



(12) 发明专利申请

(10) 申请公布号 CN 101971209 A

(43) 申请公布日 2011. 02. 09

(21) 申请号 200880128024. 8

(51) Int. Cl.

(22) 申请日 2008. 01. 11

G06T 7/20(2006. 01)

(85) PCT申请进入国家阶段日
2010. 09. 08

(86) PCT申请的申请数据
PCT/IB2008/051270 2008. 01. 11

(87) PCT申请的公布数据
W02009/087493 EN 2009. 07. 16

(71) 申请人 卓然(法国)
地址 法国马拉科夫

(72) 发明人 琼·布鲁纳埃斯特拉奇
马克亚历山大·香德

(74) 专利代理机构 上海天协和诚知识产权代理
事务所 31216

代理人 张恒康

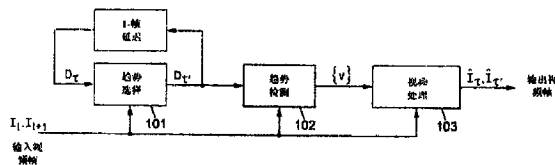
权利要求书 3 页 说明书 15 页 附图 9 页

(54) 发明名称

超分辨率视频处理的稀疏几何结构

(57) 摘要

在分析输入视频序列的方法中,输出视频序列合成图像的像素与属于趋势预定子集的各个规律趋势相关联。从预定集中确定输出序列的第一图像区域的候选趋势的第一子集。根据输入序列的图像和候选趋势的第一子集,从趋势的预定集中确定第一图像之后的输出序列第二合成图像对应区域的候选趋势的第二子集。从候选趋势的第二子集中检测第二合成图像这个区域的像素的规律趋势。候选趋势子集的递归确定为视频序列的有效分析提供了稀疏几何。该方法非常适用于超分辨率视频应用,例如去隔行扫描,帧频转换和降噪。



1. 一种分析输入视频序列的分析方法,其使输出视频序列合成图像的像素与属于趋势预定集合(Ω)的各个规律趋势相关联,所述方法包括:

- 从所述趋势预定子集中,确定所述输出序列第一图像(\hat{I}_T)区域的候选趋势的第一子集(D_T);

- 根据输入序列的图像(I_t, I_{t+1})和所述候选趋势的第一子集(D_T),从所述趋势预定集中,确定在第一图像后的第二合成图像(\hat{I}_T')输出序列对应区域的候选趋势的第二子集($D_{T'}$);以及,

- 从所述候选趋势的第二子集中,检测所述第二合成图像所述区域的像素规律趋势。

2. 根据权利要求1所述方法,其特征在于,所述候选趋势的第二子集($D_{T'}$)的确定包括:

- 检测至少一对趋势 v_r 和 v_a ,使得 v_r 属于所述候选趋势的第一子集(D_T), v_a 属于趋势预定义子集(Ω)且不属于第一子集,并且与对应第一和第二图像(\hat{I}_T, \hat{I}_T')所述区域的第一子集相关的代价函数高于与包括 v_a 和不包括 v_r 的第一子集的趋势的修改子集相关的代价函数;以及,

- 响应所述检测,从第二子集($D_{T'}$)中去除 v_r 并在第二子集中包括 v_a 。

3. 根据权利要求1所述的方法,其特征在于,所述候选趋势第二子集($D_{T'}$)的确定包括:

- 评估涉及第一子集(D_T)各个趋势对与第一子集相关的代价函数的各自贡献的第一边界($m(v)$);

- 评估涉及从预定集(Ω)在第一子集各自趋势增加所产生的代价函数分别减少的第二边界($m(v)$);以及,

- 当所述预定集合的趋势评估的第二边界超出所述第一子集趋势评估的第一边界时,将所述第一子集的趋势替换预定集的趋势。

4. 根据权利要求3所述的方法,其特征在于,用于第一子集(D_T)的趋势的第一边界($m(v)$)等于所述趋势对代价函数的贡献。

5. 根据权利要求3或4所述的方法,其特征在于,用于所述预定子集(Ω)的趋势的第二边界($m(v)$)等于由在所述趋势在第一子集(D_T)的增加所产生的代价函数减少。

6. 根据权利要求3或4所述的方法,其特征在于,用于预测集合(Ω)的趋势的第二边界($m(v)$)预测为在所述区域像素区域的局部边界之和,其中像素 x 和趋势 v 的局部边界为:

