Method for Requesting and Viewing a Zoomed Area of Detail from an Image Attachment on a Mobile Communication Device

[0001] The following is directed in general to displaying content on mobile communication devices, and more particularly to a method for viewing an area of detail from an image attachment on a mobile communication device.

[0002] Mobile communication devices are becoming increasingly popular for business and personal use due to a relatively recent increase in number of services and features that the devices and mobile infrastructures support. Handheld mobile communication devices, sometimes referred to as mobile stations, are essentially portable computers having wireless capability. and come in various forms. These include Personal Digital Assistants (PDAs), cellular phones and smart phones. While their reduced size is an advantage to portability, limitations in download bandwidth and device screen size give rise to challenges in viewing large images having high horizontal and vertical resolution (in pixels). Examples include digital pictures and fax attachments to an email, which typically can have resolutions in the 2000 x 2000 pixel range.

[0003] For wireless devices that support viewing of image attachments. this represents a large amount of information for downloading, requiring a large amount of bandwidth and associated cost thereof. Additionally, the user must wait for an extended period of time while the device is downloading the image.

[0004] Since mobile communication devices have limited screen real estate, the downloaded image must be resized on the device in order to be viewable. However. such a resizing operation requires a very large memory and CPU usage thereby affecting the performance of all other applications on the device for the duration of the resizing process.

[0005] Companies such as Mapquest® allow users to zoom in on a map from a web browser client such that the map regenrates with a higher amount of detail. This zoom feature is accomplished in a web client/server implementation using vector graphics, which is not applicable to viewing wireless image attachments.

EP1202188 discloses a method and system for viewing an enlarged area of a server stored original image on a mobile communication device. The method comprises: transmitting an image size limit from said mobile communication device to

said server indicative of screen size of said mobile communication device: downloading and displaying said original image from said server to said mobile communication device, said image being re-sized in the server in the event that the dimensions thereof exceed said image size limit; determining crop rectangle coordinates of an area of the displayed image to be enlarged; sending a request from the mobile communication device to the server to enlarge the area based on the determined coordinates: in response to receiving said request, collecting and modifying image binary data from said image to create said enlarged area of said original image; and downloading said enlarged area of said original image to said mobile communication device. The crop rectangle coordinates are determined by a user of the device from coordinates displayed around the periphery of the displayed image. Such coordinated are then manually inputted to the device by depressing appropriate buttons. It is necessary therefore that the coordinates displayed around the periphery of the displayed image have a sufficiently large granularity as to be viewable by a user.

Summary

According to an aspect of the invention, a method is provided for allowing a user to quickly view an image attachment downloaded from a server to a mobile communication device. Unlike the prior art systems discussed above, the method of the invention relates to wireless image attachment viewing, and the server side processing is performed on bitmap data and not vectors.

When a user issues a request to view a large image such as a fax or digital photo image, the server resizes the image to match the requesting device display resolution. For example, if the user requests a large fax image (e.g. 2400x1800 pixel resolution), the client device (i.e. mobile communication device) informs the server of its maximum resolution (e.g. 240x240 pixels), in response to which the server resizes the image and returns the reduced resolution image to the device for viewing. This significantly reduces the amount of data downloaded wirelessly to the device as well as device memory consumed and CPU usage required to display the image.

Additionally, the server adjusts the number of colors in the image to match the color display capabilities of the requesting device. For example, only monochrome images will be returned to a monochrome device even if the requested image is in color, thereby further reducing the amount of data downloaded wirelessly to the device.

One consequence of the server image resizing operation is decreased legibility of textual information and image details when viewed on the device, due to the reduced image resolution. Therefore, according to an additional aspect of the invention, the client includes an "Enlarge Area" feature by which the user may zoom in on an area of detail of the resized image being viewed on the device, and request the server to provide higher resolution of the enlarged detail. The enlarge area operation is performed in an iterative fashion making it possible to zoom in on a returned area of detail previously requested and thereby obtain an even higher area of detail.

