



(12) 发明专利申请

(10) 申请公布号 CN 101860734 A

(43) 申请公布日 2010.10.13

(21) 申请号 200910043056.4

(22) 申请日 2009.04.07

(71) 申请人 谢亚南

地址 410006 湖南省长沙市岳麓区湖南师大
附中广高 0701 班

(72) 发明人 谢亚南

(51) Int. Cl.

H04N 7/24 (2006.01)

H04N 7/26 (2006.01)

H04L 29/06 (2006.01)

G06F 17/30 (2006.01)

权利要求书 1 页 说明书 6 页 附图 4 页

(54) 发明名称

一种 AJAX 实现动态图像增量传输与显示的方法

(57) 摘要

传统的动态图像增量传输与显示应用一般采用 C/S 架构实现。B/S 架构的 AJAX (Asynchronous Java Script and XML, 异步 Java Script 与 XML) 应用具有客户端文件小、无需安装运行环境与可跨平台使用等诸多优秀特性,但由于 Java Script 不支持图像编码,开发人员对 AJAX 实现动态图像的增量技术望而却步,始终没有一个真正能够实现的方法。本发明的目的就是提供这样一个 AJAX 实现动态图像增量传输与显示的方法。

1. 一种浏览器用 AJAX 实现动态图像增量传输与显示的方法,其特征在于,包括以下步骤:

步骤 1:客户端向服务端发送请求;

步骤 2:服务端将动态图像当前瞬时图像的完整图像以及图像信息 XML 文档返回给客户端;

步骤 3:客户端将服务端返回的完整图像显示在网页上,并设置加载完毕触发执行步骤 4;

步骤 4:客户端向服务端发送请求;

步骤 5:服务端将增量数据图像以及包含图像信息和客户端显示方法信息的 XML 文档返回给客户端;

步骤 6:客户端按照服务端返回的 XML 文档,将增量数据图像显示在网页上,并设置加载完毕触发执行步骤 4。

2. 根据权利要求 1 所述的方法,其特征在于,所述客户端为浏览器,客户端不依靠第三方厂商提供的运行环境或是浏览器插件。

3. 根据权利要求 2 所述的方法,其特征在于,所述客户端使用 JavaScript、XML(Extensible Markup Language,可扩展标记语言)、CSS(Cascading Style Sheets,层叠样式表)和 XHTML(The Extensible Hyper Text Mark-up Language,可扩展超文本标记语言)或 HTML(Hyper Text Mark-up Language,超文本标记语言)技术实现。

4. 根据权利要求 1 所述的方法,其特征在于,所述增量数据图像由动态图像当前瞬时图像中的若干矩形图像组成的一个支持在浏览器上显示的图像。

5. 根据权利要求 4 所述的方法,其特征在于,所述若干矩形图像区域为,前后两次服务端接收请求时,动态图像瞬时图像差异部分所在的矩形区域。

6. 根据权利要求 1 所述的方法,其特征在于,所述步骤 6 中将增量数据图像显示在网页上的操作,是利用 HTML 元素定位、叠加实现的。

7. 根据权利要求 6 所述的方法,其特征在于,所述定位、叠加的实现,利用了 CSS 的 background、position、left、top、z-index 属性。

8. 根据权利要求 1 所述的方法,其特征在于,所述触发执行方法为通过修改 img 元素的 src 属性引发加载,利用 onload 事件触发加载完毕执行。

一种 AJAX 实现动态图像增量传输与显示的方法

技术领域：

[0001] 本发明是一种无需浏览器插件支持的,在 B/S 架构的客户端利用 AJAX 实现动态图像增量传输与显示的方法。涉及领域包括 B/S 前端开发、远程监控与服务器技术。

