2011 IEEE 15th International Symposium on Consumer Electronics

ACCELEROMETER-BASED SWINGING GESTURE DETECTION FOR AN ELECTRONIC HANDBELL

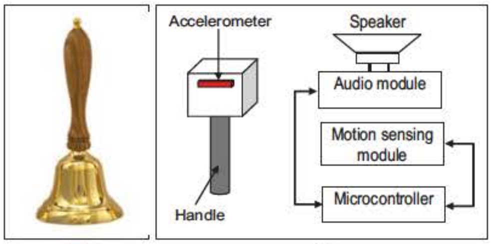
Liyanaarachchi Lekamalage Chamara Kasun, Wooi-Boon GOH
School of Computer Engineering, Nanyang Technological University, Singapore 639798
CHAMARAKASUN@ntu.edu.sg; ASWBGOH@ntu.edu.sg

ABSTRACT

This paper tackles the problem of detecting the swinging action of an electronic handbell. It describes a threshold based algorithm that is able to detect an orientation free swinging motion using only the X and Y axis signals of an accelerometer that is mounted at the end of a handle. Equations governing the accelerations of the accelerometer are defined. The equations are used to select the appropriate accelerometer axis for swing motion detection, which were X axis and Y axis. The characteristics of the swing motion are identified empirically and incorporated to the swing detection algorithm. The experimental results for swinging motion performed by 4 users on the electronic bell show that the accuracy of swing motion detection is 95.3%.

1. INTRODUCTION

Handbells are often use by pre-school educators to teach children musical concepts such as rhythm, note sequences and timing. Figure 1a shows a physical handbell, which is usually one of a set of many; each carefully tuned to ring at a specific frequency so that it plays a particular note when swung. They are generally costly and can only ring out the sound of one specific note. We are interested in developing an electronic equivalent of such handbells, which can be easily reprogrammed to ring a synthetic bell sound of any note of choice when swung in the same physical manner. Figure 1b shows the physical construction of such a device and a schematic block diagram of its major components, which consist of a MCU, audio module and motion sensing module, which comprises of a 3-axis accelerometer. The main issue addressed by this paper is the development of an appropriate algorithm that would allow this swinging motion gesture to be detected in an orientation free manner using appropriate signals from the single 3-axis accelerometer.



(a) (b)

Figure 1. (a) A physical handbell. (b) The electronic handbell and its various component modules

Gestures are expressive means of communicating between consumer devices and human. Gesture recognition is performed using video [1, 2, 3], touch screen [4] or by using micro electro mechanical systems (MEMS) such as accelerometers [5]. Accelerometers are devices which measures acceleration along a predefined axis. It is capable of measuring static (acceleration created by gravity) and dynamic acceleration (acceleration created by movement). Static acceleration is used to calculate the tilt or orientation of a device, while dynamic acceleration is used for gesture recognition and fall down detection. The advantage of using video based systems for gesture recognition is that the user does not need to wear any sensors or hold any device. However, vision based gesture recognition are complicated and prone to the problem of self-occlusion and fast gesture speeds due to the limited frame rates of the video camera.

Wearable devices for gesture recognition are easier to implement because the orientation of device can fixed by the way it is secured on the human body, for example with the use of gloves or strap-on embedded sensors. Near symmetrical hand held devices such as that shown in Figure 1b are more problematic for accelerometer-based gesture recognition since the orientation of the device can be rather arbitrary during motion execution. One way to avoid this problem is to define the physical dimensions of the hand held device in a manner that allows human ergonomics or device function to persuade the user to hold it in a consistent orientation. Some examples include gesture-capable devices such as the iPhone [6] from Apple and the Wii remote [7] from Nintendo. Phones have a screen and are usually designed to be flat, which guide users to instinctively hold them in a specific orientation that is comfortable and permit good view of the screen. Similarly, the Wii remote models after typical remote controller-type interfaces, which have buttons on only one face of an elongated device, which encourage handling in a pre-deterministic orientation. In our case, in mimicking the symmetrical form factor of typical handbells, the orientation constraint required to make accelerometer-based gesture recognition easier has been compromised. This paper presents an algorithm that is able to overcome this limitation.

