Filterbank-Based Fingerprint Matching

Anil K. Jain, *Fellow, IEEE*, Salil Prabhakar, Lin Hong, and Sharath Pankanti

***Abstract*—With identity fraud in our society reaching unprecedented proportions and with an increasing emphasis on the emerging automatic personal identification applications, biometrics-based verification, especially fingerprint-based identification, is receiving a lot of attention. There are two major shortcomings of the traditional approaches to fingerprint representation. For a considerable fraction of population, the representations based on explicit detection of complete ridge structures in the fingerprint are difficult to extract automatically. The widely used minutiae-based representation does not utilize a significant component of the rich discriminatory information available in the fingerprints. Local ridge structures cannot be completely characterized by minutiae. Further, minutiae-based matching has difficulty in quickly matching two fingerprint images containing different number of unregistered minutiae points. The proposed filter-based algorithm uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode. The fingerprint matching is based on the Euclidean distance between the two corresponding FingerCodes and hence is extremely fast. We are able to achieve a verification accuracy which is only marginally inferior to the best results of minutiae-based algorithms published in the open literature [1]. Our system performs better than a state-of-the-art minutiae-based system when the performance requirement of the application system does not demand a very low false acceptance rate. Finally, we show that the matching performance can be improved by combining the decisions of the matchers based on complementary (minutiae-based and filter-based) fingerprint information.**

***Index Terms*—Biometrics, FingerCode, fingerprints, flow pattern, Gabor filters, matching, texture, verification.**

I. Introduction

WITH THE advent of electronic banking, e-commerce, and smartcards and an increased emphasis on the privacy and security of information stored in various databases, *automatic* personal identification has become a very important topic. Accurate automatic personal identification is now needed in a wide range of civilian applications involving the use of passports, cellular telephones, automatic teller machines, and driver licenses. Traditional knowledge-based [password or personal identification number (PIN)] and token-based (passport, driver license, and ID card) identifications are prone to fraud because PIN's may be forgotten or guessed by an imposter and the tokens may be lost or stolen. As an example, Mastercard credit card fraud alone now amounts to more than 450 million U.S. dollars annually [2]. *Biometrics*, which refers to identifying an individual based on his or her physiological or behavioral characteristics has the capability to reliably distinguish between an authorized person and an imposter.

Manuscript received May 11, 1999; revised November 16, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Alan C. Bovik.

A. K. Jain and S. Prabhakar are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: jain@pilot.msu.edu).

L. Hong is with Visionics Corporation, Jersey City, NJ 07302 USA.

S. Pankanti is with the Exploratory Computer Vision and Intelligent Robotics Group, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

Publisher Item Identifier S 1057-7149(00)03864-1.

Fig. 1. Ridges and automatically detected minutiae points in a fingerprint image. The core is marked with a ×.

A biometric system can be operated in two modes: 1) verification mode and 2) identification mode. A biometric system operating in the verification mode either accepts or rejects a user's claimed identity while a biometric system operating in the identification mode establishes the identity of the user without a claimed identity information. In this work, we have focused only on a biometric system operating in the verification mode.

Among all the biometrics (e.g., face, fingerprints, hand geometry, iris, retina, signature, voice print, facial thermogram, hand vein, gait, ear, odor, keystroke dynamics, etc. [2]), fingerprint-based identification is one of the most mature and proven technique.

A fingerprint is the pattern of ridges and valleys on the surface of the finger [3]. The uniqueness of a fingerprint can be determined by the overall pattern of ridges and valleys as well as the local ridge anomalies [a ridge bifurcation or a ridge ending, called minutiae points (see Fig. 1)]. Although the fingerprints possess the discriminatory information, designing a reliable automatic fingerprint matching algorithm is very challenging (see Fig. 2). As fingerprint sensors are becoming smaller and cheaper [4], automatic identification based on fingerprints is becoming an attractive alternative/complement to the traditional methods of identification. The critical factor in the widespread use of fingerprints is in satisfying the performance (e.g., matching speed and accuracy) requirements of the emerging civilian identification applications. Some of these applications (e.g., fingerprint-based smartcards) will also benefit from a compact representation of a fingerprint

Fig. 2. Difficulty in fingerprint matching. (a) and (b) have the same global configuration but are images of two different fingers.