- 当 v 为在像素位置 x 上最小化代价函数观点中的整个预定集(Ω)中的最佳趋势时, $L_x(D)-L_x(v)$,其中 $L_x(v)$ 表示在像素位置 x 对趋势 v 的局部代价,以及 $L_x(D)$ 表示在像素位置 x 对第一子集(D_T)的趋势的局部代价最小值;以及

- 当其它时,为零。

7. 根据权利要求3至6任一所述的方法,其特征在于,包括:

- 在第一子集(D_T)中选择(501)具有最低第一边界的第一趋势(v);

- 在除了第一子集(D_T)的预定集(Ω)中选择(502)具有最高第二边界的第二趋势(w);以及,

- 如果所述选择的第二趋势的第二边界 ($m(w)$) 高于所述选择的第一趋势的第一边界 ($m(v)$), 则从第二子集 ($D_{\tau'}$) 去除所述选择的第一趋势和将所择的第二趋势包括于所述第二子集 ($D_{\tau'}$)。

8. 根据权利要求 3 至 6 任一所述的方法, 其特征在于, 包括:

- 选择并根据增加边界进行索引 (601) 在所述第一子集 (D_{τ}) 中具有最低第一边界的 n 个趋势 v_1, v_2, \dots, v_n ;

- 选择并根据减少边界进行索引 (602) 在除了第一子集 (D_{τ}) 的预定集 (Ω) 中具有最高第二边界的 n 个趋势 w_1, w_2, \dots, w_n ;

- 对除了第一子集以外的预定集索引的 n 个趋势 w_1, w_2, \dots, w_n 的各个趋势 w_i , 其中 $1 \leq i \leq n$, 如果除了第一子集的预定集索引 n 个趋势 v_1, v_2, \dots, v_n 对应趋势 v_i 的第二边界 ($m(w_i)$) 大于第一边界 ($m(v_i)$), 从第二子集 ($D_{\tau'}$) 中去除 v_i 并将 w_i 包括于第二子集 ($D_{\tau'}$)。

9. 根据权利要求 3 至 8 任一所述的方法, 其特征在于, 还包括从第二子集 ($D_{\tau'}$) 中去除具有低于预置阈值的评估边界 ($m(v)$) 的预定集 (Ω) 的趋势。

10. 根据权利要求 2 至 9 任一所述的方法, 其特征在于, 所述与趋势确定子集 (D) 相关的代价函数是在所述区域中的像素范围内对不同趋势的确定子集的局部代价最小数值之和。

11. 根据上述权利要求任一所述的方法, 其特征在于, 所述确定候选趋势的第二子集 ($D_{\tau'}$) 和从所述第二子集中检测趋势的步骤包括评估在所述输出序列的第二合成图像 (\hat{I}_{τ}) 之前和之后对具有各自时间位置的所述输入序列的至少两个连续图像 (I_t, I_{t+1}) 之间的局部代价函数, 以在确定所述第二子集的步骤中的所述局部代价函数评估比从所述第二子集中检测规律趋势的步骤中的评估更为粗略。

12. 根据上述权利要求任一所述的方法, 其特征在于, 所述至少一个预置的趋势是强制包括在第一和第二子集 ($D_{\tau}, D_{\tau'}$) 中。

13. 根据上述权利要求任一所述的方法, 其特征在于, 所述趋势的预定集 (Ω) 拆分成多个组 ($\Omega^{(r)}$), 并且在候选趋势的第二子集 ($D_{\tau'}$) 的确定中, 选择一个或有限数量的各组趋势, 用于包括在所述第二子集中。

14. 根据上述权利要求任一所述的方法, 其特征在于, 所述视频序列的图像拆分至多个窗 ($W_{p,q}$), 其中候选趋势 ($D_{\tau'}$) 的第二子集为第二合成图像整个范围对应区域进行确定, 所述方法进一步包括, 为各个窗:

- 根据为所述窗的像素确定的代价边界, 确定候选趋势的第三子集 ($D_{\tau', p,q}$) 为所述第二子集的子集; 以及,

- 从候选趋势的第三子集中, 为所述第二合成图像的所述窗的像素检测规律趋势。

15. 根据权利要求 14 所述的方法, 其特征在于, 所述至少一个窗 ($W_{p,q}$) 进一步拆分至多个子窗 ($W_{p,q,r,s}$), 为各个所述窗的子窗, 所述方法进一步包括:

