

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

### Provisional Application for Patent Cover Sheet

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

#### Inventor(s)

Inventor 1 Remove

Given Name	Middle Name	Family Name	City	State	Country ;
Grigori		Nepomniachtchi	San Diego	CA	US

Inventor 2 Remove

Given Name	Middle Name	Family Name	City	State	Country ;
James		DeBello	San Diego	CA	US

Inventor 3 Remove

Given Name	Middle Name	Family Name	City	State	Country ;
Josh		Roach	San Diego	CA	US

All Inventors Must Be Listed – Additional Inventor Information blocks may be generated within this form by selecting the **Add** button.

**Title of Invention** SYSTEMS FOR MOBILE IMAGE CAPTURE AND REMITTANCE PROCESSING

Attorney Docket Number (if applicable) 117313-011PRV

#### Correspondence Address

Direct all correspondence to (select one):

<input checked="" type="radio"/> The address corresponding to Customer Number	<input type="radio"/> Firm or Individual Name
Customer Number	27189

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

<input checked="" type="radio"/> No.
<input type="radio"/> Yes, the name of the U.S. Government agency and the Government contract number are:

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

**Entity Status**

Applicant claims small entity status under 37 CFR 1.27

- Yes, applicant qualifies for small entity status under 37 CFR 1.27  
 No

**Warning**

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

**Signature**

Please see 37 CFR 1.4(d) for the form of the signature.

Signature	/Noel C. Gillespie/			Date (YYYY-MM-DD)	2011-11-18
First Name	Noel C.	Last Name	Gillespie	Registration Number (If appropriate)	47596

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. This form can only be used when in conjunction with EFS-Web. If this form is mailed to the USPTO, it may cause delays in handling the provisional application.**

## Privacy Act Statement

**The Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that : (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to a n other federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

## **SPECIFICATION**

### **SYSTEMS FOR MOBILE IMAGE CAPTURE AND REMITTANCE PROCESSING**

#### **RELATED APPLICATIONS INFORMATION**

**[0001]** This application is a continuation in part of copending U.S. Patent Application Serial No. 12/906,036 filed on October 15, 2010, which itself is a continuation in part of copending U.S. Patent Application Serial No. 12/778,943 filed on May 12, 2010 as well as a continuation in part of U.S. Patent application Serial Number 12/346,026 filed December 30, 2008, now U.S. Patent No. 7,978,900, which in turn claims the benefit of U.S. Provisional application Serial Number 61/022,279 filed January 18, 2008, all of which are incorporated herein by reference in their entirety as if set forth in full. This application is also related to U.S. Patent Application Serial No. 12/717,080 filed March 3, 2010, which is now U.S. Patent No. 7,778,457, which is incorporated herein by reference in its entirety as if set forth in full.

#### **BACKGROUND**

##### **1. Technical Field**

**[0002]** The embodiments described herein generally relate to automated document processing and more particularly, to systems and methods for document image processing that enhances an image for data extraction from images captured on a mobile device with camera capabilities.

##### **2. Related Art**

**[0003]** Remittance processing services provide payment processing services for other businesses. A remittance processing service may be set up by a bank. Often these services set up one or more post office boxes for receiving payments from customers in the form of paper checks. Typically the user will include a remittance slip or remittance coupon with the payment.

**[0004]** Banks and other businesses often provide remittance slips or coupons that a customer can include with a payment. These coupons may be included with an invoice or provided to a

customer in advance. For example, some mortgage lenders or automobile lenders provide a book of remittance coupons to customers who then tear out a mail a remittance coupon with each scheduled payment. The remittance coupons generally include customer account information, an amount due, and a due date for the payment. The customer account information might include an account holder name, mailing address, and a customer account number. Other information, such as the mailing address of the bank or business may also be included on the remittance coupon. Some remittance coupons also include computer-readable bar codes or code lines that include text or other computer-readable symbols that can be used to encode account-related information that can be used to reconcile a payment received with the account for which the payment is being made. The bar code or code line can be detected and decoded by a computer system to extract the information encoded therein.

**[0005]** In general, the remittance coupon is mailed with a check or other form of payment to the bank or other business or to a payment processing company designated by the bank or other business. In some systems, data entry clerks may manually enter payment information into a payment processing system. In other systems, the remittance coupon and/or the check may be scanned by a computer system for automated processing.

## SUMMARY

**[0006]** Systems and method are provided for capturing and processing images of remittance coupons using a mobile device, such as a mobile phone. Embodiments of the systems and methods described herein provide image optimization and enhancement such that data can be extracted from the remittance coupon for processing by a remittance processing service. In some embodiments, an image of a check can also be captured to be processed as a payment associated with the remittance coupon. Some embodiments described herein involve a mobile communication device capturing an image of a document and transmitting the captured image to a server for image optimization and enhancement. Techniques for assessing the quality of images of documents captured using the mobile device are also provided. The tests can be selected based on the type of document that was imaged, the type of mobile application for which the image quality of the mobile image is being assessed, and/or other parameters such as

the type of mobile device and/or the characteristics of the camera of the mobile device that was used to capture the image. In some embodiments, the image quality assurance techniques can be implemented on a remote server, such as a mobile phone carrier's server or a web server, and the mobile device routes the mobile image to be assessed and optional processing parameters to the remote server processing and the test results can be passed from the remote server to the mobile device.

**[0007]** In one exemplary aspect, a computer-readable medium is disclosed. In one embodiment, the computer-readable medium comprises instructions which, when executed by a computer, perform a process comprising: receiving an image of a document captured by a mobile device; identifying a set of characteristics of the document in relation to said image; processing the image to correct one or more graphical defects, wherein said processing is adapted to generate a processed image based at least in part upon the set of characteristics; executing one or more mobile image quality assurance tests on the processed image to assess the quality of the processed image; and extracting data from the processed image.

**[0008]** In another exemplary aspect, an apparatus for processing an image captured from a mobile device is disclosed. In one embodiment, the apparatus comprises: a receiving module adapted to receive an image of a document captured by a mobile device; an identification module in electrical communication with said receiving module, the identification module adapted to identify a set of characteristics of the document in relation to said image; a processing module in electrical communication with said identification module, the processing module adapted to process the image to correct one or more graphical defects present within the image, wherein said processing is based at least in part upon said set of characteristics; a testing module in electrical communication with the processing module, the testing module adapted to execute one or more mobile image quality assurance tests on a processed image in order to assess the quality of the processed image; and a data extraction module in electrical communication with the testing module, the data extraction module adapted to extract data from the processed image.

[0009] Other features and advantages of the present invention should become apparent from the following description of the preferred embodiments, taken in conjunction with the accompanying drawings, which illustrate, by way of example, the principles of the invention.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0010] The various embodiments provided herein are described in detail with reference to the following figures. The drawings are provided for purposes of illustration only and merely depict typical or example embodiments. These drawings are provided to facilitate the reader's understanding of the invention and shall not be considered limiting of the breadth, scope, or applicability of the embodiments. It should be noted that for clarity and ease of illustration these drawings are not necessarily made to scale.

[0011] FIG. 1 is an image of a remittance coupon 100 illustrating an example remittance coupon that might be imaged with the systems and methods described herein.

[0012] FIG. 2 is a geometrically corrected image created using image processing techniques disclosed herein using the mobile image of the remittance coupon 100 illustrated in FIG. 1.

[0013] FIG. 3 is high level block diagram of a system that can be used to implement the systems and methods described herein.

[0014] FIG. 4 is a flow diagram of an example method for capturing an image of a remittance coupon using a mobile device according to an embodiment.

[0015] FIG. 5 is a flow diagram of an example method for processing an image of a remittance coupon captured using a mobile device according to an embodiment.

[0016] FIG. 6 is a flow diagram illustrating a method for correcting defects to mobile image according to an embodiment.

[0017] FIG. 7 and its related description above provide some examples of how a perspective transformation can be constructed for a quadrangle defined by the corners A, B, C, and D according to an embodiment.

[0018] FIG. 8 is a diagram illustrating an example original image, focus rectangle and document quadrangle ABCD in accordance with the example of FIG. 7.

[0019] FIG. 9 is a flow chart for a method that can be used to identify the corners of the remittance coupon in a color image according to an embodiment.

[0020] FIG. 10 is a flow diagram of a method for generating a bi-tonal image according to an embodiment.

[0021] FIG. 11 illustrates an example image binarized image of a remittance coupon generated from the remittance geometrically corrected remittance coupon image illustrated in FIG. 2.

[0022] FIG. 12 is a flow diagram of a method for converting a document image into a smaller color icon image according to an embodiment.

[0023] FIG. 13A is a mobile image of a check according to an embodiment.

[0024] FIG 13B is an example of a color icon image generated using the method of FIG. 12 on the example mobile image of a check illustrated in FIG. 13A according to an embodiment.

[0025] FIG. 14 is a flow diagram of a method for reducing the color depth of an image according to an embodiment.

[0026] FIG. 15A depicts an example of the color “icon” image of FIG. 13B after operation 1302 has divided it into a 3x3 grid in accordance with one embodiment of the invention.

[0027] FIG. 15B depicts an example of the color “icon” image of FIG. 13B converted to a gray “icon” image using the method illustrated in FIG. 14 according to an embodiment.

[0028] FIG. 16 is a flowchart illustrating an example method for finding document corners from a gray “icon” image containing a document according to an embodiment.

[0029] FIG. 17 is a flowchart that illustrates an example method for geometric correction according to an embodiment.



[0030] FIG. 18A is an image illustrating a mobile image of a check that is oriented in landscape orientation according to an embodiment.

[0031] FIG. 18B example gray-scale image of the document depicted in FIG. 13A once a geometrical correction operation has been applied to the image according to an embodiment.

[0032] FIG. 19 is a flow chart illustrating a method for correcting landscape orientation of a document image according to an embodiment.

[0033] FIG. 20 provides a flowchart illustrating an example method for size correction of an image according to an embodiment.

[0034] FIG. 21 illustrates a mobile document image processing engine (MDIPE) module 2100 for performing quality assurance testing on mobile document images according to an embodiment.

[0035] FIG. 22 is a flow diagram of a process for performing mobile image quality assurance on an image captured by a mobile device according to an embodiment.

[0036] FIG. 23 is a flow diagram of a process for performing mobile image quality assurance on an image of a check captured by a mobile device according to an embodiment.

[0037] FIG. 24A illustrates a mobile image where the document captured in the mobile document image exhibits view distortion.

[0038] FIG. 24B illustrates an example of a grayscale geometrically corrected subimage generated from the distorted image in FIG. 24A according to an embodiment.

[0039] FIG. 25A illustrates an example of an in-focus mobile document image.

[0040] FIG. 25B illustrates an example of an out of focus document.

[0041] FIG. 26 illustrates an example of a shadowed document.

[0042] FIG. 27 illustrates an example of a grayscale snippet generated from a mobile document image of a check where the contrast of the image is very low according to an embodiment.

[0043] FIG. 28 illustrates a method for executing a Contrast IQA Test according to an embodiment.

[0044] FIG. 29A is an example of a mobile document image that includes a check that exhibits significant planar skew according to an embodiment.

[0045] FIG. 29B illustrates an example of a document subimage that exhibits view skew according to an embodiment.

[0046] FIG. 30 is a flow chart illustrating a method for testing for view skew according to an embodiment.

[0047] FIG. 31 illustrates an example of a mobile document image that features an image of a document where one of the corners of the document has been cut off in the picture.

[0048] FIG. 32 illustrates a Cut-Off Corner Test that can be used for testing whether corners of a document in a document subimage have been cut off when the document was imaged according to an embodiment.

[0049] FIG. 33 illustrates an example of a mobile document image that features a document where one of the ends of the document has been cut off in the image.

[0050] FIG. 34 is a flow diagram of a method for determining whether one or more sides of the document are cut off in the document subimage according to an embodiment.

[0051] FIG. 35 illustrates an example of a mobile document image where the document is warped according to an embodiment.

[0052] FIG. 36 is a flow diagram of a method for identifying a warped image and for scoring the image based on how badly the document subimage is warped according to an embodiment.

**[0053]** FIG. 37 illustrates an example of a document subimage within a mobile document image that is relatively small in comparison to the overall size of the mobile document image according to an embodiment.

**[0054]** FIG. 38 is a flow diagram of a process that for performing an Image Size Test on a subimage according to an embodiment.

**[0055]** FIG. 39A is a flow chart of a method for executing a MICR-line Test according to an embodiment.

**[0056]** FIG. 39B is a flow chart of a method for executing a code line Test according to an embodiment.

**[0057]** FIG. 40 illustrates a method for executing an Aspect Ratio Test according to an embodiment.

**[0058]** FIG. 41 illustrates a method for performing a front-as-rear test on a mobile document image.

**[0059]** FIG 42 is a flow chart of a method for processing an image using form identification according to an embodiment.

**[0060]** FIG 43 is a flow chart of a method for processing an image using dynamic data capture according to an embodiment.

**[0061]** FIG. 44 is a block diagram of various functional elements of a mobile device that can be used with the various systems and methods described herein according to an embodiment.

**[0062]** FIG. 45 is a block diagram of functional elements of a computer system that can be used to implement the mobile device and/or the servers described in the systems and methods disclosed herein.

[0063] FIG. 46A-46F are images representing application that can reside on mobile device and be using in according with the systems and method disclosed herein according to an embodiment.

[0064] FIG. 47 is a flow diagram illustrating an exemplary method of configuring a recurring payment schedule according to an embodiment.

[0065] FIG. 48 is a flow diagram illustrating an exemplary method of selecting a specific scheduling preference according to an embodiment.

[0066] FIG. 49 is a flow diagram illustrating an exemplary method of enabling a user to set one or more reminders associated with a recurring bill payment according to an embodiment.

### DETAILED DESCRIPTION

[0067] The embodiments described herein are directed towards automated document processing and systems and methods for document image processing using mobile devices. System and methods are provided for processing a remittance coupon captured using a mobile device. Generally, in some embodiments an original color image of a document is captured using a mobile device and then converted the color image to a bi-tonal image. More specifically, in some embodiments, a color image of a document taken by a mobile device is received and converted into a bi-tonal image of the document that is substantially equivalent in its resolution, size, and quality to document images produced by “standard” scanners.

[0068] The term “standard scanners” as used herein, but is not limited to, transport scanners, flat-bed scanners, and specialized check-scanners. Some manufacturers of transport scanners include UNISYS<sup>®</sup>, BancTec<sup>®</sup>, IBM<sup>®</sup>, and Canon<sup>®</sup>. With respect to specialized check-scanners, some models include the TellerScan<sup>®</sup> TS200 and the Panini<sup>®</sup> My Vision X. Generally,

standard scanners have the ability to scan and produce high quality images, support resolutions from 200 dots per inch to 300 dots per inch (DPI), produce gray-scale and bi-tonal images, and crop an image of a check from a larger full-page size image. Standard scanners for other types of documents may have similar capabilities with even higher resolutions and higher color-depth.

[0069] The term “color images” as used herein, but is not limited to, images having a color depth of 24 bits per a pixel (24 bit/pixel), thereby providing each pixel with one of 16 million possible colors. Each color image is represented by pixels and the dimensions  $W$  (width in pixels) and  $H$  (height in pixels). An intensity function  $I$  maps each pixel in the  $[W \times H]$  area to its RGB-value. The RGB-value is a triple (R,G,B) that determines the color the pixel represents. Within the triple, each of the R(Red), G(Green) and B(Blue) values are integers between 0 and 255 that determine each respective color’s intensity for the pixel.

[0070] The term, “gray-scale images” as used herein, but is not limited to, images having a color depth of 8 bits per a pixel (8 bit/pixel), thereby providing each pixel with one of 256 shades of gray. As a person of ordinary skill in the art would appreciate, gray-scale images also include images with color depths of other various bit levels (e.g. 4 bit/pixel or 2 bit/pixel). Each gray-scale image is represented by pixels and the dimensions  $W$  (width in pixels) and  $H$  (height in pixels). An intensity function  $I$  maps each pixel in the  $[W \times H]$  area onto a range of gray shades. More specifically, each pixel has a value between 0 and 255 which determines that pixel’s shade of gray.

[0071] Bi-tonal images are similar to gray-scale images in that they are represented by pixels and the dimensions  $W$  (width in pixels) and  $H$  (height in pixels). However, each pixel within a bi-tonal image has one of two colors: black or white. Accordingly, a bi-tonal image has a color depth of 1 bit per a pixel (1 bit/pixel). The similarity transformation, as utilized by some embodiments of the invention, is based off the assumption that there are two images of  $[W \times H]$  and  $[W' \times H']$  dimensions, respectively, and that the dimensions are proportional (i.e.  $W / W' = H / H'$ ). The term “similarity transformation” may refer to a transformation  $ST$  from  $[W \times H]$  area onto  $[W' \times H']$  area such that  $ST$  maps pixel  $p = p(x,y)$  on pixel  $p' = p'(x',y')$  with  $x' = x * W' / W$  and  $y = y * H' / H$ .

**[0072]** The systems and methods provided herein advantageously allow a user to capture an image of a remittance coupon, and in some embodiments, a form of payment, such as a check, for automated processing. Typically, a remittance processing service will scan remittance coupons and checks using standard scanners that provide a clear image of the remittance coupon and accompanying check. Often these scanners produce either gray-scale and bi-tonal images that are then used to electronically process the payment. The systems and methods disclosed herein allow an image of remittance coupons, and in some embodiments, checks to be captured using a camera or other imaging device included in or coupled to a mobile device, such as a mobile phone. The systems and methods disclosed herein can test the quality of a mobile image of a document captured using a mobile device, correct some defects in the image, and convert the image to a format that can be processed by remittance processing service.

**[0073]** FIG. 1 is an image illustrating an example remittance coupon 1900 that can be imaged with the systems and methods described herein. The mobile image capture and processing systems and methods described herein can be used with a variety of documents, including financial documents such as personal checks, business checks, cashier's checks, certified checks, and warrants. By using an image of the remittance coupon 100, the remittance process can be automated and performed more efficiently. As would be appreciated by those of skill in the art, remittance coupons are not the only types of documents that might be processed using the system and methods described herein. For example, in some embodiments, a user can capture an image of a remittance coupon and an image of a check associated with a checking account from which the remittance payment will be funded.

**[0074]** FIG. 2 is a geometrically corrected image created using image processing techniques disclosed herein and using the mobile image of the remittance coupon 100 illustrated in FIG. 1. A remittance coupon may include various fields, and some fields in the documents might be considered "primary" fields. For example, some remittance coupons also include computer-readable bar codes or code lines 205 that include text or other computer-readable symbols that can be used to encode account-related information. The account-related information can be used to reconcile a payment received with the account for which the payment is being made. Code line 205 can be detected and decoded by a computer system to extract the information encoded

therein. The remittance coupon can also include an account number field 210 and an amount due field 215. Remittance coupons can also include other fields, such as the billing company name and address 220, a total outstanding balance, a minimum payment amount, a billing date, and payment due date. The examples are merely illustrative of the types of information that may be included on a remittance coupon and it will be understood that other types of information can be included on other types of remittance coupons.

**[0075]** FIG. 3 is high level block diagram of a system 300 that can be used to implement the systems and methods described herein. System 300 include a mobile device 340. The mobile device can comprise a mobile telephone handset, Personal Digital Assistant, or other mobile communication device. The mobile device can include a camera or other imaging device, such as a scanner, or might include functionality that allows it to connect to a camera or other imaging device. The connection to an external camera or other imaging device can comprise a wired or wireless connection. In this way the mobile device can connect to an external camera or other imaging device and receive images from the camera or other imaging device.

**[0076]** Images of the documents taken using the mobile device or downloaded to the mobile device can be transmitted to mobile remittance server 310 via network 330. Network 330 can comprise one or more wireless and/or wired network connections. For example, in some cases, the images can be transmitted over a mobile communication device network, such as a code division multiple access (“CDMA”) telephone network, or other mobile telephone network. Network 330 can also comprise one or more connections across the Internet. Images taken using, for example, a mobile device’s camera, can be 24 bit per pixel (24 bit/pixel) JPG images. It will be understood, however, that many other types of images might also be taken using different cameras, mobile devices, etc.

**[0077]** Mobile remittance server 310 can be configured to perform various image processing techniques on images of remittance coupons, checks, or other financial documents captured by the mobile device 340. Mobile remittance server 310 can also be configured to perform various image quality assurance tests on images of remittance coupons or financial documents captured by the mobile device 340 to ensure that the quality of the captured images is sufficient to enable

remittance processing to be performed using the images. Examples of various processing techniques and testing techniques that can be implemented on mobile remit server 210 are described in detail below.

**[0078]** Mobile remittance server 310 can also be configured to communicate with one or more remittance processor servers 315. According to an embodiment, the mobile remittance server 310 can perform processing and testing on images captured by mobile device 340 to prepare the images for processing by a third-party remittance processor and to ensure that the images are of a sufficient quality for the third-party remittance processor to process. The mobile remittance server 310 can send the processed images to the remittance processor 315 via the network 330. In some embodiments, the mobile remittance server 310 can send additional processing parameters and data to the remittance processor 315 with the processed mobile image. This information can include information collected from a user by the mobile device 340. According to an embodiment, the mobile remittance server 310 can be implemented using hardware or a combination of software and hardware. FIG. 45, describe below, illustrates a computer system that can be used to implement mobile remittance server 310 according to an embodiment.

**[0079]** According to an embodiment, the mobile remittance server 310 can be configured to communicate to one or more bank server 320 via the network 330. Bank server 320 can be configured to process payments in some embodiments. For example, in some embodiments, mobile device 340 can be used to capture an image of a remittance coupon and an image of a check that can be used to make an electronic payment of the remittance payment. For example, the remittance processor server 315 can be configured to receive an image of a remittance coupon and an image of a check from the mobile remittance server 310. The remittance processor 315 can electronically deposit the check into a bank account associated with the entity for which the electronic remittance is being performed. According to some embodiments, the bank server 320 and the remittance processor 315 can be implemented on the same server or same set of servers.



**[0080]** In other embodiments, the remittance processor 315 can handle payment. For example, the remittance processor can be operate by or on behalf of an entity associated with the coupon of FIG. 1, such as a utility or business. The user's account can then be linked with a bank, Paypal, or other account, such that when remittance processor 315 receives the remittance information, it can charge the appropriate amount to the user's account. FIGS. 46A-F illustrates example screens of an application that can run on a mobile device 340 and can be used for mobile remittance as described herein.

**[0081]** FIG. 46A is an image of an icon representing application that can reside on mobile device 340. When the application is activated, a login screen as illustrated in FIG. 46B can be displayed into which the user can input certain credentials such as a login or username and password. Once these credentials are validated, the user can be presented a screen as illustrated in FIG. 46C allowing them to pay certain bills using the application. In certain embodiments, the user can also view other information such as previous or last payments.

**[0082]** When the user elects to pay a bill, the camera application can be launched as illustrated in FIG. 46D, which can enable the user to obtain an image of the coupon. As illustrated, control buttons can be presented to allow the user to accept the image or cancel and possibly obtain a new image. Data can then be extracted from the image, as discussed in detail below, and as illustrated in FIG. 46E. The data can be presented as various fields and can enable the user to correct or alter the information as appropriate. In certain embodiments, the user can even access a grayscale image of the coupon illustrated in FIG. 46F.

**[0083]** Once the image is captured and corrected, and the data is extracted and adjusted, then the image, data, and any required credential information, such as username, password, and phone or device identifier, can be transmitted to the mobile remittance server 310 for further processing. This further processing is described in detail with respect to the remaining figure sin the description below.

**[0084]** First, FIG. 4 is a flow diagram of an example method for capturing an image of a remittance coupon using a mobile device according to an embodiment. According to an embodiment, the method illustrated in FIG. 4 can be implemented in mobile device 340.

**[0085]** An image of a remittance coupon is captured using a camera or other optical device of the mobile device 340 (step 405). For example, a user of the mobile device 340 can click a button or otherwise activate a camera or other optical device of mobile device 340 to cause the camera or other optical device to capture an image of a remittance coupon. FIG. 1 illustrates an example of a mobile image of a remittance coupon that has been captured using a camera associated with a mobile device 340.

**[0086]** According to an embodiment, the mobile device 340 can also be configured to optionally receive additional information from the user (step 410). For example, in some embodiments, the mobile device can be configured to prompt the user to enter data, such as a payment amount that represents an amount of the payment that the user wishes to make. The payment amount can differ from the account balance or minimum payment amount shown on the remittance coupon. For example, the remittance coupon might show an account balance of \$1000 and a minimum payment amount of \$100, but the user might enter a payment amount of \$400.

**[0087]** According to an embodiment, the mobile device 340 can be configured to perform some preprocessing on the mobile image (step 415). For example, the mobile device 340 can be configured to convert the mobile image from a color image to a grayscale image or to bitonal image. Other preprocessing steps can also be performed on the mobile device. For example, the mobile device can be configured to identify the corners of the remittance coupon and to perform geometric corrections and/or warping corrections to correct defects in the mobile image. Examples of various types of preprocessing that can be performed on the mobile device 340 are described in detail below.

**[0088]** Mobile device 340 can then transmit the mobile image of the remittance coupon and any additional data provided by user to mobile remittance server 310.

**[0089]** FIG. 5 is a flow diagram of an example method for processing an image of a remittance coupon captured using a mobile device according to an embodiment. According to an embodiment, the method illustrated in FIG. 4 can be implemented in mobile remittance server 310.

**[0090]** Mobile remittance server 310 can receive the mobile image and any data provided by the user from the mobile device 340 via the network 330 (step 505). The mobile remittance server 310 can then perform various processing on the image to prepare the image for image quality assurance testing and for submission to a remittance processor 315 (step 510). Various processing steps can be performed by the mobile remittance server 310. Examples of the types of processing that can be performed by mobile remittance server 310 are described in detail below.

**[0091]** Mobile remittance server 310 can perform image quality assurance testing on the mobile image to determine whether there are any issues with the quality of the mobile image that might prevent the remittance provider from being able to process the image of the remittance coupon (step 515). Various mobile quality assurance testing techniques that can be performed by mobile remittance server 310 are described in detail below.

**[0092]** According to an embodiment, mobile remittance server 310 can be configured to report the results of the image quality assurance testing to the mobile device 340 (step 520). This can be useful for informing a user of mobile device 340 that an image that the user captured of a remittance coupon passed quality assurance testing, and thus, should be of sufficient quality that the mobile image can be processed by a remittance processor server 315. According to an embodiment, the mobile remittance server 310 can be configured to provide detailed feedback messages to the mobile device 340 if a mobile image fails quality assurance testing. Mobile device 340 can be configured to display this feedback information to a user of the device to inform the user what problems were found with the mobile image of the remittance coupon and to provide the user with the opportunity to retake the image in an attempt to correct the problems identified.

**[0093]** If the mobile image passes the image quality assurance testing, the mobile remittance server 310 can submit the mobile image plus any processing parameters received from the mobile device 340 to the remittance processor server 315 for processing (step 525). According to an embodiment, mobile remittance server 310 can include a remittance processing server configured to perform the steps 525 including the methods illustrated in FIGs. 42 and 43.

According to an embodiment, the remittance processing step 525 can include identifying a type of remittance coupon found in a mobile image. According to some embodiments, coupon templates can be used to improve data capture accuracy. According to an embodiment, if form identification fails and a coupon template that matches the format of a coupon, a dynamic data capture method can be applied to extract information from the remittance coupon.

### **Image Processing**

**[0094]** Mobile device 340 and mobile remittance server 310 can be configured to perform various processing on a mobile image to correct various defects in the image quality that could prevent the remittance processor 215 from being able to process the remittance due to poor image quality.

**[0095]** For example, an out of focus image of a remittance coupon or check, in embodiments where the mobile device can also be used to capture check images for payment processing, can be impossible to read and electronically process. For example, optical character recognition of the contents of the imaged document based on a blurry mobile image could result in incorrect payment information being extracted from the document. As a result, the wrong account could be credited for the payment or an incorrect payment amount could be credited. This may be especially true if a check and a payment coupon are both difficult to read or the scan quality is poor.

**[0096]** Many different factors may affect the quality of an image and the ability of a mobile device based image capture and processing system. Optical defects, such as out-of-focus images (as discussed above), unequal contrast or brightness, or other optical defects, can make it difficult to process an image of a document, e.g., a check, payment coupon, deposit slip, etc. The quality of an image can also be affected by the document position on a surface when photographed or the angle at which the document was photographed. This affects the image quality by causing the document to appear, for example, right side up, upside down, skewed, etc. Further, if a document is imaged while upside-down it might be impossible or nearly impossible for the system to determine the information contained on the document.

**[0097]** In some cases, the type of surface might affect the final image. For example, if a document is sitting on a rough surface when an image is taken, that rough surface might show through. In some cases the surface of the document might be rough because of the surface below it. Additionally, the rough surface may cause shadows or other problems that might be picked up by the camera. These problems might make it difficult or impossible to read the information contained on the document.

**[0098]** Lighting may also affect the quality of an image, for example, the location of a light source and light source distortions. Using a light source above a document can light the document in a way that improves the image quality, while a light source to the side of the document might produce an image that is more difficult to process. Lighting from the side can, for example, cause shadows or other lighting distortions. The type of light might also be a factor, for example, sun, electric bulb, florescent lighting, etc. If the lighting is too bright, the document can be washed out in the image. On the other hand, if the lighting is too dark, it might be difficult to read the image.

**[0099]** The quality of the image can also be affected by document features, such as, the type of document, the fonts used, the colors selected, etc. For example, an image of a white document with black lettering may be easier to process than a dark colored document with black letters. Image quality may also be affected by the mobile device used. Some mobile camera phones, for example, might have cameras that save an image using a greater number of mega pixels. Other mobile cameras phones might have an auto-focus feature, automatic flash, etc. Generally, these features may improve an image when compared to mobile devices that do not include such features.

**[00100]** A document image taken using a mobile device might have one or more of the defects discussed above. These defects or others may cause low accuracy when processing the image, for example, when processing one or more of the fields on a document. Accordingly, in some embodiments, systems and methods using a mobile device to create images of documents can include the ability to identify poor quality images. If the quality of an image is determined to be poor, a user may be prompted to take another image.

**Detecting an Out of Focus Image**

[00101] Mobile device 340 and mobile remittance server 310 can be configured to detect an out of focus image. A variety of metrics might be used to detect an out-of-focus image. For example, a focus measure can be employed. The focus measure can be the ratio of the maximum video gradient between adjacent pixels measured over the entire image and normalized with respect to an image's gray level dynamic range and "pixel pitch". The pixel pitch may be the distance between dots on the image. In some embodiments a focus score might be used to determine if an image is adequately focused. If an image is not adequately focused, a user might be prompted to take another image.

[00102] According to an embodiment, the mobile device 340 can be configured to detect whether an image is out of focus using the techniques disclosed herein. In an embodiment, the mobile remittance server 310 can be configured to detect out of focus images. In some embodiments, the mobile remittance server 310 can be configured to detect out of focus images and reject these images before performing mobile image quality assurance testing on the image. In other embodiments, detecting and out of focus image can be part of the mobile image quality assurance testing.

[0100] According to an embodiment, an image focus score can be calculated as a function of maximum video gradient, gray level dynamic range and pixel pitch. For example, in one embodiment:

$$\text{Image Focus Score} = (\text{Maximum Video Gradient}) * (\text{Gray Level Dynamic Range}) * (\text{Pixel Pitch}) \text{ (eq. 1)}$$

[0101] The video gradient may be the absolute value of the gray level for a first pixel "i" minus the gray level for a second pixel "i+1". For example:

$$\text{Video Gradient} = \text{ABS}[(\text{Grey level for pixel "i"}) - (\text{Gray level for pixel "i+1"})] \text{ (eq. 2)}$$

[0102] The gray level dynamic range may be the average of the "n" lightest pixels minus the average of the "n" darkest pixels. For example:

$$\text{Gray Level Dynamic Range} = [\text{AVE}(\text{"N" lightest pixels}) - \text{AVE}(\text{"N" darkest pixels})] \text{ (eq. 3)}$$

[0103] In equation 3 above, N can be defined as the number of pixels used to determine the average darkest and lightest pixel gray levels in the image. In some embodiments, N can be chosen to be 64. Accordingly, in some embodiments, the 64 darkest pixels are averaged together and the 64 lightest pixels are averaged together to compute the gray level dynamic range value.

[0104] The pixel pitch can be the reciprocal of the image resolution, for example, in dots per inch.

$$\text{Pixel Pitch} = [1 / \text{Image Resolution}] \text{ (eq. 4)}$$

[0105] In other words, as defined above, the pixel pitch is the distance between dots on the image because the Image Resolution is the reciprocal of the distance between dots on an image.

### **Detecting and Correcting Perspective Distortion**

[0106] Figure 7 is a diagram illustrating an example of perspective distortion in an image of a rectangular shaped document. An image can contain perspective transformation distortions 2500 such that a rectangle can become a quadrangle ABCD 2502, as illustrated in the figure. The perspective distortion can occur because an image is taken using a camera that is placed at an angle to a document rather than directly above the document. When directly above a rectangular document it will generally appear to be rectangular. As the imaging device moves from directly above the surface, the document distorts until it can no longer be seen and only the edge of the page can be seen.

[0107] The dotted frame 2504 comprises the image frame obtained by the camera. The image frame is sized h x w, as illustrated in the figure. Generally, it can be preferable to contain an entire document within the h x w frame of a single image. It will be understood, however, that some documents are too large or include too many pages for this to be preferable or even feasible.

[0108] In some embodiments, an image can be processed, or preprocessed, to automatically find and “lift” the quadrangle 2502. In other words, the document that forms quadrangle 502 can be separated from the rest of the image so that the document alone can be processed. By separating quadrangle 2502 from any background in an image, it can then be further processed.

[0109] The quadrangle 2502 can be mapped onto a rectangular bitmap in order to remove or decrease the perspective distortion. Additionally, image sharpening can be used to improve the out-of-focus score of the image. The resolution of the image can then be increased and the image converted to a black-and-white image. In some cases, a black-and-white image can have a higher recognition rate when processed using an automated document processing system in accordance with the systems and methods described herein.

[0110] An image that is bi-tonal, e.g., black-and-white, can be used in some systems. Such systems can require an image that is at least 200 dots per inch resolution. Accordingly, a color image taken using a mobile device can need to be high enough quality so that the image can successfully be converted from, for example, a 24 bit per pixel (24 bit/pixel) RGB image to a bi-tonal image. The image can be sized as if the document, e.g., check, payment coupon, etc., was scanned at 200 dots per inch.

[0111] FIG. 8 is a diagram illustrating an example original image, focus rectangle and document quadrangle ABCD in accordance with the example of FIG. 7. In some embodiments it can be necessary to place a document for processing at or near the center of an input image close to the camera. All points A, B, C and D are located in the image, and the focus rectangle 2602 is located inside quadrangle ABCD 2502. The document can also have a low out-of-focus score and the background surrounding the document can be selected to be darker than the document. In this way, the lighter document will stand out from the darker background.

### **Image Correction Module**

[0112] FIG. 6 is a flow diagram illustrating a method for correcting defects to mobile image according to an embodiment. According to an embodiment, the method illustrated in FIG. 6 can be performed by an image correction module implemented on the mobile remittance server 310.



The method illustrated in FIG. 6 can be implemented as part of step 510 of the method illustrated in FIG. 5. The image correction module can also receive a mobile image and processing parameters from a mobile device (step 505 of FIG. 5). According to some embodiments, some or all of the image correction functionality of the image correction module can be implemented on the mobile device 340, and the mobile device 340 can be configured to send a corrected mobile image to the mobile remittance server 310 for further processing.

**[0113]** According to an embodiment, the image correction module can also be configured to detect an out of focus image using the technique described above and to reject the mobile image if the image focus score for the image falls below a predetermined threshold without attempting to perform other image correction techniques on the image. According to an embodiment, the image correction module can send a message to the mobile device 340 indicating that the mobile image was too out of focus to be used and requesting that the user retake the image.

**[0114]** The image correction module can be configured to first identify the corners of a coupon or other document within a mobile image (step 605). One technique that can be used to identify the corners of the remittance coupon in a color image is illustrated in FIG. 9 and is described in detail below. The corners of the document can be defined by a set of points A, B, C, and D that represent the corners of the document and define a quadrangle.

**[0115]** The image correction module can be configured to then build a perspective transformation for the remittance coupon (step 610). As can be seen in FIG. 1, the angle at which an image of a document is taken can cause the rectangular shape of the remittance coupon to appear distorted. FIG. 7 and its related description above provide some examples of how a perspective transformation can be constructed for a quadrangle defined by the corners A, B, C, and D according to an embodiment. For example, the quadrangle identified in step 605 can be mapped onto a same-sized rectangle in order to build a perspective transformation that can be applied to the document subimage, i.e. the portion of the mobile image that corresponds to the remittance coupon, in order to correct perspective distortion present in the image.