The popular fingerprint representation schemes have evolved from an intuitive system design tailored for fingerprint experts who visually match the fingerprints. These schemes are either based on predominantly local landmarks (e.g., minutiae-based fingerprint matching systems [1], [5]) or exclusively global information (fingerprint classification based on the Henry system [6]–[8]). The minutiae-based automatic identification techniques first locate the minutiae points and then match their relative placement in a given finger and the stored template [1]. A good quality fingerprint contains between 60 and 80 minutiae, but different fingerprints have different number of minutiae. The variable sized minutiae-based representation does not easily lend itself to indexing mechanisms. Further, typical graph-based [9]–[11], and point pattern-based [1], [12], [13] approaches to match minutiae from two fingerprints need to align the unregistered minutiae patterns of different sizes which makes them computationally expensive. Correlation-based techniques [14], [15] match the global patterns of ridges and valleys to determine if the ridges align. The global approach to fingerprint representation is typically used for indexing [6]–[8], and does not offer very good individual discrimination. Further, the indexing efficacy of existing global representations is poor due to a small number of categories that can be effectively identified and a highly skewed distribution of the population in each category. The natural proportion of fingerprints belonging to categories whorl (whorl and double loop put together), loop (right and left loop put together), and arch (arch and tented arch put together), is 0.279, 0.655, and 0.066, respectively. Both these approaches utilize representations which cannot be easily extracted from poor quality fingerprints.

The smooth flow pattern of ridges and valleys in a fingerprint can be viewed as an oriented texture field [16]. The image intensity surface in an ideal fingerprint image is comprised of ridges whose direction and height vary continuously, which constitutes an oriented texture. Most textured images contain a limited range of spatial frequencies, and mutually distinct textures differ significantly in their dominant frequencies [17]–[19]. Textured regions possessing different spatial frequency, orientation, or fingerprint images scanned at 500 dpi, there is a little variation in the spatial frequencies (inter-ridge distances) among different fingerprints. This implies that there is an optimal scale (spatial frequency) for analyzing the fingerprint texture. Every point in a fingerprint image is associated with a dominant local orientation and a local measure of coherence of the flow pattern. A symbolic description of a fingerprint image can be derived by computing the angle and coherence at each point in the image. Fingerprints can be identified by using quantitative measures associated with the flow pattern (oriented texture) as features.

It is desirable to explore representation schemes which combine global and local information in a fingerprint. We present a new representation for the fingerprints which yields a relatively short, fixed length code, called *FingerCode* [6] suitable for matching as well as storage on a smartcard. The matching reduces to finding the Euclidean distance between these FingerCodes and hence the matching is very fast and the representation is amenable to indexing. We utilize both the global flow of ridge and valley structures and the local ridge characteristics to generate a short fixed length code for the fingerprints while maintaining a high recognition accuracy.

The proposed scheme of feature extraction tessellates the region of interest of the given fingerprint image with respect to a reference point (Fig. 3). A feature vector is composed of an ordered enumeration of the features extracted from the (local) information contained in each subimage (sector) specified by the tessellation. Thus, the feature elements capture the local information and the ordered enumeration of the tessellation captures the invariant global relationships among the local patterns. The local discriminatory information in each sector needs to be decomposed into separate components. Gabor filterbanks are a well-known technique to capture useful information in specific bandpass channels as well as to decompose this information into biorthogonal components in terms of spatial frequencies. A feature vector, which we call FingerCode, is the collection of all the features (for every sector) in each filtered image. These features capture both the global pattern of ridges and valleys and the local characteristics. Matching is based on the Euclidean distance between the FingerCodes.