背景技术：

[0002] 随着互联网技术的发展,特别是近几年来,RIA(Rich Internet Application,富互联网应用程序)的开发模式迅速兴起,许多本地化的应用程序均开始向 RIA 开发模式发展。RIA 开发模式给用户带来了全新的互动用户界面,在无刷新页面的前提下提供快捷的响应。RIA 客户端开发技术包括 Adobe 公司的 Flex 技术、Microsoft 公司的 Silverlight 技术等。但是,用户在使用传统 RIA 技术开发的软件之前,需要在客户端安装一个应用环境软件(如 Flash Player、Silverlight 等),每次使用时还需从软件提供者服务器上下载应用程序,这些繁琐复杂的操作使得传统 RIA 客户端在移动性与跨平台能力上甚至还不如传统的本地化应用程序好。AJAX(Asynchronous JavaScript and XML,异步 JavaScript 与 XML)是一种结合了 DHTML、XML 与 JavaScript 技术的全新网页开发技术,它拥有 RIA 的大多数优秀特性,它的客户端无需预先安装环境支持软件,它的应用程序通常只有数千字节,它的软件更新后用户无需等待升级,这些都是传统 RIA 开发模式所不具备的特性。由于它是符合 W3C 标准的,得到了近乎所有的浏览器的支持。使用它构建的应用程序具备无须安装、无须刷新、快速响应等特点,应用十分广泛,最为知名的应用是 Google 公司的 Gmail 与 Google Map 产品,其中 Google Map 通过 AJAX 解析 XSLT 文档实现动态内容更新。

[0003] 远程监控技术中的图像传输与显示技术是最为核心的技术。在技术较为先进的 C/S 架构解决方案中,一般采用图像增量传输技术以降低网络要求,节省带宽开销。当前许多 RIA 产品也实现了这项技术,然而由于 AJAX 技术本身并不支持图像编码,其对图形界面更新只能通过修改、删除和插入 HTML 元素实现,开发人员对基于 AJAX 的图形增量传输与显示技术望而止步。现有的 AJAX 动态图像传输与显示技术仅只是将动态图像中的每一帧完整图像用 JPEG 格式反馈给浏览器,并简单地利用 img 元素进行显示,没有一种 AJAX 实现图像增量传输与显示的方法。

发明内容：

[0004] 本发明的目的在于提供一种在 B/S 架构下,使用 AJAX 技术实现动态图像增量传输与显示的方法,解决传统 AJAX 技术传输动态画面时网络资源消耗大与资源利用不科学的问题。

[0005] 传统的动态图像传输方法每帧的数据都包含完整的图像,网络通信量非常之大。解决这个问题的常用方法即使用图像增量传输技术。图像增量传输技术包含两类,一类是时间增量方式,另一类是坐标增量方式。这两类图像增量传输技术在 C/S 架构的软件中应用较多,但在 B/S 的纯浏览器客户端中,因为浏览器脚本没有图像处理能力,使得开发人员对在该环境中的图像增量传输与显示技术望而却步。

[0006] 该发明包括在浏览器端使用 AJAX 技术实现图像增量传输与显示的方法,包括浏览器端的 AJAX 实现以及服务端的 Web 服务、图形处理及数据编码方法。

[0007] 该发明包括在浏览器端使用 img 元素 onload 事件实现触发式通信的方法,即通过更新 img 元素的 src 属性设置 onload 事件,实现“触发事件-更新属性-触发事件-更新属性”循环触发方式的动态图像即时更新。

[0008] 该发明的浏览器端 Ajax 技术包括 XML 文档图像拆分布局信息以及对图像的拆分布局方法。

[0009] 该发明中的服务端动态图像预处理方法包括对原始图像与当前图像的对比、差异区域划分,最终拼接成一个新的图像的方法。

[0010] 该发明包括的主要步骤如下:

[0011] 步骤 a:客户端(即浏览器,下同)向服务器发送请求;

[0012] 步骤 b:服务端从动态图像中截取当前帧完整画面并保存为未经过压缩的 BMP 图像文件以及经过压缩可访问图像文件,再将包括可访问图像信息的 XML 文档返回给客户端;

[0013] 步骤 c:客户端接收到数据后解析 XML 文档,在页面中添加一个 img 元素,设置其 src 属性为步骤 b 保存的可访问图像的 URL 地址,设置其 onload 事件触发进行步骤 d,在页面中添加一个背景为步骤 b 保存的可访问图像、显示高宽与步骤 b 保存的可访问图像相同、绝对定位、样式 z-index 属性为 0 的 div 元素;

[0014] 步骤 d:由步骤 c 添加的 img 元素 onload 事件触发本步骤后,客户端再次向服务端发送请求;

[0015] 步骤 e:服务端从动态图像中截取当前帧完整画面,并保存为未经过压缩的 BMP 图像;

[0016] 步骤 f:服务端将步骤 e 保存的 BMP 图像与步骤 b 保存的 BMP 图像进行对比,将步骤 e 保存的 BMP 图像中与步骤 b 保存的 BMP 图像有差异的部分以及有差异部分的周边部分划分成若干矩形区域;