2. EXISTING APPROACHES

Existing gesture recognition algorithms using signals from accelerometer-based motion sensors can be categorized based on the following gesture classification methods:

- Hidden Markov Models (HMM)
- Dynamic Time Warping (DTW)

978-1-61284-842-6/11/$26.00 ©2011 IEEE

- Support Vector Machine (SVM)
- Machine Learning Algorithms (Fuzzy logic and Artificial Neural Networks)
- Threshold-based

Hidden Markov Models (HMM)

Joselli and Clua [8] presented a HMM based gesture recognition method for mobile phones. The authors propose that the 3–axis accelerometer values to be processed in 5 stages. The initial stage will find the start and the end of the motion. The second stage is used to reduce noise via a low pass filter. The third stage reduces the amount of data sent to the HMM by using a k-mean algorithm. The fourth stage uses an existing proven 8 states HMM for gesture identification. In the final stage a Bayesian classifier is used to remove non gesture movements. Dictionary of 10 gestures are created by the authors and the gesture recognition accuracy for the set of gestures are between 75% - 97.5%. If the HMM parameters; the number of states of HMM, number of distinct observation symbols per state, state transition probability, observation symbol probability distribution in a state and initial state distribution is known, then a gesture could be modeled by HMM. Kong et al. [12] proposes an algorithm to capture these parameters, allowing to model gestures using HMM. The authors however have not determined the gesture recognition accuracy of the proposed gesture model.

Dynamic Time Warping (DTW)

The same gesture performed by various users is not exactly same. Hence gesture recognition algorithms should be capable of detecting gestures with minimal user dependence. In order to tackle this issue Akl and Valaee [9] proposed a gesture recognition method based on DTW. The Wii remote is used as the hardware platform and the authors have created a dictionary of 18 gestures. The experimentation results show that the gesture recognition accuracy of the proposed method is between 90% - 99.79%. Same gesture performed by the same user on different days tends to change. In order to tackle this problem Liu et al. [10] proposed user personalized gesture recognition method based on DTW called µwave. For the same gesture the proposed method in contrast to [9] uses different gesture templates for each user, which changes daily. A dictionary of 8 gestures has been defined. They implemented the algorithm using Wii remote and the gesture recognition accuracy is between 98.6%-98.9%.

Support Vector Machine (SVM)

The effectiveness of any gesture recognition method depends upon the features extracted. He et al. [11] proposed 3 methods for feature extraction namely, discrete cosine transform (DCT), Fast Fourier Transform (FFT) and a hybrid method which uses wavelet packet decomposition (WPD) with FFT. Gesture recognition is performed by using SVM in all 3 cases. A dictionary of 17 gestures is defined and a mobile phone with a 3-axis accelerometer is used for implementation. DCT, FFT and the hybrid method respectively produces gesture recognition accuracy of 64.51% - 95.49%, 70.44% - 94.29% and 71.98% - 95.49%.

Machine Learning Algorithms

Gesture is nebulous by nature hence gesture recognition can be based on highly adaptable algorithms such as fuzzy logic and artificial neural networks. Helmi and Helmi [13] propose a gesture recognition method based on fuzzy logic and neural networks. Dictionary of 25 gestures are defined and implemented on a wireless accelerometer device which transmits the acceleration data to a computer. The authors showed that machine learning algorithms are the ideal solution for gesture detection because the gesture recognition accuracy is 100%. But the drawback is the required huge computational resources to perform gesture detection, which is not appropriate in an embedded processing situation like the electronic bell. Bailador et al. [14] used a Continuous Time Recurrent Neural Network (CRTNN) for gesture recognition. As CRTNN is computationally less expensive, it can perform gesture recognition in real-time. Dictionary of 8 gestures are defined and gesture recognition accuracy is 64%. But when the gestures are done in controlled manner (user sitting while performing gestures, gestures are performed non-continuously) recognition accuracy of up to 94% could be obtained.

Threshold based

Parsani and Singh [15] proposed using the running variance to find activity in the accelerometer. Once the running variance is higher than a predetermined threshold, a sub gesture detection algorithm is executed to find the specific gesture. Dictionary of 6 gestures are defined by the authors and implemented on a programmable system on chip which communicates with a PC using Bluetooth. However the authors haven't performed any experiments to determined the gesture recognition accuracy of their propose method.