II. Filter-Based Feature Extraction

It is desirable to obtain representations for fingerprints which are scale, translation, and rotation invariant. Scale invariance is not a significant problem since most fingerprint images could be scaled as per the dpi specification of the sensors. The rotation and translation invariance could be accomplished by establishing a reference frame based on the intrinsic fingerprint characteristics which are rotation and translation invariant. It is also possible to establish many frames of reference based upon several landmark structures in a fingerprint to obtain multiple representations. At the expense of additional processing and storage cost, the multiple representations offer robust matching performance when extraction algorithm fails to detect one or more frames of reference. In the proposed feature extraction scheme, translation is handled by a single reference point location during

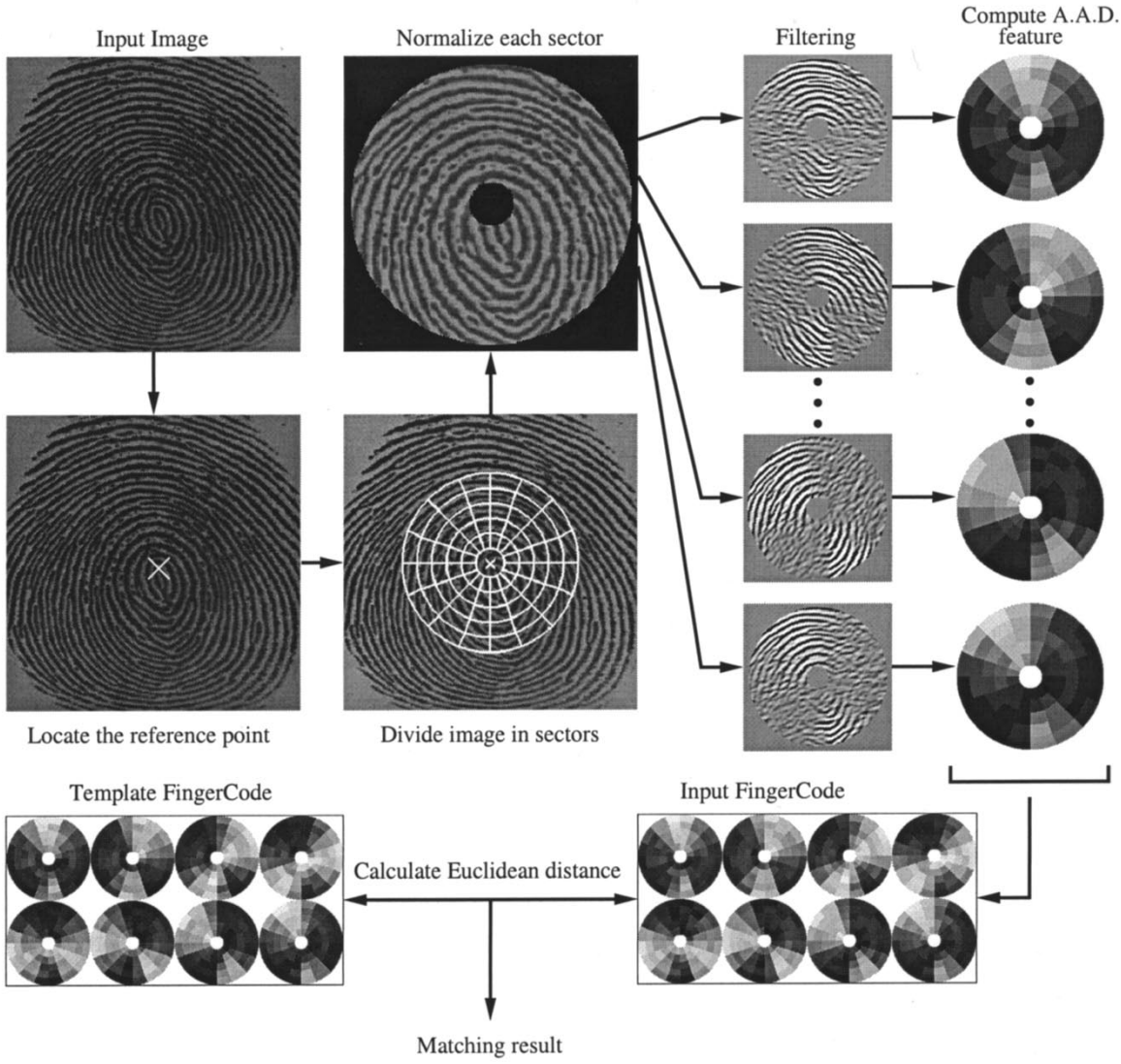

Fig. 3. System diagram of our fingerprint authentication system.

ented. In reality, the fingerprints in our database are not exactly vertically oriented; the fingerprints may be oriented up to $\pm 45^{\circ}$ away from the assumed vertical orientation. This image rotation is partially handled by a cyclic rotation of the feature values in the FingerCode in the matching stage; in future implementations, the image rotation will be correctly handled by automatically determining the fingerprint orientation from the image data. The current scheme of feature extraction tessellates the region of interest in the given fingerprint image with respect to the point of reference. The four main steps in our feature extraction algorithm are

1) determine a reference point and region of interest for the fingerprint image;
2) tessellate the region of interest around the reference point;
3) filter the region of interest in eight different directions using a bank of Gabor filters (eight directions are required to completely capture the local ridge characteristics in
4) compute the average absolute deviation from the mean (AAD) of gray values in individual sectors in filtered images to define the feature vector or the FingerCode.

In the current implementation, we have used the AAD features which give slightly better performance than variance features [6] on both the MSU_DBI and NIST 9 databases. Although AAD features perform reasonably well, we believe that a significantly better performance can be achieved by using more discriminative features.