In a main aspect, the present invention provides a process for viewing an enlarged area of a server stored original image on a mobile communication device, comprising: transmitting an image size limit from said mobile communication device to said server indicative of screen size of said mobile communication device; downloading and displaying said original image from said server to said mobile communication device, said image being re-sized in the server in the event that the dimensions thereof exceed said image size limit; determining crop rectangle coordinates of an area of the displayed image to be enlarged; sending a request from the mobile communication device to the server to enlarge the area based on the determined coordinates; in response to receiving said request, collecting and modifying image binary data from said image to create said enlarged area of said original image; and downloading said enlarged area of said original image to said mobile communication device, the process being characterised in that: it includes downloading original size data for said original image; and in that the step of determining coordinates for said area to be enlarged comprises calculating in the communications device the crop rectangle coordinates based on an image zoom area defined on said mobile communication device and translating said crop rectangle coordinates based on said downloaded original size data; and said server extracts said translated crop rectangle coordinates; and collects and modifies said image binary data from said image based on said translated crop rectangle coordinates and said image size limit to create said enlarged area of said original image.

Additional aspects and advantages will be apparent to a person of ordinary skill in the art, residing in the details of construction and operation as more fully hereinafter described and claimed, reference being had to the accompanying drawings.

Brief Description of the Drawings

[0011] A detailed description of the preferred embodiment is set forth in detail below, with reference to the following drawings, in which:

Figure 1 is a block diagram of a network environment in which the preferred embodiment may be practiced:

Figure 2 is a screen display of a menu for selecting the enlarge area command according to the preferred embodiment;

Figure 3 shows image crop rectangle coordinates established at the device and sent to the server;

Figure 4 is a flowchart showing an enlarge area request function on the device, according to the preferred embodiment;

Figure 5 is a tree diagram showing the basic structure of a Document Object Model (DOM) used in the preferred embodiment;

Figure 6 shows the top-level of the DOM structure in Figure 5;

Figure 7 shows an exemplary DOM structure for a word processing document;

Figure 8 shows an exemplary DOM structure for a table document;

Figure 9 shows an exemplary DOM structure for a word processing document containing an image subdocument; and

Figure 10 is a flowchart showing enlarge area image processing on the server, according to the preferred embodiment.

Detailed Description of the Preferred Embodiment

[0012] With reference to Figure 1, network environment 10 is shown in which the preferred embodiment may be practiced. Network environment 10 includes at least one mobile communication device 12 communicating via a wireless network/Internet 14 to a server 28, via a firewall security server 29, for downloading document attachments to the devices 12. While only one server 28 is shown for illustration purposes, a person of skill in the art will understand that network environment 10 could have many such servers for hosting web sites or graphic download sites, providing access to picture files such as

JPEG, TIFF, BMP, PNG, SGI, MP4, MOV, GIF, SVG, etc. As would be understood by one of ordinary skill in the art, wireless networks 14 include GSM/GPRS, CDPD, TDMA, iDEN Mobitex, DataTAC networks, or future networks such as EDGE or UMTS, and broadband networks like Bluetooth and variants of 802.11.

[0013] As discussed above, when the user of a mobile communication device 12 views an image attachment using an "Attachment Viewer" application, a server side resized version of the original image is downloaded to the device unless the original image has a smaller width and height than the requesting device screen size. The server 28 resizes the image to be viewed based on the requesting device screen width and height (in pixels). When the server 28 returns the resized image back to the device 12 for viewing, the dimensions (width x height) of the original image are also returned to the device. The original dimensions are required for calculating the cropping rectangle for the "Enlarge Area" command, as discussed in greater detail below.

[0014] Specifically, while viewing an image in the Attachment Viewer the user can select to zoom in on an image, which effectively magnifies the existing pixels that constitute the image on the device 12. The "Enlarge Area" command causes the server 28 to generate a new image based on the coordinates from the zoomed in image on the device 12 and returns the new image to the device 12 for display.

[0015] With reference to Figure 2, the "Enlarge Area" menu item is displayed in the device menu if the following conditions are met: (A) there is no pending "Enlarge Area" request in process for the currently viewed image and (B) the crop rectangle based on the currently zoomed image area on the device is calculated to equate to the original image coordinates as it resides on the server in pixels and the calculated crop rectangle width is greater then the device screen width or the calculated crop rectangle height is bigger than the device screen height.

