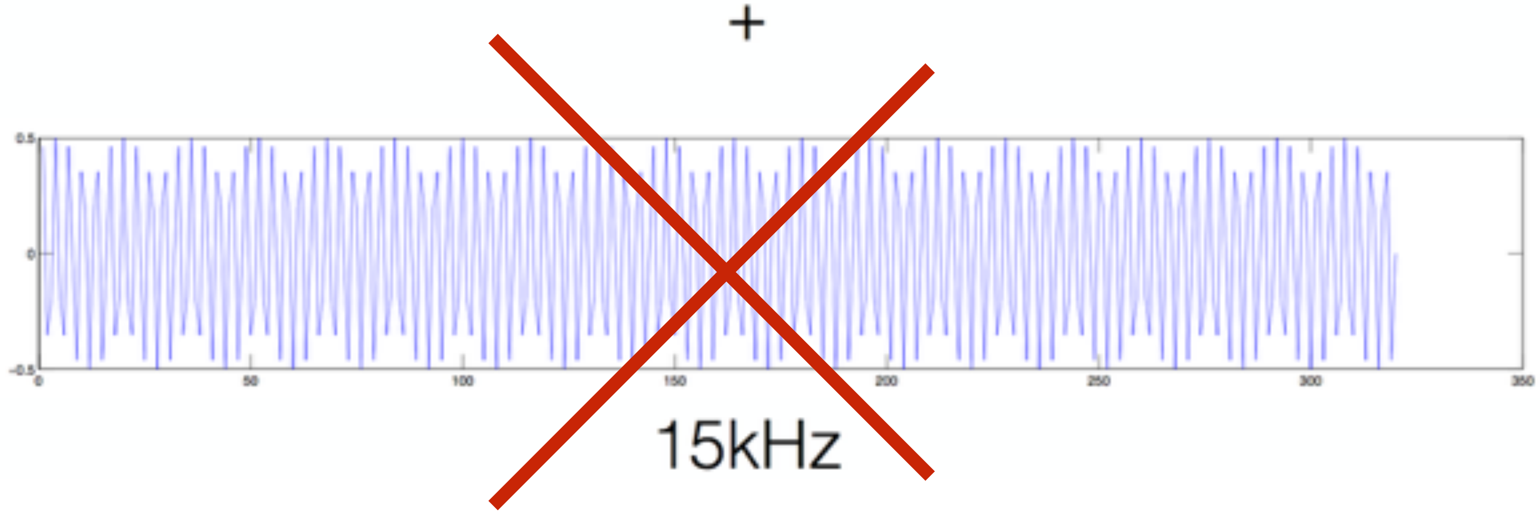
Source: Prof. Mayank Goel, CMU

Frequency Domain Filtering

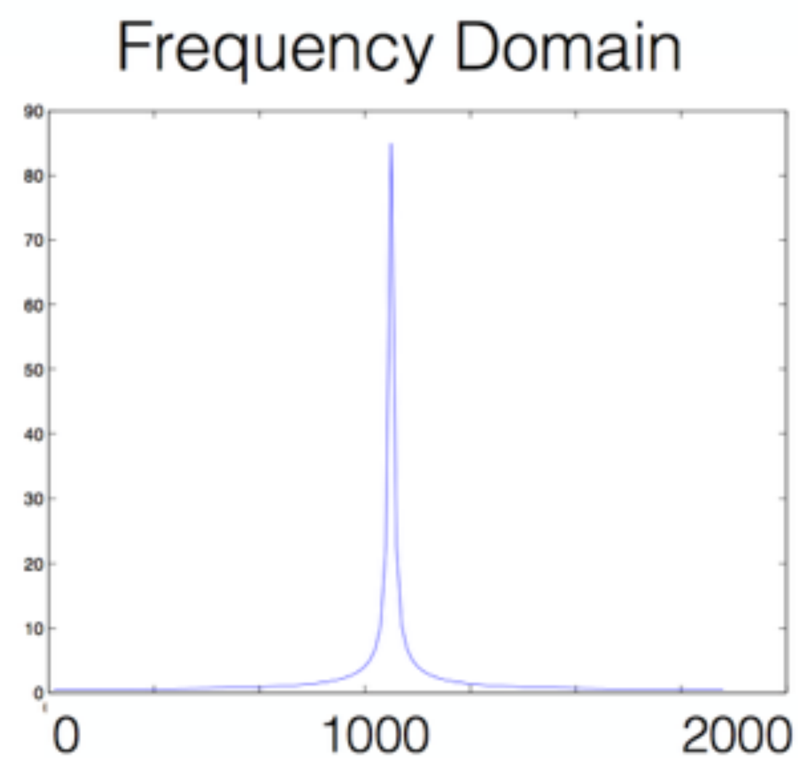


1kHz

+