Let $I(x, y)$ denote the gray level at pixel (x, y) in an $M \times N$ fingerprint image and let (x_c, y_c) denote the reference point. The *region of interest* is defined as the collection of all the sectors S_i, where the ith sector S_i is computed in terms of parameters (r, θ) as follows:

Fig. 4. Reference point (×), the region of interest, and 80 sectors superimposed on a fingerprint.

where

$$T_i = i \; div \; k \tag{2}$$

$$\theta_i = (i \mod k) \times (2\pi/k) \tag{3}$$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{4}$$

$$\theta = \tan^{-1}((y - y_c)/(x - x_c)) \tag{5}$$

b is the width of each band, k is the number of sectors considered in each band, and $i = 0 \cdots (B \times k - 1)$, where B is the number of concentric bands considered around the reference point for feature extraction. These parameters depends upon the image resolution and size. In our first experiment with MSU_DBI database (image size = 508×480 pixels, scanned at 500 dpi), we considered five concentric bands ($B = 5$) for feature extraction. Each band is 20-pixels wide ($b = 20$), and segmented into sixteen sectors ($k = 16$) (Fig. 4). A 20-pixel wide band captures an area spanning about one ridge and valley pair, on an average, in a 500 dpi fingerprint image. A band with a width of 20 pixels is necessary to capture a single minutia in a sector, allowing our low-level features to capture this local information. If the sector width is more than 20 pixels, then the local information may be modulated by more global information. The innermost band (circle) is not used for feature extraction because the flow field in a region around a very high curvature point (core) has poor coherence. Thus, absolute deviations of oriented Gabor responses to this region would be expected to be unreliable matching features. Thus, we have a total of $16 \times 5 = 80$ sectors (S_0 through S_{79}) and the region of interest is a circle of radius 120 pixels, centered at the reference point. Eighty features for each of the eight filtered images provide a total of 640 (80×8) features per fingerprint image. Each feature can be quantized into 256 values and requires 1 byte of storage, so the entire feature vector requires only 640 bytes of storage. In our second experiment with NIST 9 database (image size = 832×768 pixels, scanned at

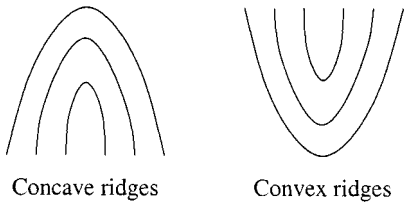

Fig. 5. Concave and convex ridges in a fingerprint image when the finger is positioned upright.