**[0116]** A geometrical transformation of the document subimage can be performed using the perspective transformation built in step 610 (step 615). The geometrical transformation corrects

the perspective distortion present in the document subimage. An example of results of geometrical transformation can be seen in FIG. 2 where a document subimage of the remittance coupon pictured in FIG. 1 has been geometrically corrected to remove perspective distortion.

**[0117]** A “dewarping” operation can also be performed on the document subimage (step 620). An example of a warping of a document in a mobile image is provided in FIG. 35. Warping can occur when a document to be imaged is not perfectly flat or is placed on a surface that is not perfectly flat, causing distortions in the document subimage. A technique for identifying warping in a document subimage is illustrated in FIG. 36.

**[0118]** According to an embodiment, the document subimage can also be binarized (step 625). A binarization operation can generate a bi-tonal image with color depth of 1 bit per a pixel (1 bit/pixel). Some automated processing systems, such as some Remote Deposit systems require bi-tonal images as inputs. A technique for generating a bi-tonal image is described below with respect to FIG. 10. FIG. 11 illustrates a binarized version of the geometrically corrected mobile document image of the remittance coupon illustrated in FIG. 2. As illustrated, in the bi-tonal image of FIG. 11, the necessary information, such as payees, amounts, account number, etc., has been preserved, while extra information has been removed. For example, background patterns that might be printed on the coupon are not present in the bi-tonal image of the remittance coupon. Binarization of the subimage also can be used to remove shadows and other defects caused by unequal brightness of the subimage.

**[0119]** Once the image has been binarized, the code line of the remittance coupon can be identified and read (step 630). As described above, many remittance coupons include a code line that comprises computer-readable text that can be used to encode account-related information that can be used to reconcile a payment received with the account for which the payment is being made. Code line 205 of FIG. 2 illustrates an example of code line on a remittance coupon.

**[0120]** Often, a standard optical character recognition font, the OCR-A font, is used for printing the characters comprising the code line. The OCR-A font is a fixed-width font where the characters are typically spaced 0.10 inches apart. Because the OCR-A font is a standardized fixed-width font, the image correction module can use this information to determine a scaling

factor for the image of the remittance coupon. The scaling factor to be used can vary from image to image, because the scaling is dependent upon the position of the camera or other image capture device relative to the document being imaged and can also be dependent upon optical characteristics of the device used to capture the image of the document. FIG. 20 illustrates a scaling method that can be used to determine a scaling factor to be applied according to an embodiment. The method illustrated in FIG. 20 is related to scaling performed on a MICR-line of a check, but can be used to determine a scaling factor for an image of a remittance coupon based on the size of the text in the code line of the image of the remittance coupon.

**[0121]** Once the scaling factor for the image has been determined, a final geometrical transformation of the document image can be performed using the scaling factor (step 635). This step is similar to that in step 615, except the scaling factor is used to create a geometrically altered subimage that represents the actual size of the coupon at a given resolution. According to an embodiment, the dimensions of the geometrically corrected image produced by step 635 are identical to the dimensions of an image produced by a flat bed scanner at the same resolution.

**[0122]** During step 635, other geometrical corrections can also be made, such as correcting orientation of the coupon subimage. The orientation of the coupon subimage can be determined based on the orientation of the text of the code line.

**[0123]** Once the final geometrical transformation has been applied, a final adaptive binarization can be performed on the grayscale image generated in step 635 (step 640). The bi-tonal image output by this step will have the correct dimensions for the remittance coupon because the bi-tonal image is generated using the geometrically corrected image generated in step 635.

**[0124]** According to an embodiment, the image correction module can be configured to use several different binarization parameters to generate two or more bi-tonal images of the remittance coupon. The use of multiple images can improve data capture results. The use of multiple bi-tonal images to improve data capture results is described in greater detail below.

**Detecting Document within Color Mobile Image**

[0125] Referring now to FIG. 9, a flowchart is provided illustrating an example method for automatic document detection within a color image from a mobile device. According to an embodiment, the method illustrated in FIG. 9 can be used to implement step 605 of the method illustrated in FIG. 6. Typically, the operations described within method of FIG. 9 are performed within an automatic document detection module of the mobile remittance server 310; however, embodiments exist where the operations reside in multiple modules. In addition, generally the automatic document detection module takes a variety of factors into consideration when detecting the document in the mobile image. The automatic document detection module can take into consideration arbitrary location of the document within the mobile image, the 3-D distortions within the mobile image, the unknown size of the document, the unknown color of the document, the unknown color(s) of the background, and various other characteristics of the mobile engine, e.g. resolution, dimensions, etc.

[0126] The method of FIG. 9 begins at step 902 by receiving the original color image from the mobile device. Upon receipt, this original color image is converted into a smaller color image, also referred to as a color “icon” image, at operation 904. This color “icon” image preserves the color contrasts between the document and the background, while suppressing contrasts inside the document. A detailed description of an example conversion process is provided with respect to FIG. 12.

[0127] A color reduction operation is then applied to the color “icon” image at step 906. During the operation, the overall color of the image can be reduced, while the contrast between the document and its background can be preserved within the image. Specifically, the color “icon” image of operation 904 can be converted into a gray “icon” image (also known as a gray-scale “icon” image) having the same size. An example, color depth reduction process is described with further detail with respect to FIG. 14.

[0128] The corners of the document are then identified within the gray “icon” image (step 910). As previously noted above with respect to FIG. 7, these corners A, B, C, and D make up the quadrangle ABCD (e.g. quadrangle ABCD 2502). Quadrangle ABCD, in turn, makes up the

perimeter of the document. Upon detection of the corners, the location of the corners is outputted (step 910).

### **Binarization**

[0129] FIG. 10 illustrates a binarization method that can be used to generate a bi-tonal image from a document image according to an embodiment. The method illustrated in FIG. 10 can be used to implement the binarization step 625 of the method illustrated in FIG. 6. In an embodiment, the steps of the method illustrated in FIG. 10 can be performed within module of the mobile remittance server 310.

[0130] A binarization operation generates a bi-tonal image with color depth of 1 bit per a pixel (1 bit/pixel). In the case of documents, such as checks and deposit coupons, a bi-tonal image is required for processing by automated systems, such as Remote Deposit systems. In addition, many image processing engines require such an image as input. The method of FIG. 10 illustrates binarization of a gray-scale image of a document as produced by geometrical operation 1004. This particular embodiment uses a novel variation of well-known Niblack's method of binarization. As such, there is an assumption that the gray-scale image received has a the dimensions  $W$  pixel x  $H$  pixels and an intensity function  $I(x,y)$  gives the intensity of a pixel at location  $(x,y)$  in terms one of 256 possible gray-shade values (8 bit/pixel). The binarization operation will convert the 256 gray-shade value to a 2 shade value (1 bit/pixel), using an intensity function  $B(x,y)$ . In addition, to apply the method, a sliding window with dimensions  $w$  pixels x  $h$  pixels is defined and a threshold  $T$  for local (in-window) standard deviation of gray image intensity  $I(x,y)$  is defined. The values of  $w$ ,  $h$ , and  $T$  are all experimentally determined.

[0131] A gray-scale image of the document is received at step 1602, the method 1600 chooses a pixel  $p(x,y)$  within the image at step 1604. In FIG. 10, the average (mean) value  $ave$  and standard deviation  $\sigma$  of the chosen pixel's intensity  $I(x,y)$  within the  $w \times h$  current window location (neighborhood) of pixel  $p(x,y)$  are computed (step 1606). If the standard deviation  $\sigma$  is determined to be too small at operation 1608 (i.e.  $\sigma < T$ ), pixel  $p(x,y)$  is considered to low-contrast and, thus, part of the background. Accordingly, at step 1610, low-contrast pixels are converted to white, i.e. set  $B(x,y)$  set to 1, which is white; however, if the deviation  $\sigma$  is

determined to be larger or equal to the threshold  $T$ , i.e.  $\sigma \geq T$ , the pixel  $p(x,y)$  is considered to be part of the foreground. In step 1612, if  $I(p) < ave - k * \sigma$ , pixel  $p$  is considered to be a foreground pixel and therefore  $B(x,y)$  is set to 0 (black). Otherwise, the pixel is treated as background and therefore  $B(x,y)$  is set to 1. In the formula above,  $k$  is an experimentally established coefficient.

[0132] Subsequent to the conversion of the pixel at either step 1610 or operation 1612, the next pixel is chosen at step 1614, and operation 1606 is repeated until all the gray-scale pixels (8 bit/pixel) are converted to a bi-tonal pixel (1 bit/pixel). However, if no more pixels remain to be converted 1618, the bi-tonal image of the document is then outputted at step 1620.

### Conversion of Color Image to Icon Image

[0133] Referring now to FIG. 12, a flowchart is provided describing an example method for conversion of a color image to a smaller “icon” image according to an embodiment. This method can be used to implement step 904 of the method illustrated FIG. 9. The smaller “icon” image preserves the color contrasts between the document depicted therein and its background, while suppressing contrasts inside the document. Upon receipt of the original color image from the mobile device (step 1201), over-sharpening is eliminated within the image (step 1202). Accordingly, assuming the color input image  $I$  has the dimensions of  $W \times H$  pixels, operation 1202 averages the intensity of image  $I$  and downscales image  $I$  to image  $I'$ , such that image  $I'$  has dimensions that are half that of image  $I$  (i.e.  $W'=W/2$  and  $H'=H/2$ ). Under certain embodiments, the color transformation formula can be described as the following:

(eq. 5)  $C(p') = ave \{C(q): q \text{ in } S \times S\text{-window of } p\}$ , where

- $C$  is any of red, green or blue components of color intensity;
- $p'$  is any arbitrary pixel on image  $I'$  with coordinates  $(x',y')$ ;
- $p$  is a corresponding pixel on image  $I$ :  $p = p(x,y)$ , where  $x=2*x'$  and  $y = 2*y'$ ;
- $q$  is any pixel included into  $S \times S$ -window centered in  $p$ ;
- $S$  is established experimentally; and
- $ave$  is averaging over all  $q$  in the  $S \times S$ -window.

[0134] Small “dark” objects within the image can then be eliminated (step 1204). Examples of such small “dark” objects include, but are not limited to, machine-printed characters and hand-printed characters inside the document. Hence, assuming operation 1204 receives image  $I'$  from step 1202, step 1204 creates a new color image  $I''$  referred to as an “icon” with width  $W''$  set to a fixed small value and height  $H''$  set to  $W''*(H/W)$ , thereby preserving the original aspect ratio of image  $I$ . In some embodiments, the transformation formula can be described as the following:

(eq. 6)  $C(p'') = \max \{C(q') : q' \text{ in } S'xS' \text{-window of } p'\}$ , where

- $C$  is any of red, green or blue components of color intensity;
- $p''$  is an arbitrary pixel on image  $I''$ ;
- $p'$  is a pixel on image  $I'$  which corresponds to  $p''$  under similarity transformation, as previously defined;
- $q'$  is any pixel on image  $I'$  included into  $S'xS'$ -window centered in  $p'$ ;
- $\max$  is maximum over all  $q'$  in the  $S'xS'$ -window;
- $W''$  is established experimentally;
- $S'$  is established experimentally for computing the intensity  $I''$ ; and
- $I''(p'')$  is the intensity value defined by maximizing the intensity function  $I'(p')$  within the window of corresponding pixel  $p'$  on image  $I'$ , separately for each color plane.

The reason for using the “maximum” rather than “average” is to make the “icon” whiter (white pixels have a RGB-value of (255,255,255)).

[0135] In the next operation 1206, the high local contrast of “small” objects, such as lines, text, and handwriting on a document, is suppressed, while the other object edges within the “icon” are preserved. Often, these other object edges are bold. In various embodiments of the invention, multiple dilation and erosion operations, also known as morphological image transformations, are utilized in the suppression of the high local contrast of “small” objects. Such morphological image transformations are commonly known and used by those of ordinary skill in the art. The sequence and amount of dilation and erosion operations used is determined experimentally. Subsequent to the suppression operation 1206, a color “icon” image is outputted at operation 1208. FIG. 13B depicts an example of the mobile image of a check illustrated in FIG. 13A after being converted into a color “icon” image according to an embodiment.

### **Color Depth Reduction**

[0136] Referring now to FIG. 14, a flowchart is provided illustrating an example method that provides further details with respect to the color depth reduction operation 906 as illustrated in FIG. 9. At step 1301, a color “icon” image for color reduction is received. The color “icon” image is divided into a grid (or matrix) of fixed length and width with equal size grid elements at operation 1302. In some embodiments, the preferred grid size is such that there is a center grid element. For example, a grid size of 3x3 may be employed. FIG. 15A depicts an example of the color “icon” image of FIG. 13B after operation 1302 has divided it into a 3x3 grid in accordance with one embodiment of the invention.

[0137] Then, at step 1304, the “central part” of the icon, which is usually the center most grid element, has its color averaged. Next, the average color of the remaining parts of the icon is computed at step 1306. More specifically, the grid elements “outside” the “central part” of the “icon” have their colors averaged. Usually, in instances where there is a central grid element, e.g. 3x3 grid, the “outside” of the “central part” comprises all the grid elements other than the central grid element.

[0138] Subsequently, a linear transformation for the RGB-space is determined at step 1308. The linear transformation is defined such that it maps the average color of the “central part” computed during operation 1304 to white, i.e. 255, while the average color of the “outside” computed during operation 1306 maps to black, i.e. 0. All remaining colors are linearly mapped to a shade of gray. This linear transformation, once determined, is used at operation 1310 to transform all RGB-values from the color “icon” to a gray –scale “icon” image, which is then outputted at operation 1312. Within particular embodiments, the resulting gray “icon” image, also referred to as a gray-scale “icon” image, maximizes the contrast between the document background, assuming that the document is located close to the center of the image and the background. FIG. 15B depicts an example of the color “icon” image of FIG. 13B once it has been converted to a gray “icon” image in accordance with one embodiment.

[0139] Referring now to FIG. 16, a flowchart is provided illustrating an example method for finding document corners from a gray “icon” image containing a document. The method



illustrated in FIG. 16 can be used to implement step 908 of the method illustrated in FIG. 9. Upon receiving a gray “icon” image at operation 1401, the “voting” points on the gray “icon” image are found in step 1402 for each side of the document depicted in the image. Consequently, all positions on the gray “icon” image that could be approximated with straight line segments to represent left, top, right, and bottom sides of the document are found.

[0140] In accordance with one embodiment, this goal is achieved by first looking for the “voting” points in the half of the “icon” that corresponds with the current side of interest. For instance, if the current side of interest is the document’s top side, the upper part of the “icon” ( $Y < H/2$ ) is examined while the bottom part of the “icon” ( $Y \geq H/2$ ) is ignored.

[0141] Within the selected half of the “icon,” the intensity gradient (contrast) in the correct direction of each pixel is computed. This is accomplished in some embodiments by considering a small window centered in the pixel and, then, breaking the window into an expected “background” half where the gray intensity is smaller, i.e. where it is supposed to be darker, and into an expected “doc” half where the gray intensity is higher, i.e. where it is supposed to be whiter. There is a break line between the two halves, either horizontal or vertical depending on side of the document sought to be found. Next the average gray intensity in each half-window is computed, resulting in an average image intensity for the “background” and an average image intensity of the “doc.” The intensity gradient of the pixel is calculated by subtracting the average image intensity for the “background” from the average image intensity for the “doc.”

[0142] Eventually, those pixels with sufficient gray intensity gradient in the correct direction are marked as “voting” points for the selected side. The sufficiency of the actual gray intensity gradient threshold for determining is established experimentally.

[0143] Continuing with method 1400, candidate sides, i.e. line segments that potentially represent the sides of the document, i.e. left, top, right, and bottom sides, are found. In order to do so, some embodiments find all subsets within the “voting” points determined in step 1402 that could be approximated by a straight line segment (linear approximation). In many embodiments, the threshold for linear approximation is established experimentally. This subset of lines is

defined as the side “candidates.” As an assurance that the set of side candidates is never empty, the gray “icon” image’s corresponding top, bottom, left, and right sides are also added to the set.

[0144] Next, in step 1406 chooses the best candidate for each side of the document from the set of candidates selected in operation 1404, thereby defining the position of the document within the gray “icon” image. In accordance with some embodiments, the following process is used in choosing the best candidate for each side of the document:

[0145] The process starts with selecting a quadruple of line segments  $\{\mathbf{L}, \mathbf{T}, \mathbf{R}, \mathbf{B}\}$ , where  $\mathbf{L}$  is one of the candidates for the left side of the document,  $\mathbf{T}$  is one of the candidates for the top side of the document,  $\mathbf{R}$  is one of the candidates for the right side of the document, and  $\mathbf{B}$  is one of the candidates for the bottom side of the document. The process then measures the following characteristics for the quadruple currently selected.

[0146] The amount of “voting” points is approximated and measured for all line segments for all four sides. This amount value is based on the assumption that the document’s sides are linear and there is a significant color contrast along them. The larger values of this characteristic increase the overall quadruple rank.

[0147] The sum of all intensity gradients over all voting points of all line segments is measured. This sum value is also based on the assumption that the document’s sides are linear and there is a significant color contrast along them. Again, the larger values of this characteristic increase the overall quadruple rank.

[0148] The total length of the segments is measured. This length value is based on the assumption that the document occupies a large portion of the image. Again, the larger values of this characteristic increase the overall quadruple rank.

[0149] The maximum of gaps in each corner is measured. For example, the gap in the left/top corner is defined by the distance between the uppermost point in the L-segment and the leftmost point in the T-segment. This maximum value is based on how well the side-candidates suit the assumption that the document’s shape is quadrangle. The smaller values of this characteristic increase the overall quadruple rank.

[0150] The maximum of two angles between opposite segments, i.e. between L and R, and between T and R, is measured. This maximum value is based on how well the side-candidates suit the assumption that the document's shape is close to parallelogram. The smaller values of this characteristic increase the overall quadruple rank.

[0151] The deviation of the quadruple's aspect ratio from the "ideal" document aspect ratio is measured. This characteristic is applicable to documents with a known aspect ratio, e.g. checks. If the aspect ratio is unknown, this characteristic should be excluded from computing the quadruple's rank. The quadruple's aspect ratio is computed as follows:

- a) Find the quadrangle by intersecting the quadruple's elements;
- b) Find middle-point of each of the four quadrangle's sides;
- c) Compute distances between middle-points of opposite sides, say  $D1$  and  $D2$ ;
- d) Find the larger of the two ratios:  $R = \max(D1/D2, D2/D1)$ ;
- e) Assuming that the "ideal" document's aspect ratio is known and Min/MaxAspectRatio represent minimum and maximum of the aspect ratio respectively, define the deviation in question as:
  - 0, if  $\text{MinAspectRatio} \leq R \leq \text{MaxAspectRatio}$
  - $\text{MinAspectRatio} - R$ , if  $R < \text{MinAspectRatio}$
  - $R - \text{MaxAspectRatio}$ , if  $R > \text{MaxAspectRatio}$ .
- f) For checks, MinAspectRatio can be set to 2.0 and MaxAspectRatio can be set to 3.0.

This aspect ratio value is based on the assumption that the document's shape is somewhat preserved during the perspective transformation. The smaller values of this characteristic increase the overall quadruple rank.

[0152] Following the measurement of the characteristics of the quadruple noted above, the quadruple characteristics are combined into a single value, called the quadruple rank, using weighted linear combination. Positive weights are assigned for the amount of "voting" points, the sum all of intensity gradients, and the total length of the segments. Negatives weights are assigned for maximum gaps in each corner, maximum two angles between opposite segments,

and the deviation of the quadruple's aspect ratio. The exact values of each of the weights are established experimentally.

[0153] The operations set forth above are repeated for all possible combinations of side candidates, eventually leading to the "best" quadruple, which is the quadruple with the highest rank. The document's corners are defined as intersections of the "best" quadruple's sides, i.e. the best side candidates.

[0154] In, step 1408 the corners of the document are defined using the intersections of the best side candidates. A person of ordinary skill in the art would appreciate that these corners can then be located on the original mobile image by transforming the corner locations found on the "icon" using the similarity transformation previously mentioned. Method 1400 concludes at step 1410 where the locations of the corners defined in step 1408 are output.

### **Geometric Correction**

[0155] FIG. 17 provides a flowchart that illustrates an example method for geometric correction in accordance with the invention according to an embodiment. According to an embodiment, the method illustrated in FIG. 17 can be used to implement steps 610, 615, and 635 of the method illustrated in FIG. 6. As previously mentioned, geometric correction is needed to correct any possibly perspective distortions that exist in the original mobile image. Additionally, geometric correction can correct the orientation of the documentation within the original mobile image, e.g. document is orientated at 90, 180, or 270 degrees where the right-side-up orientation is 0 degrees. It should be noted that in some embodiments, the orientation of the document depends on the type of document depicted in the mobile image, as well as the fields of relevance on the document.

[0156] In instances where the document is in landscape orientation (90 or 270 degrees), as illustrated by the check in FIG. 18A, geometric correction is suitable for correcting the orientation of the document. Where the document is at 180 degree orientation, detection of the 180 degree orientation and its subsequent correction are suitable when attempting to locate an object of relevance on the document. A codeline for a remittance coupon can be located in

various locations on the remittance coupon, and might not be located along the bottom of the coupon. The ability to detect a codeline in an image of the remittance coupon changes significantly after the document has been rotated 180-degrees. In contrast, the MICR-line of check is generally known to be at a specific location along the bottom of the document, and the MICR-line can be used to determine the current orientation of the check within the mobile image. In some embodiments, the object of relevance on a document depends on the document's type. For example, where the document is a contract, the object of relevance may be a notary seal, signature, or watermark positioned at a known position on the contract. Greater detail regarding correction of a document (specifically, a check) having upside-down orientation (180 degree orientation) is provided with respect to FIG. 19.

**[0157]** According to some embodiments, a mathematical model of projective transformations is built and converts the distorted image into a rectangle-shaped image of predefined size. According to an embodiment, this step corresponds to step 610 of FIG. 6. In an example, where the document depicted in mobile image is a check, the predefined size is established as 1200x560 pixels, which is roughly equivalent to the dimensions of a personal check scanned at 200 DPI. In other embodiments, where the document depicted is a remittance coupon, the size of the remittance coupons may not be standardized. However, the size and spacing of the characters comprising the code line can be used to determine a scaling factor to be applied to the image to correct the size of the image of the remittance coupon relative to a specific resolution.

**[0158]** Continuing with reference to the method of FIG. 17, there are two separate paths of operations that are either performed sequentially or concurrently, the outputs of which are eventually utilized in the final output. One path of operations begins at step 1504 where the original mobile image in color is received. In step 1508, the color depth of the original mobile image is reduced from a color image with 24 bit per a pixel (24 bit/pixel) to a gray-scale image with 8 bit per a pixel (8 bit/pixel). This image is subsequently outputted to step 1516 as a result of step 1512.

**[0159]** The other path of operations begins at step 1502, where the positions of the document's corners within the gray "icon" image are received. Based off the location of the

corners, the orientation of the document is determined and the orientation is corrected (step 1506). In some embodiments, this operation uses the corner locations to measure the aspect ratio of the document within the original image. Subsequently, a middle-point between each set of corners can be found, wherein each set of corners corresponds to one of the four sides of the depicted document, resulting in the left (L), top (T), right (R), and bottom (B) middle-points (step 1506). The distance between the L to R middle-points and the T to B middle points are then compared to determine which of the two pairs has the larger distance. This provides step 1506 with the orientation of the document.

**[0160]** In some instances, the correct orientation of the document depends on the type of document that is detected. For example, as illustrated in FIG. 18A, where the document of interest is a check, the document is determined to be in landscape orientation when the distance between the top middle-point and bottom middle-point is larger than the distance between the left middle-point and the right middle-point. The opposite might be true for other types of documents.

**[0161]** If it is determined in step 1506 that an orientation correction is necessary, then the corners of the document are shifted in a loop, clock-wise in some embodiments and counter-clockwise in other embodiments.

**[0162]** At step 1510, the projective transformation is built to map the image of the document to a predefined target image size of width of  $W$  pixels and height of  $H$  pixels. In some embodiments, the projective transformation maps the corners A, B, C, and D of the document as follows: corner A to (0,0), corner B to (W,0), corner C to (W,H), and corner D to (0,H). Algorithms for building projective transformation are commonly known and used amongst those of ordinary skill in the art.

**[0163]** At step 1516, the projective transformation created during step 1514 is applied to the mobile image in gray-scale as outputted as a result of step 1512. The projective transformation as applied to the gray-scale image of step 1512 results in all the pixels within the quadrangle ABCD depicted in the gray-scale image mapping to a geometrically corrected, gray-scale image of the document alone. FIG. 18B is an example gray-scale image of the document depicted in

FIG. 13A once a geometrical correction operation in accordance with the invention is applied thereto. The process concludes at operation 1518 where the gray-scale image of the document is outputted to the next operation.

### Correcting Landscape Orientation

[0164] FIG. 19 is a flow chart illustrating a method for correcting landscape orientation of a document image according to an embodiment. As previously noted, the geometric correction operation as described in FIG. 17 is one method in accordance with the invention for correcting a document having landscape orientation within the mobile image. However, even after the landscape orientation correction, the document still may remain in upside-down orientation. In order to the correct upside-down orientation for certain documents, some embodiments of the invention require the image containing the document be binarized beforehand. Hence, the orientation correction operation included in step 635 usually follows the binarization operation of 625. While the embodiment described herein uses the MICR-line of a check or determine the orientation of an image, the code line of a remittance coupon can be used to determine the orientation of a remittance coupon using the technique described herein.

[0165] Upon receiving the bi-tonal image of the check at operation 1702, the MICR-line at the bottom of the bi-tonal check image is read at operation 1704 and an MICR-confidence value is generated. This MICR-confidence value (*MCI*) is compared to a threshold value *T* at operation 1706 to determine whether the check is right-side-up. If  $MCI > T$  at operation 1708, then the bi-tonal image of the check is right side up and is outputted at operation 1710.

[0166] However, if  $MCI \leq T$  at operation 1708, then the image is rotated 180 degrees at operation 1712, the MICR-line at the bottom read again, and a new MICR-confidence value generated (*MC2*). The rotation of the image by 180 degree is done by methods commonly-known in the art. The MICR-confidence value after rotation (*MC2*) is compared to the previous MICR-confidence value (*MCI*) plus a *Delta* at operation 1714 to determine if the check is now right-side-up. If  $MC2 > MC2 + \text{Delta}$  at operation 1716, the rotated bi-tonal image has the check right-side-up and, thus, the rotated image is outputted at operation 1718. Otherwise, if  $MC2 \leq MC2 + \text{Delta}$  at operation 1716, the original bi-tonal image of the check is right-side-up and

outputted at operation 1710. *Delta* is a positive value selected experimentally that reflects a higher apriori probability of the document initially being right-side-up than upside-down.

### **Size Correction**

[0167] FIG. 20 provides a flowchart illustrating an example method for size correction of an image according to an embodiment. The method of FIG. 20 can be used to implement the size correction step described in relation to step 630 of FIG. 6. Specifically, FIG. 20 illustrates an example method, in accordance with one embodiment, for correcting the size of a check within a bi-tonal image, where the check is oriented right-side-up. A person of ordinary skill in the art would understand and appreciate that this method can operate differently for other types of documents, e.g. deposit coupons, remittance coupons.

[0168] Since many image processing engines are sensitive to image size, it is crucial that the size of the document image be corrected before it can be properly processed. For example, a form identification engine may rely on the document size as an important characteristic for identifying the type of document that is being processed. Generally, for financial documents such as checks, the image size should be equivalent to the image size produced by a standard scanner running at 200 DPI.

[0169] In addition, where the document is a check, during the geometric correction operation of some embodiments of the invention, the geometrically corrected predefined image size is at 1200x560 pixels (See, for e.g., FIG. 15 description), which is roughly equivalent to the size of a personal check scanned at 200 DPI; however, the size of business checks tend to vary significantly, with most business checks having a width greater than 1200 pixels when scanned at 200 DPI. Some business checks are known to be as wide as 8.75", which translates to be 1750 pixels in width when scanned at 200 DPI. Hence, in order to restore the size of business checks that have been geometrically corrected in accordance with the invention at a predefined image size of 1200x560 pixels, the size correction operation is performed.

[0170] Referring now to FIG. 20, after receiving a bi-tonal image containing a check that is orientated right-side-up at operation 1802, the MICR-line at the bottom of the check is read at



operation 1804. This allows the average width of the MICR-characters to be computed at operation 1806. In doing so, the computer average width gets compared to the average size of an MICR-character at 200 DPI at operation 1808, and a scaling factor is computed accordingly. In some embodiments of the invention, the scaling factor **SF** is computer as follows:

(eq. 7)  $SF=AW_{200}/AW$ , where

- **AW** is the average width of the MICR-character found; and
- **AW<sub>200</sub>** is the corresponding “theoretical” value based on the ANSI x9.37 standard (Specifications for Electronic Exchange of Check and Image Data) at 200 DPI.

[0171] The scaling factor is used at operation 1810 to determine whether the bi-tonal image of the check requires size correction. If the scaling **SF** is determined to be less than or equal to  $1.0 + \mathbf{Delta}$ , then the most recent versions of the check’s bi-tonal image and the check’s the gray-scale image are output at operation 1812. **Delta** defines the system’s tolerance to wrong image size.

[0172] If, however, the scaling factor **SF** is determined to be higher than  $1.0 + \mathbf{Delta}$ , then at operation 1814 the new dimensions of the check are computed as follows:

(eq. 8)  $AR = H_S/W_S$

(eq. 9)  $W' = W * SF$

(eq. 10)  $H' = AR * W'$ , where

- **H<sub>S</sub>** and **W<sub>S</sub>** are the height and width of the check snippet found on the original image;
- **AR** is the check aspect ratio which we want to maintain while changing the size;
- **W** is the width of geometrically corrected image before it’s size is adjusted;
- **W'** is the adjusted check’s width in pixels; and
- **H'** is the adjusted check’s height in pixels.

Subsequent to re-computing the new dimensions, operation 1814 repeats geometrical correction and binarization using the newly dimensioned check image. Following the repeated operations,

operation 1812 outputs the resulting bi-tonal image of the check and gray-scale image of the check.

### **Image Quality Assurance**

**[0173]** Once the mobile remittance server 310 has processed a mobile image (see step 510 of the method illustrated in FIG. 5), the mobile remittance server 310 can be configured to perform image quality assurance processing on the mobile image to determine whether the quality of the image is sufficient to submit to a remittance processor 215.

**[0174]** FIG. 21 illustrates a mobile document image processing engine (MDIPE) module 2100 for performing quality assurance testing on mobile document images according to an embodiment. The MDIPE module 2100 can receive a mobile document image captured by a mobile device, or multiple mobile images for some tests; perform preprocessing on the mobile document image; select tests to be performed on the mobile document image; and execute the selected tests to determine whether the quality of the image is high enough quality for a particular mobile application. The MDIPE module 2100 includes a preprocessing module 2110 and test execution module 2130. The preprocessing module 2110 can be configured to receive a mobile image 2105 captured using a camera of a mobile device as well as processing parameters 2107. According to an embodiment, the mobile image 2105 and the processing parameters 2107 can be passed to MDIPE 2100 by a mobile application on the mobile device where the mobile application provides the mobile image 2105 to the MDIPE 2100 to have the quality of the mobile image 2105 assessed.

**[0175]** The processing parameters 2107 can include various information that the MDIPE 2100 can use to determine which tests to run on the mobile image 2105. For example, the processing parameters 2107 can identify the type of device used to capture the mobile image 2105, the type of mobile application that will be used to process the mobile image if the mobile image passes the IQA testing, or both. The MDIPE 2100 can use this information to determine which tests to select from test data store 2132 and which test parameters to select from test parameter data store 2134. For example, if a mobile image is being tested for a mobile deposit application that expects an image of a check, a specific set of tests related to assessing the image

quality for a mobile image of a check can be selected, such as an MICR-line test, or a test for whether an image is blurry, etc. The MDIPE 2100 can also select test parameters from test parameters data store 2134 that are appropriate for the type of image to be processed, or for the type of mobile device that was used to capture the image, or both. In an embodiment, different parameters can be selected for different mobile phones that are appropriate for the type of phone used to capture the mobile image. For example, some mobile phones might not include an autofocus feature.

**[0176]** The preprocessing module 2110 can process the mobile document image to extract a document snippet that includes the portion of the mobile document that actually contains the document to be processed. This portion of the mobile document image is also referred to herein as the document subimage. The preprocessing module 2110 can also perform other processing on the document snippet, such as converting the image to a grayscale or bi-tonal document snippet, geometric correction of the document subimage to remove view distortion, etc. Different tests can require different types of preprocessing to be performed, and the preprocessing module 2110 can produce mobile document snippets from a mobile document image depending on the types of mobile IQA tests to be executed on the mobile document image.

**[0177]** The test execution module 2130 receives the selected tests and test parameters 2112 and the preprocessed document snippet (or snippets) 120 from the preprocessing module 2110. The test execution module 2130 executes the selected tests on the document snippet generated by the preprocessing module 2110. The test execution module 2130 also uses the test parameters provided by the preprocessing module 2110 when executing the test on the document snippet. The selected tests can be a series of one or more tests to be executed on the document snippets to determine whether the mobile document image exhibits geometrical or other defects.

**[0178]** The test execution module 2130 executes each selected test to obtain a test result value for that test. The test execution module 2130 then compares that test result value to a threshold value associated with the test. If the test result value is equal to or exceeds the threshold, then the mobile image has passed the test. Otherwise, if the test result value is less

than the threshold, the mobile document image has failed the test. According to some embodiments, the test execution module 2130 can store the test result values for the tests performed in test results data store 2138.

**[0179]** According an embodiment, the test threshold for a test can be stored in the test parameters data store 2134 and can be fetched by the preprocessing module 2110 and included with the test parameters 2112 provided to the test execution module 2130. According to an embodiment, different thresholds can be associated with a test based on the processing parameters 2107 received by the preprocessing module 2110. For example, a lower threshold might be used for an image focus IQA test for image capture by camera phones that do not include an autofocus feature, while a higher threshold might be used for the image focus IQA test for image capture by camera phones that do include an autofocus feature.

**[0180]** According to an embodiment, a test can be flagged as “affects overall status.” These tests are also referred to here as “critical” tests. If a mobile image fails a critical test, the MDIPE 2100 rejects the image and can provide detailed information to the mobile device user explaining why the image was not of a high enough quality for the mobile application and that provides guidance for retaking the image to correct the defects that caused the mobile document image to fail the test, in the event that the defect can be corrected by retaking the image.

**[0181]** According to an embodiment, the test result messages provided by the MDIPE 2100 can be provided to the mobile application that requested the MDIPE 2100 perform the quality assurance testing on the mobile document image, and the mobile application can display the test results to the user of the mobile device. In certain embodiments, the mobile application can display this information on the mobile device shortly after the user takes the mobile document image to allow the user to retake the image if the image is found to have defects that affect the overall status of the image. In some embodiments, where the MDIPE 2100 is implemented at least in part on the mobile device, the MDIPE 2100 can include a user interface module that is configured to display the test results message on a screen of the mobile device.

**[0182]** FIG. 21 merely provides a description of the logical components of the MDIPE 2100. In some embodiments, the MDIPE 2100 can be implemented on the mobile device 340, in

software, hardware, or a combination thereof. In other embodiments, the MDIPE 2100 can be implemented on the mobile remittance server 310, and the mobile device can send the mobile image 2105 and the processing parameters 2107, e.g., via a wireless interface, to the mobile remittance server 310 for processing, and the mobile remittance server 310 can send the test results and test messages 2140 to the mobile device to indicate whether the mobile image passed testing. In some embodiments, part of the functionality of the MDIPE 2100 can be implemented on the mobile device while other parts of the MDIPE 2100 are implemented on the remote server. The MDIPE 2100 can be implemented in software, hardware, or a combination thereof. In still other embodiments, the MDIPE 2100 can be implemented entirely on the remote server, and can be implemented using appropriate software, hardware, or a combination there.

**[0183]** FIG. 22 is a flow diagram of a process for performing mobile image quality assurance on an image captured by a mobile device according to an embodiment. The process illustrated in FIG. 22 can be performed using the MDIPE 2100 illustrated in FIG. 21.

**[0184]** The mobile image 2105 captured by a mobile device is received (step 2205). The mobile image 2105 can also be accompanied by one or more processing parameters 2107.

**[0185]** As described above, the MDIPE 2100 can be implemented on the mobile device, and the mobile image can be provided by a camera that is part of or coupled to the mobile device. In some embodiments, the MDIPE 2100 can also be implemented at least in part on a remote server, and the mobile image 2105 and the processing parameters 2107 can be transmitted to the remote server, e.g., via a wireless interface included in the mobile device.