- 根据为所述子窗像素确定的代价边界, 确定候选趋势的第四子集 ($D_{\tau', p,q,r,s}$) 是为所述窗确定的第三子集 ($D_{\tau', p,q}$) 的子集; 以及,

- 从候选趋势的第四子集中, 为第二合成图像的所述子窗的像素检测所述规律趋势。

16. 一种视频处理装置, 其特征在于, 包括根据权利要求 1 至 15 任一所述的方法分析视

频序列所设置的计算电路。

17. 一种计算机程序产品,其特征在于,包括当所述程序产品在计算机处理单元中运行时执行根据权利要求 1 至 15 任一的分析视频序列方法的指令。

18. 一种视频处理方法,其特征在于包括:

- 接收输入视频序列的连续图像 (I_t 、 I_{t+1});

- 应用根据权利要求 1 至 15 任一所述方法分析输入视频序列,使得输出视频序列合成图像的像素与各自的规律趋势相关联;以及,

- 使用检测到的规律趋势从输入视频序列中产生输出视频序列。

19. 根据权利要求 18 所述方法,其特征在于,所述产生输出视频序列的步骤包括使用所述检测到的规律趋势在输入视频序列的连续图像之间进行插值。

20. 根据权利要求 19 所述的方法,其特征在于,所述插值包括视频去隔行扫描。

21. 根据权利要求 19 所述的方法,其特征在于,所述插值包括转换输入视频序列的帧频。

22. 根据权利要求 18 所述的方法,其特征在于,所述产生输出视频序列的步骤包括使用检测到的规律趋势对输入视频序列应用降噪处理。

23. 一种视频处理装置,其特征在于,包括根据权利要求 18 至 22 任一所述的方法的对视频序列进行处理所设置的计算电路。

超高分辨率视频处理的稀疏几何结构

背景技术

[0001] 本发明涉及数字视频处理的方法。具体地,本发明能够用在超高分辨率视频处理领域。超高分辨率视频处理方法应用于包括超高分辨率插值(例如,帧频转换,超高分辨率视频的缩放和去隔行扫描)以及减少压缩伪像和/或噪声的各种应用中。

[0002] 在数字系统中,视频序列通常表示为像素值的列阵 $I_t(x)$, 式中 t 为整数时间索引, x 为一个表示图像中的像素位置的二维整数索引 (x_1, x_2) 。该像素数值可以是例如单一数字(例如,灰度值)或表示在颜色空间(例如, RGB, YUV, YCbCr, 等)中的颜色坐标的三数集。

[0003] 超高分辨率视频处理方法通过结合在多个时间相邻的视频帧的像素值来计算新像素值(用于插值)或者已有像素的新值(降低噪声)。

[0004] 文献 WO 2007/115583 A1 阐述了一种显示很少伪像的超高分辨率视频处理方法。该方法包括为每个将要计算的新像素选择最适合计算该像素的插值。然而,对某些特定序列,就需要通过增加被考虑插值的总数量来改善该方法。但是,质量的提高是以较高复杂度为代价的。

[0005] 在视频插值的应用中,现有技术为动态自适应或动态补偿。

[0006] 当视频不运动时,动态自适应视频的去隔行扫描只提供全分辨率的去隔行扫描帧。否则,去隔行扫描帧会出现锯齿状的轮廓或者较低分辨率纹理以及闪烁。例如,美国专利 No. 5, 428, 398 描述了先进动态自适应技术的实例。

[0007] 众所周知,动态补偿技术可达到较高的质量等级,但与动态自适应技术相比,是以较低的鲁棒性和在有些情况下显示较为严重的伪像为代价的。这种情况尤其发生在动态预测不能很好工作的视频的位置,如遮挡物,透明物体或阴影。美国专利 No. 6, 940, 557 讨论了动态补偿去隔行扫描技术的实例。

[0008] 执行帧频转换的标准方法包括用于计算密集动态区域的预测两帧之间的动态预测,以及通过动态补偿插值计算新的帧。由于上述相同的原因,根据这样步骤的帧频转换会有一些缺点。密集的动态预测对周期模式、轮廓、或平坦区域失去效果。

[0009] 动态预测的常用技术称为“块匹配”。在块匹配技术中,预测在 x 和 t 上的动态包括通过窗 W 的最小化匹配能量 $E_x(v)$, 其中一组偏移量为 $d = (d_1, d_2)$ 。匹配能量 (L_1 -energy)