[0017] 步骤 g:服务端将步骤 f 中从步骤 e 保存的 BMP 图像中提取出来的若干矩形区域拼接成新的图像,并保存为经过压缩后的可访问的图像文件;

[0018] 步骤 h:服务端将步骤 b 中保存的 BMP 图像文件替换成步骤 e 中保存的 BMP 图像文件,将步骤 b 中保存的经过压缩可访问图像文件删除,并将包含步骤 g 保存的可访问图像信息、步骤 f 的划分方法信息以及步骤 g 拼接方法信息的 XML 文档返回给客户端;

[0019] 步骤 i:客户端收到数据后解析 XML 文档,在页面中添加与步骤 f 划分矩形区域数量相同数量的 div 元素,利用 div 元素样式 position、left、top 属性定位到页面相应位置,其样式 z-index 属性值均按本步骤的执行次数依次递增,实现图层的叠加显示。修改步骤 c 添加的 img 元素的 src 属性为步骤 g 保存的可访问图像的 URL 地址, img 元素的 onload 事件会自动触发执行步骤 d。

[0020] 服务端动态图像可持续更新,实现客户端图像与动态图像源在时间上一致。

[0021] 服务端保存经过压缩的可访问图像时,可选用 JPEG 格式进行压缩,亦可选用 PNG 格式或其他浏览器可支持的图像格式进行保存。

[0022] 步骤 b 由服务端返回给客户端的 XML 文档包含的可访问图像信息包括图像的 URL

地址、图像的高和宽。

[0023] 步骤 i 由服务端返回给客户端的 XML 文档包含的可访问图像信息包括图像的 URL 地址、图像的高和宽,包含的步骤 f 划分方法信息包括每次划分矩形区域在原 BMP 图像中的位置,包括每个矩形区域的左上角坐标和矩形区域长、宽,包含的步骤 g 拼接方法信息包括每个矩形区域在步骤 g 保存的图像中的位置,包括每个矩形区域的左上角坐标。

[0024] 在步骤 d 中,若当前页面上 div 元素过多,可将所有添加的 div 元素以及 img 元素清除,重新从步骤 a 开始执行,降低客户端的内存消耗。

[0025] 利用该方法实现的远程监控软件或在线视频播放器,客户端无需另外安装任何软件或浏览器插件,用户直接打开网页进行使用,将客户端支持设备的范围扩充到一切拥有标准互联网浏览器的设备(除个人电脑外还有部分移动智能设备),同时借助浏览器的安全设置,保证了客户端免受其他因素的安全威胁。通过只对图像有变化的矩形区域进行更新,实现了快速高效的动态图像增量传输与显示。

[0026] 替换上述步骤中涉及到的属性、元素等信息或在上述步骤中插入其它步骤达到相同或类似目的的方法均属于本发明专利所要求保护的专利范围。

附图说明：

[0027] 图 1 是本发明所描述方法构建的总体架构的示意图。

[0028] 图 2 是本发明所描述方法涉及结构的层次图。

[0029] 图 3 是本发明所描述方法客户端工作的流程图。

[0030] 图 4 是本发明所描述方法服务端工作的流程图。

[0031] 图 5 是本发明所描述方法中部分刷新操作图形处理的一个实例示意图。

具体实施方式：

[0032] 以下,参照附图来说明本发明所描述的方法。在下面的说明中,未详细说明公知的技术细节或其他术语,因为它们会在不必要的细节上混淆本发明。

[0033] 图 1 是本发明所描述方法构建的总体架构的示意图。参见图 1,客户端与服务端通过标准的 HTTP 协议进行通信,数据以 Post 或 Get 方法提交,其中客户端可为个人电脑、移动智能设备或其他拥有支持或基本支持 W3C 标准浏览器的设备,服务端为具备图形处理、信息编码、网络通信、Web 服务功能的电子计算机或者微电脑等设备。服务端提供 Web 服务,包括数据传输、请求处理等。

[0034] 客户端首次访问服务端时,浏览器从服务端下载页面 XHTML、CSS 以及 JavaScript 代码并呈现给用户,同时执行 JavaScript 代码中的相关程序,通过 XMLHttpRequest(微软环境)对象或 XMLHttpRequest(其他环境)对象(下文将这 2 类对象称作 XMLHttpRequest 对象)与服务端进行余下的通信工作,其通信方法以及原理是公知的,这里不再作描述。