It is difficult to rely on feature extraction based on explicit detection of structural features in fingerprints, especially in poor quality images. Features based on statistical properties of images are likely to degrade gracefully with the image quality deterioration. For this study, we use grayscale variance-based features. The average absolute deviation of the gray levels from the mean value in an image sector is indicative of the overall ridge activity in that sector which we claim to be useful for fingerprint verification. As noted in Section IV, our matcher based on this simple statistical feature performs well and we expect to achieve significantly better accuracies with more discriminative attributes.

A. Reference Point Location

Fingerprints have many conspicuous landmark structures and a combination of them could be used for establishing a reference point. We define the reference point of a fingerprint as the point of maximum curvature of the concave ridges (see Fig. 5) in the fingerprint image.

Many previous approaches to determination of a reference point (x_c, y_c) critically relied on the local features like Poincaré index or some other similar properties of the orientation field. While these methods work well in good quality fingerprint images, they fail to correctly localize reference points in poor quality fingerprints with cracks and scars, dry skin, or poor ridge and valley contrast. Recently, Hong and Jain have attempted to judiciously combine the orientation field information with available ridge details in a fingerprint [8]. However, this method does not reliably handle poor quality fingerprints when the orientation field is very noisy and can be misled by poor structural cues in the presence of finger cracks.

In order that a reference point algorithm gracefully handle local noise in a poor quality fingerprint, the detection should necessarily consider a large neighborhood in the fingerprint. On the other hand, for an accurate localization of the reference point, the approach should be sensitive to the local variations in a small neighborhood. To meet these conflicting requirements of an accurate and reliable localization, we propose a new method of reference point determination based on multiple resolution analysis of the orientation fields. Our new method locates the reference point more precisely than the algorithm proposed by Hong and Jain [8].

Let us first define the *orientation field*, $\mathcal{O}$, for a fingerprint image. The orientation field, $\mathcal{O}$, is defined as a $P \times Q$ image, where $\mathcal{O}(i, j)$ represents the local ridge orientation at pixel (i, j). Local ridge orientation is usually specified for a block

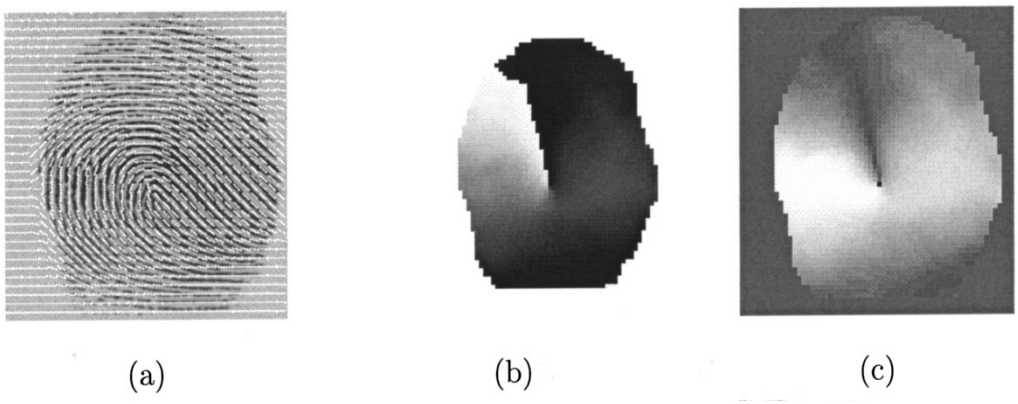

Fig. 6. Estimating the reference point. (a) Smoothed orientation field overlapped on the original image. (b) Orientation field ($w = 10$) shown as intensity distribution; the background has been segmented. (c) *sine* component of the orientation field; the darkest pixel marks the detected reference point. Images have been scaled for viewing.