**[0186]** Once the mobile image 2105 and the processing parameters 2107 have been received, the mobile image is processed to generate a document snippet or snippets (step 2210). For example, preprocessing module 2110 of MDIPE 2100 can be used to perform various preprocessing on the mobile image. One part of this preprocessing includes identifying a document subimage in the mobile image. The subimage is the portion of the mobile document image that includes the document. The preprocessing module 2110 can also perform various preprocessing on the document subimage to produce what is referred to herein as a “snippet.” For example, some tests can require that a grayscale image of the subimage be created. The

preprocessing module 2110 can create a grayscale snippet that represents a grayscale version of the document subimage. In another example, some tests can require that a bitonal image of the subimage be created. The preprocessing module 2110 can create a bitonal snippet that represents a bitonal version of the document subimage. In some embodiments, the MDIPE 2100 can generate multiple different snippets based on the types of tests to be performed on the mobile document image.

**[0187]** After processing the mobile document image to generate a snippet, the MDIPE 2100 then selects one or more tests to be performed on the snippet or snippets (step 2215). In an embodiment, the tests to be performed can be selected from test data store 2132. In an embodiment, the MDIPE 2100 selects the one or more tests based on the processing parameters 2107 that were received with the mobile image 2105.

**[0188]** After selecting the tests from the test data store 2132, test parameters for each of the tests can be selected from the test parameters data store 2134 (step 2220). According to an embodiment, the test parameters can be used to configure or customize the tests to be performed. For example, different test parameters can be used to configure the tests to be more or less sensitive to certain attributes of the mobile image. In an embodiment, the test parameters can be selected based on the processing parameters 2107 received with the mobile image 2105. As described above, these processing parameters can include information, such as the type of mobile device used to capture the mobile image as well as the type of mobile application that is going to be used to process the mobile image if the mobile image passes scrutiny of the mobile image IQA system.

**[0189]** Once the tests and the test parameters have been retrieved and provided to the test execution module 2130, a test is selected from tests to be executed, and the test is executed on the document snippet to produce a test result value (step 2225). In some embodiments, more than one document snippet may be used by a test. For example, a test can be performed that tests whether images of a front and back of a check are actually images of the same document can be performed. The test engine can receive both an image of the front of the check and an image of

the back of the check from the preprocessing module 2110 and use both of these images when executing the test.

**[0190]** The test result value obtained by executing the test on the snippet or snippets of the mobile document is then compared to test threshold to determine whether the mobile image passes or fails the test (step 2230) and a determination is made whether the test results exceed the threshold (step 2235). According to an embodiment, the test threshold can be configured or customized based on the processing parameters 2107 received with the mobile image. For example, the test for image blurriness can be configured to use a higher threshold for passing if the image is to be used to for a mobile deposit application where the MICR-line information needs to be recognized and read from the document image. In contrast, the test for blurriness can be configured use a lower threshold for passing the mobile image for some mobile applications. For example, the threshold for image quality may be lowered for if a business card is being imaged rather than a check. The test parameters can be adjusted to minimize the number of false rejects and false accept rate, the number of images marked for reviewing, or both.

**[0191]** The “affects overall status” flag of a test can also be configured based on the processing parameters 2107. For example, a test can be marked as not affecting the overall status for some types of mobile applications or for documents being processed, or both. Alternatively, a test can also be marked as affecting overall status for other types of mobile applications or documents being processed, or both. For example, a test that identifies the MICR-line of a check can be marked as “affecting overall status” so that if the MICR-line on the check cannot be identified in the image, the image will fail the test and the image will be rejected. In another example, if the mobile application is merely configured to receive different types of mobile document image, the mobile application can perform a MICR-line test on the mobile document image in an attempt to determine whether the document that was imaged was a check. In this example, the MICR-line may not be present, because a document other than a check may have been imaged. Therefore, the MICR-line test may be marked as not “affecting overall status,” and if a document fails the test, the transaction might be flagged for review but not marked as failed.

**[0192]** Since different camera phones can have cameras with very different optical characteristics, image quality may vary significantly between them. As a result, some image quality defects may be avoidable on some camera phones and unavoidable on the others and therefore require different configurations. To mitigate the configuration problem, Mobile IQA test can be automatically configured for different camera phones to use different tests, or different thresholds for the tests, or both. For example, as described above, a lower threshold can be used for an image focus IQA test on mobile document images that are captured using a camera phone that does not include an autofocus feature than would be used for camera phones that do include an autofocus feature, because it can be more difficult for a user to obtain as clear an image on using a device that doesn't an autofocus feature.

**[0193]** In certain embodiments, if the test result exceeded or equaled the threshold, the image passed the test and a determination is made whether there are more tests to be executed (step 2240). If there are more tests to be executed, the next test can be selected and executed on the document snippet (step 2225). Otherwise, if there were not more tests to be executed, the test results, or test messages, or both are output by MDIPE 2100 (step 2270). There can be one or more test messages included with the results if the mobile image failed one more of the tests that were executed on the image.

**[0194]** In such embodiments, if the test result was less than the threshold, then the mobile image has failed the test. A determination is made whether the test affects the overall status (step 250). If the test affects the overall status of the image, detailed test result messages that explain why the image failed the test can be loaded from the test message data store 134 (step 2255) and the test result messages can be added to the test results (step 2260). The test results and test messages can then be output by the MDIPE 2100 (step 2270).

**[0195]** Alternatively, if the test did not affect the overall status, the test results can be loaded noted and the transaction can be flagged for review (step 2265). By flagging the transaction for review, a user of a mobile device can be presented with information indicating that a mobile image has failed at least some of the test that were performed on the image, but the image still may be of sufficient quality for use with the mobile application. The user can then be presented



with the option to retake the image or to send the mobile image to the mobile application for processing. According to some embodiments, detailed test messages can be loaded from the test message data store 134 for all tests that fail and can be included with the test results, even if the test is not one that affects the overall status of the mobile image.

**[0196]** According to some embodiments, the mobile IQA test can also be configured to eliminate repeated rejections of a mobile document. For example, if an image of a check is rejected as have too low a contrast by a contrast test, the image is rejected, and the user can retake and resubmit the image via the mobile application, the processing parameters 2107 received with the mobile image can include a flag indicating that the image is being resubmitted. In some embodiments, the thresholds associated with the tests that the image failed can be lowered to see if the image can pass the test with a lower threshold. In some embodiments, the thresholds are only lowered for non-critical tests. According to an embodiment, the processing parameters 2107 can also include a count of the number of times that an image has been resubmitted and the thresholds for a test are only lowered after a predetermined number of times that the image is resubmitted.

**[0197]** FIG. 23 is a flow diagram of a process for performing mobile image quality assurance on an image of a check captured by a mobile device according to an embodiment. Like the process illustrated in FIG. 22, the process illustrated in FIG. 23 can be performed using the MDIPE 2100 illustrated in FIG. 21. The method illustrated in FIG. 23 can be used where an image of a check is captured in conjunction with a remittance payment. The method illustrated in FIG. 23 can be used to assess the quality of the image of the check.

**[0198]** The method illustrated in FIG. 23 illustrates how the mobile IQA and MDIPE 2100 can be used with the electronic check processing provided under the Check Clearing for the 21st Century Act. The Check Clearing for the 21st Century Act (also referred to as the “Check 21 Act”) is a United States federal law (Pub.L. 108-100) that was enacted on October 28, 2003. The law allows the recipient of a paper check to create a digital version of the original check called a “substitute check,” which can be processed, eliminating the need to process the original physical document. The substitute check includes an image of the front and back sides of the original

physical document. The mobile IQA tests can be used check the quality of the images captured by a mobile device. The snippets generated by the MDIPE 2100 can then be further tested by one or more Check 21 mobile IQA tests that perform image quality assurance on the snippets to determine whether the images meet the requirements of the Check 21 Act as well.

**[0199]** The mobile image 2105 captured by a mobile device is received (step 2305). In an embodiment, image of the front and back sides of the check can be provided. The mobile image 2105 can also be accompanied by one or more processing parameters 2107. Check data can also be optionally received (step 2307). The check data can be optionally provided by the user at the time that the check is captured. This check data can include various information from the check, such as the check amount, check number, routing information from the face of the check, or other information, or a combination thereof. In some embodiments, a mobile deposition application requests this information from a user of the mobile device, allows the user to capture an image of a check or to select an image of a check that has already been captured, or both, and the mobile deposit information provides the check image, the check data, and other processing parameters to the MDIPE 2100.

**[0200]** Once the mobile image 2105, the processing parameters 2107, and the check data have been received, the mobile image is processed to generate a document snippet or snippets (step 2310). As described above, the preprocessing can produce one or more document snippets that include the portion of the mobile image in which the document was located. The document snippets can also have additional processing performed on them, such as conversion to a bitonal image or to grayscale, depending on the types of testing to be performed.

**[0201]** After processing the mobile document image to generate a snippet, the MDIPE 2100 then selects one or more tests to be performed on the snippet or snippets (step 2315). In an embodiment, the tests to be performed can be selected from test data store 2132. In an embodiment, the MDIPE 2100 selects the one or more tests based on the processing parameters 2107 that were received with the mobile image 2105.

**[0202]** After selecting the tests from the test data store 2132, test parameters for each of the tests can be selected from the test parameters data store 2134 (step 2320). As described above, the test parameters can be used to configure or customize the tests to be performed.

**[0203]** Once the tests and the test parameters have been retrieved and provided to the test execution module 2130, a test is selected from tests to be executed, and the test is executed on the document snippet to produce a test result value (step 2325). In some embodiments, more than one document snippet can be used by a test. For example, a test can be performed that tests whether images of a front and back of a check are actually images of the same document can be performed. The test engine can receive both an image of the front of the check and an image of the back of the check from the preprocessing module 2110 and use both of these images when executing the test. Step 2325 can be repeated until each of the tests to be executed is performed.

**[0204]** The test result values obtained by executing each test on the snippet or snippets of the mobile document are then compared to test threshold with that test to determine whether the mobile image passes or fails the test (step 2330) and a determination can be made whether the mobile image of the check passed the test indicating that image quality of mobile image is acceptable (step 2335). If the mobile document image of the check passed, the MDIPE 2100 passes then executes one or more Check 21 tests on the snippets (step 2340).

**[0205]** The test result values obtained by executing the Check 21 test or tests on the snippet or snippets of the mobile document are then compared to test threshold with that test to determine whether the mobile image passes or fails the test (step 2345) and a determination can be made whether the mobile image of the check passed the test indicating that image quality of mobile image is acceptable under the requirements imposed by the Check 21 Act (step 2350). Step 345 can be repeated until each of the Check 21 tests is performed. If the mobile document image of the check passed, the MDIPE 2100 passes the snippet or snippets to the mobile application for further processing (step 2370).

**[0206]** If the mobile document image of the check failed one or more mobile IQA or Check 21 tests, detailed test result messages that explain why the image failed the test can be loaded from the test message data store 134 (step 2355) and the test result messages can be added to the

test results (step 2360). The test results and test messages are then output to the mobile application where they can be displayed to the user (step 2365). The user can use this information to retake the image of the check in an attempt to remedy some or all of the factors that caused the image of the check to be rejected.

### **Mobile IQA Tests**

[0207] Figs. 24A-41 illustrate various sample mobile document images and various testing methods that can be performed when assessing the image quality of a mobile document image. As described above, the preprocessing module 2110 can be configured to extract the document subimage, also referred to herein as the subimage, from the mobile document image. The subimage generally will be non-rectangular because of perspective distortion; however, the shape of the subimage can generally be assumed to be quadrangular, unless the subimage is warped. Therefore, the document can be identified by its four corners.

[0208] In some embodiments, a mobile IQA test generates a score for the subimage on a scale that ranges from 0-1000, where “0” indicates a subimage having very poor quality while a score of “1000” indicates that the image is perfect according to the test criteria.

[0209] Some tests use a geometrically corrected snippet of the subimage to correct view distortion. The preprocessing module 2110 can generate the geometrically corrected snippet. FIG. 24A illustrates a mobile image where the document captured in the mobile document image exhibits view distortion. FIG. 24B illustrates an example of a grayscale geometrically corrected subimage generated from the distorted image in FIG. 24A.

### **Image Focus IQA Test**

[0210] According to some embodiments, an Image Focus IQA Test can be executed on a mobile image to determine whether the image is too blurry to be used by a mobile application. Blurry images are often unusable, and this test can help to identify such out-of-focus images and reject them. The user can be provided detailed information to assist the user in taking a better quality image of the document. For example, the blurriness may have been the result of motion

blur caused by the user moving the camera while taking the image. The test result messages can suggest that the user hold the camera steadier when retaking the image.

[0211] Mobile devices can include cameras that have significantly different optical characteristics. For example, a mobile device that includes a camera that has an auto-focus feature can generally produce much sharper images than a camera that does not include such a feature. Therefore, the average image focus score for different cameras can vary widely. As a result, the test threshold can be set differently for different types of mobile devices. As described above, the processing parameters 2107 received by MDIPE 2100 can include information that identifies the type of mobile device and/or the camera characteristics of the camera used with the device in order to determine what the threshold should be set to for the Image Focus IQA Test.

[0212] An in-focus mobile document image, such as that illustrated in FIG. 25A will receive a score of 1000, while an out of focus document, such as that illustrated in FIG. 25B will receive a much lower score, such as in the 50-100 range. Most of the time, images are not completely out of focus. Therefore, a score of 0 is uncommon.

[0213] According to an embodiment, the focus of the image can be tested using various techniques, and the results can then be normalized to the 0-1000 scale used by the MDIPE 2100.

[0214] In an embodiment, the Image Focus Score can be computed using the following technique: The focus measure is a ratio of maximum video gradient between adjacent pixels, measured over the entire image and normalized with respect to image's gray level dynamic range and "pixel pitch." According to an embodiment, the image focus score can be calculated using the following equation described in "The Financial Services Technology Consortium," Image Defect Metrics, IMAGE QUALITY & USABILITY ASSURANCE: Phase 1 Project, Draft Version 1.0.4. May 2, 2005, which is hereby incorporated by reference:

$$\text{Image Focus Score} = (\text{Maximum Video Gradient}) / [(\text{Gray Level Dynamic Range}) * (\text{Pixel Pitch})]$$

$$\text{where Video Gradient} = \text{ABS} [(\text{Gray level for pixel "i"}) - (\text{Gray level for pixel "i+1"})]$$

*Gray Level Dynamic Range = [(Average of the “N” Lightest Pixels) – (Average of the “N” Darkest Pixels)]*

*Pixel Pitch = [ 1 / Image Resolution (in dpi)]*

[0215] The variable  $N$  is equal to the number of pixels used to determine the average darkest and lightest pixel gray levels in the image. According to one embodiment, the value of  $N$  is set to 64. Therefore, the 64 lightest pixels in the image are averaged together and the 64 darkest pixels in the image are averaged together, to compute the “Gray Level Dynamic” range value. The resulting image focus score value is multiplied by 10 in order to bring the value into the 0-1000 range used for the test results in the mobile IQA system.

[0216] The Image Focus Score determined using these techniques can be compared to an image focus threshold to determine whether the image is sufficiently in focus. As described above, the threshold used for each test may be determined at least in part by the processing parameters 2107 provided to MDIPE 2100. The Image Focus score can be normalized to the 0-1000 range used by the mobile IQA tests and compared to a threshold value associated with the test. If the Image Focus Score meets or exceeds this threshold, then the mobile document image is sufficiently focused for use with the mobile application.

### **Shadow Test**

[0217] According to some embodiments, a Shadow Test can be executed on a mobile image to determine whether a portion of the image is covered by a shadow. A shadow can render parts of a mobile image unreadable. This test helps to identify whether a shadow coverage a least a portion of a subimage in a mobile document image, and to reject images if the shadow has too much of an effect on the image quality, so that the user can attempt to take a better quality image of the document where the shadow is not present.

[0218] According to an embodiment, the presence of a shadow is measured by examining boundaries in the mobile image that intersect two or more sides of the document subimage. FIG. 26 illustrates an example of a shadowed document. The document subimage has been extracted

from the mobile document image and converted to a grayscale snippet in this example. The shadow boundary clearly intersects the top and the bottom of the check pictured in the snippet.

[0219] The presence of shadows can be measured using the area and contrast. If a shadow covers the entire image, the result is merely an image that is darker overall. Such shadows generally do not worsen image quality significantly. Furthermore, shadows having a very small surface area also do not generally worsen image quality very much.

[0220] According to an embodiment, the Image Shadowed Score can be calculated using the following formula to determine the score for a grayscale snippet:

*Image Shadowed score* = 1000 if no shadows were found,  
otherwise

*Image Shadowed score* = 1000 -  $\min(\text{Score}(S[i]))$ , where  
 $\text{Score}(S[i])$  is computed for every shadow  $S[i]$  detected on the  
grayscale snippet

[0221] In an embodiment, the Score for each shadow can be computed using the following formula:

Given shadow  $S[i]$  in the grayscale image, the score can be calculated  $\text{Score}(S[i])$  as  $\text{Score}(S[i]) = 2000 * \min(A[i]/A, 1 - A[i]/A) * (\text{Contrast}/256)$ , where  $A[i]$  is the area covered by shadow  $S[i]$  (in pixels),  $A$  is the entire grayscale snippet area (in pixels), and  $\text{Contrast}$  is the difference of brightness inside and outside of the shadow (the maximum value is 256).

[0222] Due to the normalization factor 2000,  $\text{Score}(S[i])$  fits into 0-1000 range. It tends to assume larger values for shadows that occupy about 1/2 of the snippet area and have high *contrast*.  $\text{Score}(S[i])$  is typically within 100-200 range. In an embodiment, the Image Shadowed score calculated by this test falls within a range of 0-1000 as do the test results from other tests. According to an embodiment, a typical mobile document image with few shadows will have a test result value in a range from 800-900. If no shadows are on are found the document subimage, then the score will equal 1000. The Image Shadowed score can then be compared to a threshold associated with the test to determine whether the image is of sufficiently high quality

for use with the mobile application requesting the assessment of the quality of the mobile document image.

### **Contrast Test**

**[0223]** According to some embodiments, a Contrast Test can be executed on a mobile image to determine whether the contrast of the image is sufficient for processing. One cause of poor contrast is images taken with insufficient light. A resulting grayscale snippet generated from the mobile document image can have low contrast, and if the grayscale snippet is converted to a binary image, the binarization module can erroneously white-out part of the foreground, such as the MICR-line of a check, the code line of a remittance coupon, an amount, or black-out part of the background. The Contrast Test measures the contrast and rejects poor quality images, and instructs the user to retake the picture under brighter light to improve the contrast of the resulting snippets.

**[0224]** FIG. 28 illustrates a method for executing a Contrast IQA Test according to an embodiment. The Contrast IQA Test illustrated in FIG. 28 is performed on a grayscale snippet generated from a mobile document image. The MDIPE 2100 receives the mobile image (step 2805) and generates a grayscale snippet that comprises a grayscale version of the document subimage (step 2810). FIG. 27 is an example of a grayscale snippet generated from a mobile document image of a check. As can be seen from FIG. 27, the contrast of the image is very low.

**[0225]** A histogram of the grayscale values in the grayscale snippet can then be built (step 2815). In an embodiment, the x-axis of the histogram is divided into bins that each represents a “color” value for the pixel in the grayscale image and the y-axis of the histogram represents the frequency of that color value in the grayscale image. According to an embodiment, the grayscale image has pixel in a range from 0-255, and the histogram is built by iterating through each value in this range and counting the number of pixels in the grayscale image having this value. For example, frequency of the “200” bin would include pixels having a gray value of 200.

**[0226]** A median black value can then be determined for the grayscale snippet (step 2820) and a median white value is also determined for the grayscale snippet (step 2825). The median



black and white values can be determined using the histogram that was built from the grayscale snippet. According to an embodiment, the median black value can be determined by iterating through each bin, starting with the “0” bin that represents pure black and moving progressively toward the “250” bin which represents pure white. Once a bin is found that includes at least 20% of the pixels included in the image, the median black value is set to be the color value associated with that bin. According to an embodiment, the median white value can be determined by iterating through each bin, starting with the “255” bin which represents pure white and moving progressively toward the “0” bin which represents pure black. Once a bin is found that includes at least 20% of the pixels included in the image, the median white value is set to be the color value associated with that bin.

[0227] Once the median black and white values have been determined, the difference between the median black and white values can then be calculated (step 2830). The difference can then be normalized to fall within the 0-1000 test range used in the mobile IQA tests executed by the MDIPE 2100 (step 2835). The test result value can then be returned (step 2840). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. See for example, FIG. 22, step 2230, described above. If the mobile image fails the Contrast IQA Test, the MDIPE 2100 can reject the image, and load detailed test messages from the test message data store 134 that include detailed instructions that how the user might retake the image.

### **Planar Skew Test**

[0228] According to some embodiments, a Planar Skew Test can be executed on a mobile image to determine whether the document subimage is skewed within the mobile image. See FIG. 29A for an example of a mobile document image that includes a remittance coupon or check that exhibits significant planar skew. Planar skew does not result in distortion of the document subimage; however, in an embodiment, the subimage detection module included in the preprocessing module assumes that the document subimage is nearly horizontal in the mobile document image. If the skew becomes too extreme, for example approaching 45 degrees from

horizontal, cropping errors could occur when the document subimage is extracted from the mobile document image.

[0229] According to an embodiment, document skew can be measured by first identifying the corners of the document subimage using one of the techniques described above. The corners of the documents subimage can be identified by the preprocessing module 130 when performing projective transformations on the subimage, such as that described above with respect to Figs. 24A and 24B. Various techniques for detecting the skew of the subimage can be used. For example, techniques for detecting skew disclosed in the related '071 and '091 Applications, can be used to detect the skew of the subimage. The results from the skew test can then be to fall within the 0-1000 test range used in the mobile IQA tests executed by the MDIPE 2100. The higher the skew of the document subimage, the lower the normalized test value. If the normalized test value falls below the threshold value associated with the test, the mobile document image can be rejected and the user can be provided detailed information from the test result messages data store 136 for how to retake the image and reduce the skew.

### **View Skew Test**

[0230] “View skew” denotes a deviation from direction perpendicular to the document in mobile document image. Unlike planar skew, the view skew can result in the document subimage having perspective distortion. FIG. 29B illustrates an example of a document subimage that exhibits view skew. View skew can cause problems in processing the subimage if the view skew becomes too great, because view skew changes the width-to-height ratio of the subimage. This can present a problem, since the true dimensions of the document pictured in the subimage are often unknown. For example, remittance coupons and business checks can be various sizes and can have different width-to-height ratios. View skew can result in content recognition errors, such as errors in recognition of the MICR-line data on a check or CAR/LAR recognition (which stands for Courtesy Amount Recognition and Legal Amount Recognition) or errors in recognition of the code line of a remittance coupon. By measuring the view skew, the view skew test can be used to reject images that have too much view skew, which can help

reduce false rejects and false accepts rates by addressing an issue that can be easily corrected by a user retaking the mobile document image.

[0231] FIG. 30 is a flow chart illustrating a method for testing for view skew according to an embodiment. The MDIPE 2100 receives the mobile image (step 3005) and identifies the corners of the document within the subimage (step 3010). A skew test score can then be determined for the document subimage (step 3015) and skew test score can then be returned (3040). As described above, the test result value can then be provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test.

[0232] According to an embodiment, the view skew of a mobile document can be determined using the following formula:

*View Skew score = 1000 - F(A, B, C, D), where*

$$F(A, B, C, D) = 500 * \max(\frac{\text{abs}(|AB| - |CD|)}{(|DA| + |BC|)}, \frac{\text{abs}(|BC| - |DA|)}{(|AB| + |CD|)}),$$

where  $|PQ|$  denotes the distance from point  $P$  to point  $Q$ , and the corners of the subimage are denoted as follows:  $A$  represents the top-left corner,  $B$  represents the top-right corner of the subimage,  $C$  represents the bottom-right corner of the subimage, and  $D$  represents the bottom-left corner of the subimage.

[0233] One can see that View Skew score can be configured to fit into  $[0, 1000]$  range used in the other mobile IQA tests described herein. In this example, the View Skew score is equal to 1000 when  $|AB| = |CD|$  and  $|BC| = |DA|$ , which is the case when there is no perspective distortion in the mobile document image and camera-to-document direction was exactly perpendicular. The View Skew score can then be compared to a threshold value associated with the test to determine whether the image quality is sufficiently high for use with the mobile application.

### **Cut Corner Test**

[0234] Depending upon how carefully the user framed a document when capturing a mobile image, it is possible that one or more corners of the document can be cut off in the mobile document image. As a result, important information can be lost from the document. For

example, if the lower left-hand corner of a check is cut off in the mobile image, a portion of the MICR-line of a check or the code line of a remittance coupon might be cut off, resulting in incomplete data recognition. FIG. 31 illustrates an example of a mobile document image that features a receipt where one of the corners has been cut off.

**[0235]** FIG. 32 illustrates a Cut-Off Corner Test that can be used with embodiments of the MDIPE 2100 for testing whether corners of a document in a document subimage have been cut off when the document was imaged. The mobile image including height and width parameters are received (step 3205). In an embodiment, the height and width of the mobile image can be determined by the preprocessing module 2110. The corners of the document subimage are then identified in the mobile document image (step 3210). Various techniques can be used to identify the corners of the image, including the various techniques described above. In an embodiment, the preprocessing module 2110 identifies the corners of the document subimage. As illustrated in FIG. 11, one or more of the corners of a document can be cut off. However, the preprocessing module 2110 can be configured to determine what the location of the corner should have been had the document not been cut off using the edges of the document in the subimage. FIG. 31 illustrates how the preprocessing module 2110 has estimated the location of the missing corner of the document by extending lines from the sides of the document out to the point where the lines intersect. The preprocessing module 2110 can then provide the corners information for the document to the test execution module 2130 to execute the Cut-Off Corner IQA Test. In an embodiment, test variables and the test results values to be returned by the test are set to default values: the test value  $V$  to be returned from the test is set to a default value of 1000, indicating that all of the corners of the document are within the mobile document image, and a maximum cut off variable (MaxCutOff) is set to zero indicating that no corner was cut off.

**[0236]** A corner of the document is selected (step 3220). In an embodiment, the four corners are received as an array of  $x$  and  $y$  coordinates  $C[I]$ , where  $I$  is equal to the values 1-4 representing the four corners of the document.

**[0237]** A determination is made whether the selected corner of the document is within the mobile document image (step 3225). The  $x$  &  $y$  coordinates of the selected corner should be at

or between the edges of the image. According to an embodiment, the determination whether a corner is within the mobile document image can be determined using the following criteria: (1)  $C[I].x \geq 0 \ \& \ C[I].x \leq Width$ , where  $Width$  = the width of the mobile document image and  $C[I].x$  = the x-coordinate of the selected corner; and (2)  $C[I].y \geq 0 \ \& \ C[I].y \leq Height$ , where  $Height$  = the height of the mobile document image and  $C[I].y$  = the y-coordinate of the selected corner.

**[0238]** If the selected corner fails to satisfy the criteria above, the corner is not within the mobile image and has been cut-off. A corner cut-off measurement is determined for the corner (step 3230). The corner cut-off measurement represents the relative distance to the edge of the mobile document image. According to an embodiment, the corner cut-off measurement can be determined using the following:

(1) Set  $H[I]$  and  $V[I]$  to zero, where  $H[I]$  represents the horizontal normalized cut-off measure and  $V[I]$  represents the vertical normalized cut-off measure.

(2) If  $C[I].x < 0$ , then set  $H[I] = -1000 * C[I].x / Width$

(3) If  $C[I].x > Width$ , set  $H[I] = 1000 * (C[I].x - Width) / Width$ , where  $Width$  is the width of the mobile image

(4) If  $C[I].y < 0$ , set  $V[I] = -1000 * C[I].y / Height$ , where  $Height$  is the height of the mobile image

(5) If  $C[I].y > Height$ , set  $V[I] = 1000 * (C[I].y - Height) / Height$

(6) Normalize  $H[I]$  and  $V[I]$  to fall within the 0-1000 range used by the mobile IQA tests by setting  $H[I] = \min(1000, H[I])$  and  $V[I] = \min(1000, V[I])$ .

(7) Set  $CutOff[I] = \min(H[I], V[I])$ , which is the normalized cut-off measure of the corner. One can see that the  $CutOff[I]$  lies within [0-1000] range used by the mobile IQA tests and the value increases as the corner moves away from mobile image boundaries.

**[0239]** An overall maximum cut-off value is also updated using the normalized cut-off measure of the corner (step 3235). According to an embodiment, the following formula can be

used to update the maximum cut-off value:  $MaxCutOff = \max(MaxCutOff, CutOff[I])$ . Once the maximum cut-off value is determined, a determination is made whether more corners are to be tested (step 3225).

**[0240]** If the selected corner satisfies the criteria above, the corner is within the mobile document image and is not cut-off. A determination is then made whether there are additional corners to be tested (step 3225). If there are more corners to be processed, a next corner to be test is selected (step 3215). Otherwise, if there are no more corners to be tested, the test result value for the test is computing using the maximum test cut-off measurement. In an embodiment, the test result value  $V = 1000 - MaxCutOff$ . One can see that  $V$  lies within [0-1000] range for the mobile IQA tests and is equal to 1000 when all the corners are inside the mobile image and decreases as one or more corner move outside of the mobile image.

**[0241]** The test result value is then returned (3245). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. If the test result value falls below the threshold associated with the test, detailed test result messages can be retrieved from the test result message data store 136 and provided to the user to indicate why the test failed and what might be done to remedy the test. The user may simply need to retake the image with the document corners within the frame.

### **Cut-Side Test**

**[0242]** Depending upon how carefully the user framed a document when capturing a mobile image, it is possible that one or more sides of the document can be cut off in the mobile document image. As a result, important information can be lost from the document. For example, if the bottom a check is cut off in the mobile image, the MICR-line might be cut off, rendering the image unusable for a Mobile Deposit application that uses the MICR information to electronically deposit checks. Furthermore, if the bottom of a remittance coupon is cut off in the mobile image, the code line may be missing, the image may be rendered unusable by a Remittance Processing application that uses the code information to electronically process the remittance.

[0243] FIG. 33 illustrates an example of a mobile document image that features a receipt where one of the ends of the receipt has been cut off in the image. Unlike the Cut-Corner Test described above which can be configured to allow a document to pass if the amount of cut-off falls is small enough that the document image still receives a test score that meets or exceeds the threshold associated with the test, the Cut-Side Test is either pass or fail. If one or more sides of the document subimage are cut off in the mobile document image, the potential to lose critical information is too high, and mobile document is marked as failing.

[0244] FIG. 34 is a flow diagram of a method for determining whether one or more sides of the document are cut off in the document subimage according to an embodiment. The mobile image is received (step 3405). In an embodiment, the height and width of the mobile image can be determined by the preprocessing module 2110. The corners of the document subimage are then identified in the mobile document image (step 3410). Various techniques can be used to identify the corners of the image, including the various techniques described above. In an embodiment, the preprocessing module 2110 identifies the corners of the document subimage.

[0245] A side of the document is selected (step 3420). In an embodiment, the four corners are received as an array of x and y coordinates  $C[I]$ , where  $I$  is equal to the values 1-4 representing the four corners of the document.

[0246] A determination is made whether the selected corner of the document is within the mobile document image (step 3425). According to an embodiment, the document subimage has four side and each side  $S[I]$  includes two adjacent corners  $C1[I]$  and  $C2[I]$ . A side is deemed to be cut-off if the corners comprising the side are on the edge of the mobile image. In an embodiment, a side of the document is cut-off if any of the following criteria are met:

- (1)  $C1[I].x = C2[I].x = 0$ , where  $x$  = the x-coordinate of the corner
- (2)  $C1[I].x = C2[I].x = Width$ , where  $Width$  = the width of the mobile image
- (3)  $C1[I].y = C2[I].y = 0$ , where  $y$  = the y-coordinate of the corner
- (4)  $C1[I].y = C2[I].y = Height$ , where  $Height$  = the height of the mobile image

[0247] If the side does not fall within the mobile image, the test result value is set to zero indicating that the mobile image failed the test (step 3430), and the test results are returned (step 3445).

[0248] If the side falls within the mobile image, a determination is made whether there are more sides to be tested (step 3425). If there are more sides to be tested, an untested side is selected (step 3415). Otherwise, all of the sides were within the mobile image, so the test result value for the test is set to 1000 indicating the test passed (step 3440), and the test result value is returned (step 3445).

### **Warped Image Test**

[0249] The warped image test identifies images where document is warped. FIG. 35 illustrates an example of a mobile document image where the document is warped. In some embodiments, the preprocessing module 2110 can be configured to include de-warping functionality for correcting warped images. However, in some embodiments, a Warped Image Test is provided to detect and reject warped images. One solution for correcting warped images is to instruct the user to retake the image after flattening the hardcopy of the document being imaged.

[0250] FIG. 36 is a flow diagram of a method for identifying a warped image and for scoring the image based on how badly the document subimage is warped according to an embodiment. A warped image test score value is returned by the test, and this value can be compared with a threshold value by the test execution module 2130 to determine whether the image warping is excessive.

[0251] The mobile image is received (step 3605). In an embodiment, the height and width of the mobile image can be determined by the preprocessing module 2110. The corners of the document subimage are then identified in the mobile document image (step 3610). Various techniques can be used to identify the corners of the image, including the various techniques described above. In an embodiment, the preprocessing module 2110 identifies the corners of the document subimage.



[0252] A side of the document is selected (step 3615). According to an embodiment, the document subimage has four side and each side  $S[I]$  includes two adjacent corners  $C1[I]$  and  $C2[I]$ .

[0253] A piecewise linear approximation is built for the selected side (step 3620). According to an embodiment, the piecewise-linear approximation is built along the selected side by following the straight line connecting the adjacent corners  $C1[I]$  and  $C2[I]$  and detecting position of the highest contrast starting from any position within  $[C1[I], C2[I]]$  segment and moving in orthogonal direction.

[0254] After the piecewise linear approximation is built along the  $[C1[I], C2[I]]$  segment, the  $[C1[I], C2[I]]$  segment is walked to compute the deviation between the straight line and the approximation determined using piecewise linear approximation (step 3625). Each time the deviation is calculated, a maximum deviation value ( $MaxDev$ ) is updated to reflect the maximum deviation value identified during the walk along the  $[C1[I], C2[I]]$  segment.

[0255] The maximum deviation value for the side is then normalized to generate a normalized maximized deviation value for the selected size of the document image (step 3630). According to an embodiment, the normalized value can be determined using the following formula:

$$NormMaxDev[I] = 1000 * MaxDev[I] / Dim, \text{ where } Dim \text{ is the mobile image dimension perpendicular to side } S[I].$$

[0256] An overall normalized maximum deviation value is then updated using the normalized deviation value calculated for the side. According to an embodiment, the overall maximum deviation can be determined using the formula:

$$OverallMaxDeviation = \max(OverallMaxDeviation, NormMaxDev[I])$$

[0257] A determination is then made whether there are anymore sides to be tested (step 3640). If there are more sides to be tested, an untested side is selected for testing (step 3615).

Otherwise, if no untested sides remain, the warped image test value is computed. According to an embodiment, the warped image test value can be determined using the following formula:

$$V = 1000 - OverallMaxDeviation$$

[0258] One can see that  $V$  lies within [0-1000] range used by the image IQA system and is equal to 1000 when the sides  $S[I]$  are straight line segments (and therefore no warp is present). The computed test result is then returned (step 3650). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. If the test result value falls below the threshold associated with the test, detailed test result messages can be retrieved from the test result message data store 136 and provided to the user to indicate why the test failed and what might be done to remedy the test. For example, the user may simply need to retake the image after flattening out the hardcopy of the document being imaged in order to reduce warping.

### **Image Size Test**

[0259] The Image Size Test detects the actual size and the effective resolution of the document subimage. The perspective transformation that can be performed by embodiments of the preprocessing module 2110 allows for a quadrangle of any size to be transformed into a rectangle to correct for view distortion. However, a small subimage can cause loss of detail needed to process the subimage.

[0260] FIG. 37 illustrates an example of a document subimage within a mobile document image that is relatively small. Small size of the subimage can cause the loss of important foreground information. This effect is similar to digital zooming in a digital camera where image of an object becomes larger, but the image quality of object can significantly degrade due to loss of resolution and important details can be lost.

[0261] FIG. 38 is a flow diagram of a process that for performing an Image Size Test on a subimage according to an embodiment. The mobile image is received (step 3805). In an embodiment, the height and width of the mobile image can be determined by the preprocessing module 2110. The corners of the document subimage are then identified in the mobile document

image (step 3810). Various techniques can be used to identify the corners of the image, including the various techniques described above. In an embodiment, the preprocessing module 2110 identifies the corners of the document subimage. In the method the corners of the subimage are denoted as follows: *A* represents the top-left corner, *B* represents the top-right corner of the subimage, *C* represents the bottom-right corner of the subimage, and *D* represents the bottom-left corner of the subimage.