Many of the gesture recognition methods reviewed require training phase in which templates of gestures to be classified are generated when sufficient exemplar gestures are supplied. The requirement of training can make the method somewhat inconvenient. Most importantly, of the methods reviewed, the threshold-based method requires the least computational resources such as memory and processing power, making it an ideal approach for an embedded solution which has constrained resources. In this paper we propose a novel threshold-based gesture detection algorithm to detect the orientation-free swinging gestures on a physical construction such as that shown in Figure 1b.

3. SWINGING MOTION

The physical construction of the electronic bell and the respective three axes of the embedded accelerometer are shown in Figure 2. For the ease of discussion, 3 perpendicular reference axes called X, Y and Z are created with respect to the user. Any forward-backward movement performed by the user is called X-axis movement, left-right movement performed by the user is called Y-axis movement and any up-down or rotation movement performed by the user is called Z-axis movement. The red dot shown in Figure 2 describes the orientation of the accelerometer X, Y, Z axis. If the user holds the electronic bell as shown in figure 2 where the red dot is in the upper right corner, then the accelerometer axes and reference axes coincide and called as the ideal position. Because X, Y, Z acceleration created by movement correlates to the X, Y, Z accelerations measured by the accelerometer.

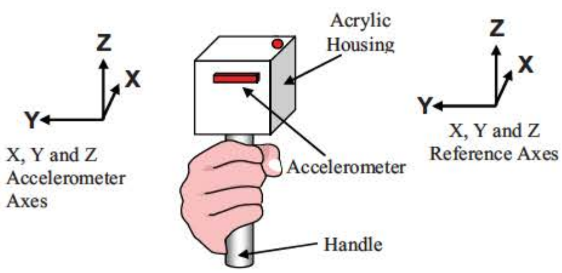


Figure 2. Physical construction of the electronic bell and the directions of the various accelerometer axes.

When the electronic bell is rotated around the Z reference axis, angle μ defines the rotation from the ideal position as show in figure 4a. The swing angle α defines the movement of the electronic bell when swinging motion is performed from the pivot point as shown in Figure 4b. This electronic bell is used to analyze the static and dynamic acceleration created along the 3 reference axis when a swinging motion is performed. The swinging motion is essentially an X axis and/or Y axis movement, but not a Z axis movement due to the physical characteristics of the electronic bell as shown in Figure 3.

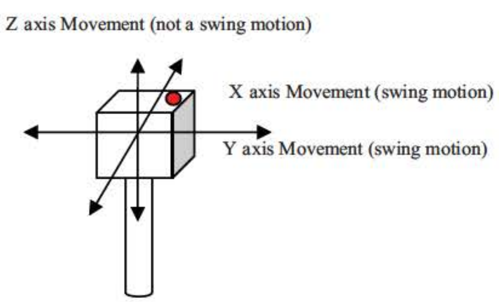


Figure 3. Valid swinging motions in the X and Y directions

The user can perform the swing motion by using the wrist or the elbow and the pivot point will be respectively wrist or elbow. It is most unlikely that a person will use both the elbow and wrist simultaneously to perform a swing motion. Hence the person will create a graceful arc when the swing motion is performed.

The instantaneous tangential acceleration of a circular motion can be described by using equation 1.

$$a_i = r.\theta \tag{1}$$

Where,
a_i is the instantaneous linear tangential acceleration,
r is the radius of the swing motion and
θ is the instantaneous angular acceleration.

The instantaneous linear tangential acceleration is equivalent to the dynamic acceleration detected by the accelerometer when a swinging motion is performed.

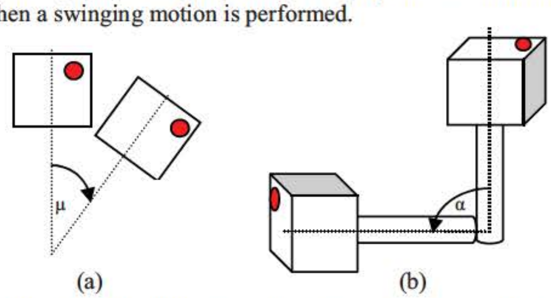


Figure 4. (a) Rotation along the Z axis given by μ and (b) the swing angle α when a swinging motion performed