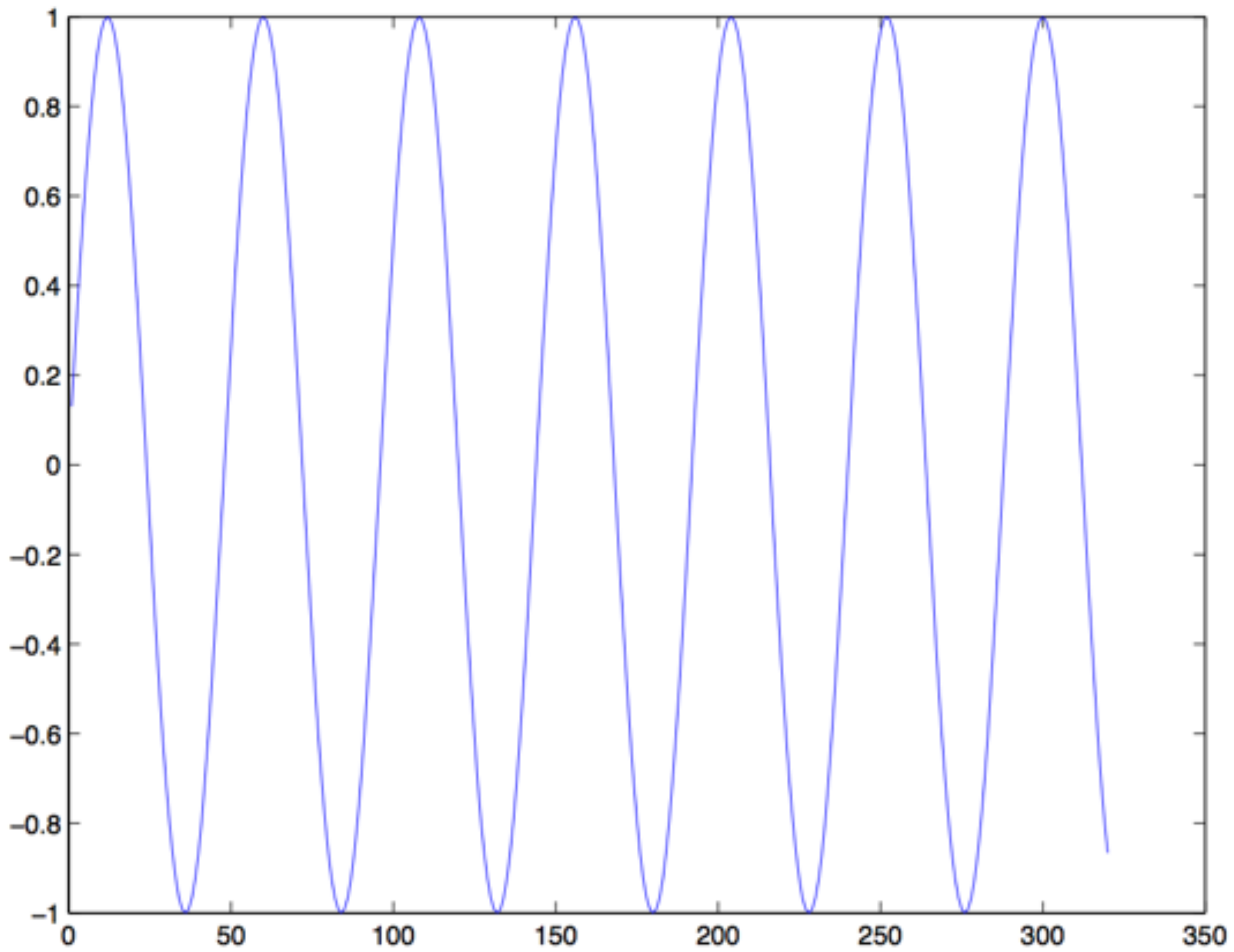
15kHz



Low-Pass Filter

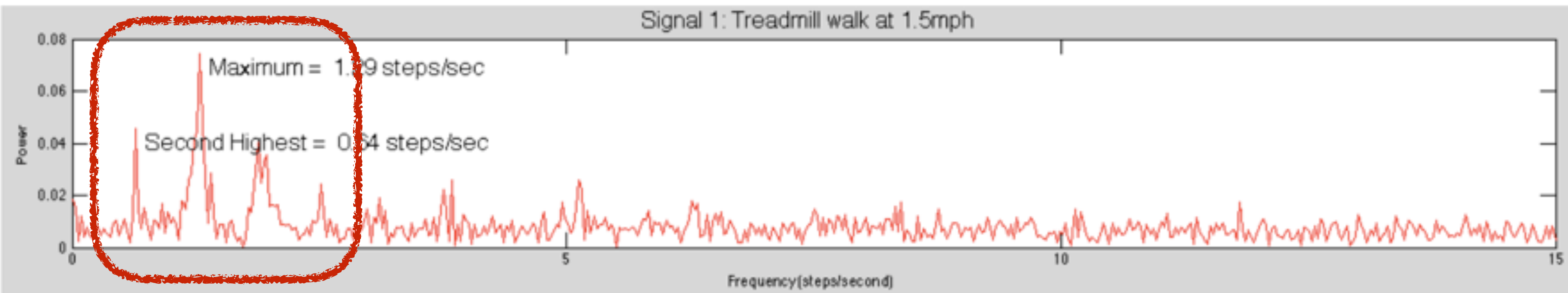
Source: Prof. Mayank Goel, CMU

Frequency Domain Filtering