**[0262]** A subimage average width is computed (step 3815). In an embodiment, the subimage average width can be calculated using the following formula:

$$\text{Subimage average width as } AveWidth = (|AB| + |CD|)/2, \text{ where } |PQ| \text{ represents the Euclidian distance from point } P \text{ to point } Q.$$

**[0263]** A subimage average height is computed (step 3820). In an embodiment, the subimage average height can be calculated using the following formula:

$$AveHeight = (|BC| + |DA|)/2$$

**[0264]** The average width and average height values are then normalized to fit the 0-1000 range used by the mobile IQA tests (step 3822). The following formulas can be used determine the normalize the average width and height:

$$\text{NormAveWidth} = 1000 * AveWidth/Width$$

$$\text{NormAveHeight} = 1000 * AveHeight/Height$$

**[0265]** A minimum average value is then determined for the subimage (step 3825). According to an embodiment, the minimum average value is the smaller of the normalized average width and the normalized average height values. The minimum average value falls within the 0-1000 range used by the mobile IQA tests. The minimum average value will equal 1000 if the document subimage fills the entire mobile image.

**[0266]** The minimum average value is returned as the test result (step 3865). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. If the test result value falls

below the threshold associated with the test, detailed test result messages can be retrieved from the test result message data store 2136 and provided to the user to indicate why the test failed and what might be done to remedy the test. For example, the user may simply need to retake the image by positioning the camera closer to the document.

### **MICR-line Test**

[0267] The MICR-line Test is used to determine whether a high quality image of a check front has been captured using the mobile device according to an embodiment. The MICR-line Test can be used in conjunction with a Mobile Deposit application to ensure that images of checks captures for processing with the Mobile Deposit information are of a high enough quality to be processed so that the check can be electronically deposited. Furthermore, if a mobile image fails the MICR-line Test, the failure may be indicative of incorrect subimage detections and/or poor overall quality of the mobile image, and such an image should be rejected anyway.

[0268] FIG. 39A is a flow chart of a method for executing a MICR-line Test according to an embodiment. A mobile image is received (step 3905) and a bitonal image is generated from the mobile image (step 3910). In an embodiment, preprocessor 110 extracts the document subimage from the mobile image as described above, including preprocessing such as geometric correction. The extracted subimage can then be converted to a bitonal snippet by the preprocessor 110. The MICR line is then identified in the bitonal snippet (step 3915). According to an embodiment, a MICR recognition engine is then applied to identify the MICR-line and to compute character-level and overall confidence values for the image (step 3920). These confidences can then be normalized to the 0-1000 scale used by the mobile IQA tests where 1000 means high quality and 0 means poor MICR quality. The confidence level is then returned (step 3925). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. If the test result value falls below the threshold associated with the test, detailed test result messages can be retrieved from the test result message data store 136 and provided to the user to indicate why the test failed and what might be done to remedy the test. For example, the user may simply need to retake the image to adjust for geometrical or other factors, such as poor lighting or a shadowed document.

In some instances, the user may not be able to correct the errors. For example, if the MICR line on the document is damaged or incomplete and the document will continue to fail the test even if the image were retaken.

### **Code Line Test**

[0269] The Code Line Test can be used to determine whether a high quality image of a remittance coupon front has been captured using the mobile device according to an embodiment. The Code Line Test can be used in conjunction with a Remittance Processing application to ensure that images of remittance coupon captures for processing with the Remittance Processing information are of a high enough quality to be processed so that the remittance can be electronically processed. Furthermore, if a mobile image fails the Code Line Test, the failure may be indicative of incorrect subimage detections and/or poor overall quality of the mobile image, and such an image should be rejected anyway.

[0270] FIG. 39B is a flow chart of a method for executing a Code Line Test according to an embodiment. A mobile image of a remittance coupon is received (step 3955) and a bitonal image is generated from the mobile image (step 3960). In an embodiment, preprocessor 110 extracts the document subimage from the mobile image as described above, including preprocessing such as geometric correction. The extracted subimage can then be converted to a bitonal snippet by the preprocessor 110. The code line is then identified in the bitonal snippet (step 3965). According to an embodiment, a code line recognition engine is then applied to identify the code line and to compute character-level and overall confidence values for the image (step 3970). These confidences can then be normalized to the 0-1000 scale used by the mobile IQA tests where 1000 means high quality and 0 means poor code line quality. The confidence level is then returned (step 3975). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. If the test result value falls below the threshold associated with the test, detailed test result messages can be retrieved from the test result message data store 136 and provided to the user to indicate why the test failed and what might be done to remedy the test. For example, the user may simply need to retake the image to adjust for geometrical or other

factors, such as poor lighting or a shadowed document. In some instances, the user may not be able to correct the errors. For example, if the code line on the document is damaged or incomplete and the document will continue to fail the test even if the image were retaken.

### **Aspect Ratio Tests**

[0271] The width of a remittance coupon is typically significantly longer than the height of the document. According to an embodiment, an aspect ratio test can be performed on a document subimage of a remittance coupon to determine whether the aspect ratio of the document in the image falls within a predetermined ranges of ratios of width to height. If the document image falls within the predetermined ranges of ratios, the image passes the test. An overall confidence value can be assigned to different ratio values or ranges of ratio values in order to determine whether the image should be rejected.

[0272] According to some embodiments, the mobile device can be used to capture an image of a check in addition to the remittance coupon. A second aspect ratio test is provided for two-sided documents, such as checks, where images of both sides of the document may be captured. According to some embodiments, a remittance coupon can also be a two-sided document and images of both sides of the document can be captured. The second aspect ratio test compares the aspect ratios of images that are purported to be of the front and back of a document to determine whether the user has captured images of the front and back of the same document according to an embodiment. The Aspect Ratio Test could be applied to various types two-sided or multi-page documents to determine whether images purported to be of different pages of the document have the same aspect ratio.

[0273] FIG. 40 illustrates a method for executing an Aspect Ratio Test for two-sided documents according to an embodiment. In the embodiment illustrated in FIG. 40, the test is directed to determining whether the images purported to be of the front and back side of a document have the same aspect ratio. However, the method could also be used to test whether two images purported to be from a multi-page and/or multi-sided document have the same aspect ratio.

[0274] A front mobile image is received (step 4005) and a rear mobile image is received (step 4010). The front mobile image is supposed to be of the front side of a document while the rear mobile image is supposed to be the back side of a document. If the images are really of opposite sides of the same document, the aspect ratio of the document subimages should match. Alternatively, images of two different pages of the same document may be provided for testing. If the images are really of pages of the same document, the aspect ratio of the document subimages should match.

[0275] The preprocessing module 2110 can process the front mobile image to generate a front-side snippet (step 4015) and can also process the back side image to generate a back-side snippet (step 4020).

[0276] The aspect ratio of the front-side snippet is then calculated (step 4025). In an embodiment, the  $AspectRatioFront = Width / Height$ , where  $Width$  = the width of the front-side snippet and  $Height$  = the height of the front-side snippet.

[0277] The aspect ratio of the back-side snippet is then calculated (step 4030). In an embodiment, the  $AspectRatioBack = Width / Height$ , where  $Width$  = the width of the back-side snippet and  $Height$  = the height of the back-side snippet.

[0278] The relative difference between the aspect ratios of the front and rear snippets is then determined (step 4035). According to an embodiment, the relative difference between the aspect ratios can be determined using the following formula:

$$RelDiff = 1000 * \frac{abs(AspectRatioFront - AspectRatioBack)}{\max(AspectRatioFront, AspectRatioBack)}$$

[0279] A test result value is then calculated based on the relative difference between the aspect ratios (step 4040). According to an embodiment, the test value  $V$  can be computed using the formula  $V = 1000 - RelDiff$ .

[0280] The test results are then returned (step 4045). As described above, the test result value is provided to the test execution module 2130 where the test result value can be compared to a threshold value associated with the test. If the test result value falls below the threshold

associated with the test, detailed test result messages can be retrieved from the test result message data store 136 and provided to the user to indicate why the test failed and what might be done to remedy the test. For example, the user may have mixed up the front and back images from two different checks having two different aspect ratios. If the document image fails the test, the user can be prompted to verify that the images purported to be the front and back of the same document (or images of pages from the same document) really are from the same document.

### **Front-as-Rear Test**

[0281] FIG. 41 illustrates a method for performing a front-as-rear test on a mobile document image. The front-as-rear test can be adapted to be performed on images purported to be the front and back side of a check. The Front-as-Rear Test can be used to determine whether an image that is purported to be the back of a check is actually an image of the front of the document according to an embodiment. The Front-as-Rear Test can be used in embodiments where a user has captured an image of a check to be processed as payment for a remittance.

[0282] The Front-as-Rear Test is a check specific Boolean test. The test returns a value of 0 if an image fails the test and a value of 1000 if an image passes the test. According to an embodiment, if a MICR-line is identified on what is purported to be an image of the back of the check, the image will fail the test and generate a test message that indicates that the images of the check have been rejected because an image of the front of the check was mistakenly passed as an image of the rear of the check. Similarly, if a code line is identified on what is purported to be the back of a remittance coupon, the image will fail the test and generate a test message that indicates that the images of the remittance coupon have been rejected because an image of the front of the coupon was mistakenly passed as an image of the rear of the coupon.

[0283] An image of the rear of the document is received (step 4105) and the image is converted to a bitonal snippet by preprocessor 110 of the MDIPE 2100 (step 4110). The image may be accompanied by data indicating whether the image is of a check or of a remittance coupon. In some embodiments, no identifying information may be provided, and the testing will be performed to identify either a code line or an MICR line in the bitonal snippet.



[0284] If the document is identified as a check, a MICR recognition engine can then be applied to identify a MICR-line in the bitonal snippet (step 4115). Various techniques for identifying the MICR-line in an image of a check are described above. The results from the MICR recognition engine can then be normalized to the 0-1000 scale used by the mobile IQA tests, and the normalized value compared to a threshold value associated with the test. If the document is identified as a remittance coupon, a code line recognition engine can be applied to identify the code line in the image of the coupon. Various techniques for identifying the code line in an image of a remittance coupon are described above, such as identifying text in OCR-A font within the image. If no information as to whether the image to be tested includes a check or a remittance coupon is provided, both MICR-line and code line testing can be performed to see if either a MICR-line or code line can be found. In an embodiment, the highest normalized value from the MICR-line and code line tests can be selected for comparison to the threshold.

[0285] According to an embodiment, the test threshold can be provided as a parameter to the test along with the with mobile document image to be tested. According to an embodiment, the threshold used for this test is lower than the threshold used in the MICR-line Test described above.

[0286] If the normalized test result equals or exceeds the threshold, then the image includes an MICR-line or code line and the test is marked as failed (test result value = 0), because a MICR-line or code line was identified in what was purported to be an image of the back of the document. If the normalized test result is less than the threshold, the image did not include a MICR line and the test is marked as passed (test result value = 1000). The test results value is then returned (step 4125).

### **Form Identification**

[0287] Various embodiments of the present invention may utilize a novel technique of form identification in order to expeditiously identify key features of a captured mobile image. The form identification can be provided by a user, or it can be automatically determined by reading a captured mobile image. This captured mobile image may include any type of document including, without limitation: remittance coupons, employment forms, store receipts, checks,

bills or sales invoices, business cards, medical and dental records, store coupons, educational information such as progress reports and report cards, birth and death certificates, insurance policies, legal documents, magazine and newspaper clippings, forms of personal identification such as passports and driver licenses, police records, real estate records, etc. In the form identification step, a template is identified that is associated with a document that has been captured in a mobile image. The template identifies the layout of information contained within the document. This layout information can be used to improve data capture accuracy because data should be in known locations on the document.

**[0288]** Form identification can be helpful in a number of different situations. If the layout of the document is known, capturing the data from known locations on the document can be more accurate than relying on a dynamic data capture technique to extract the data from the document. Additionally, according to some embodiments, the identification of a prerequisite minimum number of data fields associated with only one type of document can enable a faster lookup of data from other data fields as soon as the specific type of document has been identified.

**[0289]** Form identification can also be used for documents that lack keywords that could otherwise be used to identify key data on the document. For example, if a document does not include an “Account Number” label for an account number field, the dynamic data capture may misidentify the data in that field. Misidentification can become even more likely if multiple fields have similar formats. Form identification can also be used for documents having ambiguous data. For example, a document might include multiple fields that include data having a similar format. If a document includes multiple unlabeled fields having similar formats, dynamic data capture may be more likely to misidentify the data. However, if the layout of the document is known, the template information can be used to extract data from known positions in the document image.

**[0290]** According to some embodiments, form identification can also be used for documents having a non-OCR friendly layout. For example, a document may use fonts where identifying keywords and/or form data is printed using a non-OCR friendly font. Form identification can

also be used to improve the chance of correctly capturing data when a poor quality image is presented. A poor quality image of a document can make it difficult to locate and/or read data.

[0291] FIG 42 is a flow chart of a method for processing an image using form identification according to an embodiment. At step 4205, a binarized/bi-tonal document image is received. Various techniques for creating a bi-tonal subimage from a mobile image are provided above. For example, step 625 of FIG. 6 describes binarization of a document subimage. FIG. 10 also illustrates a method of binarization that can be used to generate a bi-tonal image according to one embodiment of the present invention.

[0292] A matching algorithm is executed on the bi-tonal image of the document in an attempt to find a matching template (step 4210). According to an embodiment, one or more computing devices can include a template data store that can be used to store templates of the layouts of various types of documents. Various matching techniques can be used to match a template to a document image. For example, optical character recognition can be used to identify and read text content from the image. The types of data identified and the positions of the data on the document can be used to identify a matching template. According to another embodiment, a document can include a unique symbol or identifier that can be matched to a particular document template. In yet other embodiments, the image of the document can be processed to identify “landmarks” on the image that may correspond to labels and/or data. In some embodiments, these landmarks can include, but are not limited to: positions of horizontal and/or vertical lines on the document, the position and/or size of boxes and/or frames on the document, and/or the location of pre-printed text. The position of these landmarks on the document may be used to identify a template from the plurality of templates in the template data store. According to some embodiments, a cross-correlation matching technique can be used to match a template to an image of a document. In some embodiments, the positions of frames/boxes found on image and/or other such landmarks, can be cross-correlated with landmark information associated a template to compute the matching confidence score. If the confidence score exceeds a predetermined threshold, the template is considered to be a match and can be selected for use in extracting information from the mobile image.

[0293] A determination is made whether a matching template has been found (step 4215). If no matching template is found, a dynamic data capture can be performed on the image of the document (step 4225). Dynamic data capture is described in detail below and an example method for dynamic data capture is illustrated in the flow chart of FIG. 43.

[0294] If a matching template is found, data can be extracted from the image of the document using the template (step 4220). The template can provide the location of various data within the document, such as the document's author(s), the document's publication date, the names of any corporate, governmental, or educational entities associated with the document, an amount due, an account holder name, an account number, a payment due date, etc. In some embodiments, various OCR techniques can be used to read text content from the locations specified by the template. Since the location of various data elements is known, ambiguities regarding the type of data found can be eliminated. That is, use of the template enables the system to distinguish among data elements which have a similar data type.

### **Dynamic Data Capture**

[0295] FIG 43 is a flow chart of a dynamic data capture method for extracting data from an image according to an embodiment. The dynamic data capture method illustrated in FIG. 43 can be used if a form ID for identifying a particular format of a document is not available. The method illustrated in FIG. 43 can also be used if the form ID does not match any of the templates stored in the template data store. The method begins with receiving a binarized/bi-tonal document image (step 4305). Various optical character recognition techniques can then be used to locate and read fields from the bi-tonal image (step 4310). Some example OCR techniques are described below. Once data fields have been located, the data can be extracted from the bi-tonal image (step 4315). In some embodiments, steps 4310 and 4315 can be combined into a single step where the field data is located and the data extracted in a combined OCR step. Once the data has been extracted from the image, the data can be analyzed to identify what data has been extracted (step 4320). The data can also be analyzed to determine whether any additional data is required in order to be able to process the image.

**[0296]** According to an embodiment, a keyword-based detection technique can be used to locate and read the data from the bi-tonal image in steps 4310 and 4315 of the method of FIG. 43. The method uses a set of field-specific keywords to locate fields of interest in the bitonal image. For example, if the captured image is an image of a remittance coupon, the keywords “Account Number,” “Account #,” “Account No.,” “Customer Number,” and/or other variations can be used to identify the customer’s account number. According to an embodiment, text located proximate to the keyword can be associated with the keyword. For example, text located within a predetermined distance to the right of or below an “Account Number” keyword may be identified and extracted from the image using OCR and the text found in this location can then be treated as the account number. According to an embodiment, the distance and directions in relation to the keyword in which the field data can be located can be configured based on the various parameters, such as locale or language. The position of the keyword in relation to field that includes the data associated with the keyword may vary based on the language being used, e.g. written right to left versus left to right.

**[0297]** According to an embodiment, a format-based detection technique can be used to locate and read the data from the bi-tonal image in steps 4310 and 4315. For example, an OCR technique can be used to recognize text in the document image. A regular expression mechanism can then be applied to the text extracted from the bitonal image. A regular expression can be used to formalize the format description for a particular field, such as “contains 7-12 digits,” “may start with 1 or 2 uppercase letters,” or “contains the letter “U” in the second position.” According to an embodiment, multiple regular expressions may be associated with a particular field, such as an account number, in order to increase the likelihood of a correct match.

**[0298]** According to yet another embodiment, a combination of keyword-based and format-based matching can be used to identify and extract field data from the bi-tonal image (steps 4310 and 4315). This approach can be particularly effective where multiple fields of the same or similar format are included within the image. A combination of keyword-based and format-based matching can be used to identify field data can be used to disambiguate the data extracted from the bi-tonal image.

[0299] According to an embodiment, a code-line validation technique can be used to locate and read the data from the bi-tonal image of in steps 4310 and 4315. One or more fields may be embedded into a code-line. In some embodiments, the code-line characters may be cross-checked against fields recognized in other parts of the document. In the event that a particular field is different from a known corresponding value in the code line, the value in the code line may be selected over the field value due to the relative difference in the reliabilities of reading the code line versus reading the field value.

[0300] According to an embodiment, a cross-validation technique can be used where multiple bi-tonal images of the same document have been captured, and one or more OCR techniques are applied to each of the bi-tonal images (such as by any of the techniques described above). The results from the one or more OCR technique from one bi-tonal image can be compared to the results of OCR techniques applied to one or more other bitonal images in order to cross-validate the field data extracted from the images. If conflicting results are found, a set of results having a higher confidence value can be selected to be used for document image processing.

### **Exemplary Hardware Embodiments**

[0301] FIG. 44 is an exemplary embodiment of a mobile device 4400 according to an embodiment. Mobile device 4400 can be used to implement the mobile device 340 of Fig. 3. Mobile device 4200 includes a processor 4410. The processor 4410 can be a microprocessor or the like that is configurable to execute program instructions stored in the memory 4420 and/or the data storage 4440. The memory 4420 is a computer-readable memory that can be used to store data and or computer program instructions that can be executed by the processor 4410. According to an embodiment, the memory 4420 can comprise volatile memory, such as RAM and/or persistent memory, such as flash memory. The data storage 4440 is a computer readable storage medium that can be used to store data and or computer program instructions. The data storage 4440 can be a hard drive, flash memory, a SD card, and/or other types of data storage.

[0302] The mobile device 4400 also includes an image capture component 4430, such as a digital camera. According to some embodiments, the mobile device 4400 is a mobile phone, a

smart phone, or a PDA, and the image capture component 4430 is an integrated digital camera that can include various features, such as auto-focus and/or optical and/or digital zoom. In an embodiment, the image capture component 4430 can capture image data and store the data in memory 4220 and/or data storage 4440 of the mobile device 4400.

**[0303]** Wireless interface 4450 of the mobile device can be used to send and/or receive data across a wireless network. For example, the wireless network can be a wireless LAN, a mobile phone carrier's network, and/or other types of wireless network.

**[0304]** I/O interface 4460 can also be included in the mobile device to allow the mobile device to exchange data with peripherals such as a personal computer system. For example, the mobile device might include a USB interface that allows the mobile to be connected to USB port of a personal computer system in order to transfers information such as contact information to and from the mobile device and/or to transfer image data captured by the image capture component 4430 to the personal computer system.

**[0305]** As used herein, the term module might describe a given unit of functionality that can be performed in accordance with one or more embodiments of the present invention. As used herein, a module might be implemented utilizing any form of hardware, software, or a combination thereof. For example, one or more processors, controllers, ASICs, PLAs, logical components, software routines or other mechanisms might be implemented to make up a module. In implementation, the various modules described herein might be implemented as discrete modules or the functions and features described can be shared in part or in total among one or more modules. In other words, as would be apparent to one of ordinary skill in the art after reading this description, the various features and functionality described herein may be implemented in any given application and can be implemented in one or more separate or shared modules in various combinations and permutations. Even though various features or elements of functionality may be individually described or claimed as separate modules, one of ordinary skill in the art will understand that these features and functionality can be shared among one or more common software and hardware elements, and such description shall not require or imply that separate hardware or software components are used to implement such features or functionality.

**[0306]** Where components or modules of processes used in conjunction with the operations described herein are implemented in whole or in part using software, in one embodiment, these software elements can be implemented to operate with a computing or processing module capable of carrying out the functionality described with respect thereto. One such example-computing module is shown in FIG. 43. Various embodiments are described in terms of this example-computing module 1900. After reading this description, it will become apparent to a person skilled in the relevant art how to implement the invention using other computing modules or architectures.

**[0307]** Referring now to FIG. 43, computing module 1900 may represent, for example, computing or processing capabilities found within desktop, laptop and notebook computers; mainframes, supercomputers, workstations or servers; or any other type of special-purpose or general-purpose computing devices as may be desirable or appropriate for a given application or environment. Computing module 1900 might also represent computing capabilities embedded within or otherwise available to a given device. For example, a computing module might be found in other electronic devices. Computing module 1900 might include, for example, one or more processors or processing devices, such as a processor 1904. Processor 1904 might be implemented using a general-purpose or special-purpose processing engine such as, for example, a microprocessor, controller, or other control logic.

**[0308]** Computing module 1900 might also include one or more memory modules, referred to as main memory 1908. For example, random access memory (RAM) or other dynamic memory might be used for storing information and instructions to be executed by processor 1904. Main memory 1908 might also be used for storing temporary variables or other intermediate information during execution of instructions by processor 1904. Computing module 1900 might likewise include a read only memory (“ROM”) or other static storage device coupled to bus 1902 for storing static information and instructions for processor 1904.

**[0309]** The computing module 1900 might also include one or more various forms of information storage mechanism 1910, which might include, for example, a media drive 1912 and a storage unit interface 1920. The media drive 1912 might include a drive or other mechanism to



support fixed or removable storage media 1914. For example, a hard disk drive, a floppy disk drive, a magnetic tape drive, an optical disk drive, a CD or DVD drive (R or RW), or other removable or fixed media drive. Accordingly, storage media 1914 might include, for example, a hard disk, a floppy disk, magnetic tape, cartridge, optical disk, a CD or DVD, or other fixed or removable medium that is read by, written to or accessed by media drive 1912. As these examples illustrate, the storage media 1914 can include a computer usable storage medium having stored therein particular computer software or data.

**[0310]** In alternative embodiments, information storage mechanism 1910 might include other similar instrumentalities for allowing computer programs or other instructions or data to be loaded into computing module 1900. Such instrumentalities might include, for example, a fixed or removable storage unit 1922 and an interface 1920. Examples of such storage units 1922 and interfaces 1920 can include a program cartridge and cartridge interface, a removable memory (for example, a flash memory or other removable memory module) and memory slot, a PCMCIA slot and card, and other fixed or removable storage units 1922 and interfaces 1920 that allow software and data to be transferred from the storage unit 1922 to computing module 1900.

**[0311]** Computing module 1900 might also include a communications interface 1924. Communications interface 1924 might be used to allow software and data to be transferred between computing module 1900 and external devices. Examples of communications interface 1924 might include a modem or softmodem, a network interface (such as an Ethernet, network interface card, WiMedia, IEEE 802.XX or other interface), a communications port (such as for example, a USB port, IR port, RS232 port Bluetooth® interface, or other port), or other communications interface. Software and data transferred via communications interface 1924 might typically be carried on signals, which can be electronic, electromagnetic (which includes optical) or other signals capable of being exchanged by a given communications interface 1924. These signals might be provided to communications interface 1924 via a channel 1928. This channel 1928 might carry signals and might be implemented using a wired or wireless communication medium. These signals can deliver the software and data from memory or other storage medium in one computing system to memory or other storage medium in computing system 1900. Some examples of a channel might include a phone line, a cellular link, an RF

link, an optical link, a network interface, a local or wide area network, and other wired or wireless communications channels.

[0312] Computing module 1900 might also include a communications interface 1924. Communications interface 1924 might be used to allow software and data to be transferred between computing module 1900 and external devices. Examples of communications interface 1924 might include a modem or softmodem, a network interface (such as an Ethernet, network interface card, WiMAX, 802.XX or other interface), a communications port (such as for example, a USB port, IR port, RS232 port, Bluetooth interface, or other port), or other communications interface. Software and data transferred via communications interface 1924 might typically be carried on signals, which can be electronic, electromagnetic, optical or other signals capable of being exchanged by a given communications interface 1924. These signals might be provided to communications interface 1924 via a channel 1928. This channel 1928 might carry signals and might be implemented using a wired or wireless medium. Some examples of a channel might include a phone line, a cellular link, an RF link, an optical link, a network interface, a local or wide area network, and other wired or wireless communications channels.

### **Recurring Payment Scheduling**

[0313] According to various embodiments, a user of a remittance application (see, e.g., FIGs. 46A-46F) can set up one or more recurring payment schedules. A recurring payment schedule may have a variety of advantages over a series of single payments, including: i.) utilizing persistent data in order to make the process of paying a bill more expeditious for the user (i.e., less input may be required from the user before each bill is submitted), ii.) enabling a fast lookup of a remittance coupon template associated with a specified payee (thereby decreasing search time); and iii.) enabling the remittance application to send one or more payment reminders to the user so as to safeguard against a payment default.

[0314] FIG. 47 is a block diagram illustrating an exemplary method of configuring a recurring bill payment schedule according to one embodiment of the present invention. At block 4702, a user launches a remittance application. In some embodiments, the remittance application

is resident within the mobile device 340 (see FIG. 3). In other embodiments, the remittance application is resident within a remote computing device, such as a mobile remittance server 310 (see FIG. 3). Once the remittance application is launched, a splash screen may appear (block 4704) indicating the name and/or software version of the remittance application (FIG. 46A).

**[0315]** At block 4706, a login screen can then be displayed, prompting the user to input one or more security credentials (e.g., username and a password). In some embodiments, the security credentials of all users of the remittance application may be encrypted and stored locally, for example, within a non-volatile storage device associated with the mobile device 350. In other embodiments, the security credentials may be encrypted and stored in a non-volatile device present at a remote location.

**[0316]** Once the credentials have been validated, a main menu is then displayed (block 4708). The main menu may list a number of functions associated with the remittance application, including the option to “pay a bill” or to “view the last bill paid.” An option to “configure recurring payments” is also presented to the user as one of the options, and the application will listen for the user’s selection of this option at decision block 4710.

**[0317]** At block 4712, a listing of all recurring payment schedules associated with the user is then displayed. For example, if the user had previously set up a recurring payment with Time Warner Cable and San Diego Gas and Electric, these two entries will be selectable within this listing. However, if no entries had been previously entered and saved by the user, a message such as: “No recurring payments have been scheduled” may appear in the display window in the alternative. An additional option to “set up a new recurring payment” is also presented to the user, for example, at the bottom of the display screen.

**[0318]** At blocks 4714 and 4716, the user will decide whether to update an existing recurring bill payment or to otherwise set up a new recurring payment. In the event that the user selected a preexisting recurring payment entry, previously stored data regarding this entry will be loaded at block 4718 (such as the name of the recurring payment entry, the payor, the payee, the selected payment method, a bank account or check routing number, a credit card number, and any other

preferred payment options). Otherwise, in the event that the user had selected to set up a new recurring payment, these data fields may be blank by default.

**[0319]** At block 4720, a sub-menu is then displayed including various data fields associated with this recurring payment entry. In some embodiments, the user may have an option to auto-populate at least some of these fields by instructing the system to extract data from a bill that has already been paid. Other fields can be modified, for example, by a keyboard, touchpad, mouse, or other such input device.

**[0320]** At block 4722, the user may then update these fields accordingly. In some embodiments, a “save” or “apply changes” option enables the user to save his input after the recurring payment entry has been updated. In other embodiments, the remittance application automatically saves the recurring payment entry after any data field has been modified by the user. Also, according to some embodiments, the remittance application may prevent the user from saving changes to the recurring bill payment entry if a certain minimum number of prerequisite data fields have not been filled out, or otherwise, if the data entered within any of these fields is of an invalid format.

**[0321]** According to some embodiments, the user may be presented the option of how he wishes to schedule recurring payments with the payee. FIG. 48 is a flow diagram illustrating this process. At block 4802, the user may be prompted to select among the options of: “Immediately,” “Manually,” “By Schedule,” or “Return to Previous Menu.” The remittance application may then check which option was selected at respective decision blocks 4810, 4820, 4830, and 4840.

**[0322]** If the user selected to schedule bill payments with the payee “Immediately,” then at block 4812, the remittance application configures itself to attempt to make a payment soon after receiving an image of a check and/or remittance coupon from the user. The document images can be preprocessed by the mobile device 350 and/or processed by the mobile remittance server 310 in any of the manners already described above. After the images have been successfully processed, one or more of the image quality assurance tests already described can then be run in

real-time in order to ensure that the user has taken an image with a quality sufficient to process a payment.

**[0323]** If the user selected to schedule bill payments with the payee “Manually,” then at block 4822, the remittance application configures itself to attempt to make a payment only upon a specific input from the user. This input might be, for example, a “Pay Bill” button located in one or more menus or sub-menus of the remittance application. Images of any remittance coupons/checks received from the user may then be persistently stored within a non-volatile storage device until the user acknowledges he is ready to pay a certain bill by providing the specific input required by the remittance application.

**[0324]** If the user selected to schedule payments with the payee “By Schedule,” then at block 4832, a submenu may appear prompting the user to specify certain scheduling options. In some embodiments, the user may specify how many days he wishes the application to submit the payment before (or after) a certain payment due date. For example, if a utility bill is always due the 15<sup>th</sup> of every month, the user may elect to have these recurring bills paid on the 10<sup>th</sup> of every month. Images of any remittance coupons/checks received from the user may then be persistently stored within a non-volatile storage device until the scheduled date of payment. In some embodiments, any preprocessing, processing, or image quality and assurance tests are run on the document images soon after they are received from the user. This enables the user to detect and correct any defects with the image documents well before the scheduled date of payment.

**[0325]** Irrespective of the option selected, the user will be returned to scheduling menu after providing the input from the recurring payment sub-menu. If the user selected to “Return to Previous Menu,” then at block 4842 the user will be directed to the previous menu and the process will end.

**[0326]** According to some embodiments, the user may be presented the option of whether he wishes to have the remittance application send him one or more reminders about upcoming payment due dates. The reminders may thus serve to assist the user in preventing a payment default due to inattention, inadvertence, or neglect.

[0327] FIG. 49 is a flow diagram illustrating an exemplary process of enabling a user to set one or more reminders associated with a recurring bill payment according to one embodiment of the present invention. At block 4902, a menu is displayed to the user, the menu including an option (such as a hyperlink or selectable button) for setting one or more payment reminders associated with a recurring payment schedule.

[0328] Once this option is selected at block 4904, then at block 4906, a sub-menu may then be displayed to the user. In some embodiments, the sub-menu presents the user with a number of configurable options regarding payment reminders. For example, the user may decide whether to set up a single reminder or a series of periodic reminders. Additionally, the user may specify when the reminders are to be sent (for example, on a regularly occurring day each month, such as on the 5<sup>th</sup>, or instead on a day that is always measured relative to the payment due date, such as 7 days before the bill is due). In some embodiments, the user may also specify how frequently the reminders are to be sent (e.g., daily, every third day, weekly, bi-weekly, etc.).

[0329] Additionally, according to some embodiments, the user may specify the type of reminders to be provided to the user by the remittance application. Any number of mechanisms for informing the user about an upcoming payment may be used according to embodiments of the present invention (including, but not limited to: e-mail, popup windows, SMS messages, “push”/PAP messaging, calendar alerts, scheduled printing, and phone messages/voicemail). Once the user has finished inputting preferred options at block 4908, the options are saved at block 4910, and the process then ends. Subsequently, the remittance application can provide payment reminders to the user in any manner or manner(s) that the user has specified.

[0330] In this document, the terms “computer program medium” and “computer usable medium” are used to generally refer to physical storage media such as, for example, memory 1908, storage unit 1920, and media 1914. These and other various forms of computer program media or computer usable media may be involved in storing one or more sequences of one or more instructions to a processing device for execution. Such instructions embodied on the medium, are generally referred to as “computer program code” or a “computer program product” (which may be grouped in the form of computer programs or other groupings). When executed,

such instructions might enable the computing module 1900 to perform features or functions of the present invention as discussed herein.

**[0331]** While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not of limitation. The breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments. Where this document refers to technologies that would be apparent or known to one of ordinary skill in the art, such technologies encompass those apparent or known to the skilled artisan now or at any time in the future. In addition, the invention is not restricted to the illustrated example architectures or configurations, but the desired features can be implemented using a variety of alternative architectures and configurations. As will become apparent to one of ordinary skill in the art after reading this document, the illustrated embodiments and their various alternatives can be implemented without confinement to the illustrated example. One of ordinary skill in the art would also understand how alternative functional, logical or physical partitioning and configurations could be utilized to implement the desired features of the present invention.

**[0332]** Furthermore, although items, elements or components of the invention may be described or claimed in the singular, the plural is contemplated to be within the scope thereof unless limitation to the singular is explicitly stated. The presence of broadening words and phrases such as “one or more,” “at least,” “but not limited to” or other like phrases in some instances shall not be read to mean that the narrower case is intended or required in instances where such broadening phrases may be absent.

## CLAIMS

What is claimed is:

1. A computer-readable medium comprising instructions which, when executed by a computer, perform a process comprising:
  - receiving an image of a document captured by a mobile device;
  - identifying a set of characteristics of the document in relation to said image;
  - processing the image to correct one or more graphical defects, wherein said processing is adapted to generate a processed image based at least in part upon the set of characteristics;
  - executing one or more mobile image quality assurance tests on the processed image to assess the quality of the processed image; and
  - extracting data from the processed image.
2. The computer-readable medium of Claim 1, wherein identifying a set of characteristics comprises identifying the corners of the document.
3. The computer-readable medium of Claim 1, wherein identifying a set of characteristics comprises identifying the orientation of the document.
4. The computer-readable medium of Claim 1, wherein said processing the image comprises geometrically correcting the image.
5. The computer-readable medium of Claim 1, wherein said processing the image comprises performing a dewarping operation on the image.
6. The computer-readable medium of Claim 1, wherein said extracting data from the processed image comprises generating a bi-tonal version of the image.



7. The computer-readable medium of Claim 6, wherein said extracting data from the processed image further comprises attempting to identify a matching document template corresponding to the layout of said document.
8. The computer-readable medium of Claim 7, wherein said attempting to identify a matching document template comprises reading a portion of text from the image by applying an optical character recognition technique.
9. The computer-readable medium of Claim 7, wherein said attempting to identify a matching document template comprises identifying a unique symbol present within the document.
10. The computer-readable medium of Claim 7, wherein said attempting to identify a matching document template comprises identifying a unique identifier present within the document.
11. The computer-readable medium of Claim 7, wherein said attempting to identify a matching document template comprises analyzing a region of the document measured at some position relative to a landmark symbol present within said document.
12. The computer-readable medium of Claim 6, wherein said extracting data from the processed image further comprises identifying the presence of one or more characters by utilizing an optical character recognition technique to at least a portion of said processed image.
13. The computer-readable medium of Claim 6, wherein said extracting data from the processed image further comprises utilizing a keyword-based detection technique.
14. The computer-readable medium of Claim 6, wherein said extracting data from the processed image further comprises utilizing a format-based detection technique.
15. The computer-readable medium of Claim 14, wherein said format-based detection technique is based at least in part upon a regular expression.

16. The computer-readable medium of Claim 6, wherein said extracting data from the processed image further comprises utilizing a code-line validation technique.

17. An apparatus for processing an image captured from a mobile device, the apparatus comprising:

    a receiving module adapted to receive an image of a document captured by a mobile device;

    an identification module in electrical communication with said receiving module, the identification module adapted to identify a set of characteristics of the document in relation to said image;

    a processing module in electrical communication with said identification module, the processing module adapted to process the image to correct one or more graphical defects present within the image, wherein said processing is based at least in part upon said set of characteristics;

    a testing module in electrical communication with the processing module, the testing module adapted to execute one or more mobile image quality assurance tests on a processed image in order to assess the quality of the processed image; and

    a data extraction module in electrical communication with the testing module, the data extraction module adapted to extract data from the processed image.

18. The apparatus of Claim 17, wherein the apparatus is configured to interface with an application comprising a recurring payment scheduler.

19. The apparatus of Claim 18, wherein the recurring payment scheduler is adapted to enable a user to configure one or more scheduling options.