[0035] 图 2 是上述 XHTML 与 CSS 布局中涉及到该发明部分结构的层次图。参见图 2,布局中涉及 3 个涉及到该发明的层(这里层的概念并不等同于 XHTML 中的 div 标签),分别为“控制感应层”、“远程图像层”以及“辅助功能层”。其中控制感应层在 XHTML 文档中是一个 div 标签,使用层叠样式表(Cascading Style Sheets)的 position 属性、left 属性、top 属性、z-index 属性固定在页面中特定位置,其 z-index 属性值取最大值(一般 256)。远程

图像层定位在控制感应层的下方,辅助功能层则独立于控制感应层与远程图像层,可用层叠样式表的 display 属性隐藏。

[0036] 远程图像层中可包含多个 div 标签,这里将这些 div 标签称作图层,通过设置图层 position、left、top 样式属性实现图层在远程图像层中的定位,通过设置图层 z-index 样式属性实现图层的叠加。该发明利用图层的定位与叠加实现图像的区域更新,即增量显示。控制感应层用于提供设备对图像操作位置的感应功能,若将本发明应用到电子计算机远程屏幕监控中,控制感应层可用于获取鼠标相对于远程图像层的相对位置以及感应单击、双击等操作,以实现鼠标的模拟远程控制。辅助功能层是用于存放该方法实施过程中需要引入的而无需与用户进行交互 XHTML 对象的容器,通过设置 display 样式属性进行隐藏,是实施该发明必不可少的一部分。

[0037] 图 3 是本发明所描述方法客户端工作的流程图。参见图 3,客户端具体工作步骤如下:

[0038] B1 步骤:用户打开网页,触发 JavaScript 脚本。这里的触发方式有多种,可为网页加载完毕后自动触发,也可由用户单击按钮触发;

[0039] B2 步骤:浏览器通过 XMLHttpRequest 对象向服务端发送请求(Request)。XMLHttpRequest 对象是浏览器内置的对象,可通过 JavaScript 直接利用,是 AJAX 的核心技术。本次请求提交的信息包括一个表示本次请求为完全刷新操作的参数;

[0040] B3 步骤:客户端接收到服务端返回 XML 文档。在 XML 文档接收完成后,使用 DOM(Document Object Model,文档对象模型)解析其中后续步骤所需要的信息;

[0041] B4 步骤:判断本次操作是否是完全刷新操作。判断本次操作是否为完全刷新操作依据是 B3 步骤中从 XML 文档中解析出来的信息。若本次操作是完全刷新操作则进入 B5 步骤,否则执行 B8 步骤;

[0042] B5 步骤:清空远程图像层、辅助功能层内的元素。该操作仅在远程图像层(可能包含 div 图层)、辅助功能层内(可能包含 img 元素)有元素的情况下执行;

[0043] B6 步骤:在远程图像层内添加图层,在辅助功能层内添加 img 元素并设置触发。图层可用 div 元素,通过设置其样式 background、position、left、top、z-index 属性实现图层的显示定位与叠加,这里的 z-index 属性值为 0。添加的 img 元素的 src 属性为 B3 步骤中解析得到的图像 URL 地址,设置其 onload 事件触发执行 B7 步骤;

[0044] B7 步骤:发送部分刷新请求。该步骤由步骤 B6 或步骤 B9 更新 img 元素触发执行。该步骤执行完毕后执行步骤 B3;

[0045] B8 步骤:去除新图层完全覆盖的图层。判断远程图像区现有的图层中是否有能被 B9 步骤中将要添加图层完全覆盖的图层,若有则将其去除。图层信息在 B3 步骤中从 XML 文档中解析取得;

[0046] B9 步骤:在远程图像层添加图层,更新辅助功能层 img 元素。图层添加方式信息,在 B3 步骤中从 XML 文档中取得,添加的图层利用样式 position、left、top、z-index 定位在远程图像层的位置,用样式 background 定位显示的区域在 B3 步骤解析得到图像中的位置。设置 img 元素 src 属性为在 B3 步骤解析得到的图像 URL 地址。该步骤中修改 img 元素 src 属性会使 img 元素重新加载,img 元素加载完毕后会再次触发 onload 事件执行 B7 步骤。