Source: Prof. Mayank Goel, CMU

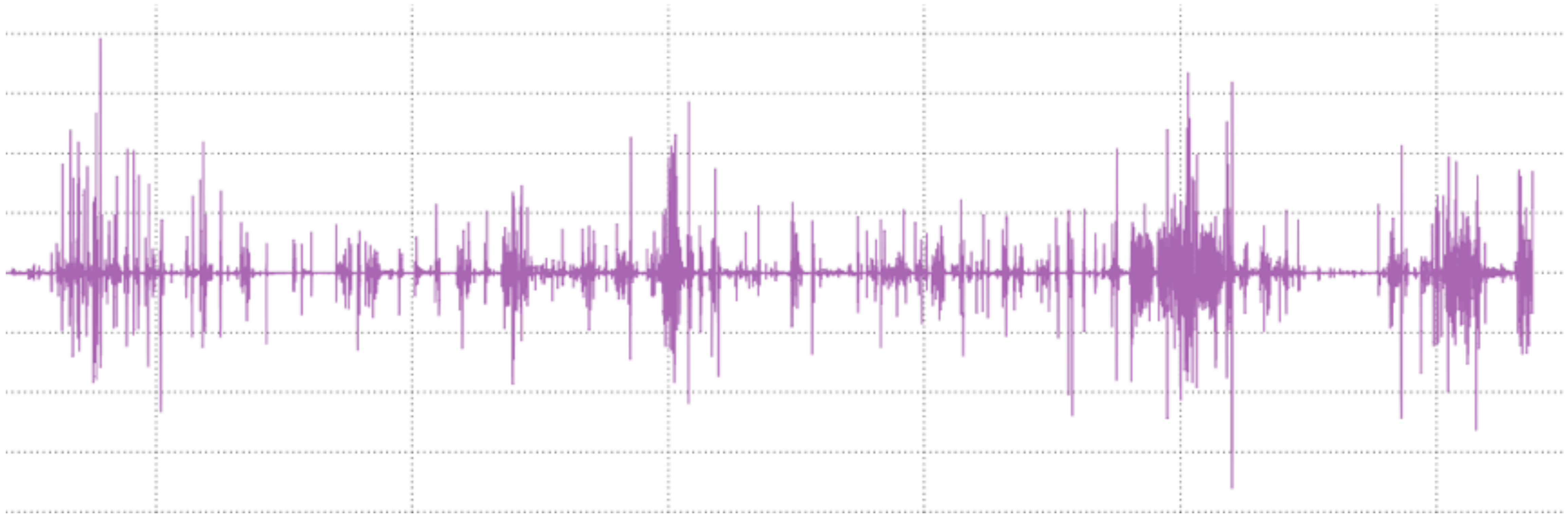
Frequency Domain Filtering



Walking on a treadmill

Source: Prof. Deepak Ganesan, Computer Science@UMASS Amherst