The total accelerations detected by the accelerometer along the 3 axes X, Y, Z are respectively described by equations 2, 3 and 4.

$$a_{Tx} = a_{dY}\sin\mu + a_{sY}\sin\mu + a_{dX}\cos\mu + a_{sX}\cos\mu \tag{2}$$

$$a_{Ty} = a_{dX}\sin\mu + a_{sX}\sin\mu + a_{dY}\cos\mu + a_{sY}\cos\mu \tag{3}$$

$$a_{Tz} = a_{sZ}\cos\alpha \tag{4}$$

Where,
a_{Tx} :- Total acceleration detected by the accelerometer along the accelerometer X axis.
a_{Ty} :- Total acceleration detected by the accelerometer along the accelerometer Y axis.
a_{Tz} :- Total acceleration detected by the accelerometer along the accelerometer Z axis.
a_{dX}:- Dynamic acceleration along the reference X axis.
a_{dY}:- Dynamic acceleration along the reference Y axis.
a_{sX}:- Static acceleration along the reference X axis.
a_{sY}:- Static acceleration along the reference Y axis.
a_{sZ}:- Static acceleration along the reference Z axis.
α:- The angle of the swing motion performed.
μ:- The angle rotated along the Z axis of the reference axis

By analyzing equations 2, 3 and 4, it can be inferred that the easiest way to detect a swinging motion is to use equation 4. According to equation 4, when the electronic bell is kept in the upright position (as shown in Figure 2), the Z axis of the accelerometer will detect -1g and when horizontal, it will measure 0g. When a forward-backward swinging is performed with α less than π/18 radians (10 degrees) and μ equal to 0 radians, the acceleration generated in the Z axis of the accelerometer will be insignificant compared to the X axis accelerometer values as shown in Figure 5. This is because according to equation 2, the X axis accelerometer output is affected by dynamic acceleration unlike the Z axis. Hence the solution is to use X axis and Y axis accelerometer values to detect the swinging motion instead of the Z axis accelerometer values.

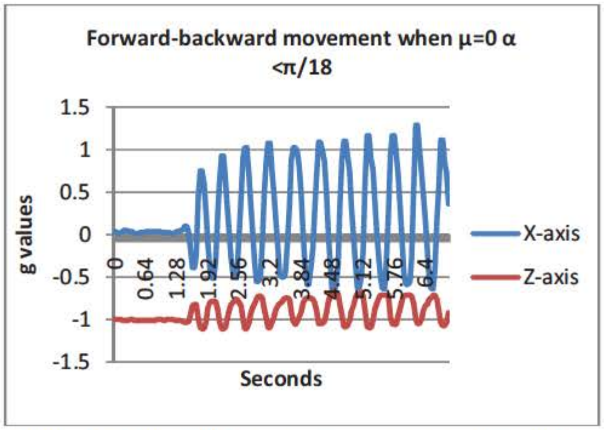


Figure 5. Z axis and X axis accelerometer g values when forward-backward swing is performed

3.1. CHARACTERISTICS OF SWING MOTION

The developed algorithm utilizes the following basic characteristics of the swinging motion for identification.

Accelerometer Signal Noise

The augmented signal of the X and Y axis accelerometer will be twice as noisy as the original individual signals, because the accelerometer is not noise free. Hence the X and Y axis of the accelerometer must be evaluated individually. Figure 6 shows X axis and Y axis accelerometer g values when forward-backward swing motion is performed with μ equal to 0 radians and α is less than $\pi/2$ radians (90 degrees).

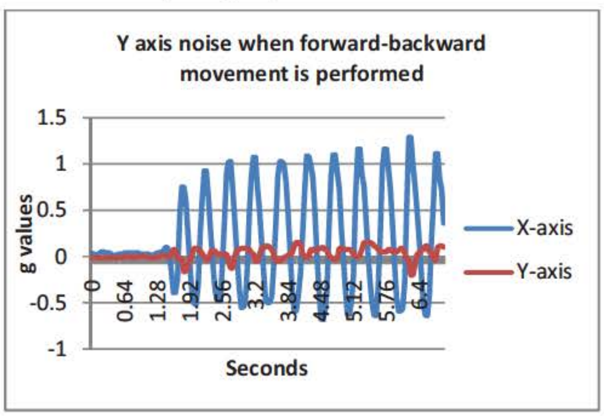


Figure 6. Noise generated when forward-backward movement swing motion with μ equal to 0 radians and α less than $\pi/2$ radians.