[0047] img 元素的 onload 事件会在该元素 src 属性所指向的图像加载完毕后自动触发。

其中 B7 步骤始终由 img 元素 onload 事件触发,该发明利用“发送请求 - 更新 img 对象 - 触发发送请求 - 更新 img 对象 - 触发发送请求”这样的循环操作实现图像实时传输。

[0048] 这里所说的“完全刷新操作”指的是客户端首次向服务端发送请求获得完整图像、或客户端与服务端图像不同步时进行的操作,其发送的请求被称作“完全刷新请求”。“部分刷新操作”与“完全刷新操作”相对,指的是客户端在现有图像的基础上向服务端发送请求获的增量图像及相关信息时的操作,其发送的请求被称作“部分刷新请求”。

[0049] 图 4 是本发明所描述方法服务端工作的流程图。参见图 4,服务端具体工作步骤如下:

[0050] S1 步骤:服务器收到来自客户端的请求;

[0051] S2 步骤:通过 S1 步骤收到的请求所带的参数判断此次请求是完全刷新请求或是部分刷新请求,若是完全刷新请求则执行 S3 步骤,否则执行 S6 步骤;

[0052] S3 步骤:从动态图像中截取瞬时图像并保存为 BMP 图像文件。这里的 BMP 图像文件的文件路径与文件名是固定不变的,后续步骤中将会对该文件进行替换;

[0053] S4 步骤:将 BMP 图像文件压缩并保存在可访问路径下。一般情况下可用 JPEG 格式保存,也可用 PNG 或其他浏览器支持的格式保存,该文件保存在可访问的路径下,以便客户端能够正常访问;

[0054] S5 步骤:将包含压缩后图像文件信息的 XML 文档返回给客户端。该 XML 文档包括图像文件 URL 地址、图像高、宽以及表示该次操作为完全刷新操作的文本标志;

[0055] S6 步骤:从动态图像中截取当前瞬时图像并保存为 BMP 图像文件。该步骤保存的文件与 S3 步骤保存的文件路径与文件名不同,后续步骤将对两图像进行对比;

[0056] S7 步骤:将 S6 步骤所保存 BMP 图像与上次请求保存 BMP 图像的差异部分划分成矩形并重新拼接。将两图像的差异部分及差异部分周边区域划分成若干矩形区域,并将这些矩形区域拼接成一个新的图像;

[0057] S8 步骤:将拼接后的图像压缩保存在可访问路径下。拼接后的图像指的是 S7 步骤中拼接出的新图像;

[0058] S9 步骤:将包含压缩后图像文件信息与划分拼接信息的 XML 文档返回给客户端。

[0059] 图 5 是本发明所描述方法中部分刷新操作图形处理的一个实例示意图,如图 5 所示,部分刷新操作对图像的处理(即图 4 所示 S7 步骤)分 3 步完成:

[0060] 第一步:逐像素点对比当前图像与上帧图像(指上次操作所保存的图像)并用 FloodFill 算法找出差异区域;

[0061] 第二步:将在当前图像上的这些差异区域划分成若干矩形,这里将这些矩形区域称作差异矩形;

[0062] 第三步:将差异矩形图像拼接成一个新的矩形图像,并保存为一个静态图像文件,替换原来的原始图像文件。

[0063] 在第一步中,只要像素点颜色不完全相同就认为该像素点属于差异区域,在使用 FloodFill 寻找差异区域时,可定义曼哈顿距离小于高宽之和的 1% 的两个像素点是相互连接的,以平衡差异区域数量,提高图像处理速度与客户端解析拆分速度;在第二步中,可简单地将每块差异区域所在的最小矩形直接划成差异矩形,但这种方法在同时存在长差异矩形与宽差异矩形特殊情况下会使得第三步生成的新矩形过大,导致增量数据过大,因此

在拆分时需要把每块的高度与宽度限定在一定范围内,范围的确定可由当前图像的高宽确定;一般情况下,第三步可用深度优先搜索算法寻找最优的拼接方法。

[0064] 以上所涉及到的公知算法不包括在本发明范围内,在这些步骤使用类似算法达到相同目的方法也属于本发明所描述的方法。

[0065] 当客户端发送的请求为完全刷新请求时,服务端返回的 XML 文档应当包含图像文件 URL 地址、图像高宽以及完全刷新文本标志。当客户端发送的请求为部分刷新请求时,服务端返回的 XML 文档应包含图像文件 URL 地址、部分刷新文本标志以及各差异矩形在原 BMP 图像中的位置信息和在拼接后图像中的位置信息。位置信息包括差异矩形所在的像素位置(即矩形左侧像素点个数和矩形上方像素点个数)以及差异矩形的大小(即差异矩形横向像素点个数及纵向像素点个数)。