20. The apparatus of Claim 18, wherein the recurring payment scheduler is adapted to provide automatic payment reminders to one or more users associated with said application.

**ABSTRACT**

The present invention relates to automated document processing and more particularly, to methods and systems for document image capture and processing using mobile devices. In accordance with various embodiments, methods and systems for document image capture on a mobile communication device are provided such that the image is optimized and enhanced for data extraction from the document as depicted. These methods and systems may comprise capturing an image of a document using a mobile communication device; transmitting the image to a server; and processing the image to create a bi-tonal image of the document for data extraction. Additionally, these methods and systems may comprise capturing a first image of a document using the mobile communication device; automatically detecting the document within the image; geometrically correcting the image; binarizing the image; correcting the orientation of the image; correcting the size of the image; and outputting the resulting image of the document.

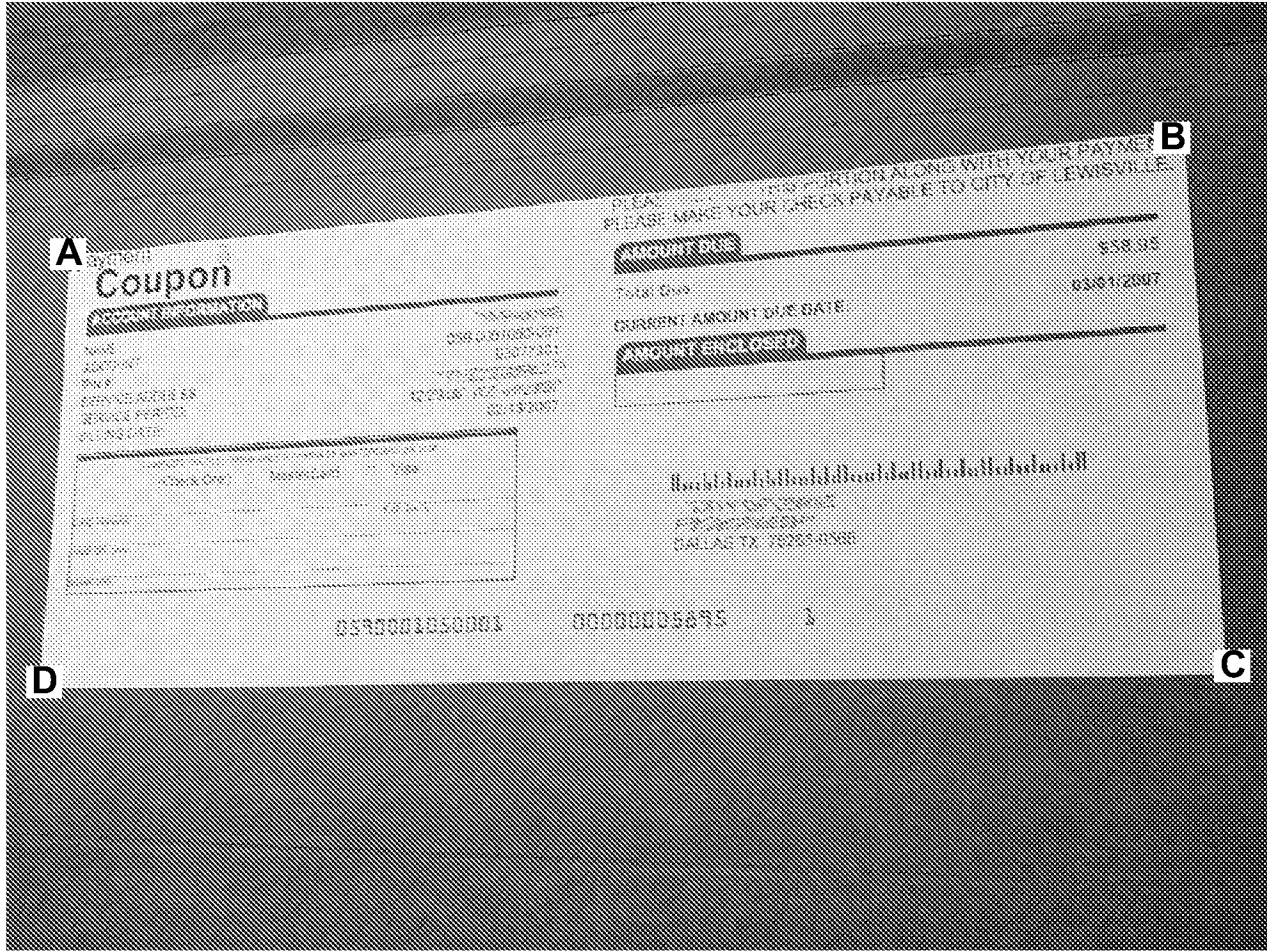


FIG. 1

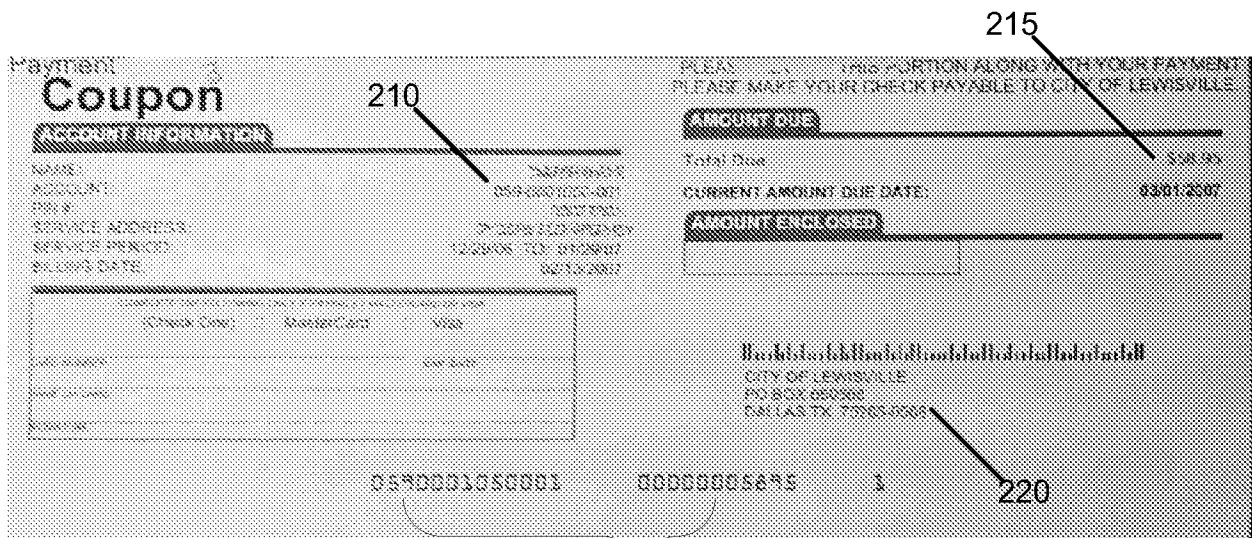
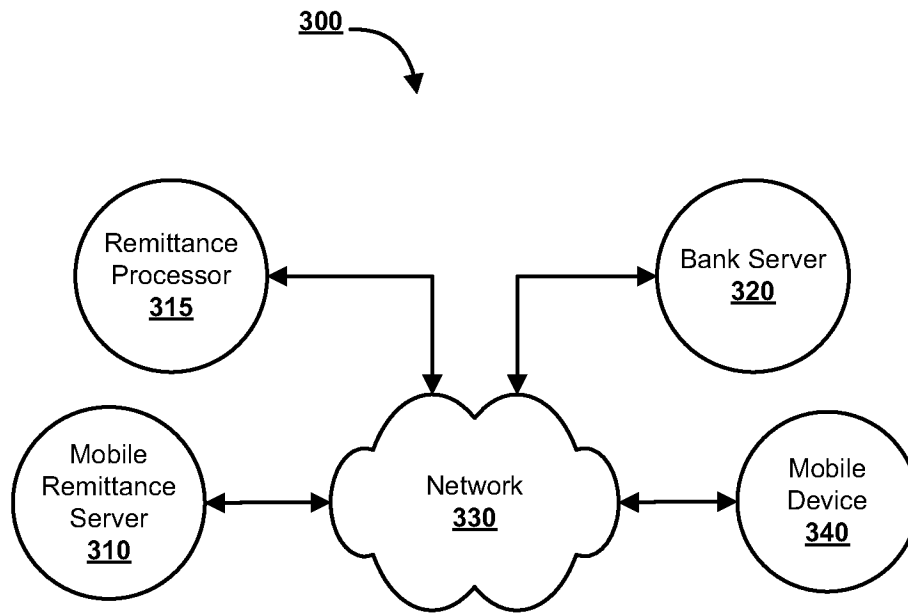
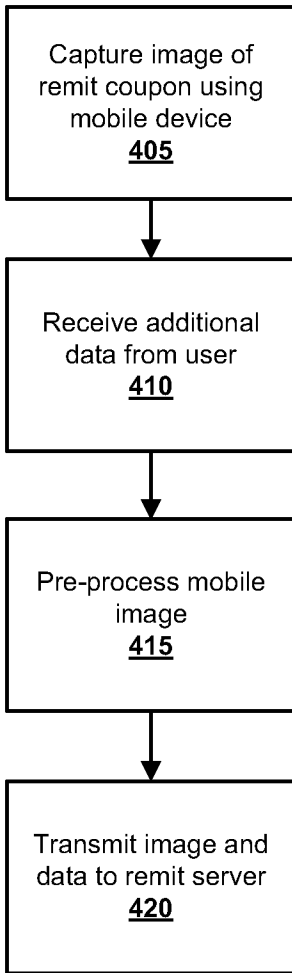


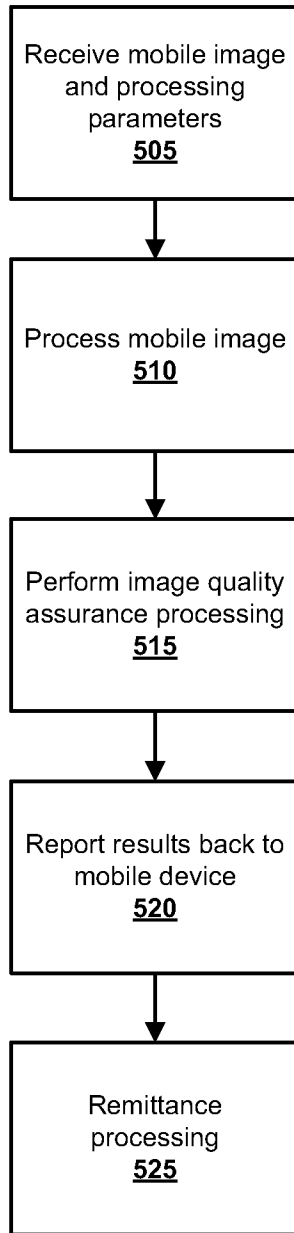
FIG. 2



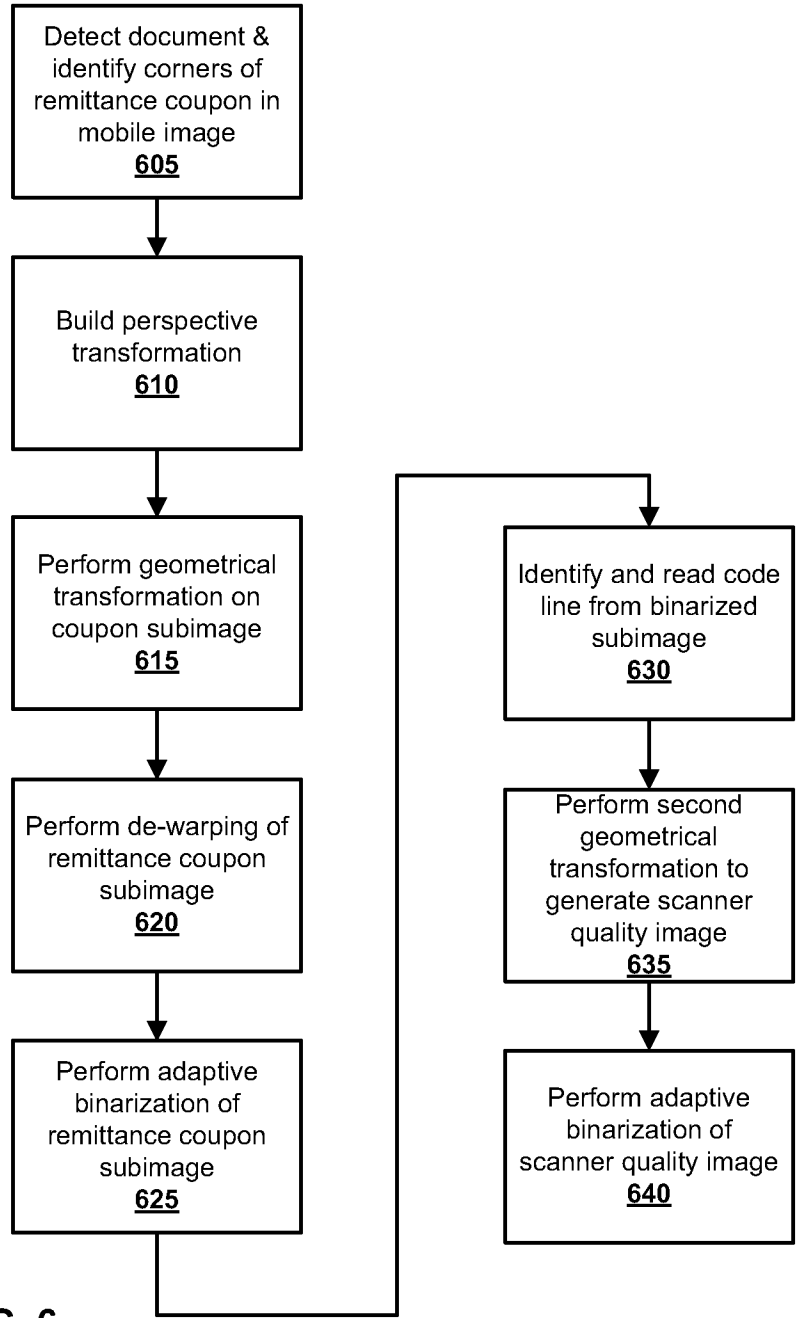
**FIG. 3**



**FIG. 4**  
Mobile Device Processing



**FIG. 5**  
Server Side Processing



**FIG. 6**  
Image  
Correction  
Module

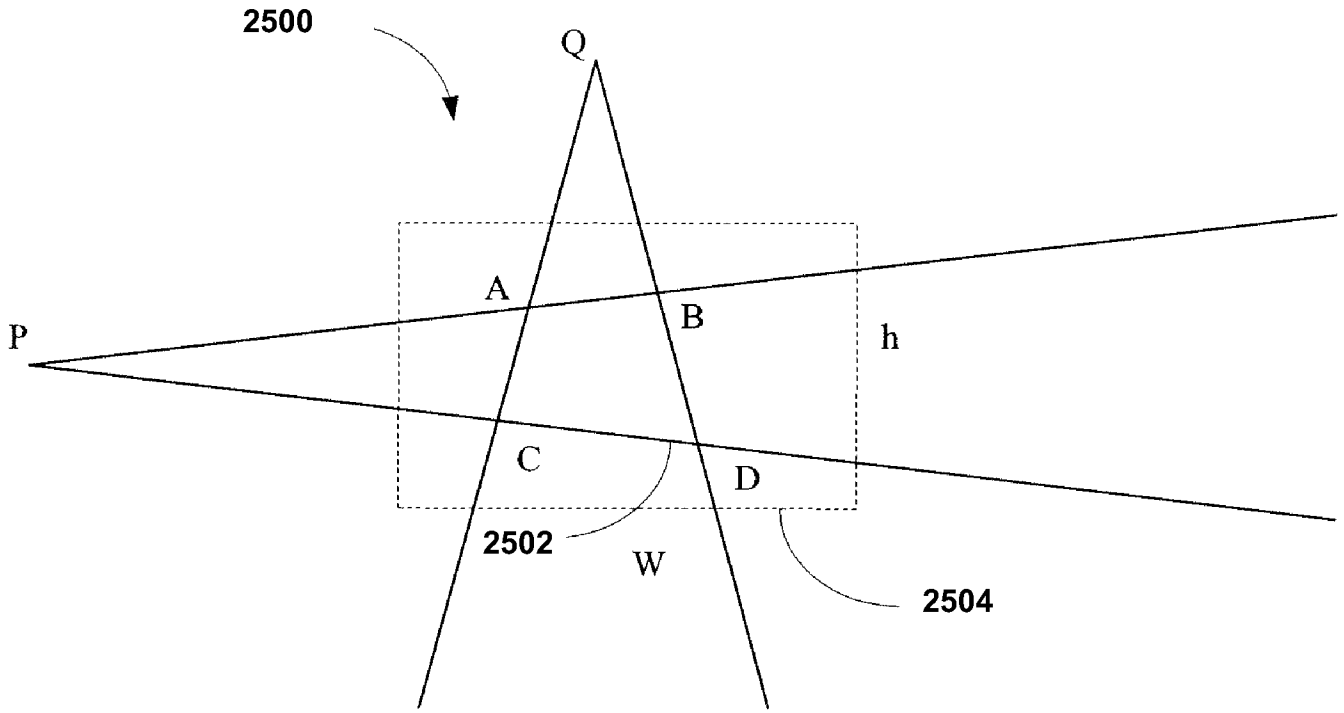


FIG. 7

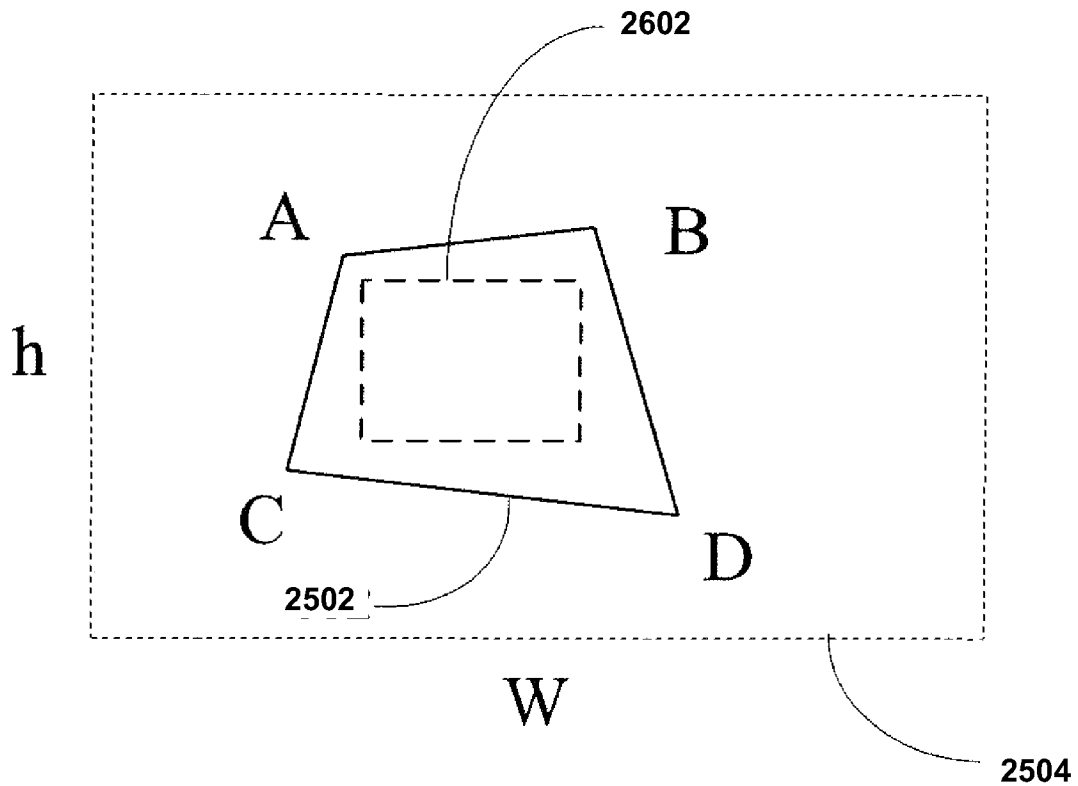
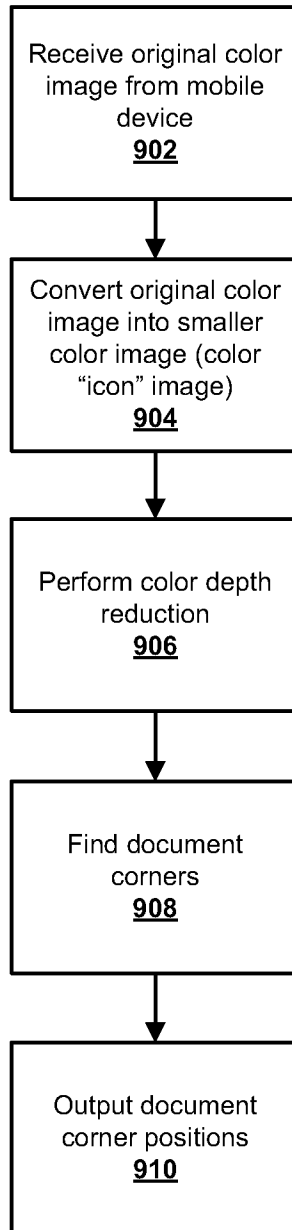


FIG. 8





**FIG. 9**  
Detect  
Document in  
Image

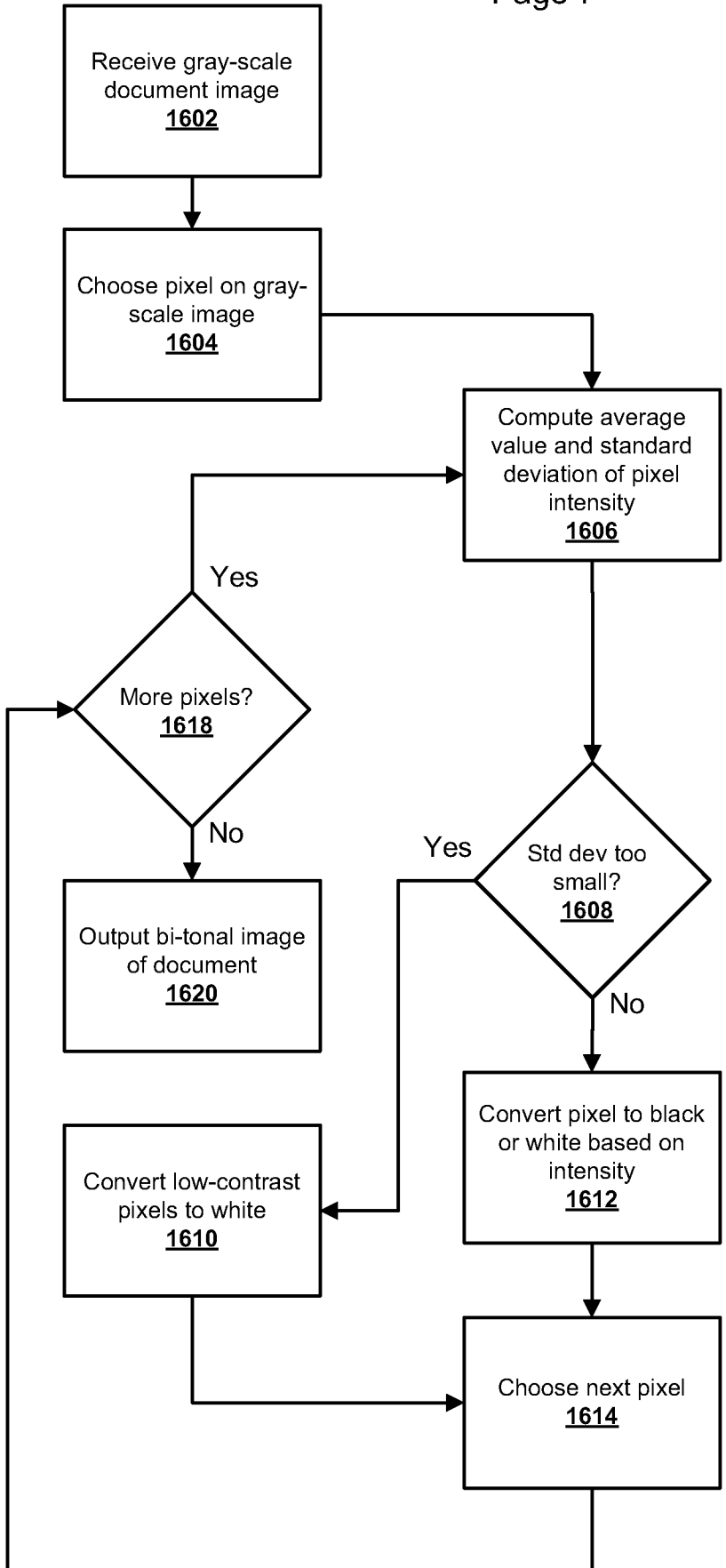
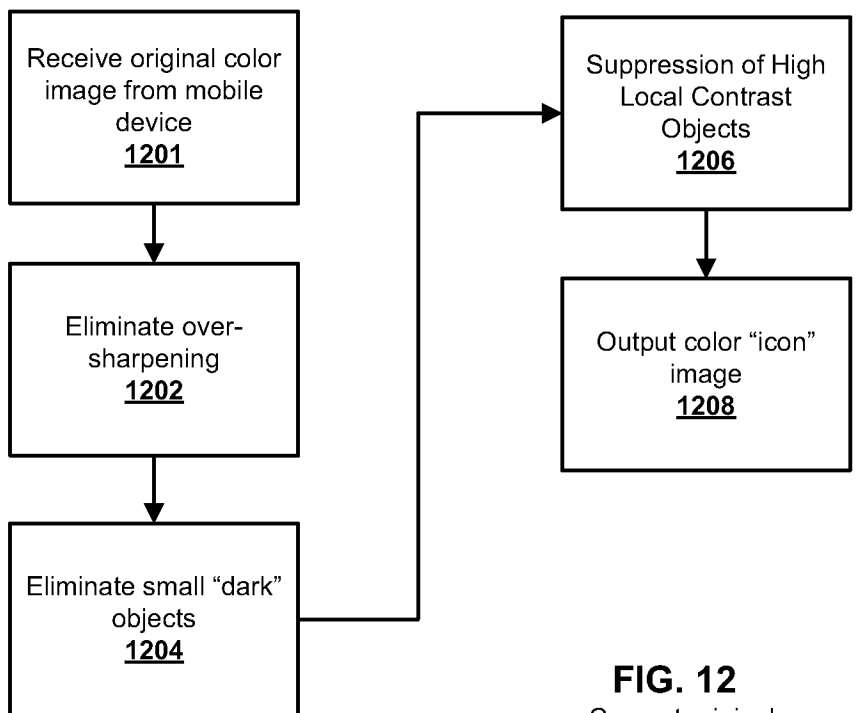


FIG. 10

Payment <b>Coupon</b>		PLEASE ATTACH THIS PORTION ALONG WITH YOUR PAYMENT PLEASE MAKE YOUR CHECK PAYABLE TO CITY OF LEWISVILLE.	
<b>ACCOUNT INFORMATION</b>		<b>AMOUNT DUE</b>	
NAME:		Total Due	\$58.95
ACCOUNT:	059-0001050-001	CURRENT AMOUNT DUE DATE:	03/01/2007
PIN #:	03077301	<b>AMOUNT ENCLOSED</b>	
SERVICE ADDRESS:		[Empty Box]	
SERVICE PERIOD:	12/29/06 TO: 01/29/07		
BILLING DATE:	02/13/2007		
COMPLETE THE FOLLOWING ONLY IF PAYING BY MASTERCARD OR VISA (Check One) MasterCard Visa			
CARD NUMBER	_____	EXPIRE DATE	_____
NAME ON CARD	_____		
SIGNATURE	_____		
0590001050001		00000005895	
		CITY OF LEWISVILLE PO BOX 650568 DALLAS TX 75265-0568	

FIG. 11



**FIG. 12**  
 Convert original color image into smaller color image (color "icon" image)

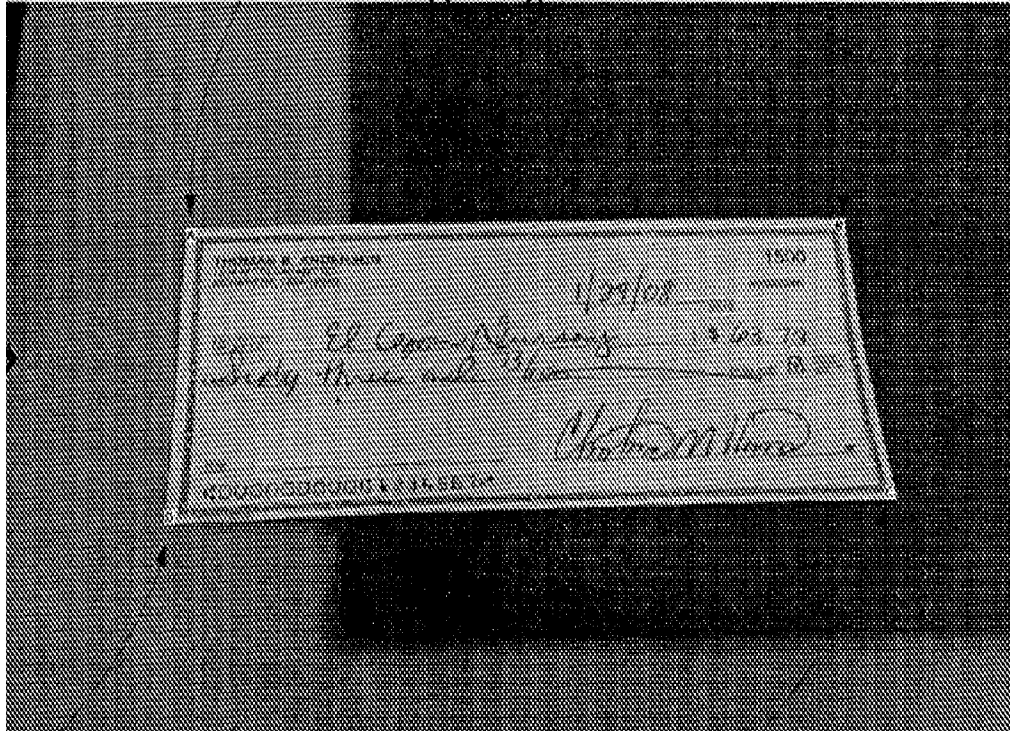


FIG. 13A

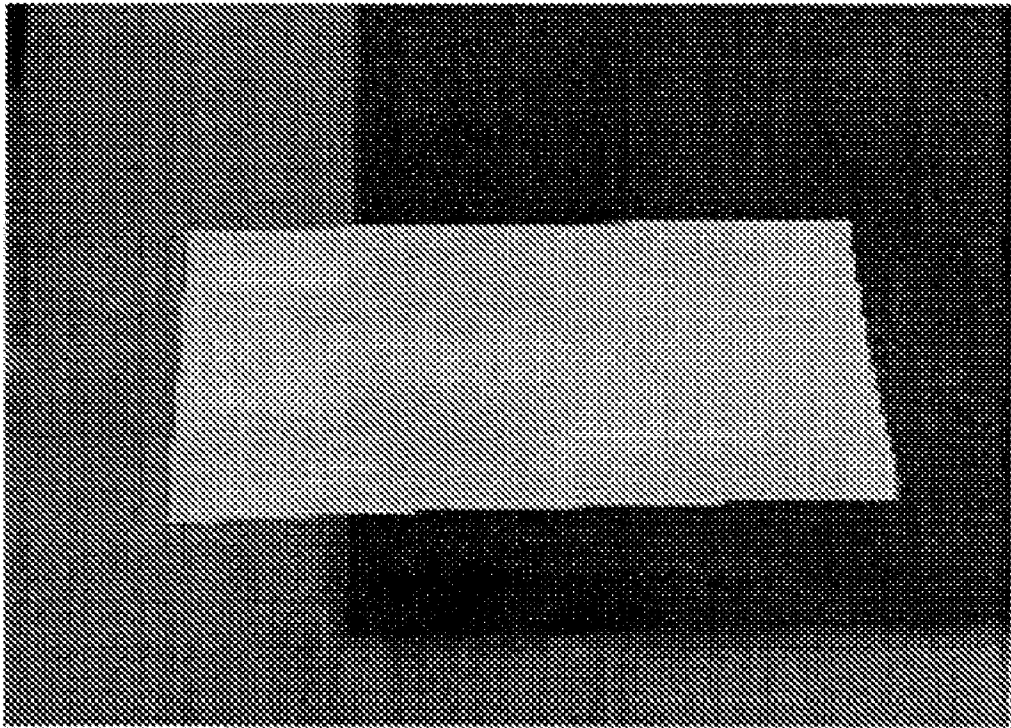


FIG. 13B

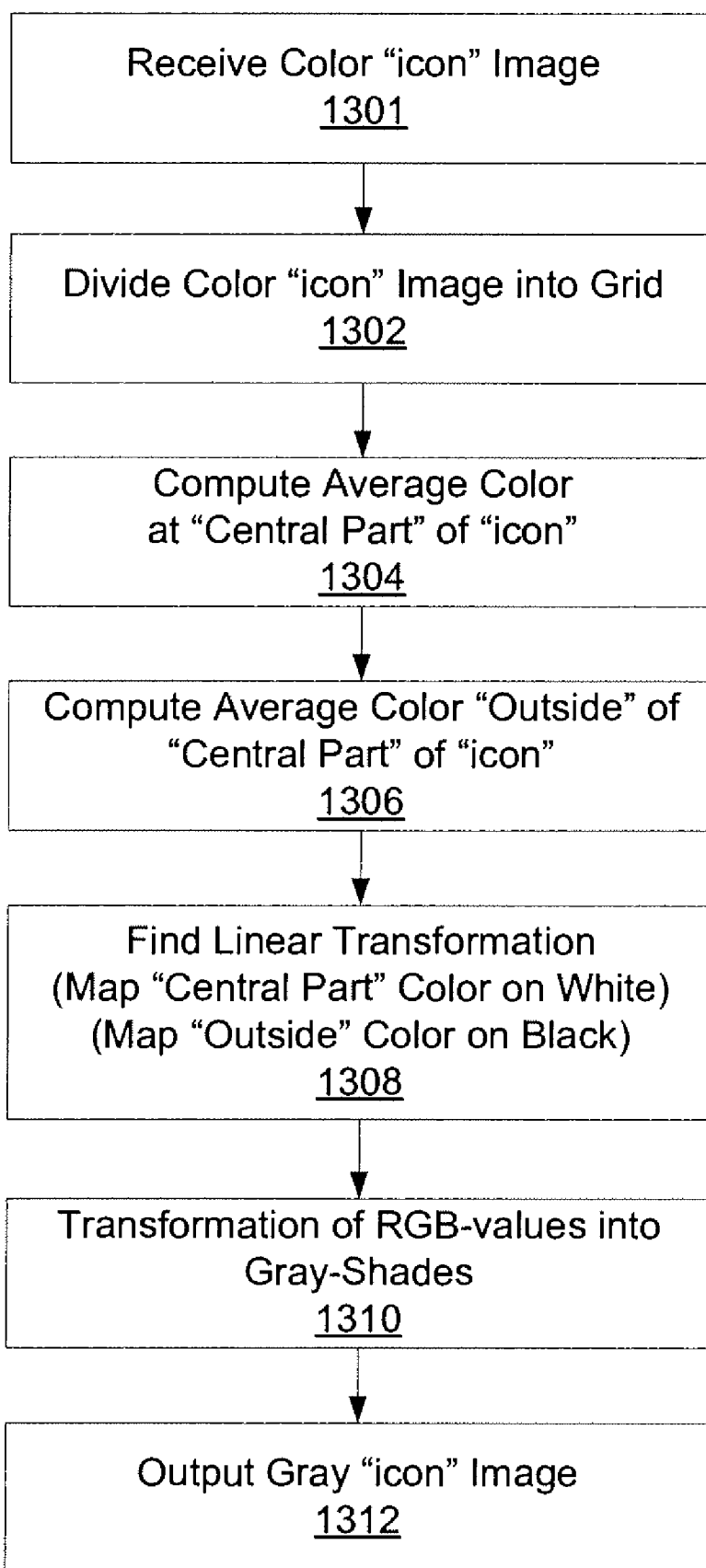
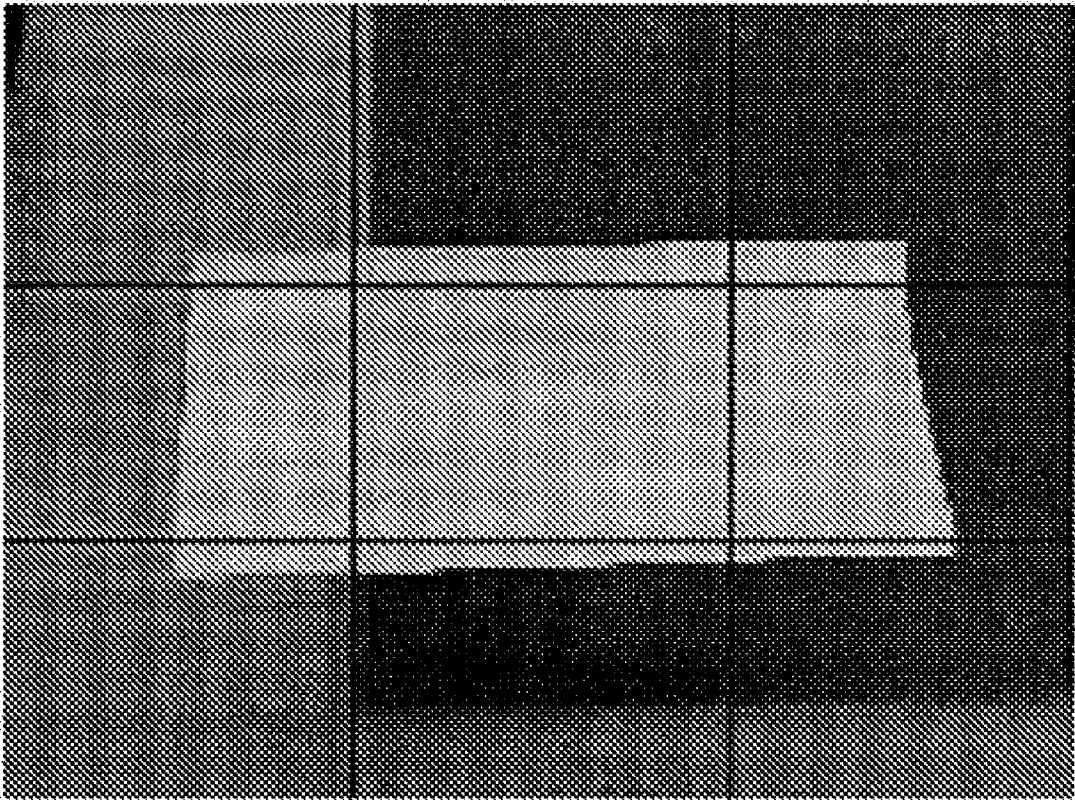
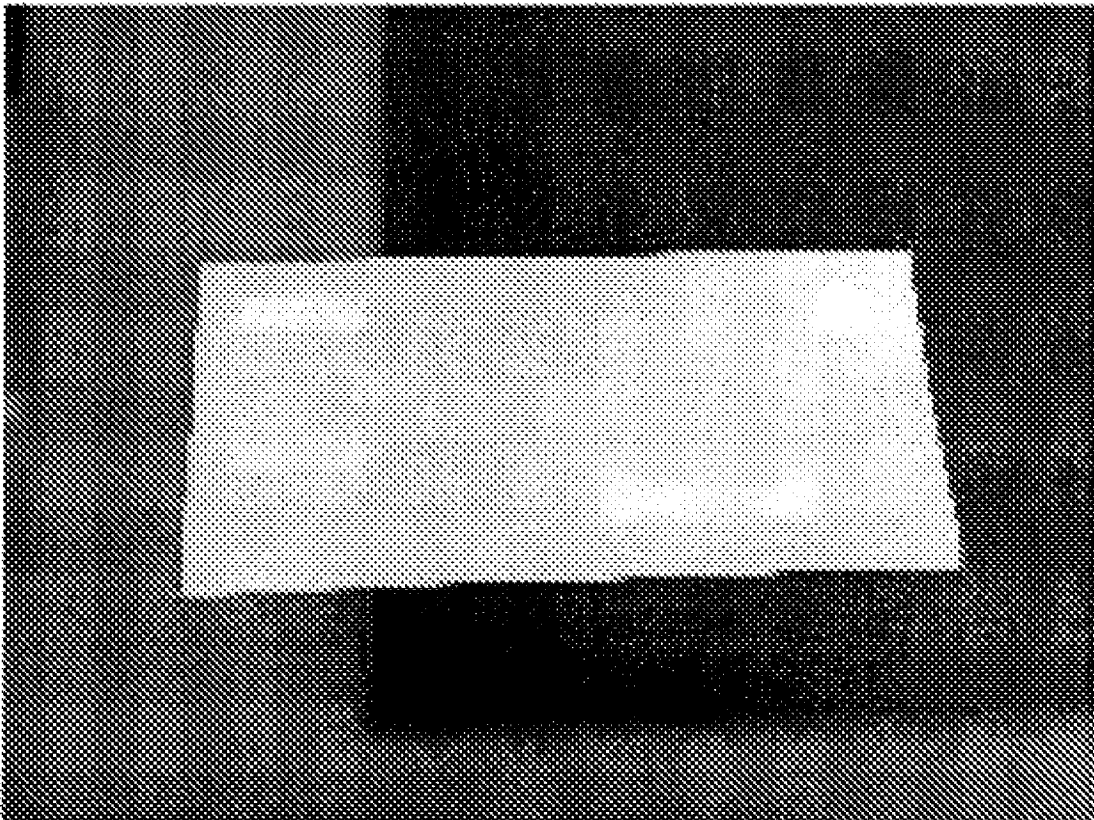