的可能形式为 $E_x(v) = \sum_{d \in W} |I_t(x+d) - I_{t+1}(x+d+v)|$ 。另一种经常使用的形式为 L_2 能量

(L_2 -energy) 或欧几里得距离 (Euclidean distance) : $E_x(v) = \sum_{d \in W} |I_t(x+d) - I_{t+1}(x+d+v)|^2$ 。

[0010] 块匹配非常适合视频补偿方案中的动态补偿,例如,使用根据块变换的 MPEG。如果匹配算法匹配图像中有两个窗口,该图像具有相似性但不是表示相同对象(例如,在单词“sweet”的图像中,用第二个‘e’匹配第一个‘e’),则压缩效果不会有损失。但是,当执行视频插值时,不能与实际相同对相对应的匹配像素组会产生插值伪像,这是由于图像中所显示的对象中的空间相关性使得插值像素会反映出“不正确的动态”。

[0011] 块匹配的方法是计算密集型的,且与各个像实际考虑的可能位移的数量成比例。在视频压缩中,“快速”块匹配方案包括使用预定动态子集合来限制可能位移的范围。这在

视频插值中是不能接受的,在视频插值中,使用不太精确的位移向量会导致模糊插值图像或者形成伪像。

[0012] 为了避免在运动预测中的这些问题,已开发多种不同的方法。第一类方法强调在动态区域中的平滑性约束,例如,通过相邻的像素进行平滑处理,因而对相应的动态向量是相互接近的。该方法可利用多个尺度的动态预测或递归块匹配来实现。另一类解决该问题的方法是相位相关。

[0013] 美国专利 No. 5, 742, 710 讨论了根据多个尺度的块匹配的方法。在两个尺度的情况下,块匹配在各个维度减少 2 倍的尺寸因子(即,像素减少 4 倍)的 I_t 和 I_{t+1} 的副本之间执行,并细化所产生的位移图从而获得好两倍的分辨率。细化处理是在粗尺寸结果周围的限制范围的搜索。因此,因为仅仅只对较小的图像进行全范围的搜索,可以降低位移搜索的代价。所产生的位移区域同样更加平滑,因为其进精细化了低分辨的图(map)。但是,在视景(scene)中的动态不能通过平滑的位移图进行准确的计算:动态区域基本上不连续,尤其是在遮挡对象的附近。实施位移图的平滑约束不是解决鲁棒性问题的合适方法。

[0014] 以相似方法来解决这个问题的另一类方法是递归块匹配,如文献" True-Motion with 3D Recursive Search Block Matching", G. De Haan et al., IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, No. 5, October 1993, pp. 368-379 所阐述的。这种方法显著降低了计算动态图的代价,但是仍然会受到周期模式或者甚至遮挡物误导。

[0015] 文献 GB-A-2 188 510 阐述了称为相位相关的方法,其为候选位移集合通过一个大图像窗计算位移能量图。该图可以通过快速傅立叶变换进行有效的计算。对应于在该能量图中峰值的位移子集确定为这个窗所包括的最具代表性的位移。随后,块匹配仅以考虑这个位移子集的第二步骤像素(pixelwise)进行执行。

[0016] 这个方法降低了动态预测的复杂性,并且还能够检测不连续的动态图。根据相位相关技术,动态图也是规范和约束的,但非常不同于空间规范方法。取代了对动态图进行局部平滑,相位相关限制了在动态图中的不同可能向量集合的固定数量。

[0017] 但是,相位相关还要求根据二维快速傅里叶变换的相对复杂计算,其硬件实施的代价很大。此外,该方法根据动态向量的相位相关有关的各自特性来选择动态向量。所以,其具有限制提供最小动态向量集合的能力。当然,当一个移动模式具有周期性结构或者位移不变时,不同的向量具有可比较特性量,并且难以对它们之间的相位相关进行判断。因此,产生的动态补偿视频插值处理方法具有次优的鲁棒性。这也具有复杂性的代价,因为对于所有的像素都需要考虑多于必要数量的候选动态向量。

[0018] 其它类型的方法包括通过在候选向量中计算低复杂性匹配能量进行选择第一位移子集。这样可以在某种程度上降低计算的复杂性,但是,这不是一种使动态补偿插值更加可靠的合适方法。

[0019] 减少视频序列噪声的典型和最普遍的方法包括动态补偿递归或非递归时间滤波。可以参考,例如文献" Noise reduction in Image Sequences Using Motion-Compensated Temporal Filtering", E. Dubois and S. Sabri, IEEE Transactions on Communications, Vol. COM-32, No. 7, July 1984, pp. 826-832。该方法包括了在帧和先前帧之间的动态预测,以及通过时间滤波器随着预测动态对视频序列进行滤波。

