



(12) 发明专利申请

(10) 申请公布号 CN 104469400 A

(43) 申请公布日 2015. 03. 25

(21) 申请号 201410779048. 7

(22) 申请日 2014. 12. 17

(71) 申请人 浪潮软件集团有限公司

地址 250100 山东省济南市高新区孙村镇科
航路 2877 号

(72) 发明人 张家重 董毅 李光瑞 王玉奎
张涛 孙成通

(74) 专利代理机构 济南信达专利事务有限公
司 37100

代理人 姜明

(51) Int. Cl.

H04N 21/2343(2011. 01)

H04N 21/4402(2011. 01)

H04N 19/42(2014. 01)

权利要求书1页 说明书4页

(54) 发明名称

一种基于 RFB 协议的图像数据压缩方法

(57) 摘要

本发明公开了一种基于 RFB 协议的图像数据压缩方法,属于图像数据压缩方法,本发明要解决的技术问题为现有技术中庞大的图像数据要通过网络进行传输时,给用户所带来的延迟感。技术方案为:在服务器端或者在客户端将图像数据帧进行空间化编码,空间化编码方式包括静态编码和动态编码;静态编码是以时间为单位的定量帧编码,静态编码针对的是图像持续快速变化,前后帧内容缺乏一致性的使用环境;动态编码是以帧图像所包含内容为依据的动态帧编码,动态编码所针对的是内容变化频率较低、变化范围相对有限的使用环境。

1. 一种基于 RFB 协议的图像数据压缩方法,其特征在于在服务器端或者在客户端将图像数据帧进行空间化编码,空间化编码方式包括静态编码和动态编码;静态编码是以时间为单位的定量帧编码,静态编码针对的是图像持续快速变化,前后帧内容缺乏一致性的使用环境;动态编码是以帧图像所包含内容为依据的动态帧编码,动态编码所针对的是内容变化频率较低、变化范围相对有限的使用环境。

2. 根据权利要求 1 所述的一种基于 RFB 协议的图像数据压缩方法,其特征在于在服务器端将图像数据帧进行空间化编码,步骤如下:

步骤 1:调用函数 SystemInfo() 获取系统配置信息;

步骤 2:调用函数 SetFramesStep() 设置空间编码默认所包含的帧数量;

步骤 3:调用函数 GetKeyframe() 获取初始关键帧;

步骤 4:调用函数 CatchChangingZone() 获取每一帧的变化区域;

步骤 5:当抓取的帧计数器达到 SetFramesStep() 所设置的上限时,调用函数 CreateSpaceEncoding(),将所有变化区域编码后发送到客户端;

步骤 6:读取变量 staticEncoding 的值,判断当前模式;

步骤 7:若步骤 6 得到的值为真,则调用 GetKeyframe() 重置关键帧,否则跳过该步骤;

步骤 8:重复步骤 3 到步骤 7 的过程,直到步骤 6 所得到的值为假;

步骤 9:读取变量 changeRate,并与函数 GetChangeRate() 所返回的值相比较;

步骤 10:若步骤 9 比较的结果是 changeRate 的值更大,则重复步骤 4 到步骤 7 的过程,否则进入下一步;

步骤 11:调用函数 GetCacheKeyframe() 尝试获取缓存的关键帧;

步骤 12:若步骤 11 返回的值为空,则调用函数 SetCacheKeyframe() 缓存当前使用的关键帧,然后重复步骤 3 到步骤 7 的过程,否则进入下一步;

步骤 13:调用函数 CompareKeyframe() 依次比较当前关键帧与之前缓存关键帧的差异,若存在相同关键帧,则调用函数 SetKeyframe() 设置该关键帧为当前关键帧,同时向客户端发送该关键帧编号,使其重置当前桌面,否则进入下一步;

步骤 14:重复步骤 3 到步骤 7 的过程。

3. 根据权利要求 1 所述的一种基于 RFB 协议的图像数据压缩方法,其特征在于在客户端将图像数据帧进行空间化编码,步骤如下:

步骤 1:调用函数 GetServerInfo() 获取服务端的基础参数;

步骤 2:调用函数 ReadSpaceMessage() 读取服务端的消息包类型,若消息类型为 SET_KEYFRAME,则使用读取到的关键帧数据重置桌面,否则进入下一步;

步骤 3:调用函数 SpaceMessageDeconding() 解析空间编码数据;

步骤 4:调用函数 FillFrame() 将步骤 3 获得的数据依次填充到相应位置;

步骤 5:重复步骤 4 的过程,直到将所有帧数据填充完毕;

步骤 6:重复步骤 2 到步骤 5 的过程。

一种基于 RFB 协议的图像数据压缩方法

技术领域

[0001] 本发明涉及一种图像数据压缩方法,具体地说是一种基于 RFB 协议的图像数据压缩方法。

背景技术

[0002] 目前,市场上所流行的远程帧缓存(RFB 协议)领域所采用的编码方法往往基于前后帧变化的区域而生成相应的帧缓存数据流,数据传输量往往取决于图形变化区域的大小与帧速率。而当画面的变化范围超过某种零界点,又或所变化的区域较为零散时,这种方法将无法有效减少数据的实际传输量。