FIG. 14



**FIG. 15A**



**FIG. 15B**

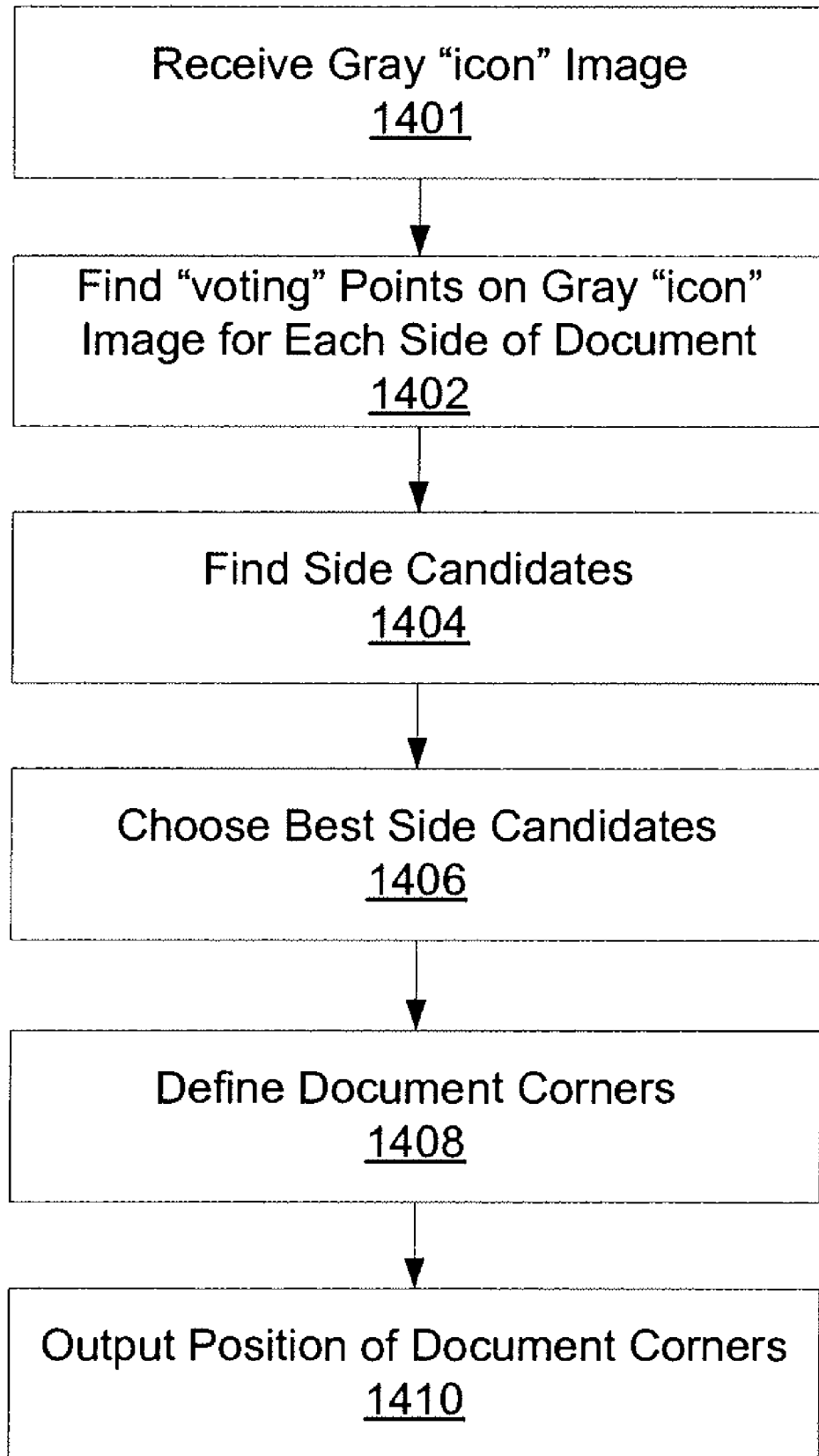
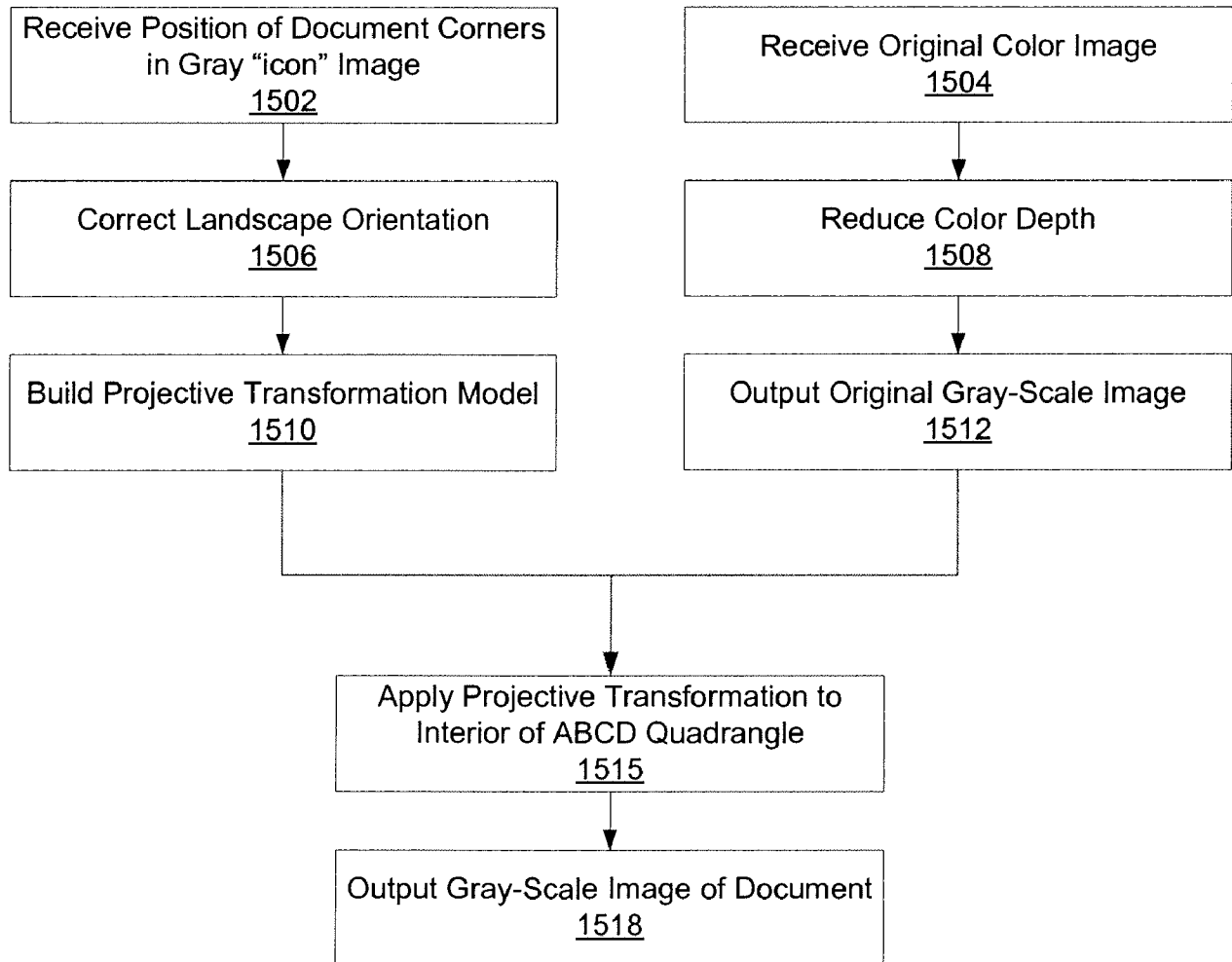
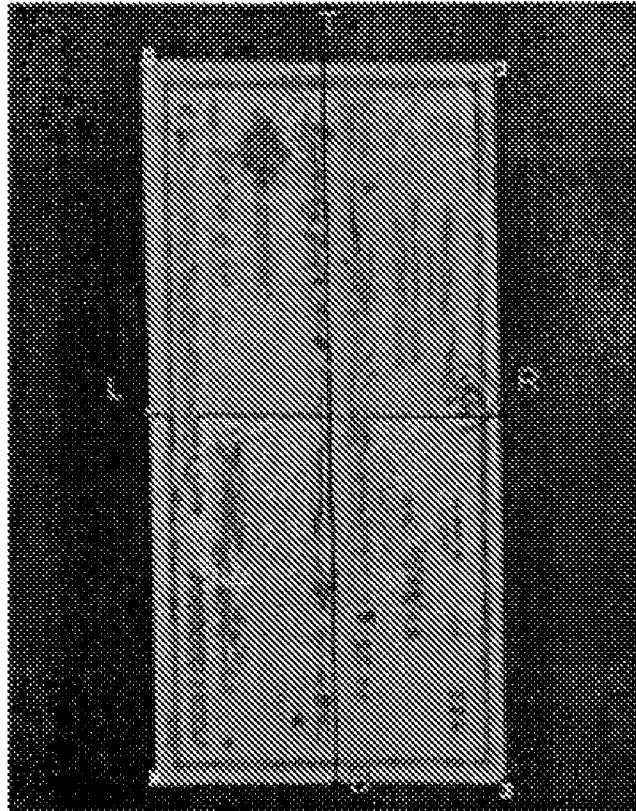


FIG. 16

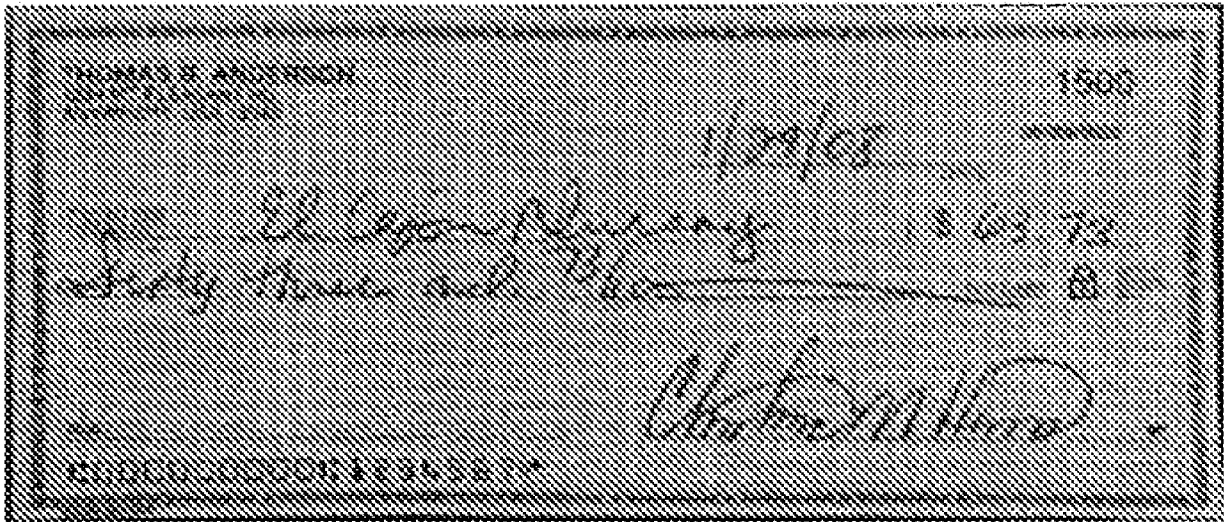


**FIG. 17**





**FIG. 18A**  
The check in this image is intentionally depicted in landscape orientation.



**FIG. 18B**

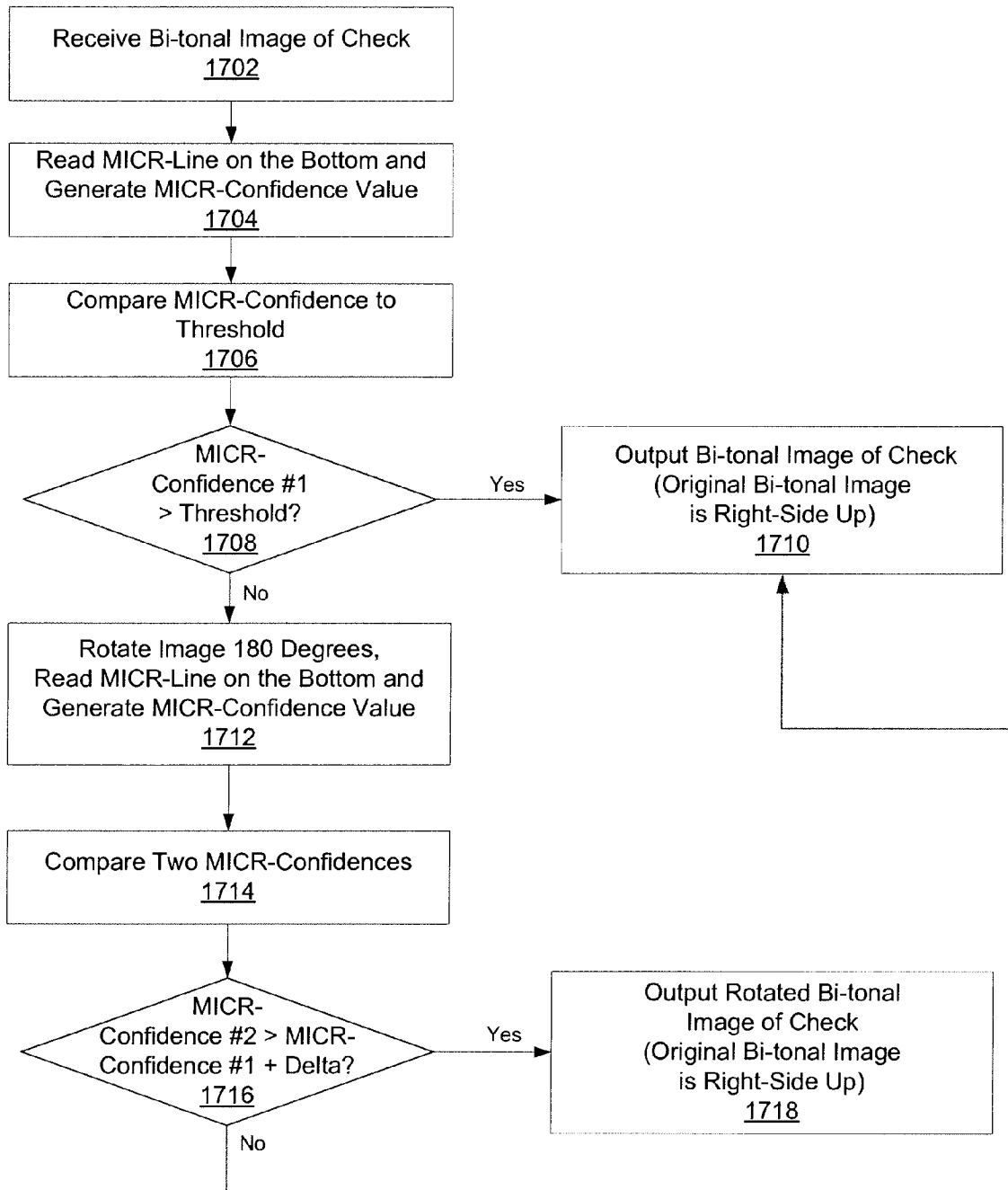


FIG. 19

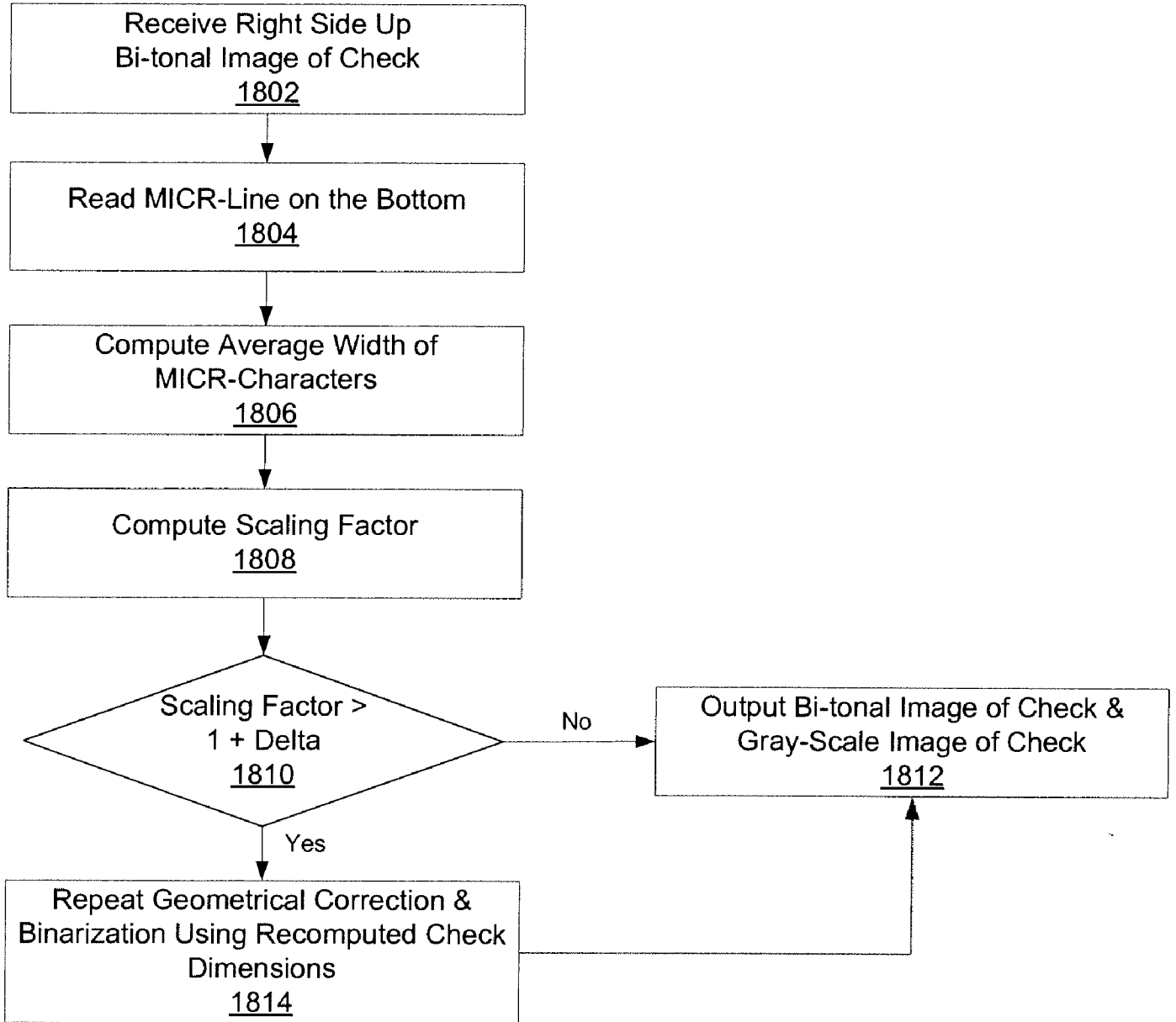


FIG. 20

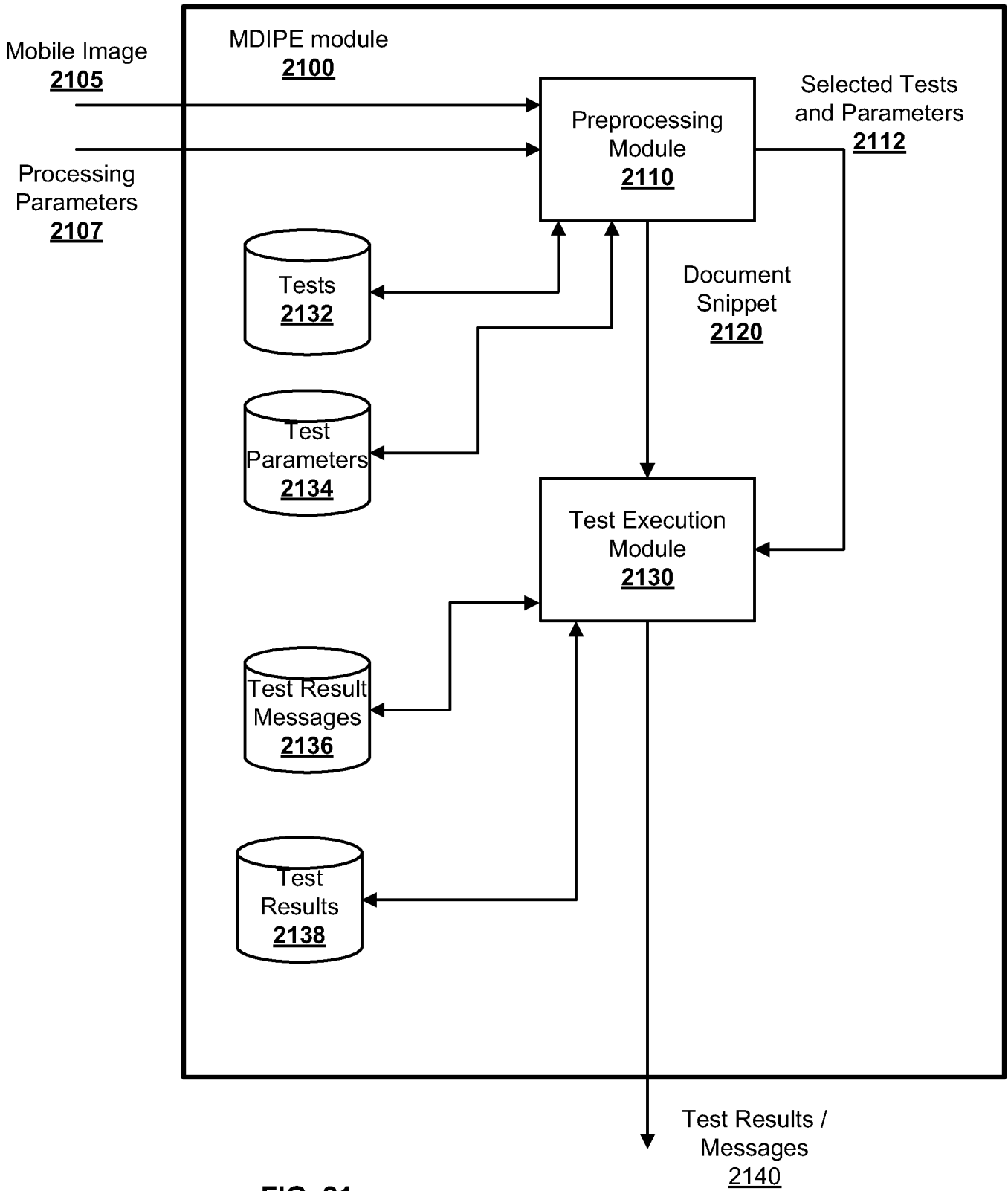
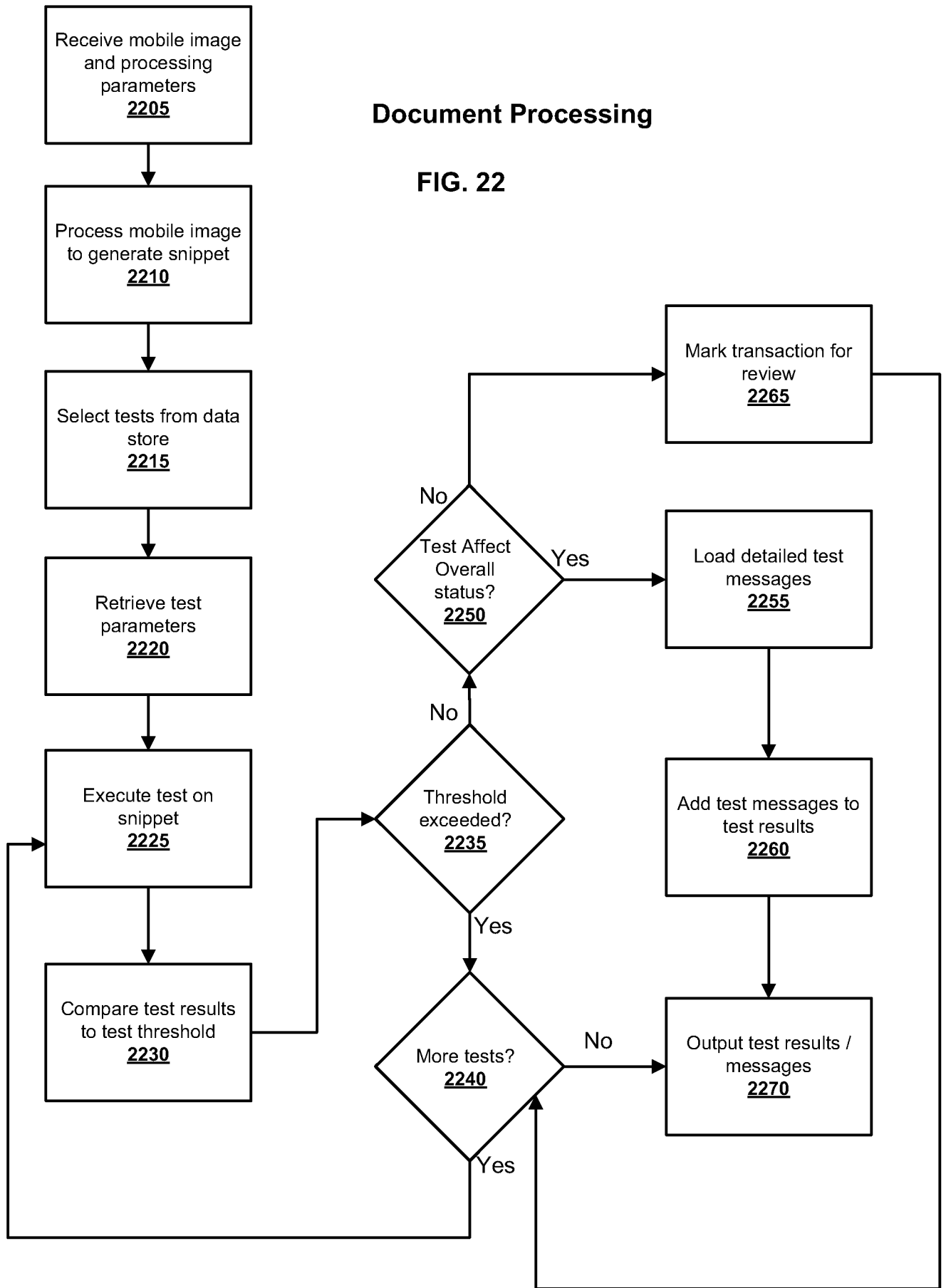
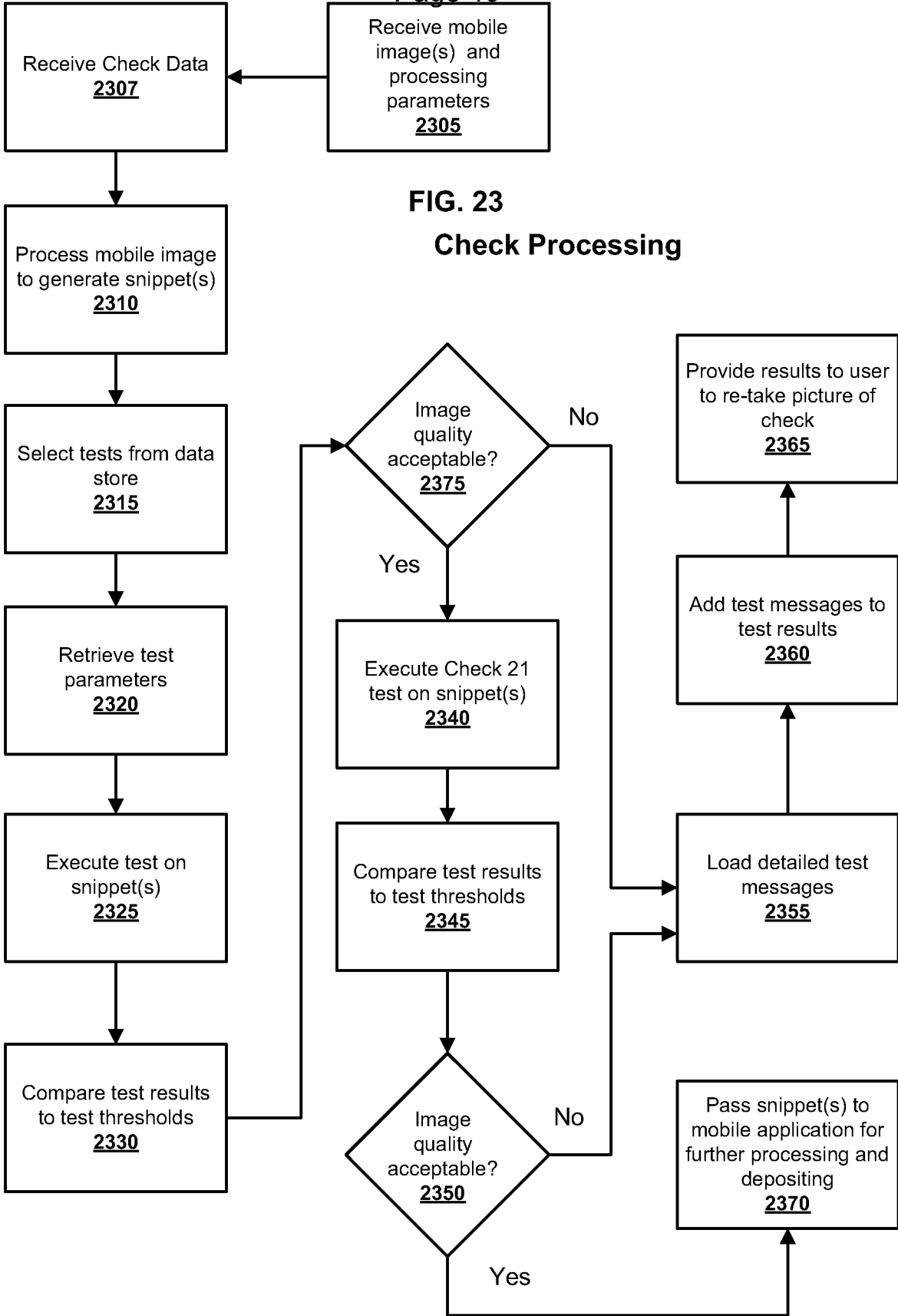