[0016] Provided the foregoing conditions are met, the Attachment Viewer application on device 12 issues a request to the server 28 for the Enlarge Area operation to be performed on the original image stored on the server.

[0017] When sending the request, the Attachment Viewer application includes a cropping rectangle for the original image stored on the server 28, which corresponds to the currently viewed zoomed image area on the device 12. Thus, with reference to Figure 2,

the cropping rectangle is the viewable area (left-hand side) that the user wishes to see returned from the server 28 with higher quality/resolution (right-hand side).

[0018] The crop rectangle coordinates are the zoomed area coordinates on the device 12 calculated so they translate to the equivalent crop rectangle for the original image stored on the server 28, as shown in Figure 3.

[0019] The crop rectangle calculation also correctly allows for device side operations such as "Rotate" of the device side image prior to the "Enlarge Area" command being issued. The Attachment Viewer also sends the device screen characteristics of width, height and depth along with the calculated crop rectangle for the original image when sending the Enlarge Area command to the server 28. The Attachment Server application then returns the new image resulting from the enlarge area command for matching these device characteristics.

[0020] Figure 4 shows the device side "Enlarge Area" command request procedure of the Attachment Viewer application running on device 12. At step 30, the Attachment Viewer displays an image attachment downloaded from the server 28. As discussed above, the image is re-sized in the server if the dimensions exceed the screen size of the viewing device 12. Also, as discussed above, the server 28 returns the original image size to the device for calculating the crop rectangle.

[0021] Next, at step 32, the application determines whether precondition (A), above, is met for generating the "Enlarge Area" menu item (i.e. whether there is any pending Enlarge Area request in progress).

[0022] The crop rectangle coordinates are then calculated based on the image zoom area defined on the device and translated using the original image dimensions to crop rectangle coordinates for the original image stored on the server 28 (step 34).

[0023] Condition (B), discussed above, is then tested at step 36 and, provided that the image width or height exceeds the device display width or height, respectively, then the "Enlarge Area" menu is displayed (step 37), and a determination is made as to whether the "Enlarge Area" command has been invoked (step 38).

[0024] At step 40, the device sends a request to the server 28 to enlarge the displayed image based on the translated crop rectangle coordinates and the device screen

dimensions.

[0025] If the enlarged image is received from the server 28 (a YES at decision block 41), then process control returns to step 30. Otherwise, an error code is displayed (step 42) and the process terminates (step 43).

[0026] Before turning to the server-side application for image area enlargement, a description is provided herein of the structure of a Document Object Model (DOM) for a document attachment to be viewed on the mobile communication device 12.

[0027] The attachment server 28 uses a file-parsing distiller in the preferred embodiment, for a specific document type, to build an in-memory Document Object Model (DOM) structure representing an attachment of that document type. The document DOM structure is stored in a memory cache of server 28, and can be iterated bi-directionally.

[0028] As shown in Figure 5, the graph-based document DOM structure consists of nodes and leaves. The nodes serve as the parents of leaves and nodes, while leaves are end points of a branch in the graph. Each node and leaf can have a set of attributes to specify its own characteristics. For example, a paragraph node can contain attributes to specify its alignment, style, entry of document TOC, etc. In addition, each of the nodes and the leaves has a unique identifier, called a DOM ID, to identify itself in the document DOM structure.

[0029] The document DOM structure is divided into three parts: top-level, component and references. The top level refers to the document root structure, while the main document is constructed in the component and the references represent document references to either internal or external sub-document parts. The following paragraphs examine each part in detail.

[0030] The root node of a document DOM structure, referred to as "Document", contains several children nodes, referred to as "Contents", which represent different aspects of the document contents. Each "Contents" node contains one or multiple "Container" nodes used to store various document global attributes. The children of the "Container" nodes are components, which store the document structural and navigational information. When the attachment server 28 builds the DOM structure for an attachment file for the first time, the top-level structure is a single parent-child chain as shown in

Figure 6:

[0031] Three types of components are defined by the attachment server 28: text components, table components and image components, which represent text, tables and images in a document, respectively. The text and table components are described in detail below, and the image component structure is identical.