each block [see Fig. 6(a) and (b)]. Note that there is an ambiguity of π in fingerprint orientation, i.e., local ridges oriented at $\pi/2$ and ridges oriented at $3\pi/2$ cannot be differentiated from each other. A number of methods have been developed to estimate the orientation field in a fingerprint [20]–[23]. The least mean square orientation estimation algorithm [33] has the following steps.

1) Divide $\mathcal{I}$, the input image, into nonoverlapping blocks of size $w \times w$.
2) Compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel (i, j). Depending on the computational requirement, the gradient operator may vary from the simple *Sobel* operator to the more complex *Marr–Hildreth* operator [24].
3) Estimate the local orientation of each block centered at pixel (i, j) using the following equations [23]:

$$\mathcal{V}_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\partial_x(u, v)\partial_y(u, v) \tag{6}$$

$$\mathcal{V}_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (\partial_x^2(u, v) - \partial_y^2(u, v)) \tag{7}$$

$$\mathcal{O}(i, j) = \frac{1}{2} \tan^{-1}\left(\frac{\mathcal{V}_y(i, j)}{\mathcal{V}_x(i, j)}\right) \tag{8}$$

where $\mathcal{O}(i, j)$ is the least square estimate of the local ridge orientation at the block centered at pixel (i, j). Mathematically, it represents the direction that is orthogonal to the dominant direction of the *Fourier spectrum* of the $w \times w$ window.

A summary of our reference point location algorithm is presented below.

1) Estimate the orientation field $\mathcal{O}$ as described above using a window size of $w \times w$.
2) Smooth the orientation field in a local neighborhood. Let the smoothed orientation field be represented as $\mathcal{O}'$. In order to perform smoothing (low-pass filtering), the orientation image needs to be converted into a *continuous vector field*, which is defined as follows:

and

$$\Phi_y(i, j) = \sin(2\mathcal{O}(i, j)) \tag{10}$$

andwhere Φ_x and Φ_y, are the x and y components of the vector field, respectively. With the resulting vector field, the low-pass filtering can then be performed as follows:

$$\Phi'_x(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \Phi_x(i - uw, j - vw) \tag{11}$$

and

$$\Phi'_y(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \Phi_y(i - uw, j - vw) \tag{12}$$

where W is a two-dimensional low-pass filter with unit integral and $w_\Phi \times w_\Phi$ specifies the size of the filter. Note that the smoothing operation is performed at the block level. For our experiments, we used a 5×5 mean filter. The smoothed orientation field $\mathcal{O}'$ at (i, j) is computed as follows:

$$\mathcal{O}'(i, j) = \frac{1}{2} \tan^{-1}\left(\frac{\Phi'_y(i, j)}{\Phi'_x(i, j)}\right). \tag{13}$$

3) Compute $\mathcal{E}$, an image containing only the *sine* component of $\mathcal{O}'$ [see Fig. 6(c)]

$$\mathcal{E}(i, j) = \sin(\mathcal{O}'(i, j)). \tag{14}$$

4) Initialize $\mathcal{A}$, a label image used to indicate the reference point.
5) For each pixel (i, j) in $\mathcal{E}$, integrate pixel intensities (sine component of the orientation field) in regions R_I and R_{II} shown in Fig. 7 and assign the corresponding pixels in $\mathcal{A}$ the value of their difference

$$\mathcal{A}(i, j) = \sum_{R_I} \mathcal{E}(i, j) - \sum_{R_{II}} \mathcal{E}(i, j). \tag{15}$$

The regions R_I and R_{II} (see Fig. 7) were determined empirically by applying the reference point location algo-

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.