[0020] 其它已知的方法是用动态补偿的三维(3D)小波变换。参考,例如文献"Three-Dimensional Embedded Subband Coding with Optimized Truncation(3D-ESCOT)", Xu, et al., Applied and Computational Harmonic Analysis, Vol. 10, 2001, pp. 290-315。该文献所阐述的动态补偿三维小波变换可用于通过对这样的三维变换进行小波阈值来减少噪声。这种使用根据随着动态趋势提升小波变换的方法的限制是它对由噪声产生的动态图的衰减非常敏感。

[0021] 文献 WO 2007/059795 A1 阐述讨论了超高分辨率处理方法,它可用于长程降噪或超高分辨率的缩放。该方法根据使用小波系数的多尺度分组的 Bandlet 变换 (bandlet transform)。这种表达方法比 3D-ESCOT 文献所阐述的三维变换更加适用于降噪或超高分辨的缩放。多尺度分组执行变化范围的图像配准,其可通过例如块匹配或者现有技术的图像配准过程进行计算。对超高分辨率缩放和降噪来说,所使用的图像配准图不被噪声或伪像损坏是非常重要的。

[0022] 不论怎样的应用(插值或噪声减小),具有密流场(dense flowfield)的动态补偿的使用都会有些限制:光圈、对透明或阴影部分内容的单一动态模型。正如文献 WO 2007/115583 A1 所讨论的,通过检测各个像素在视频信号空间和时间中的一个或多个规律趋势对视频的局部不变结构进行分析,从而为视频插值提供更为普遍和鲁棒的方法。因此,就需要一种能够以有效方法和具有增强鲁棒性的方式来检测这种趋势的技术。

[0023] 本发明的一个目的是提供了一种具有高精度度和高鲁棒性在输入视频流中检测规律趋势的有效方法。尤其是,在超高分辨率视频插值中,能够避免通常由不连续的插值趋势所形成的伪像。在视频降噪中,能够选择不被噪声损坏的平均趋势。

[0024] 另一个目的是显著降低实施超高分辨率插值或降噪处理方法的复杂性。

发明内容

[0025] 阐述一种分析输入视频序列的方法,其中输出视频序列合成图像的像素与属于趋势的预定集的各个规律趋势相关联。该方法包括:从趋势的预定集中确定输出序列的第一图像范围的候选趋势的第一子集;根据输入序列的图像和候选趋势的第一子集,从趋势的预定集中确定第一图像后的第二合成图像序列对应范围的候选趋势的第二子集;以及从所述候选趋势的第二子集中检测所述第二合成图像范围的像素规律趋势。

[0026] 候选趋势的子集通过考虑先前时间所决定的子集在时间递归中进行确定。典型的方法是,在子集中的趋势的增加或去除是根据由这样的增加或去除产生的代价函数的增加变化进行的。如下文所述,图像的“范围”可以包含整个图像区域或者仅仅是其一部分。

[0027] 候选趋势的第二子集的确定可以包括:检测至少一对趋 v_r 和 v_a ,使得 v_r 属于候选趋势的第一子集, v_a 属于趋势预定义子集且不属于第一子集,并且与涉及第一和第二图像的第一子集相关的代价函数高于与包括 v_a 和不包括 v_r 的第一子集的趋势的改进子集相关的代价函数;以及对检测的响应,从第二子集中去除 v_r 并在第二子集中包括 v_a 。

[0028] 该技术可使用简单的操作和结构,来加速规律趋势的检测或者降低其实施的代价。它减少了动态补偿视频插值所产生的伪像的数量。

[0029] 一些实施例的特征包括评估由新的趋势提供给已有趋势的子集的相关边界增益。相反,在动态预测的某些具体领域中的大多现有方法仅仅使用位移向量的绝对效率测量,

并不考虑已使用的位移。本方法选择稀疏趋势集 (sparser direction sets), 并且还对不同的伪影进行保留管理。

[0030] 例如, 已知的相位相关方法包括根据总的相位相关测量来在图像范围内寻找最佳位移。在一定的图像范围内, 所有的候选位移 V_i 都具有相关的相位相关值, 并表示为 $P(V_i)$, $i = 1, \dots, n$ 。因此, 一个理想的子集是由具有最高相位相关值的位移所组成。其可以与所选择的 m 个趋势的子集 $(V_i)_{i \in S}$ 进行比较, 使得 $\sum_{i \in S} P(V_i)$ 最大化。在趋势子集中的函数