[0032] A component consists of a hierarchy of command nodes. Each command represents a physical entity, a property, or a reference defined in a document. For the text component, the physical entity commands are page, section, paragraph, text segments, comments, footnote and endnote commands, which by name define the corresponding entity contained in a document. The property commands for the text component are font, text color, text background color, hyperlink start/end and bookmark commands. The text component has only one reference command, referred to as the text reference command, which is used to reference a subdocument defined in the main body of a document. Usually, the children of a text component are page or section command nodes that, in turn, comprise a set of paragraph command nodes. The paragraph command can contain one or multiple nodes for the remaining command types.

[0033] Using the following sample text document, the corresponding document DOM structure is shown in Figure 7:

```
First paragraph.
Second paragraph with bold and red text.
```

[0034] As Figure 7 demonstrates, the section command, which is the child of the text component, consists of two paragraph commands. The first paragraph command contains one text segment command and the text content for that paragraph is added as an attribute to the text segment command. The second paragraph command has a relatively more complex structure, as the text properties in the paragraph are much richer. Each time a text property (font, text color, etc) changes, a corresponding text property command is created and the change value is added to that command as an attribute. The subsequent text segment command records the text with the same text property as an attribute. As document structure gets richer and more complex, more commands of corresponding types are created and the document properties are added as attributes to those commands.

[0035] The table component has the same three types of commands as the text component, but different command names. The document DOM structure for the sample table document below is shown in Figure 8:

Cell One	Cell Two
Cell Three	Cell Four

[0036] As shown in the Figure 8, the table component has physical entity type commands of table, tablerow and tablecell, where the tablecell command can contain all available commands for the text component. In the example above, the first child TableRow command of the table command has an attribute "Index" defined by value of 0. This indicates that the indicated table row is the first one defined in the table. The attribute of the leftmost table cell command in Figure 8 has the same meaning.

[0037] A document sometimes contains subdocuments, for example images, tables, text boxes etc. The DOM structure set forth herein uses a reference command to point to the graph of such subdocuments. Thus, for the following sample document, the attachment server 28 generates the DOM structure shown in Figure 9:

This document has subdocument of images like this one .

Second paragraph contains the same image .

The structure shown in Figure 9 is identical to that discussed above in connection with Figures 7 and 8, except for the attributes of the two reference commands. The attachment server 28 constructs the image in "Sample Three" as a separate image component, which contains all of the image data in its own DOM hierarchy. In the DOM structure for the main document, the values of the "Ref" attributes of those two reference commands point to the image component, as indicated by the dashed lines, such that the DOM structure connects together all parts of the document.

[0038] Returning now to the preferred embodiment with reference to Figure 10, when the attachment server 28 receives an "Enlarge Area" image request for an attachment (step 44), it first extracts the screen size information for the mobile communication device 12 and the crop rectangle parameters from the request (step 46). After retrieving the document DOM structure for the attachment from the in-memory document DOM cache

(step 48), the server 28 traverses through the document DOM structure to find out the corresponding image component in the DOM structure for that image (step 50). The server 28 then iterates through the attributes of the image component to determine if a separate image component specifically for that screen size has already been constructed (step 52).

[0039] This separate image component contains the same graphic information as the original image but is usually of smaller size (equal to or less than the device screen size), thus consuming much less device memory (usually in ratio of 1:15 to 1:20). If the separate image component does not exist, the attachment server 28 collects the image binary data from the original image component in memory (Step 54) and constructs the new image component from the binary data (step 56).

[0040] At step 58, the server 28 caches the DSS image component as an attribute of the original image component in the DOM structure. These two image components contain essentially the same graphic information except that the original image component has a much higher resolution. A person skilled in the art will appreciate that this solution will consume appreciable memory if there are a lot of different mobile device DSS definitions. Therefore, as an alternative, it would be possible to construct the DSS image component each time the server receives an "Enlarge Area" request, without caching it. However, based on the premise that time is a more critical factor in wireless operation than the hardware deployment cost, caching the DSS image component is employed in the preferred embodiment.