[0066] 一般情况下,完全刷新操作只执行一次,此后触发的操作均为部分刷新操作,在如下的 2 种特殊情况下,部分刷新操作的处理方式可进行变动:

[0067] 1、在原始图像与当前图像完全一致时,返回的 XML 文档不再包含新的图像访问地址,图像更新文本标志更换为表示无须更新的文本标志,客户端处理时直接触发新的部分刷新操作;

[0068] 2、在客户端远程图像层内图层数达到一定量时,客户端不触发执行部分刷新操作,而触发执行完全刷新操作。

[0069] 图 4 中 S7 步骤的图像处理具体实现方式可能会对本发明所提供方法的实际应用效果产生影响,本发明专利不包括该步骤中图像处理的具体算法。

[0070] 本发明所提供的方法可用于制作 B/S 架构的远程计算机桌面监控软件和远程视频监控软件,在制作 B/S 架构远程计算机桌面监控软件上带来的速度与性能的提升更明显。在制作远程计算机桌面监控软件之需要在以上方法的基础上,在控制感应层设置 onmousemove、onclick、onmousedown 和 onmouseup 等事件,记录客户端鼠标的动作,将动作信息作为图 3 中 B7 步骤发送请求的参数之一,服务端在图 4 中 S1 操作后根据参数模拟相关动作,即可实现带鼠标控制功能的 B/S 远程计算机桌面监控。

[0071] 使用本发明提供的方法制作的 B/S 架构远程计算机桌面监控软件,客户端用户无需下载安装应用程序,打开网页即可使用,同时可跨平台使用,支持移动智能设备。同时在性能方面,较未使用本发明所提供的方法制作的 B/S 无需插件支持的同类软件,本发明所提供的方案最高能节省 90% 以上的网络通信量,在特定网络环境下可提供数十倍的图像更新速度。