As illustrated in Figure 6, even though the Y axis of the accelerometer must be 0g, the values are observed to be fluctuating.

Swinging Motion Characteristics

A breakdown of the forward-backward swinging motion (with μ equal to 0 radians and α is less than $\pi/2$ radians) signal is show in Figure 7.

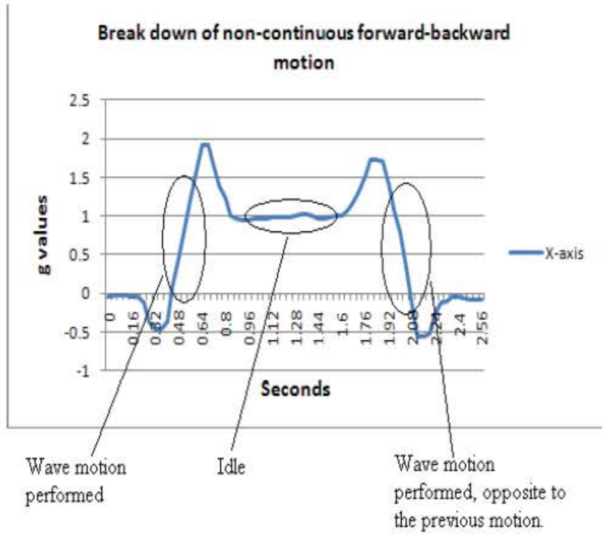


Figure 7. Break down of a non-continuous forward-backward swing motion with μ equal to 0 radians and α less than $\pi/2$.

According to Figure 7, the circled acclivity shows the swing motion occurring to one side while the circled declivity shows the swing motion occurring to the opposite side of the initial swing motion. When μ is between 0 and 2π, X and Y axis signal values can merely infer the direction of the swinging motion relative to the previous swing.

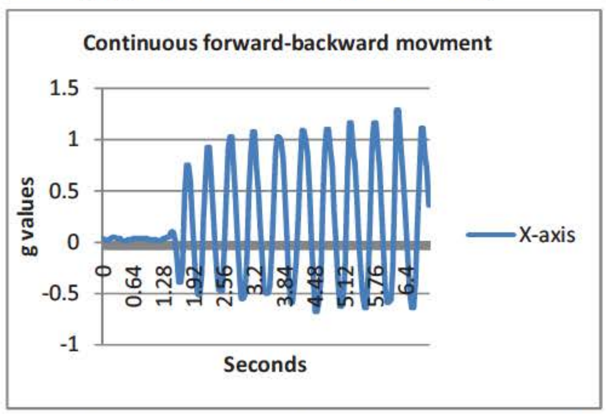


Figure 8. Continuous forward-backward swing motion with μ equal to 0 radians and α less than $\pi/2$.

In order to find the exact direction of the swinging motion, the relationship between the reference axes and the accelerometer axes must be established by using the value μ and the starting position of the electronic bell. When the swinging motion is performed continuously, the circled acclivity and circled declivity in Figure 7 merges together to create a sinusoidal swing as shown in Figure 8.

Different values of μ (rotation around reference Z)

When the value of μ is not 0, π/2, π and 3π/2 accelerations are measured in both X and Y accelerometer axes. In this case the axis which produces the largest acceleration values is used to decide the swinging motion, as shown in Figure 9.

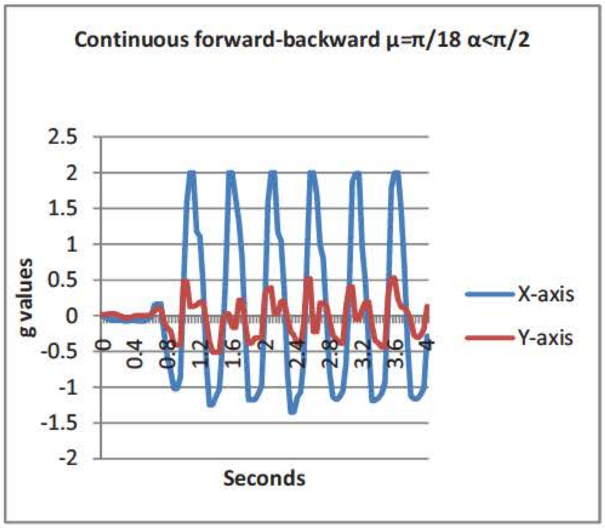


Figure 9. Continuous forward-backward swing motion with μ equal to π/18 radians and α less than π/2.