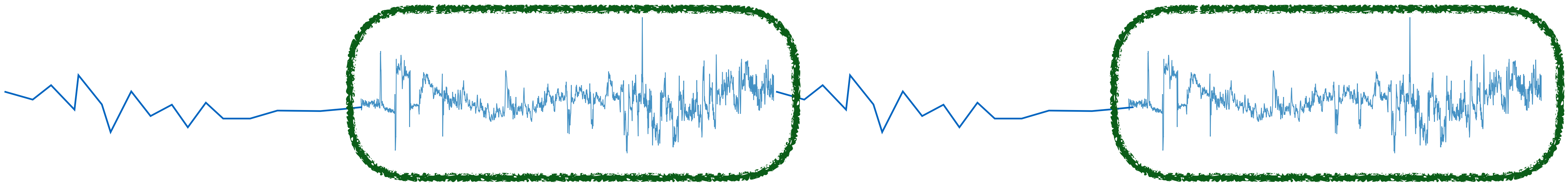
Extracting Features



Ready?

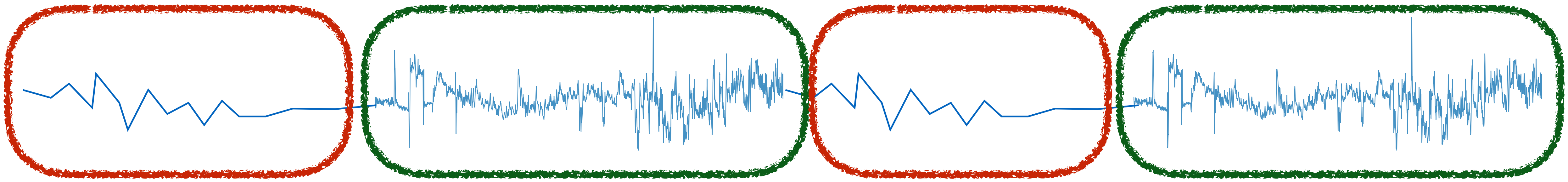
Don’t forget mean removal, normalization…