[0072] 开发人员使用本发明所提供的方法进行软件开发,无需具备图形编码知识。服务端可使用微软 IIS 软件、ASP 技术以及 COM+ 技术实现。

[0073] 通过对本发明所提供的方法进行改造和扩充,也可以实现一服务器对多客户、动态图像源与服务器分离、三端混合架构的多用户对多动态图像源等多种应用。

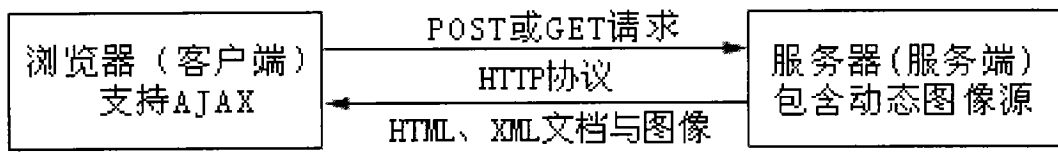


图 1

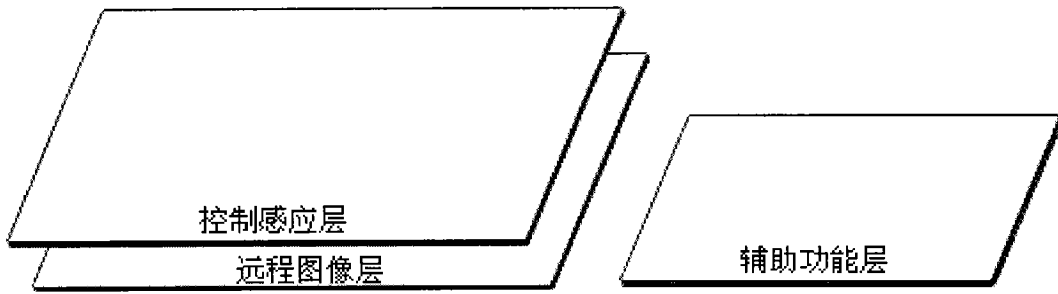


图 2

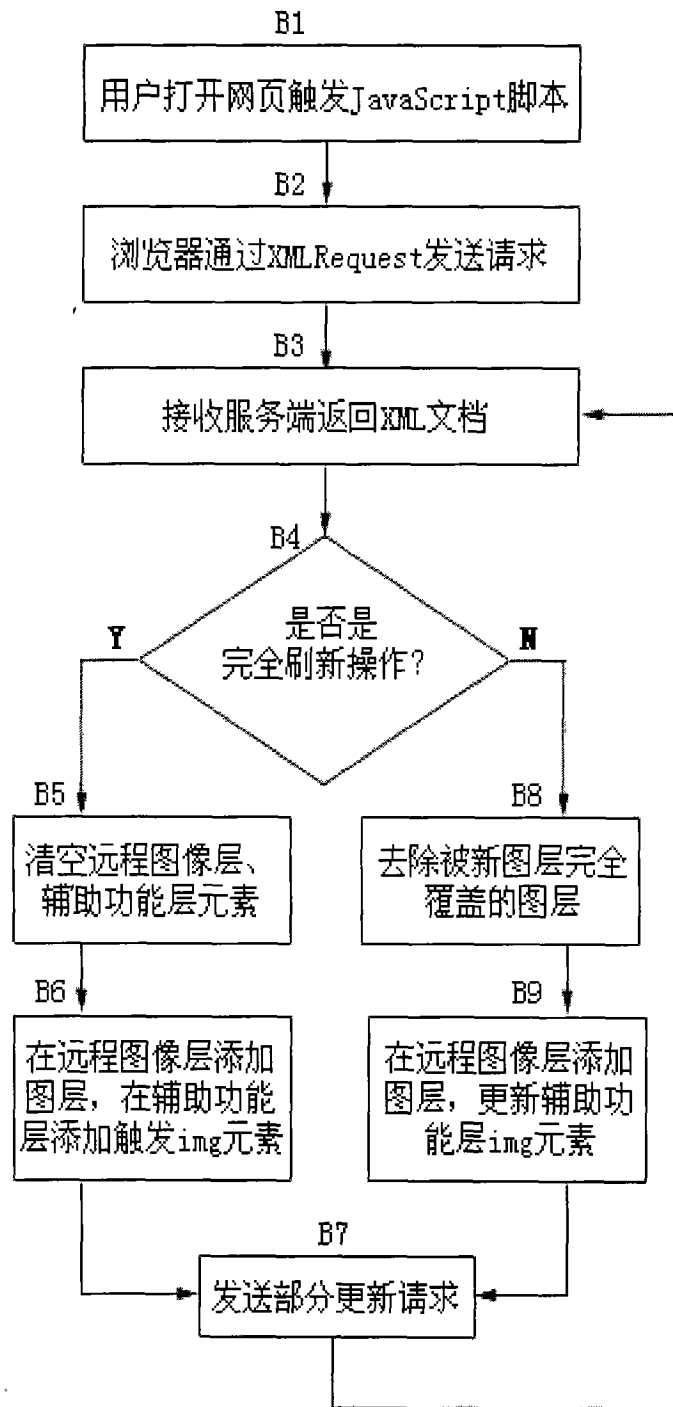


图 3

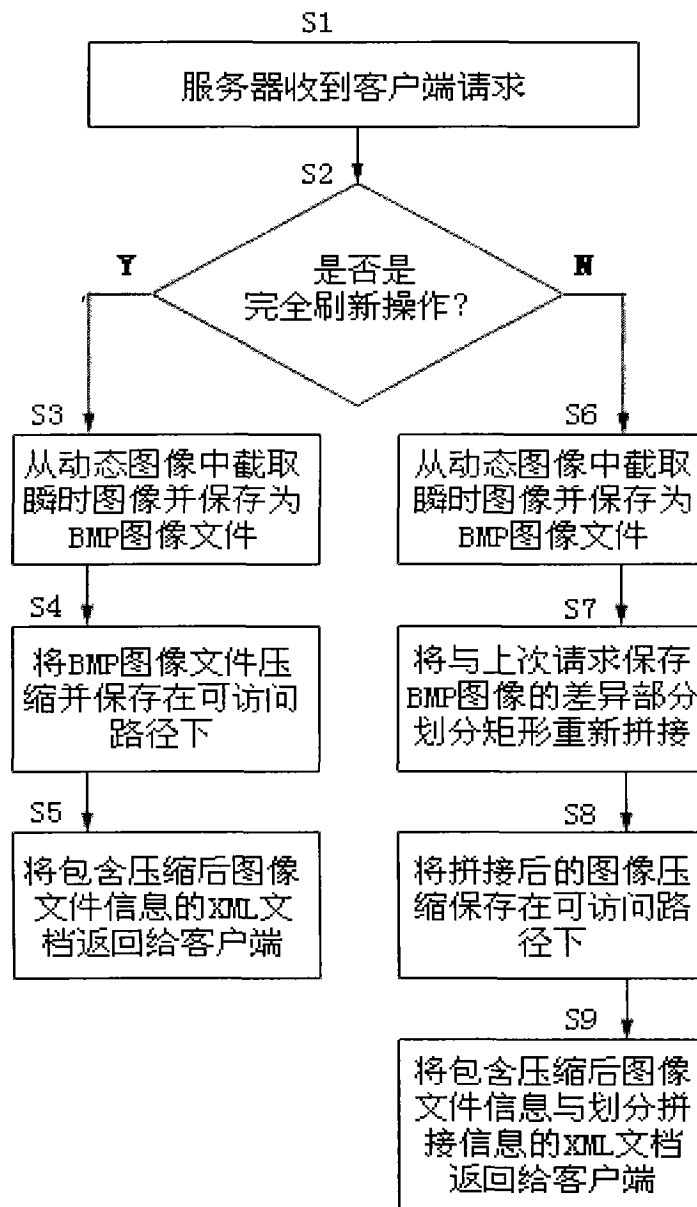


图 4

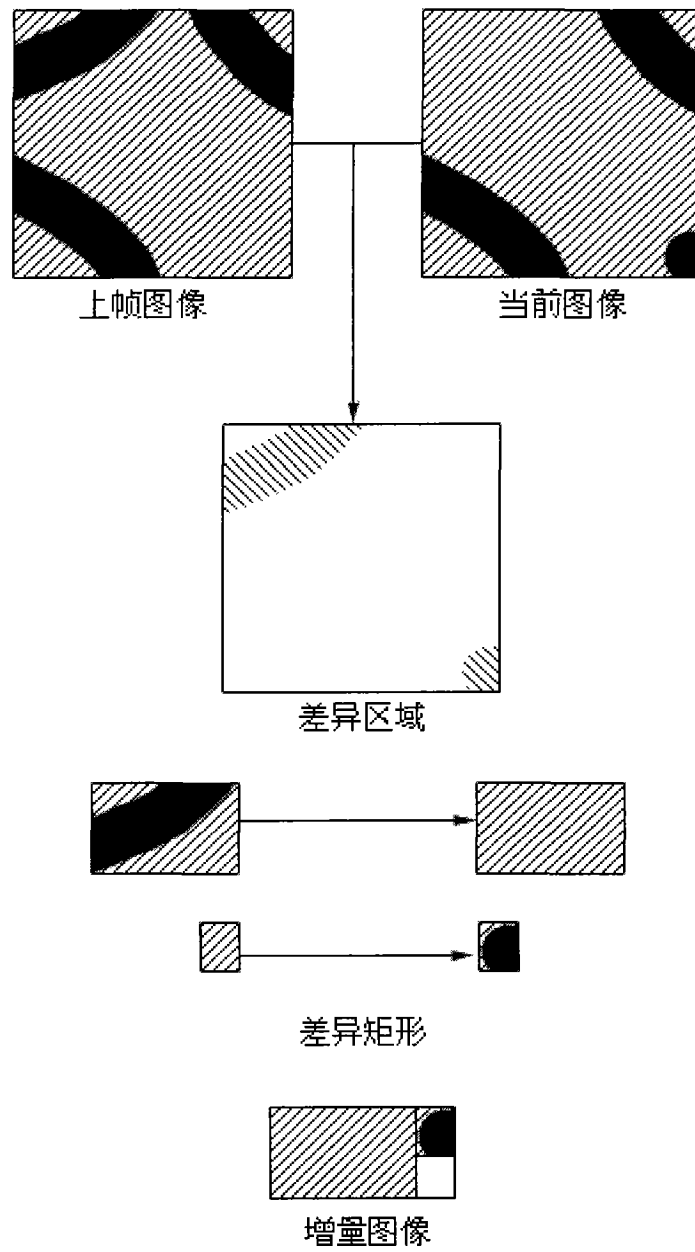


图 5