[0003] 帧速率相应增大的时候,帧图像之间的变化过程将会不可避免的变得更加细腻,所产生的传输数据也就会相应增加,当这些庞大的数据要通过网络进行传输时,给用户所带来的延迟感也就变得越发严重。

发明内容

[0004] 本发明的技术任务是针对以上不足之处,提供一种在减少数据传输量的同时提升画面显示的帧速率,从而有效降低图像延迟,提升用户的操作体验的一种基于 RFB 协议的图像数据压缩方法。

[0005] 本发明解决其技术问题所采用的技术方案是:

一种基于 RFB 协议的图像数据压缩方法,在服务器端或者在客户端将图像数据帧进行空间化编码,空间化编码方式包括静态编码和动态编码;静态编码是以时间为单位的定量帧编码,静态编码针对的是图像持续快速变化,前后帧内容缺乏一致性的使用环境;动态编码是以帧图像所包含内容为依据的动态帧编码,动态编码所针对的是内容变化频率较低、变化范围相对有限的使用环境。

[0006] 在服务器端将图像数据帧进行空间化编码,步骤如下:

- 步骤 1:调用函数 SystemInfo() 获取系统配置信息,如分辨率,色深等;
- 步骤 2:调用函数 SetFramesStep() 设置空间编码默认所包含的帧数量;
- 步骤 3:调用函数 GetKeyframe() 获取初始关键帧;
- 步骤 4:调用函数 CatchChangingZone() 获取每一帧的变化区域;
- 步骤 5:当抓取的帧计数器达到 SetFramesStep() 所设置的上限时,调用函数 CreateSpaceEncoding(),将所有变化区域编码后发送到客户端;
- 步骤 6:读取变量 staticEncoding 的值,判断当前模式;
- 步骤 7:若步骤 6 得到的值为真,则调用 GetKeyframe() 重置关键帧,否则跳过该步骤;
- 步骤 8:重复步骤 3 到步骤 7 的过程,直到步骤 6 所得到的值为假;
- 步骤 9:读取变量 changeRate,并与函数 GetChangeRate() 所返回的值相比较;
- 步骤 10:若步骤 9 比较的结果是 changeRate 的值更大,则重复步骤 4 到步骤 7 的过程,否则进入下一步;

步骤 11 :调用函数 GetCacheKeyframe () 尝试获取缓存的关键帧 ;

步骤 12 :若步骤 11 返回的值为空,则调用函数 SetCacheKeyframe () 缓存当前使用的关键帧,然后重复步骤 3 到步骤 7 的过程,否则进入下一步 ;

步骤 13 :调用函数 CompareKeyframe () 依次比较当前关键帧与之前缓存关键帧的差异,若存在相同关键帧,则调用函数 SetKeyframe () 设置该关键帧为当前关键帧,同时向客户端发送该关键帧编号,使其重置当前桌面,否则进入下一步 ;

步骤 14 :重复步骤 3 到步骤 7 的过程。

[0007] 在客户端将图像数据帧进行空间化编码,步骤如下 :

步骤 1 :调用函数 GetServerInfo () 获取服务端的基础参数,如分辨率、色深、默认编码帧数等 ;

步骤 2 :调用函数 ReadSpaceMessage () 读取服务端的消息包类型,若消息类型为 SET_KEYFRAME,则使用读取到的关键帧数据重置桌面,否则进入下一步 ;

步骤 3 :调用函数 SpaceMessageDeconding () 解析空间编码数据 ;

步骤 4 :调用函数 FillFrame () 将步骤 3 获得的数据依次填充到相应位置 ;

步骤 5 :重复步骤 4 的过程,直到将所有帧数据填充完毕 ;

步骤 6 :重复步骤 2 到步骤 5 的过程。

[0008] 本发明的一种基于 RFB 协议的图像数据压缩方法和现有技术相比,具有以下优点 :

1、将图像数据帧进行空间化编码的解决方案,通过该方案,计算机可以最大限度的压缩多个帧图像之间的共同区域,并可以将那些产生变化的区域进行分拆后独立编码,而且这种编码的细粒度可以精确到桌面像素级别,从而最大程度上减少冗余数据的数量 ;

2、本将 RFB 协议中的帧缓存数据以时间坐标轴为序进行排列,将每一个像素的变化抽象为一条相对独立的曲线,再以此精确更新变化区域,从而最大程度的减少单位时间内的数据传输量 ;

3、打破了该领域中以单帧图像为基准的数据编码模式,而是将多帧图像数据缓存为单位体积的空间数据后再进行统一处理,因此可以有效提升用户的操作体验 ;