After Preprocessing…



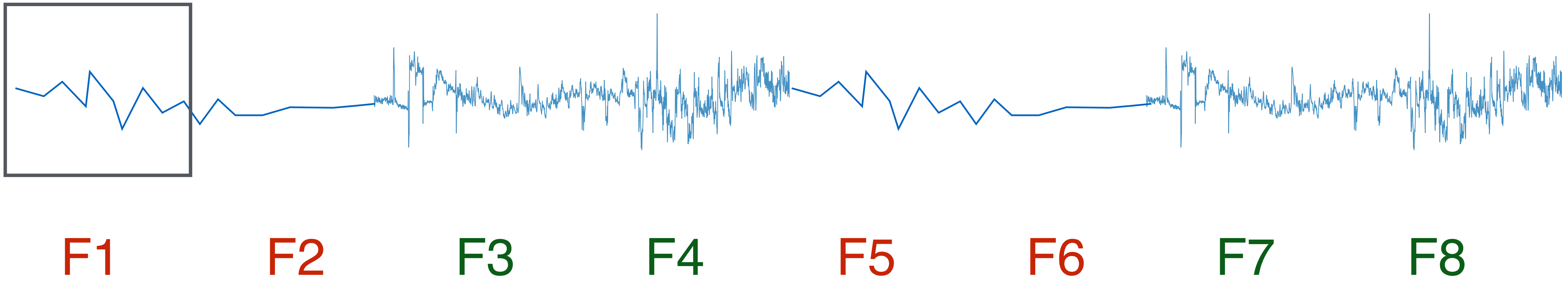
We want to count **these** activities

After Preprocessing…

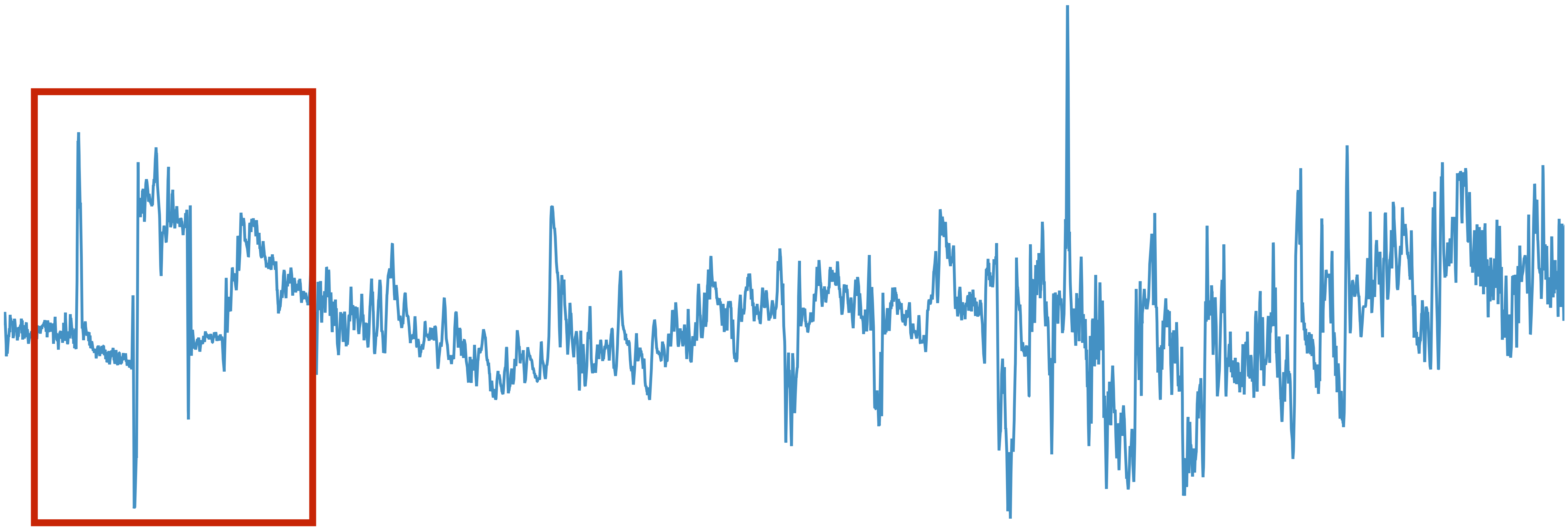


We want to count **these** activities

Frame and Feature Extraction



Frame and Feature Extraction

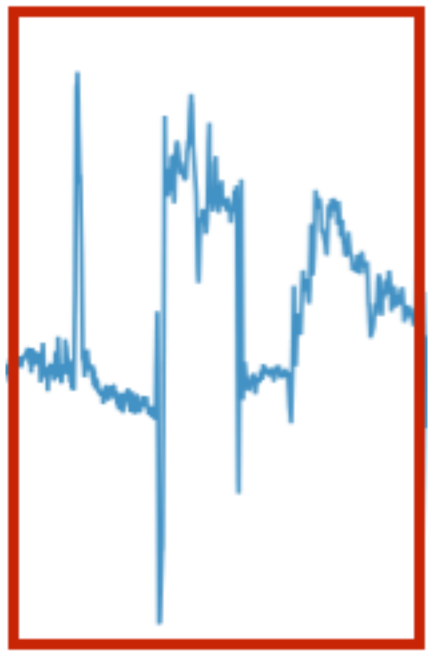


F_1 $\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$

F_2 $\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$

F_3 $\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$

Frame and Feature Extraction



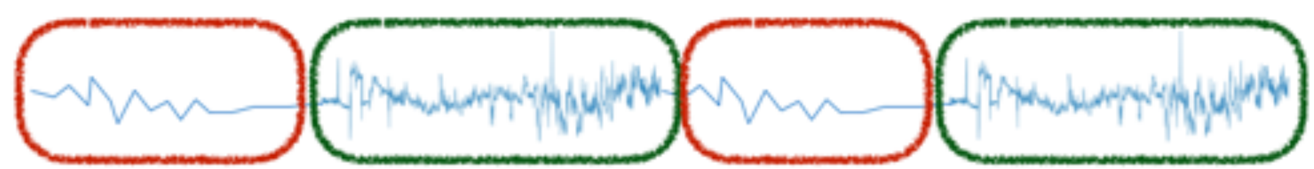
F_1

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

What should be the size of the window?