FIG. 21

### Document Processing

FIG. 22





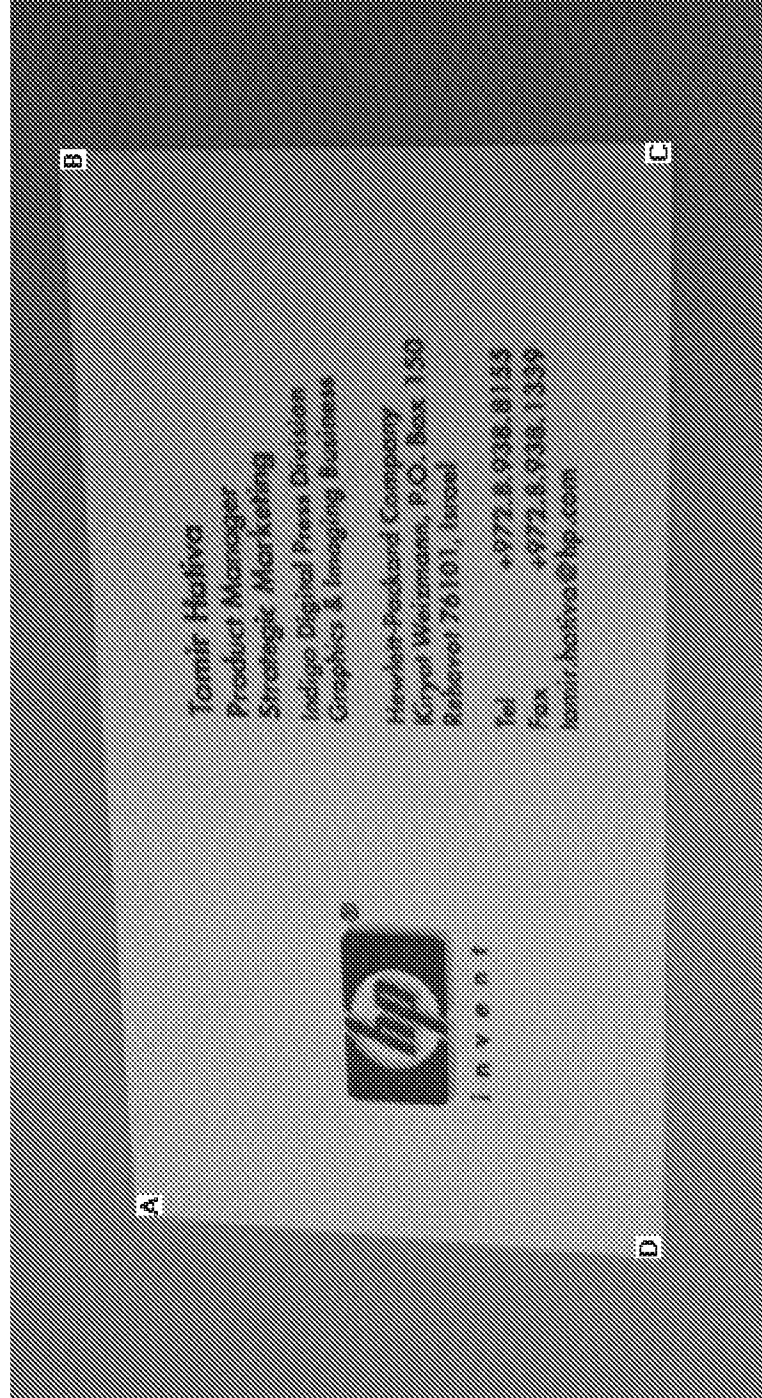


FIG. 24A

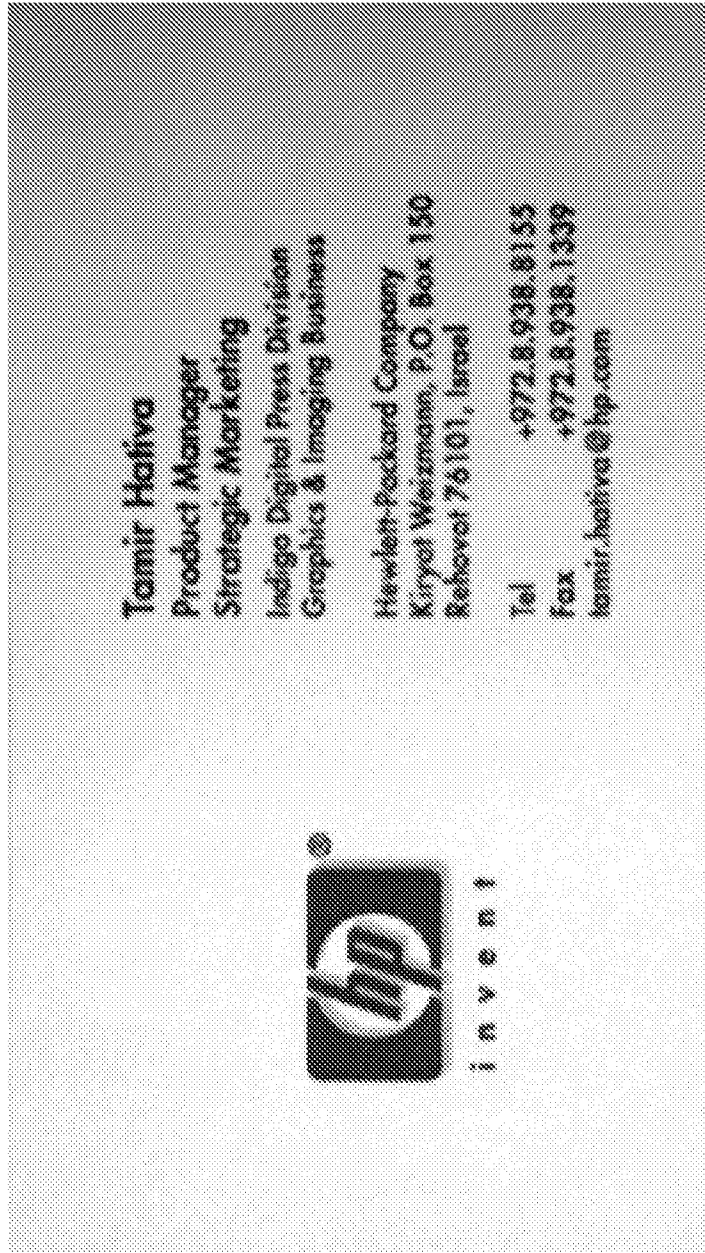


FIG. 24B





FIG. 25A

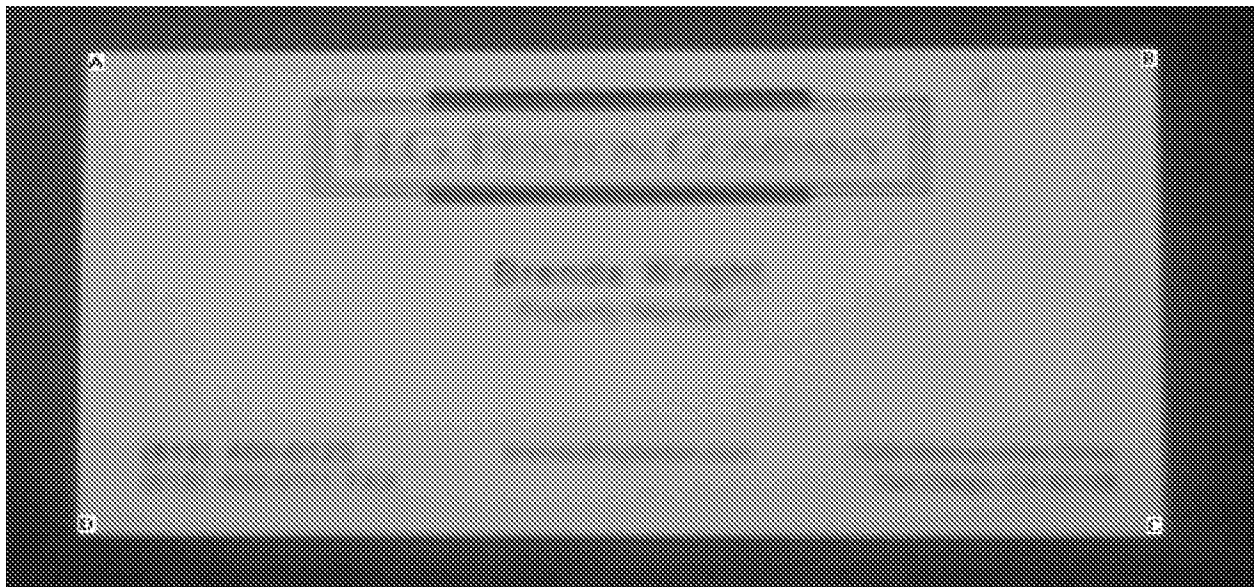


FIG. 25B

**This figure is intentionally blurry.**

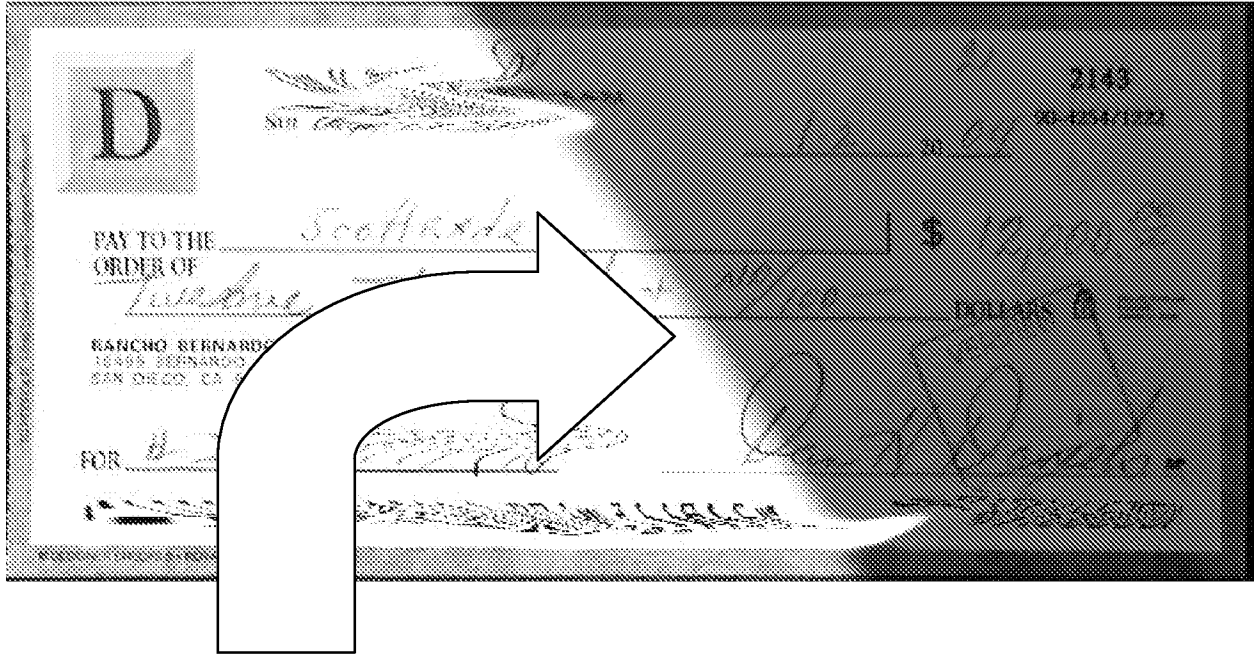
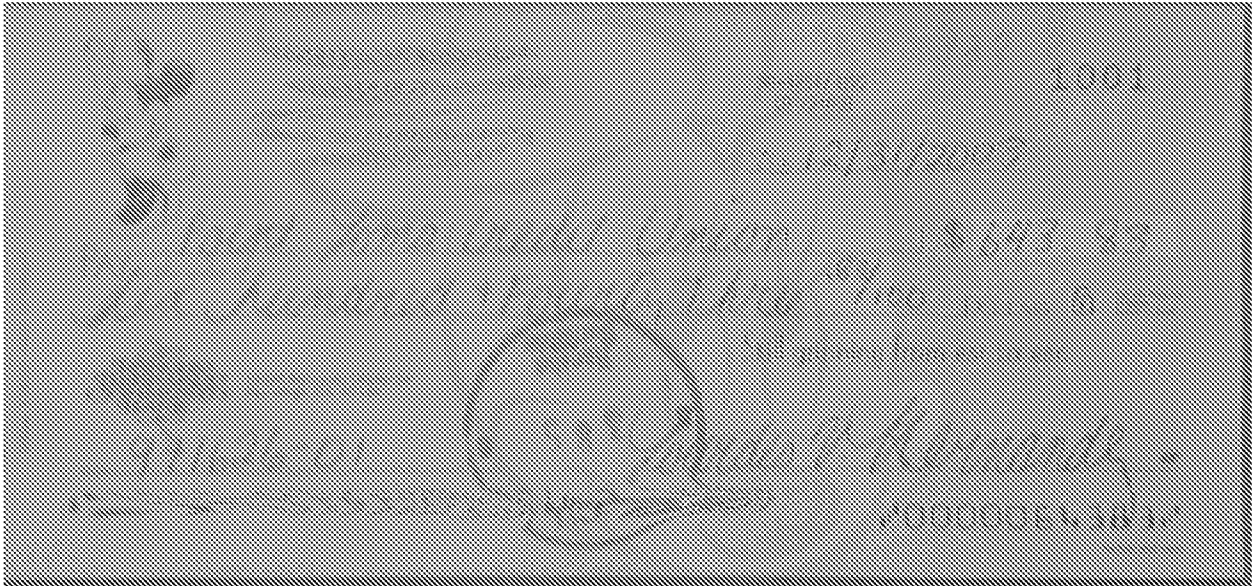


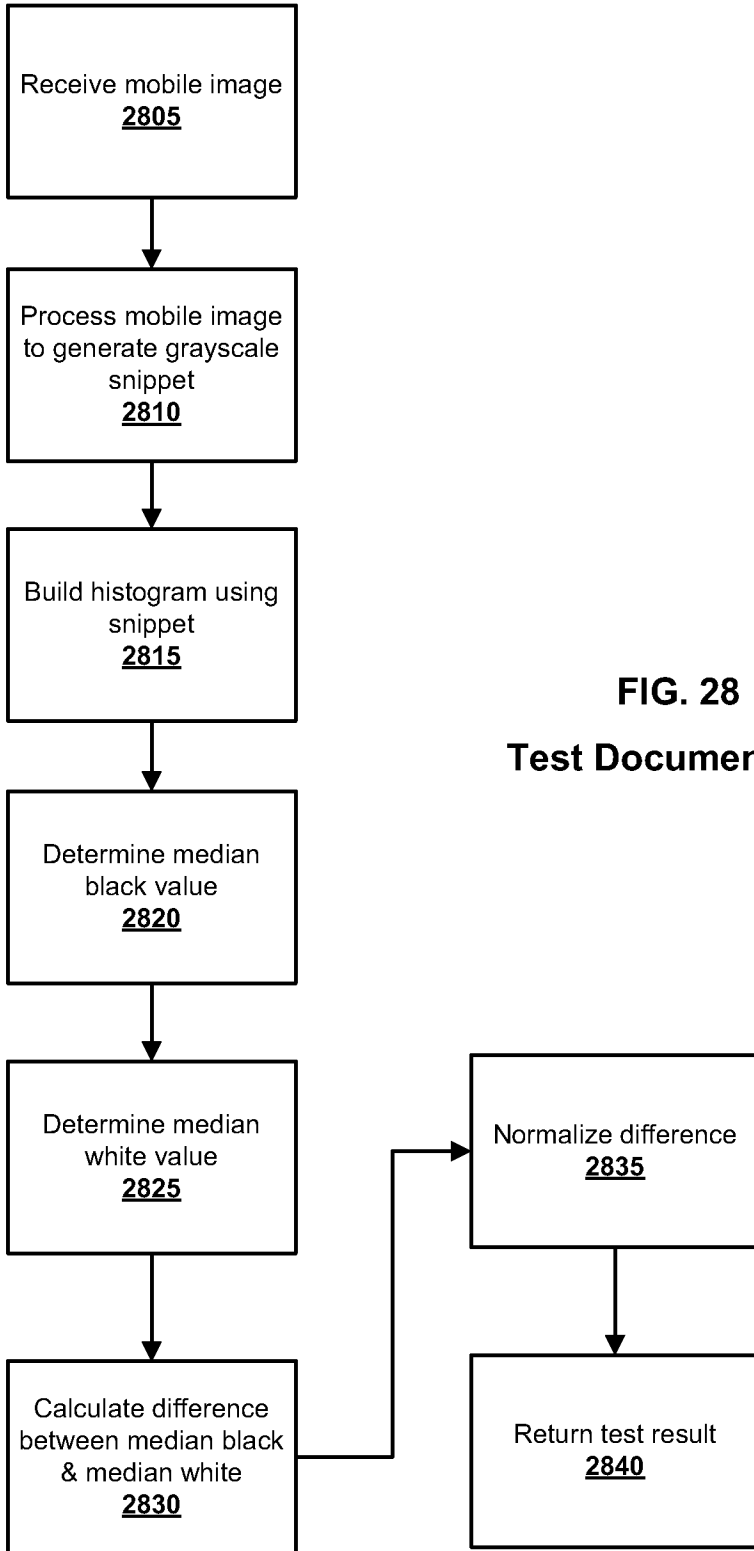
FIG. 26

This figure is intentionally blurry.

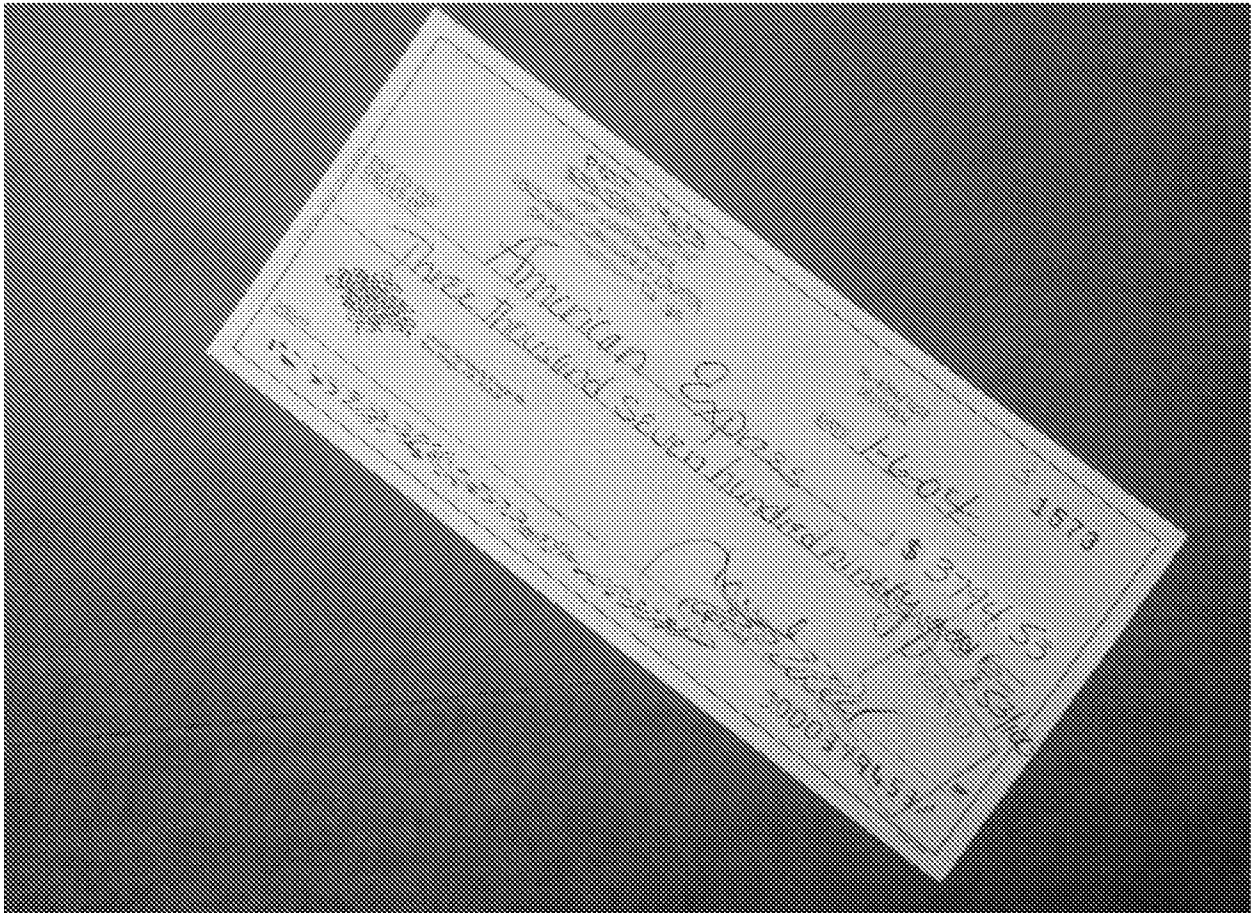


**FIG. 27**

**This figure is intentionally blurry.**



**FIG. 28**  
**Test Document Contrast**

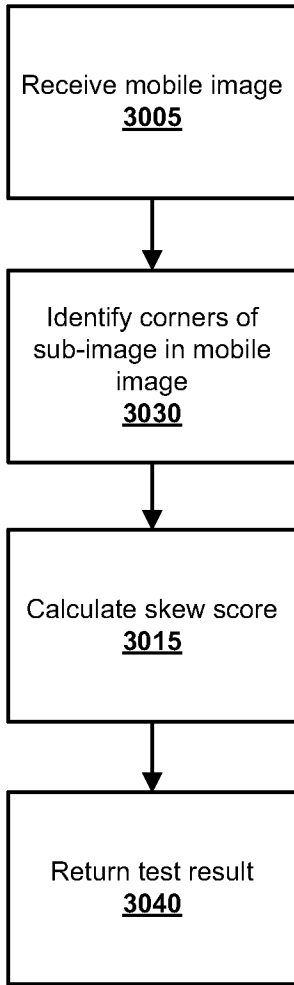


**FIG. 29A**

**This figure is intentionally blurry.**



FIG. 29B



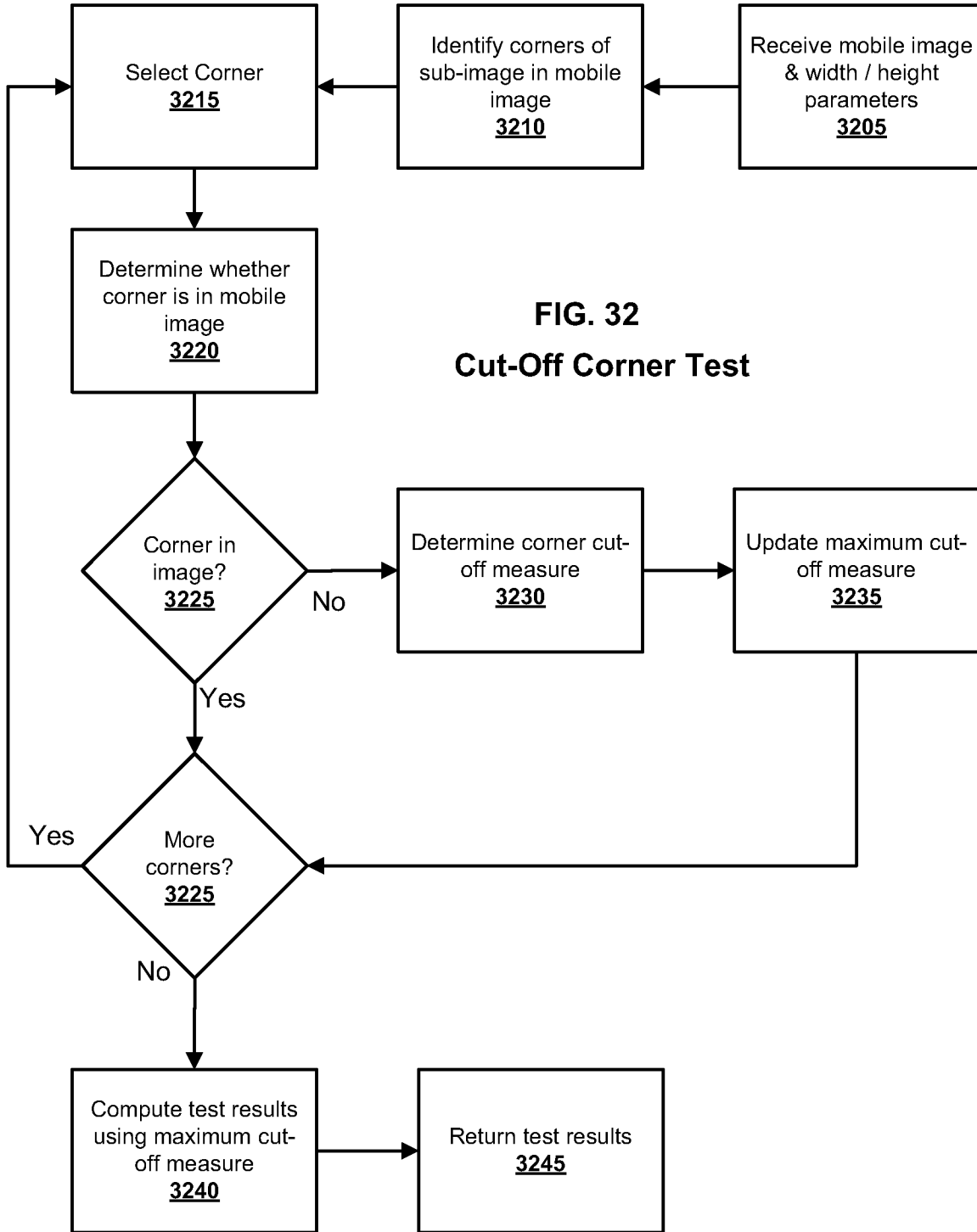
**FIG. 30**  
**Test Document Skew**



C[1]

FIG. 31





**FIG. 32**  
**Cut-Off Corner Test**

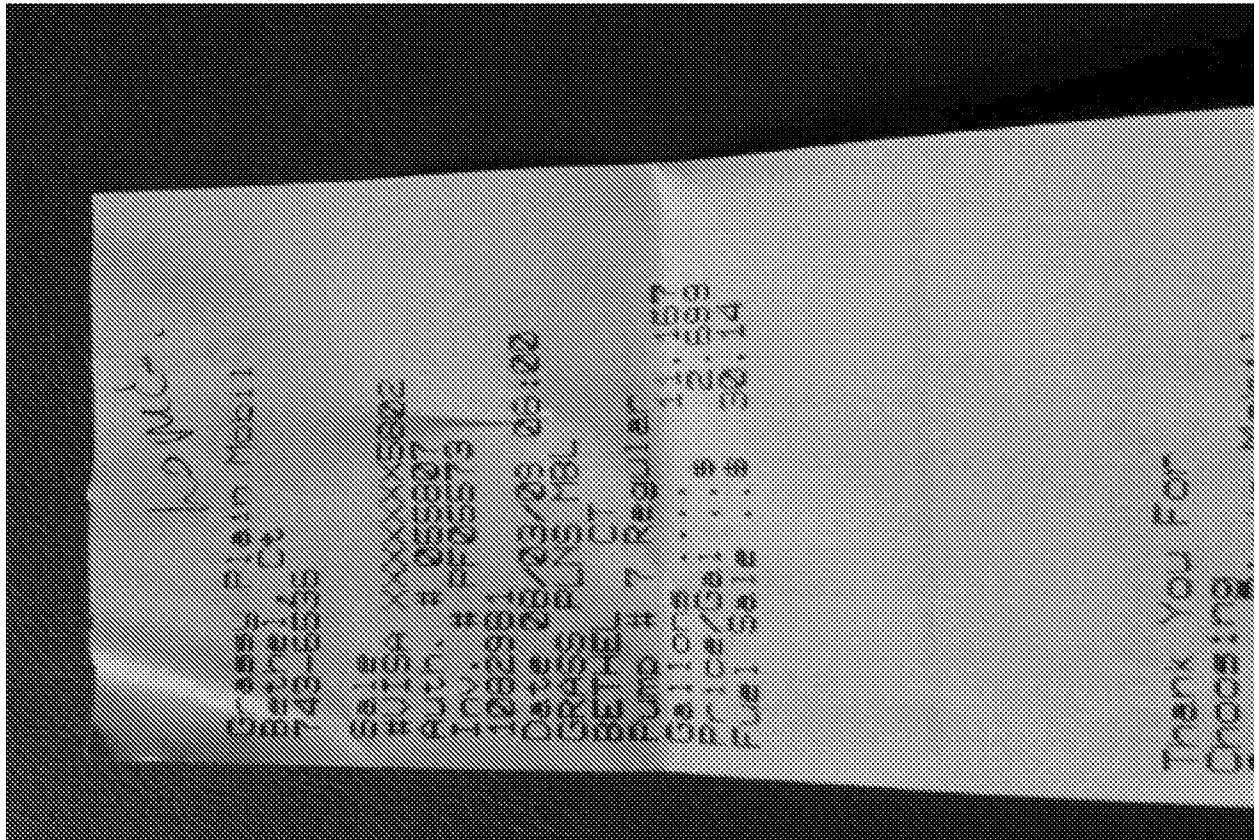
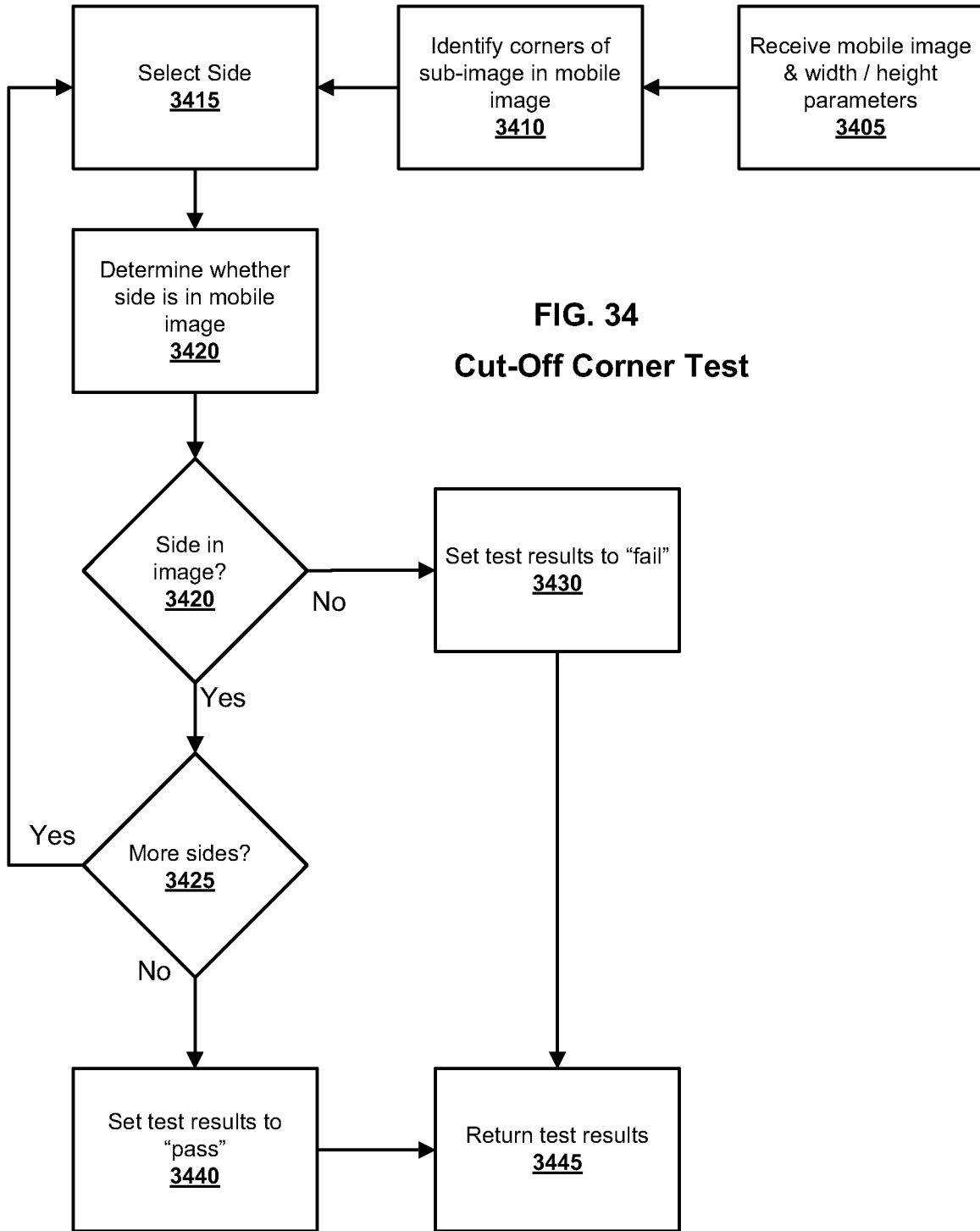


FIG. 33



**FIG. 34**  
**Cut-Off Corner Test**

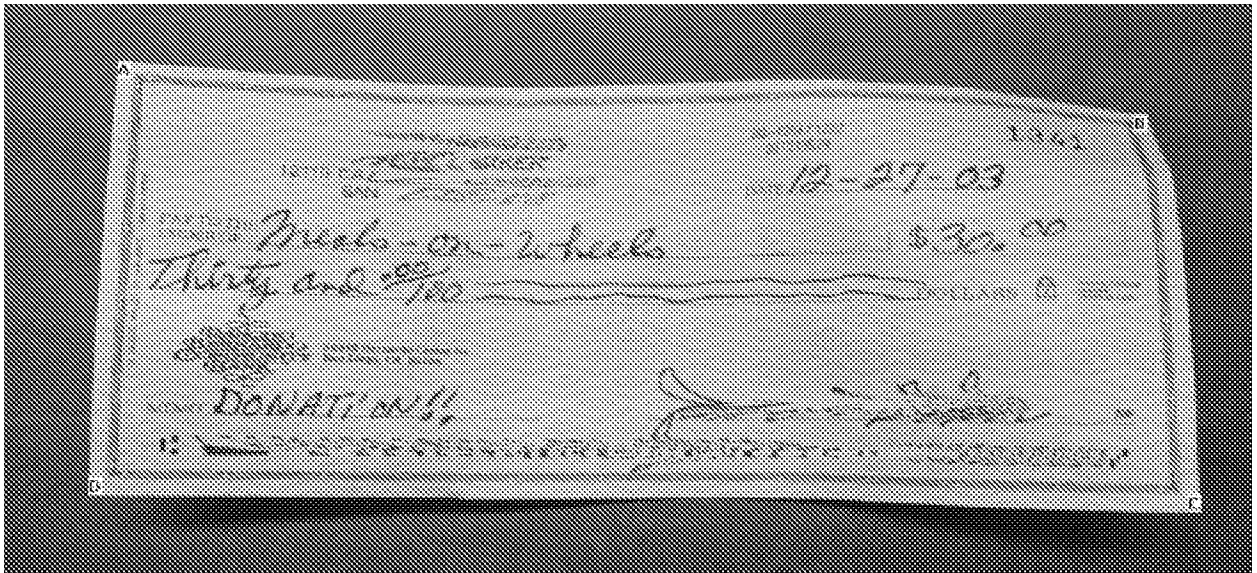
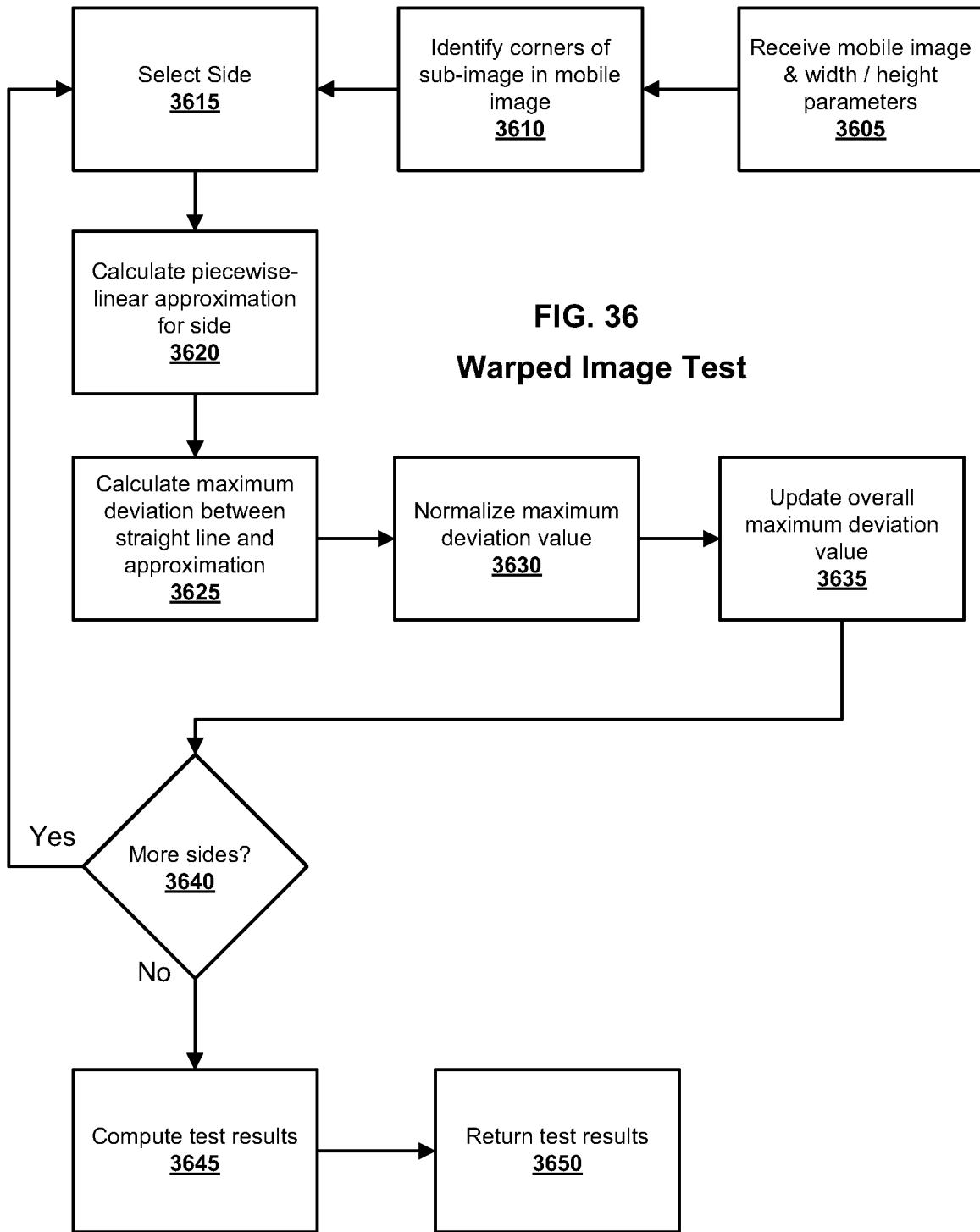


FIG. 35

This figure is intentionally blurry.