But when μ is π/4, 3π/4, 5π/4 and 7π/4 both X and Y accelerometer axes will generate acceleration values which are equal. In this case the signal of the X axis is used to decide the swing motion as shown in Figure 10.

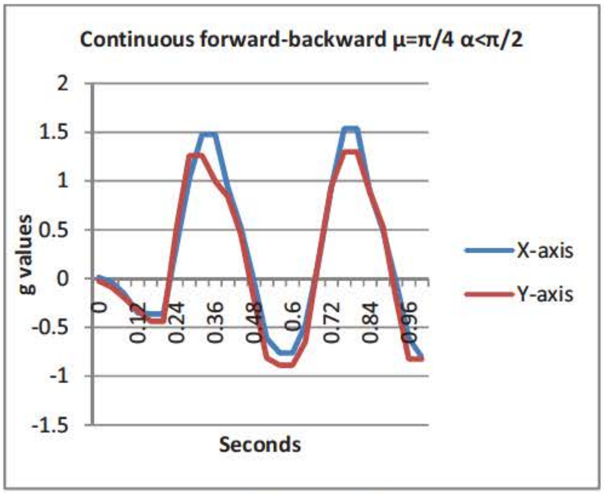


Figure 10. Continuous forward-backward swing motion with μ equal to π/4 radians and α less than π/2.

3.2. SWING DETECTION ALGORITHM

The algorithm developed is capable of detecting swinging motion when μ is between 0 and 2π radians (rotation along the reference Z axis from the ideal position).

Pseudo Code

1. Perform median filtering (windows size is 3) on X and Y axis accelerometer values.
2. Is Acclivity or declivity detected on X or Y axis? If yes go to step 3 else go to step 5.
3. Is the acclivity or declivity detected on X and/or Y axis larger than threshold? If yes got to step 4 else go to step 5.
4. Is the amplitude of the acclivity or declivity detected on X and/or Y axis not twice smaller than the amplitude of the previous acclivity or declivity in the respective axis? If yes go to step 5 else go to step 5.
5. Record the activity (acclivity, declivity or no activity) occurred in X and Y axis.
6. Cross compare the amplitude of the acclivity or declivity occurred in X and Y axis. Choose the axis showing the highest amplitude to decide the swing motion.

Median filtering is used to reduce the noise generated by the accelerometer and also to remove noise due to slight unconscious hand movements. Acclivity or declivity generated on an axis means a swinging motion has occurred. Intra axis amplitude comparison in step 4 of the pseudo code is done to reduce false triggers. Finally a cross axis amplitude comparison of the acclivity or declivity generated is performed in order to negate the effects caused by rotation along the Z reference axis (μ is not equal to 0).

4. EXPERIMENTATION RESULTS

The recognition accuracy of the proposed swinging motion detection algorithm was tested on an embedded processor system consisting of a microcontroller (Texas Instrument CC2510 [16]) powered by a 7.7V Lithium-ion battery that is voltage regulated to 3.3V operating voltage. The embedded processor system was used to build a wireless portable interactive device (PID) featuring various input and output modalities. The input peripherals include a 3-axis accelerometer (MMA7260 from Freescale) and a serial-accessed micro-SD memory card. The output peripherals include six tri-color LEDs and a mini vibration speaker. The PID activates these peripherals when required by enabling the appropriate CC2510 component (USART, DSM, ADC). The hardware components of the PID were fitted to the cuboid acrylic housing section (see Figure 2) of the electronic bell.

Four users tested the swing motion recognition accuracy when μ is 0, π/4, π/2, 3π/4, π, 5π/4, 3π/2 and 7π/4. The PID was programmed to generate a sound and light up the LEDs when swing motion performed. According to the direction of the swinging motion (relative to the previous swing

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase® Smarter legal research.