[0041] Similar to the operation of the separate image component construction, the attachment server 28 creates a new image data segment in memory by tailoring the original image binary data based on the crop rectangle parameters (step 60), and resizes the new image data segment based on the device screen size to create the final image data segment (step 62).

[0042] In summary, the "Enlarge Area" device/server feature allows users to quickly retrieve any relevant part of a large image attachment that has been resized by the server. This minimizes bandwidth usage, device memory/CPU consumption, and request/response latency while still allowing the user to view an image area in its original (native) level of detail.

[0043] A person skilled in the art, having read this description of the preferred

embodiment, may conceive of variations and alternative embodiments.

[0044] All such variations and alternative embodiments are believed to be within the ambit of the claims appended hereto.

Claims:

1. A process for viewing an enlarged area of a server stored original image on a mobile communication device (12), comprising:

transmitting an image size limit from said mobile communication device (12) to said server (28) indicative of screen size of said mobile communication device;

downloading and displaying said original image from said server to said mobile communication device, said image being re-sized in the server in the event that the dimensions thereof exceed said image size limit;

determining crop rectangle coordinates of an area of the displayed image to be enlarged:

sending a request from the mobile communication device to the server to enlarge the area based on the determined coordinates;

in response to receiving said request, collecting and modifying image binary data from said image to create said enlarged area of said original image; and

downloading said enlarged area of said original image to said mobile communication device,

the process being characterised in that:

it comprises downloading from the server to the mobile communications device original size data for said original image;

said step of determining coordinates for said area to be enlarged comprises calculating in the communications device the crop rectangle coordinates based on an image zoom area defined on said mobile communication device and translating said crop rectangle coordinates based on said downloaded original size data: and

said server extracts said translated crop rectangle coordinates: and collects and modifies said image binary data from said image based on said translated crop rectangle coordinates and said image size limit to create said enlarged area of said original image.

2. The process of claim 1, further comprising:

building a graph structure within said server representing a map of said original image;

retrieving and traversing said graph structure within the server (28) to locate any corresponding image component for said original image and, upon locating said corresponding image component iterating through attributes of the image component to determine if a separate image component has been constructed for said image size limit;

in the event that said separate image component has not been constructed then collecting initial image binary data from said original image within said server and constructing a new image component from said binary data; and

caching said new image component as an attribute of the original image in said graph structure.

3. The process of claim 1 or claim 2, further comprising generating a menu selection at said mobile communication device (12) for user enabling of said calculating of the crop rectangle coordinates, said menu selection being generated in the event predetermined conditions are met.

4. The process of claim 3, wherein said predetermined conditions comprise:

absence of any request being processed for viewing an enlarged area of said original image; and any of:

width of said original image being greater than screen width of said mobile communication device (12) and said original image being re-sized so as to exceed said screen width; or

height of said original image being greater than screen height of said mobile communication device and said original image being re-sized so as to exceed said screen height.

5. The process of any one of claims 1 to 4, wherein said graph structure is a Document Object Model (DOM).

6. A process of enlarging an area of an original image stored on a server (28), the process comprising the steps at the server of:

receiving an image size limit for an original image stored on said server (28);

transmitting said original image for display, said image being re-sized in the event that the dimensions thereof exceed said image size limit;

receiving a request to enlarge the displayed image comprising crop rectangle coordinates of an area of said original image to be enlarged;

collecting and modifying image binary data from said image based on said received coordinates to create said enlarged area of said original image; and

transmitting said enlarged area of said original image for display

the process being characterized in that:

it comprises transmitting original size data for said original image;

and in that said server extracts from said received request translated crop rectangle coordinates, said translated crop rectangle coordinates being derived from said area to be enlarged and said original size data; and collects and modifies said image binary data from said image based on said translated crop rectangle coordinates and said image size limit to create said enlarged area of said original image.

7. The process of claim 6, further comprising:

building a graph structure representing a map of said original image;

retrieving and traversing said graph structure to locate any corresponding image component for said original image and, upon locating said corresponding image component iterating through attributes of the image component to determine if a separate image component has been constructed for said image size limit;

in the event that said separate image component has not been constructed then collecting initial image binary data from said original image and constructing a new image component from said binary data: and

caching said new image component as an attribute of the original image in said graph structure.