**FIG. 36**  
**Warped Image Test**

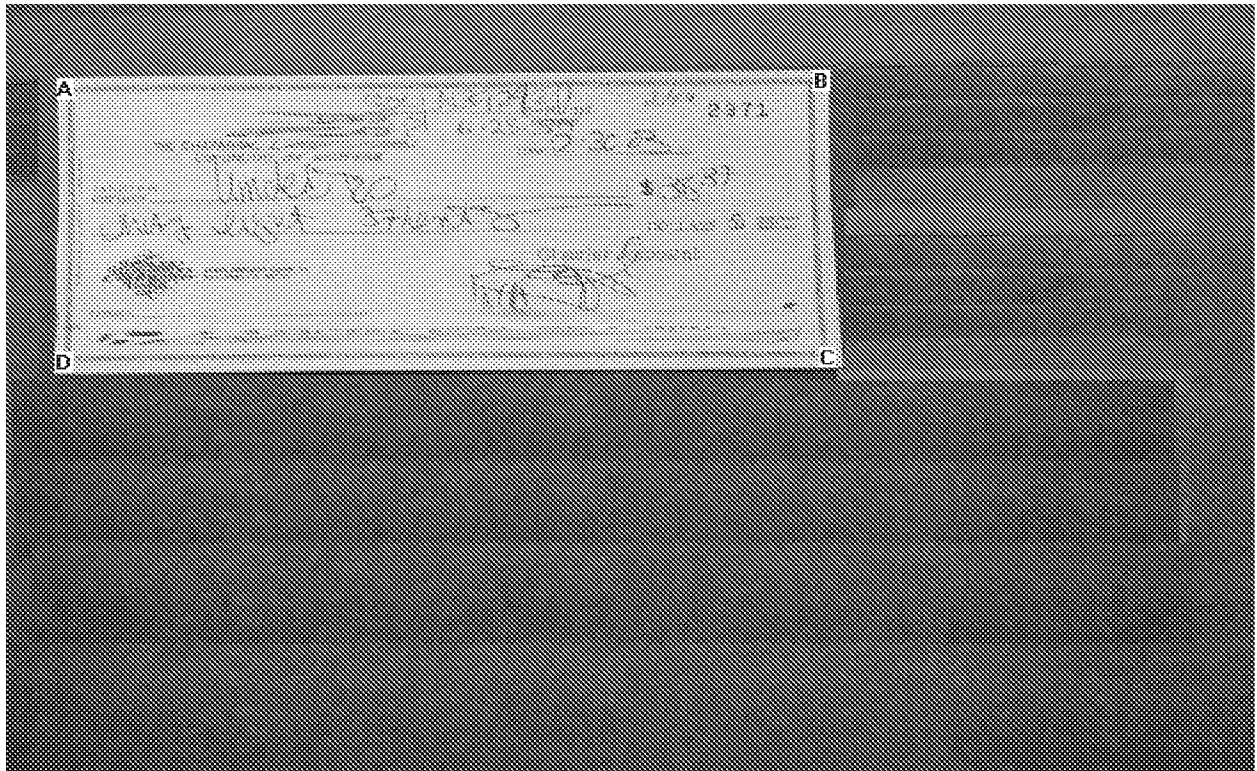
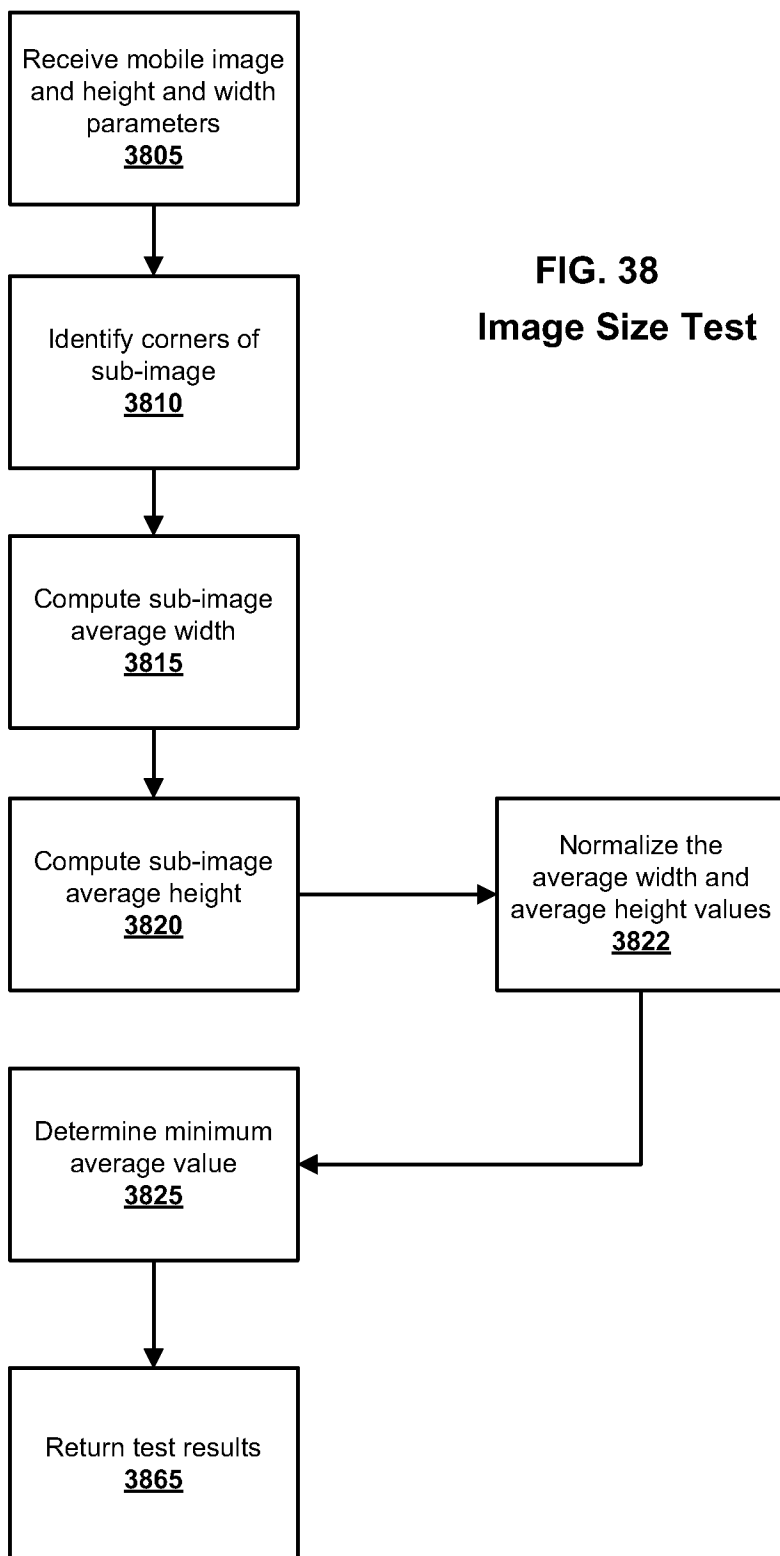
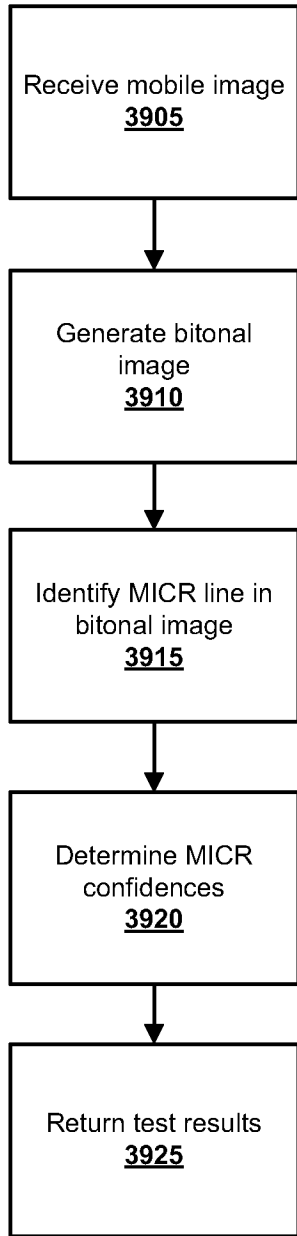
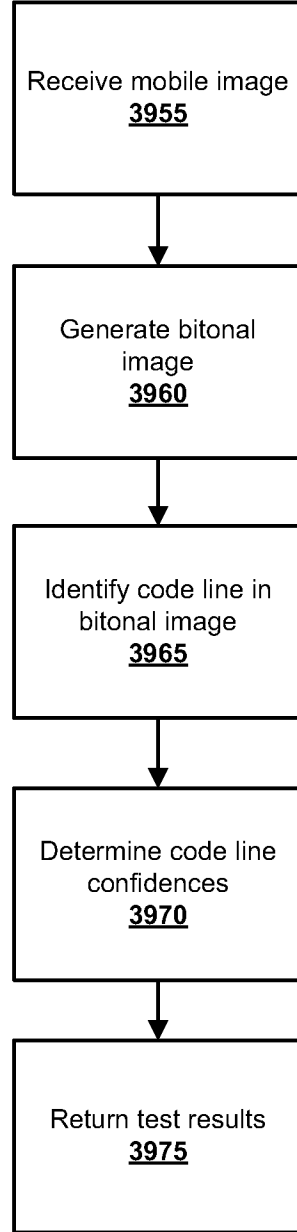


FIG. 37



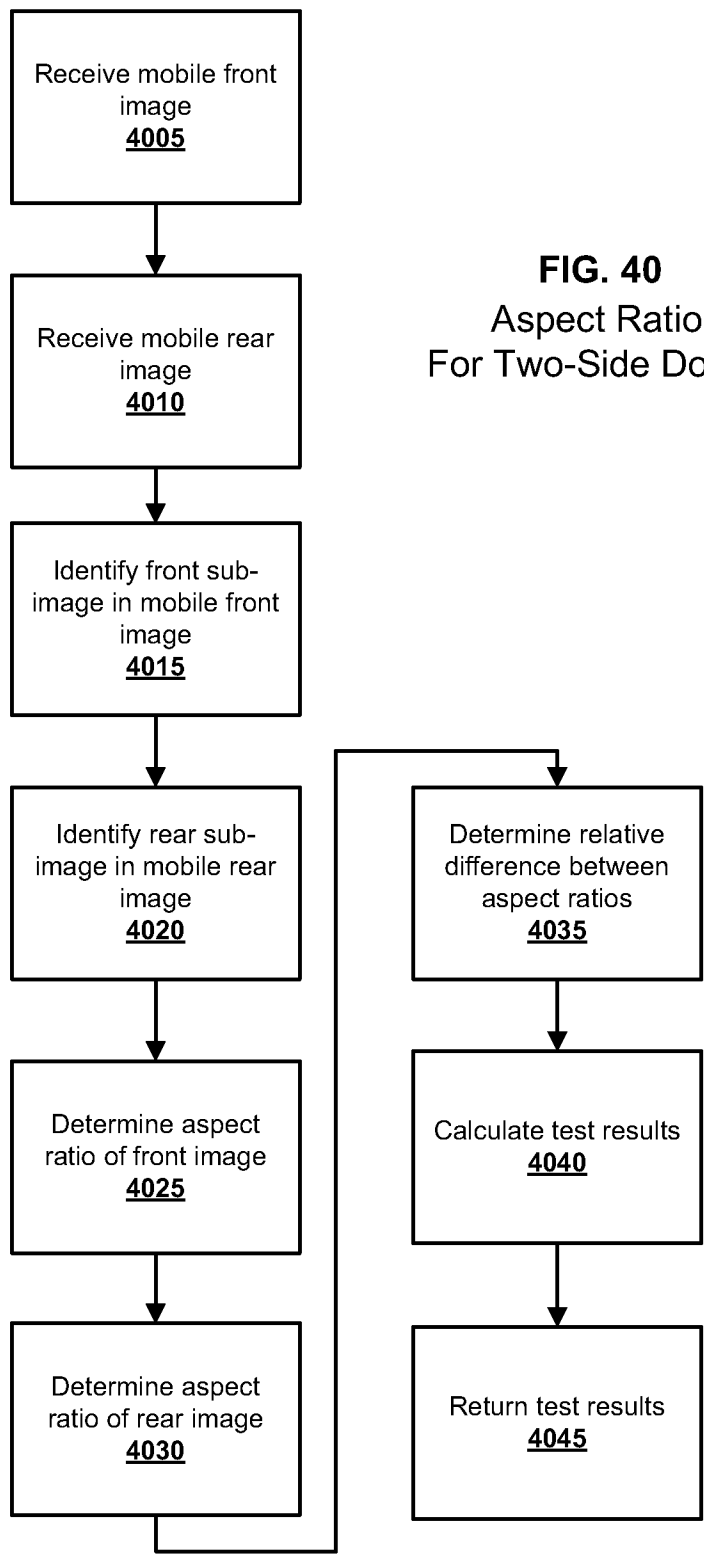


**FIG. 39A**  
MICR-line Test

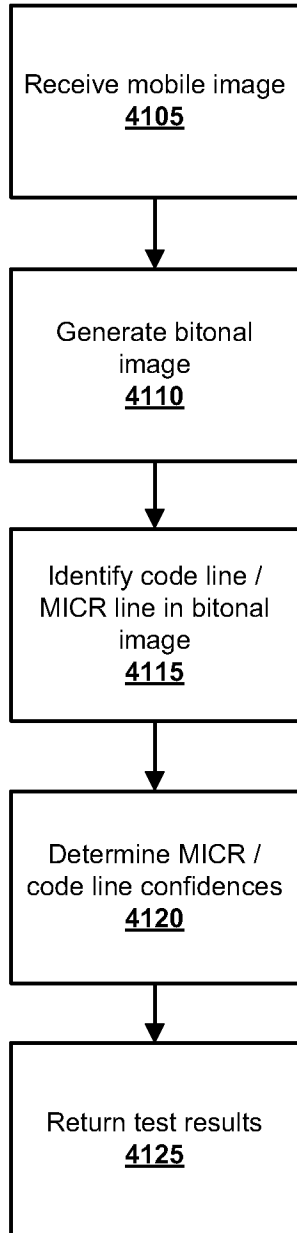


**FIG. 39B**  
Code line Test





**FIG. 40**  
Aspect Ratio Test  
For Two-Side Documents



**FIG. 41**  
Front-as-Rear Test

Receive bi-tonal  
image of document  
4205

Execute matching  
4210

Matching  
template?  
4215  
N

Extract data from  
document using topic  
information and  
associate data  
with field values  
4220

Perform dynamic  
data capture to  
extract information  
from document 4225

FIG. 42  
Form Identification  
and Data Extraction

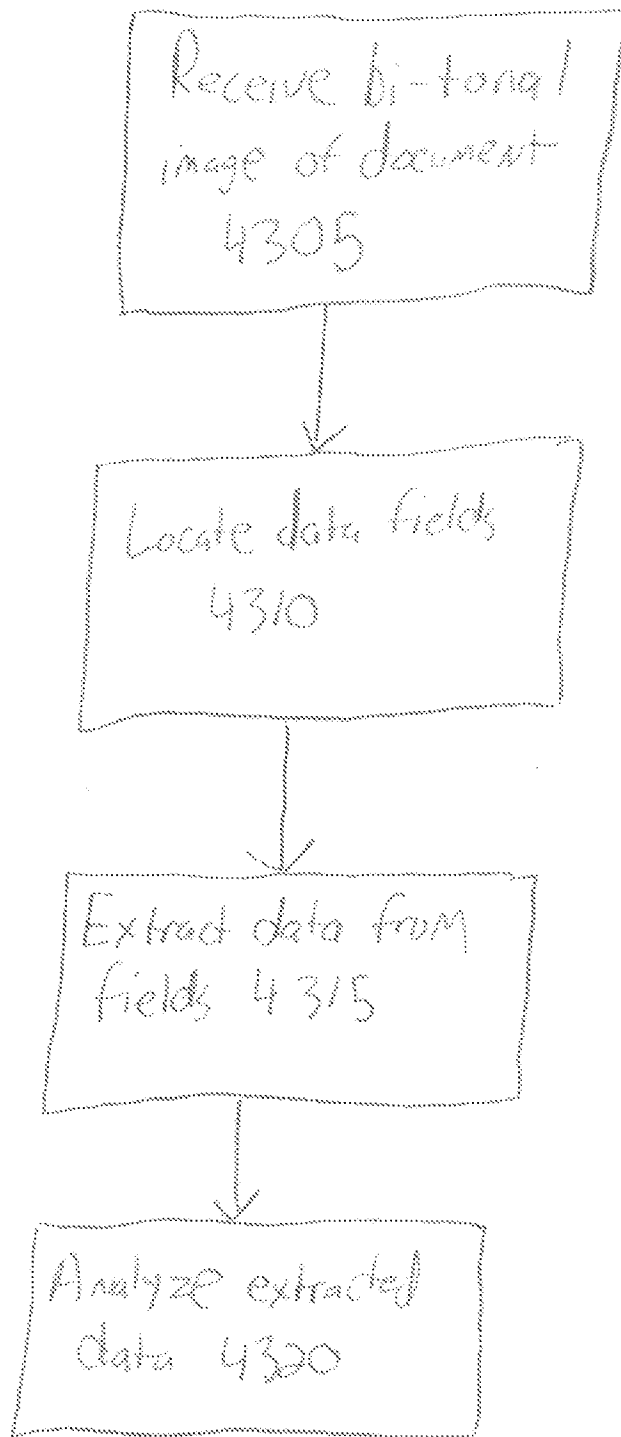
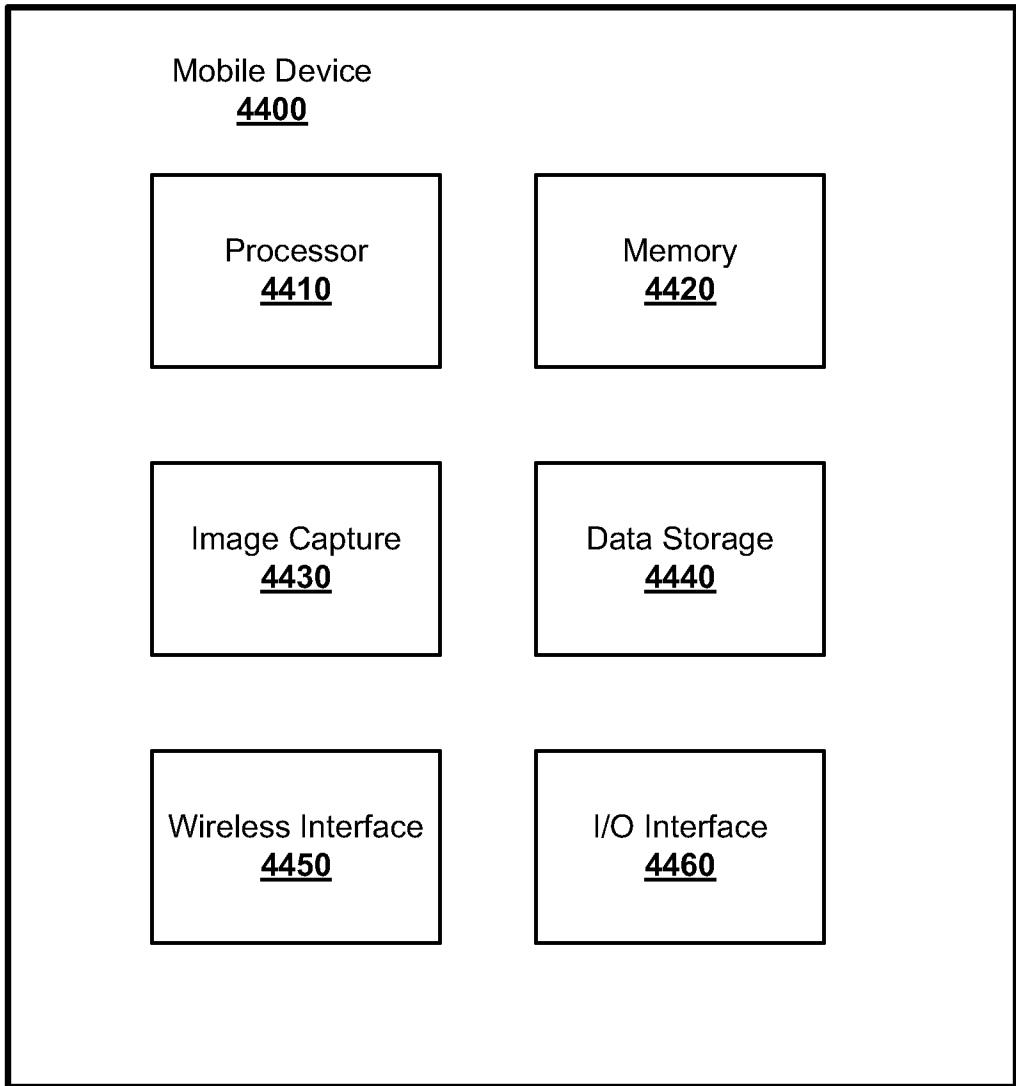


FIG. 43  
Dynamic Data Capture



**FIG. 44**

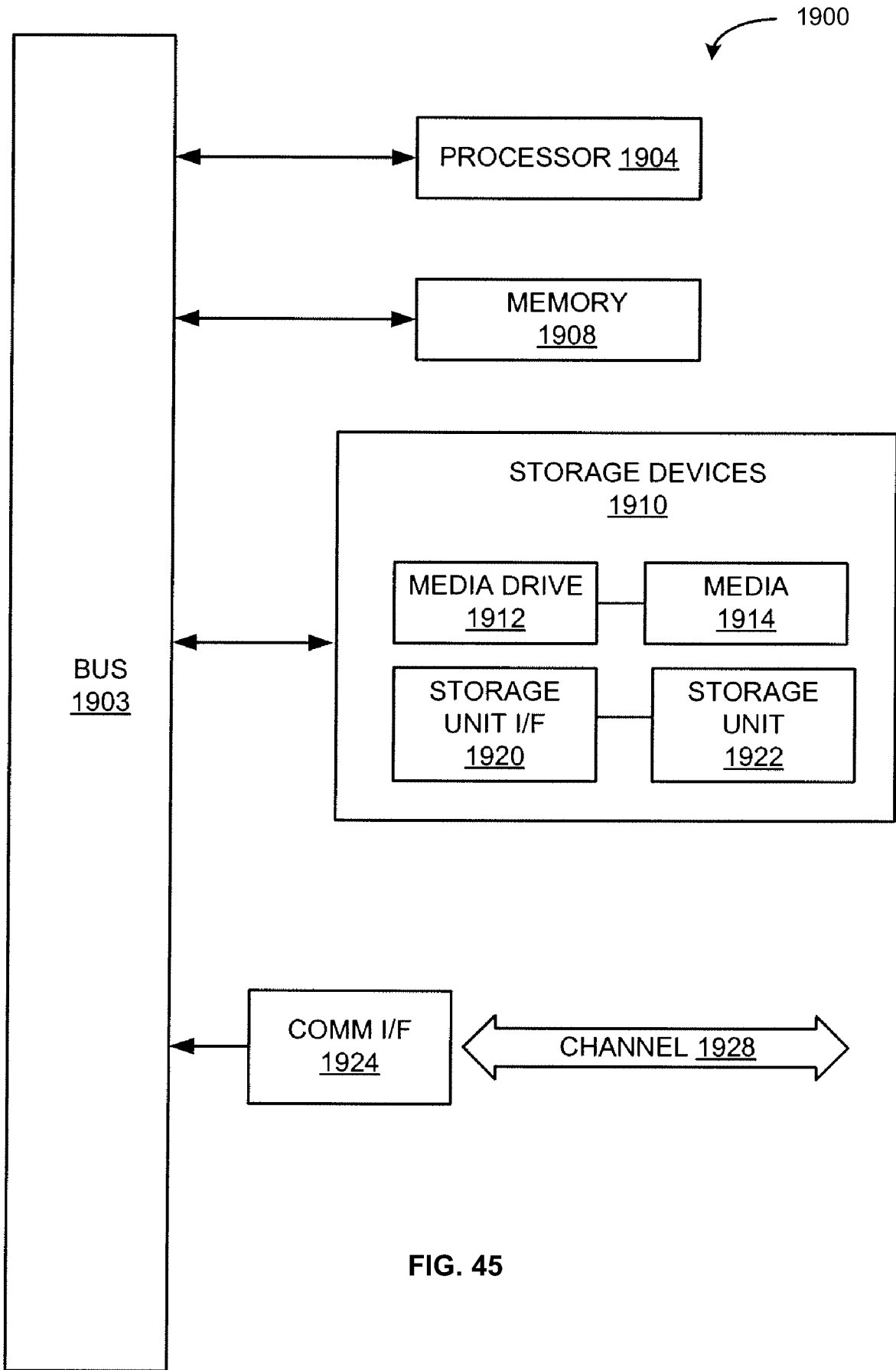


FIG. 45

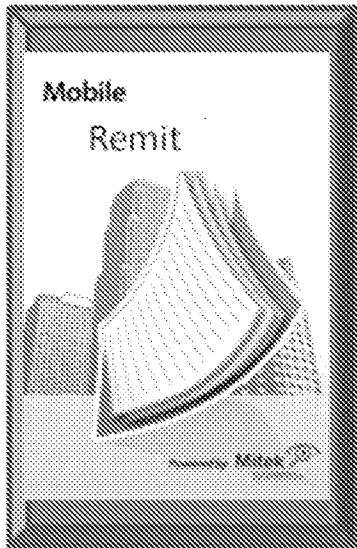


FIG. 46A

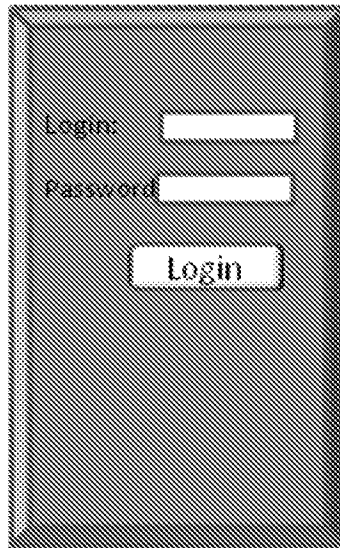


FIG. 46B

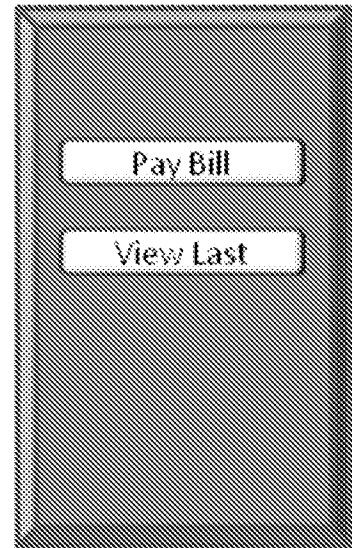


FIG. 46C

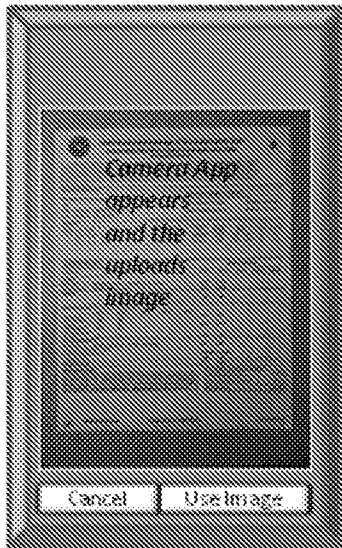


FIG. 46D

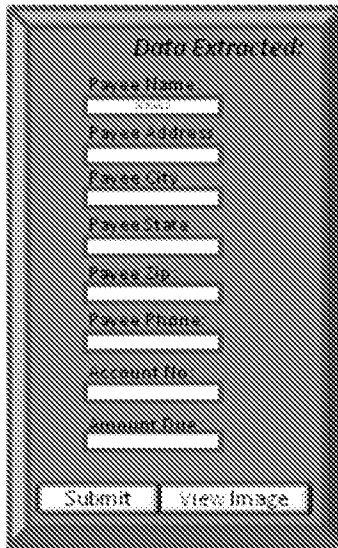


FIG. 46E

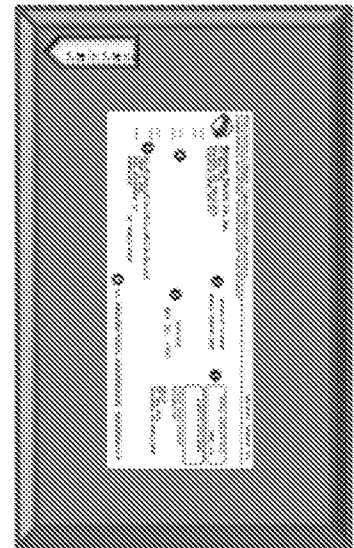


FIG. 46F



FIG. 47

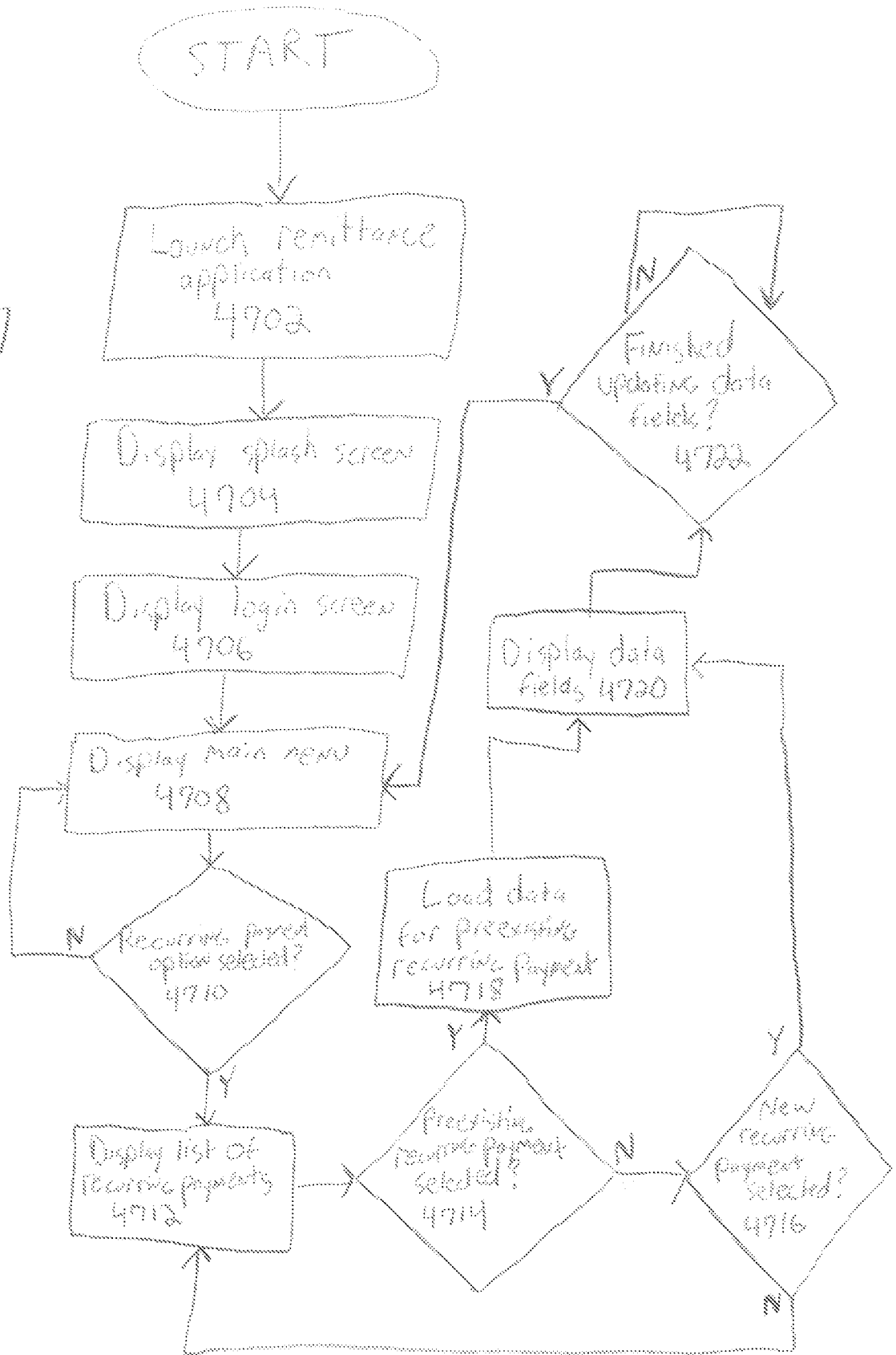


FIG. 48

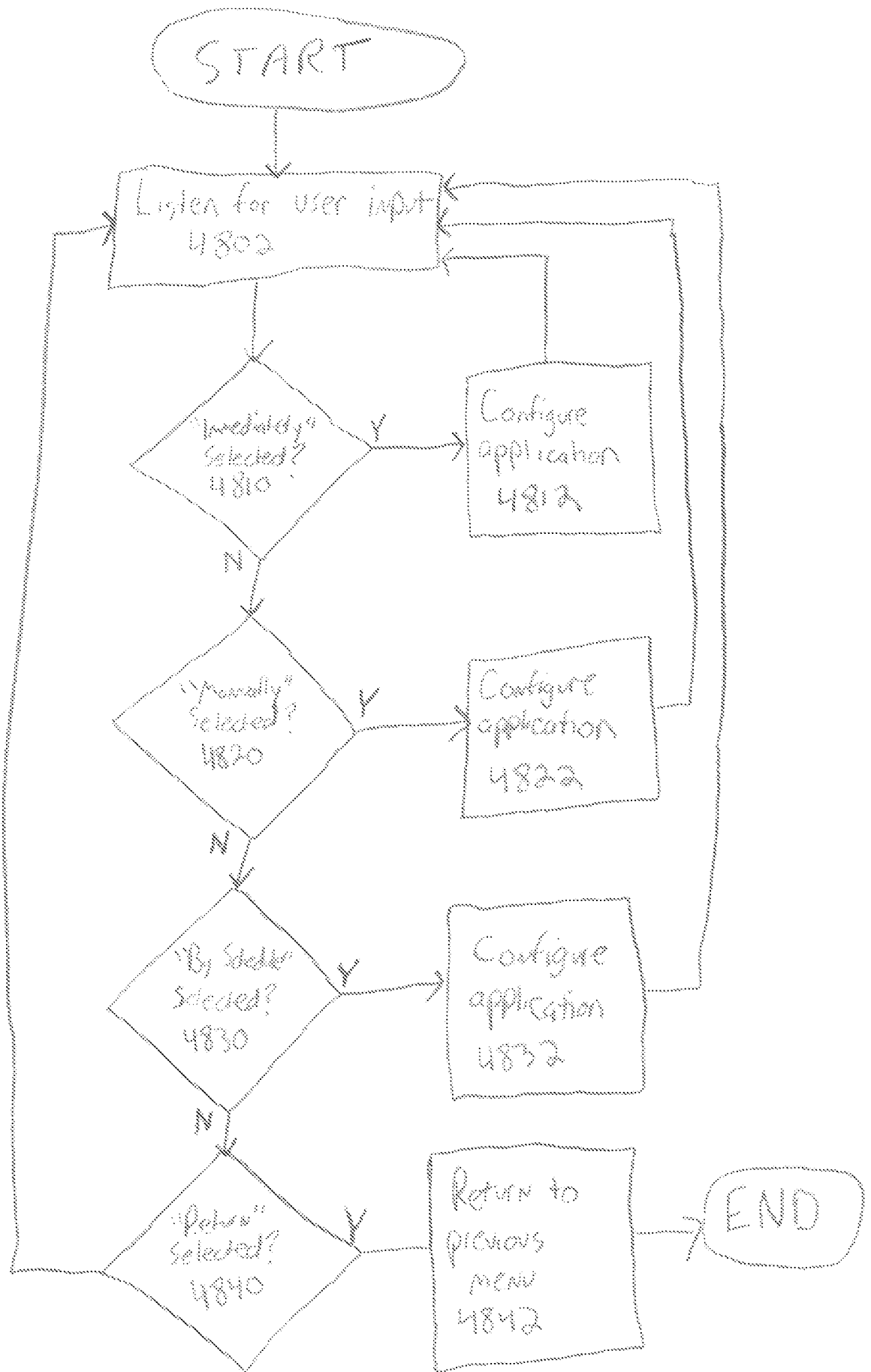


FIG. 49