4、将单位时间内多帧图像进行统一处理的编 / 解码的相关方法 ;

5、通过使用本方法,可以有效提升公司相关产品在该领域中的核心竞争力,获取更高的市场价值 ;而且如能将该算法形成相应的协议标准,更可以在竞争中保持产品在技术上的领先地位。

具体实施方式

[0009] 下面结合具体实施例对本发明作进一步说明。

[0010] 实施例 1 :

本发明的一种基于 RFB 协议的图像数据压缩方法,在服务器端或者在客户端将图像数据帧进行空间化编码,空间化编码方式包括静态编码和动态编码 ;静态编码是以时间为单位的定量帧编码,静态编码针对的是图像持续快速变化,前后帧内容缺乏一致性的使用环境,如视频播放、远程监控等 ;动态编码是以帧图像所包含内容为依据的动态帧编码,动态编码所针对的是内容变化频率较低、变化范围相对有限的使用环境,如远程桌面的普通桌

面环境。

[0011] 静态编码 :以单位编码间隔 1 秒为例,假设桌面采样的帧速率为 30,则所传输的每一个数据包中所包含的都是 30 帧的图像数据,此时的数据延迟约为 1 秒。

[0012] 动态编码 :以动态采样的变化率设定 10% 为例,当帧数据的变化率不超过该值时,将初始化图像作为关键帧,而当变化率超出上限时则重新生成关键帧。

[0013] 在服务器端将图像数据帧进行空间化编码,步骤如下 :

步骤 1 :调用函数 SystemInfo() 获取系统配置信息,如分辨率,色深等 ;

步骤 2 :调用函数 SetFramesStep() 设置空间编码默认所包含的帧数量 ;

步骤 3 :调用函数 GetKeyframe() 获取初始关键帧 ;

步骤 4 :调用函数 CatchChangingZone() 获取每一帧的变化区域 ;

步骤 5 :当抓取的帧计数器达到 SetFramesStep() 所设置的上限时,调用函数 CreateSpaceEncoding(),将所有变化区域编码后发送到客户端 ;

步骤 6 :读取变量 staticEncoding 的值,判断当前模式 ;

步骤 7 :若步骤 6 得到的值为真,则调用 GetKeyframe() 重置关键帧,否则跳过该步骤 ;

步骤 8 :重复步骤 3 到步骤 7 的过程,直到步骤 6 所得到的值为假 ;

步骤 9 :读取变量 changeRate,并与函数 GetChangeRate() 所返回的值相比较 ;

步骤 10 :若步骤 9 比较的结果是 changeRate 的值更大,则重复步骤 4 到步骤 7 的过程,否则进入下一步 ;

步骤 11 :调用函数 GetCacheKeyframe() 尝试获取缓存的关键帧 ;

步骤 12 :若步骤 11 返回的值为空,则调用函数 SetCacheKeyframe() 缓存当前使用的关键帧,然后重复步骤 3 到步骤 7 的过程,否则进入下一步 ;

步骤 13 :调用函数 CompareKeyframe() 依次比较当前关键帧与之前缓存关键帧的差异,若存在相同关键帧,则调用函数 SetKeyframe() 设置该关键帧为当前关键帧,同时向客户端发送该关键帧编号,使其重置当前桌面,否则进入下一步 ;

步骤 14 :重复步骤 3 到步骤 7 的过程。

[0014] 实施例 2 :

本发明的一种基于 RFB 协议的图像数据压缩方法,在服务器端或者在客户端将图像数据帧进行空间化编码,空间化编码方式包括静态编码和动态编码;静态编码是以时间为单位的定量帧编码,静态编码针对的是图像持续快速变化,前后帧内容缺乏一致性的使用环境;动态编码是以帧图像所包含内容为依据的动态帧编码,动态编码所针对的是内容变化频率较低、变化范围相对有限的使用环境。

[0015] 在客户端将图像数据帧进行空间化编码,步骤如下 :

步骤 1 :调用函数 GetServerInfo() 获取服务端的基础参数,如分辨率、色深、默认编码帧数等 ;

步骤 2 :调用函数 ReadSpaceMessage() 读取服务端的消息包类型,若消息类型为 SET_KEYFRAME,则使用读取到的关键帧数据重置桌面,否则进入下一步 ;

步骤 3 :调用函数 SpaceMessageDeconding() 解析空间编码数据 ;

步骤 4 :调用函数 FillFrame() 将步骤 3 获得的数据依次填充到相应位置 ;

步骤 5 :重复步骤 4 的过程,直到将所有帧数据填充完毕 ;

步骤 6 :重复步骤 2 到步骤 5 的过程。

[0016] 上述具体实施方式仅是本发明的具体个案,本发明的专利保护范围包括但不限于上述具体实施方式,任何符合本发明的一种基于 RFB 协议的图像数据压缩方法的权利要求书的且任何所属技术领域的普通技术人员对其所做的适当变化或替换,皆应落入本发明的专利保护范围。