8. The process of claim 7, wherein said graph structure is a Document Object Model (DOM).

9. A process for displaying on a mobile communications device (12) an enlarged area of an original image displayed on said device, the process comprising the steps at the mobile communications device (12) of:

transmitting an image size limit indicative of screen size of said mobile communication device;

displaying said original image, said image being re-sized in the event that the dimensions thereof exceed said image size limit;

determining crop rectangle coordinates of an area of said displayed image to be enlarged;

sending a request from the mobile communication device to enlarge an area of the image displayed on said device based on the determined coordinates;

receiving an enlarged area of said original image comprising modified binary data based on said determined coordinates; and

displaying said enlarged area of said original image on said screen

the process being characterized in that:

it comprises downloading original image size data of the original image;

and in that the step of determining coordinates for said area to be enlarged comprises calculating in the communications device the crop rectangle coordinates based on an image zoom area defined on said mobile communication device and translating said crop rectangle coordinates based on said downloaded original size data; and

transmitting said translated crop rectangle coordinates in said request to enlarge an area of the image displayed on said mobile communications device.

10. The process of claim 9, further comprising generating a menu selection for user enabling of said calculating of the crop rectangle coordinates, said menu selection being generated in the event predetermined conditions are met.

11. The process of claim 10, wherein said predetermined conditions comprise:

absence of any request being processed for viewing an enlarged area of said original image; and any of:

width of said original image being greater than screen width of said mobile communication device and said original image being re-sized so as to exceed said screen width; or

height of said original image being greater than screen height of said mobile communication device and said original image being re-sized so as to exceed said screen height.

12. A server (28) for enlarging an area of an original image stored on the server, the server comprising:

means for receiving an image size limit for an original image stored on said server;

means for transmitting said original image for display, said image being re-sized in the event that the dimensions thereof exceed said image size limit;

means for receiving a request to enlarge the displayed image comprising crop rectangle coordinates of an area of said original image to be enlarged;

means for collecting and modifying image binary data from said image based on said received coordinates to create said enlarged area of said original image; and

means for transmitting said enlarged area of said original image for display

the server being characterized in that:

it comprises means for transmitting original size data for said original image; means for extracting from said received request translated crop rectangle coordinates, said translated crop rectangle coordinates being derived from said area to be enlarged and said original size data; and in that said means for collecting and modifying image binary data collects and modifies said image binary data from said image based on said translated crop rectangle coordinates and said image size limit to create said enlarged area of said original image.

13. Computer code embodied on a machine readable medium of the server (28) of claim 12, said code being executable in a processor of said server to implement the method of any of claims 6 to 8.

14. A mobile communications device (12) for displaying an enlarged area of an original image displayed on said device, the device comprising:

means for transmitting an image size limit indicative of screen size of said mobile communication device;

means for downloading an original image;

means for displaying said original image, said image being re-sized in the event that the dimensions thereof exceed said image size limit;

means for determining crop rectangle coordinates of an area of said displayed image to be enlarged;

means for sending a request from the mobile communication device to enlarge an area of the image displayed on said device based on the determined coordinates;

means for receiving an enlarged area of said original image comprising modified binary data based on said determined coordinates; and

means for displaying said enlarged area of said original image on said screen

the device being characterized in that:

it comprises means for downloading original size data for said original image;

and in that the means for determining coordinates for said area to be enlarged comprises means for calculating the crop rectangle coordinates based on an image zoom area defined on said mobile communication device and translating said crop rectangle coordinates based on said downloaded original size data; and

said means for sending a request is arranged to transmit said translated crop rectangle coordinates in said request to enlarge an area of the image displayed on said mobile communications device.

15. Computer code embodied on a machine readable medium of the device (12) of claim 14, said code being executable in a processor of said device to implement the method of any of claims 9 to 11.

16. A telecommunications network (10) comprising the server (28) of claim 12 and at least one mobile communications device (12) in accordance with claim 14.