How much should the windows overlap?

Which features are commonly used?



GOAL: Distinguish numerically the frames we want to model

Common Features

Mean	Average value of the samples of signal **x**	$\mu_{\mathbf{x}} = \frac{1}{N} \sum_{n=0}^{N-1} x_n$
Variance	Power of values of signal **x** with its mean removed	$\sigma_x^2 = \frac{1}{N} \sum_{n=0}^{N-1} \lvert x_n - \mu_x \rvert^2$
Skewness	Measure of (lack of) symmetry in data distribution	$\frac{\sum_{n=1}^{N} (x_n - x)^3}{(N-1)s^3}$
Kurtosis	Measure of the shape of the data distribution	$\frac{\sum_{n=1}^{N} (x_n - x)^4}{(N-1)s^4}$
Total energy	Sum of squared moduli of signal **x**	$E_x = \sum_{n=0}^{N-1} \lvert x_n \rvert^2$
RMS	Square root of the average power of signal **x**	$\sqrt{P_x}$, where $P_x = \frac{E_x}{N} = \frac{1}{N} \sum_{n=0}^{N-1} \lvert x_n \rvert^2$
SMA	Signal Magnitude Area	$SMA = \frac{1}{t}\left(\int_0^t \lvert x(t) \rvert dt + \int_0^t \lvert y(t) \rvert dt + \int_0^t \lvert z(t) \rvert dt \right)$
ZCR	Zero-Crossing Rate	

Why are these features interesting in a mobile/wearable context?

Common Features

Time-domain metric	Ref(s)	Comp. cost	Storage req.	Precision	Mobile device
Mean	[5, 27, 50, 59]	very low	very low	single/int	Yes
Std. Deviation	[15, 22, 23, 30, 59]	very low	very low	double/single	Yes
Median	[2, 3]	medium	very low	single/int	Yes
Range	[11]	very low	very low	single/int	Yes
Maximum	[4, 59]	very low	very low	single/int	Yes
Minimum	[4, 59]	very low	very low	single/int	Yes
RMS	[8, 42, 53]	very low	very low	double/single	Yes
Integration	[15, 30, 42]	very low	very low	double/single	Yes
Correlation	[43, 57]	medium	low	double/single	Moderate
Cross-Correlation	[5, 16, 18, 28]	medium	low	double/single	Moderate
Differences	[55]	very low	very low	single/int	Yes
Zero-Crossings	[8, 11, 30, 49]	very low	very low	single/int	Yes
SMA	[6, 21, 37, 38]	low	low	single/int	Yes
SVM	[21, 44]	low	low	double/single	Yes
DSVM	[19]	low	low	double/single	Yes

Low Computation and Storage Cost

Papers

A Practical Approach for Recognizing Eating Moments with Wrist-Mounted Inertial Sensing

Edison Thomaz, Irfan Essa, Gregory D. Abowd

School of Interactive Computing
Georgia Institute of Technology
Atlanta, Georgia, USA

ABSTRACT

Recognizing when eating activities take place is one of the key challenges in automated food intake monitoring. Despite progress over the years, most proposed approaches have been largely impractical for everyday usage, requiring multiple on-body sensors or specialized devices such as neck collars for swallow detection. In this paper, we describe the implementation and evaluation of an approach for inferring eating moments based on 3-axis accelerometry collected with a popular off-the-shelf smartwatch. Trained with data collected in a semi-controlled laboratory setting with 20 subjects, our system recognized eating moments in two free-living condition studies (7 participants, 1 day; 1 participant, 31 days), with F-scores of 76.1% (66.7% Precision, 88.8% Recall), and 71.3% (65.2% Precision, 78.6% Recall). This work represents a contribution towards the implementation of a practical, automated system for everyday food intake monitoring, with applicability in areas ranging from health research and food journaling.

Author Keywords

Activity recognition; Food Journaling; Dietary Intake; Automated Dietary Assessment; Inertial Sensors

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Dietary habits have been studied by health researchers for many decades, and it is now well-understood that diet plays a critical role in overall human health [19]. To elucidate the mapping between diet and disease, nutritional epidemiologists have typically relied on validated dietary assessment instruments driven by self-reported data including food frequency questionnaires and meal recalls [33]. Unfortunately, these instruments suffer from several limitations, ranging from biases to memory recollection issues [15, 22]. For this reason, over the last 15 years, a large body of research has aimed at fully automating the task of food intake monitoring [11, 12, 21, 34]. Although significant progress has been achieved, most proposed systems have required individuals to wear specialized devices such as neck collars for swallow detection [2], or microphones inside the ear canal to detect chewing [20]. These form-factor requirements have severely limited the immediate practicality of automated food intake monitoring in health research.

There are two key technical challenges in building a fully automated food intake monitoring system: (1) recognizing *when* an individual is performing an eating activity, and then (2) inferring *what* and *how much* the individual eats. In this paper we focus on recognizing when an *eating moment* is taking place, which includes having a sit-down meal with utensils, eating a sandwich, or having a snack.

Our aim with this work is to explore a *practical* solution for eating moment detection; we describe an approach leveraging the inertial sensor (3-axis accelerometer) contained in a popular off-the-shelf smartwatch. This approach contrasts with methods that require either multiple sensors or specialized forms of sensing.

Our eating moment recognition method consists of two steps. First, we perform food intake gesture spotting on the stream of inertial sensor data coming from the smartwatch, reflecting arm and hand movements. Secondly, we cluster these gestures across the time dimension to unearth eating moments. To evaluate our approach, we first ran a formative study with 20 participants to validate our experimental design protocol and instrumentation. Informed by this pilot, we conducted user studies that resulted in three datasets, (1) a laboratory semi-controlled study with 20 participants, (2) an in-the-wild study with 7 participants, and (3) 422 hours of in-the-wild data for one participant collected over the course of 31 days.

The contributions of this work are:

- A practical system for eating moment estimation leveraging the inertial sensor (3-axis accelerometer) of a popular off-the-shelf smartwatch.
- An evaluation of a lab-trained eating moment classification model in-the-wild with two datasets: 7 participants over one day (76.1% F-score, 66.7% Precision, 88.8% Recall), and one participant over 31 days (71.3% F-score, 65.2%

Activity Recognition from User-Annotated Acceleration Data

Ling Bao and Stephen S. Intille

Massachusetts Institute of Technology
1 Cambridge Center, 4FL
Cambridge, MA 02142 USA
intille@mit.edu

Abstract. In this work, algorithms are developed and evaluated to detect physical activities from data acquired using five small biaxial accelerometers worn simultaneously on different parts of the body. Acceleration data was collected from 20 subjects without researcher supervision or observation. Subjects were asked to perform a sequence of everyday tasks but not told specifically where or how to do them. Mean, energy, frequency-domain entropy, and correlation of acceleration data was calculated and several classifiers using these features were tested. Decision tree classifiers showed the best performance recognizing everyday activities with an overall accuracy rate of 84%. The results show that although some activities are recognized well with subject-independent training data, others appear to require subject-specific training data. The results suggest that multiple accelerometers aid in recognition because conjunctions in acceleration feature values can effectively discriminate many activities. With just two biaxial accelerometers – thigh and wrist – the recognition performance dropped only slightly. This is the first work to investigate performance of recognition algorithms with multiple, wire-free accelerometers on 20 activities using datasets annotated by the subjects themselves.

1 Introduction

One of the key difficulties in creating useful and robust ubiquitous, context-aware computer applications is developing the algorithms that can detect context from noisy and often ambiguous sensor data. One facet of the user's context is his physical activity. Although prior work discusses physical activity recognition using acceleration (e.g. [17,5,23]) or a fusion of acceleration and other data modalities (e.g. [18]), it is unclear how most prior systems will perform under real-world conditions. Most of these works compute recognition results with data collected from subjects under artificially constrained laboratory settings. Some also evaluate recognition performance on data collected in natural, out-of-lab settings but only use limited data sets collected from one individual (e.g. [22]). A number of works use naturalistic data but do not quantify recognition accuracy. Lastly, research using naturalistic data collected from multiple subjects has focused on

A. Ferscha and F. Mattern (Eds.): PERVASIVE 2004, LNCS 3001, pp. 1–17, 2004.