$\sum_{i \in S} P(V_i)$ 是独立的, 也就是说, 它可以写成单独用于各个趋势的函数之和。通常进行这样的选择, 因为这是可直接最小化函数且不会导致组合爆增的唯一情况。为了从相位相关观点出发来获得理想的子集 S , 使函数 P 获得最大值的 m 个趋势能在该序列中被简单地挑选出。

[0031] 然而, 如果这个函数不是独立的并且只可写成 $P(\{V_i\}_{i \in S})$, 就仅使用这样简单算法不能完成最小化。直接获得最佳候选子集具有高的组合复杂性。但是, 在某些情况中, 当从例如 S 和 S' 仅仅只有一个元素不同的 $P(\{V_i\}_{i \in S}) - P(\{V_i\}_{i \in S'})$ 的已选择子集中增加或删除一个向量或趋势时, 函数的变化计算仍然可以完成。那么, 这样得到了能够在时间递归方法中实现函数增量优化的方法。

[0032] 因此, 在某些实施例, 候选趋势的第二子集的确定包括: 评估涉及第一子集各个趋势的对第一子集相关的代价函数各自贡献的第一边界; 评估涉及从预定集的各自趋势增加所产生的对第一子集代价函数的分别减少的第二边界; 以及, 当用于所述预定集的趋势评估的第二边界超出用于所述第一子集的趋势评估的第一边界时, 将第一子集的趋势替换成预定子集的趋势。值得注意的是, 尽管诸如相位相关的技术最大化了整体的相关性测量, 但最小化了整体的代价函数。

[0033] 视频序列的超高分辨率处理可以是插值或降噪。简单的减噪也是可行的。

[0034] 输入视频序列 $I_t(x)$ 由称为“原始像素”的点网格 (x, t) 定义。输出视频序列 $\hat{I}_\tau(\xi)$ 由称为“目标像素”的点网格 (ξ, τ) 定义。像素由位置 (x, t) 或 (ξ, τ) 和在该位置上称为“像素值”的视频图像数值 $I_t(x)$ 或 $\hat{I}_\tau(\xi)$ 定义。

[0035] 在视频插值具体情况中, 一些分布在空间和 / 或时间中的目标像素 $\hat{I}_\tau(\xi)$ 还可以为原像素 $I_t(x)$ ($\tau = t, \xi = x$), 并且不需要进行重新计算, 因为我们可以选取 $\hat{I}_\tau(\xi) = I_t(x)$ 。必须要计算数值的像素为那些不是原始像素 $I_t(x)$ 、称为“新的像素” ($\tau \neq t$ 或 $\xi \neq x$) 的目标像素 $\hat{I}_\tau(\xi)$ 。

[0036] 在视频去隔行扫描的情况中, 输入和输出视频序列的帧频通常是相同的, 使得在输出序列中的时间索引 τ 可以与在输入序列的时间索引 t 相同; 它们通常可以由例如整数索引 $t, t+1$ 表示。视频去隔行扫描处理方法包括在输入序列的连续帧中增加插值缺少的行。典型的是, 当偶数帧只有偶数行时, 输入序列的奇数帧只具有奇数行, 例如, 对于 $x = (x_1, x_2)$, 只有当 t 和 x_2 同时为奇或同时为偶时, 输入视频序列才提供 $I_t(x)$ 。输出的去隔行扫描的视频序列的合成帧 \hat{I}_t 由具有 $\xi = (x_1, x_2)$ 并不具有任何整数行索引 x_2 的奇偶约束的像素 $\hat{I}_t(\xi)$ 构成, 这样当 t 和 x_2 同时为奇或同时为偶时, 使得 $\hat{I}_t(\xi) = I_t(\xi)$ 。视频去隔行扫

描的目的是为 $\hat{I}_t(\xi)=\hat{I}_t(x_1,x_2)$ 执行“最佳”数值插值,其中 t 和 x_2 中的一个为奇数且另一个为偶数。为了实现这样的插值,检测帧间和 / 或帧内的规律趋势是帮助的。

[0037] 在帧频转换的情况中,输入和输出视频序列中的时间索引 t 、 τ 是不同的。例如,可以使用整数 t 、 $t+1$ 等等作为输入序列的帧索引,并因此有些帧 \hat{I}_τ 为非整数值 τ 而合成。输入和输出帧 I_t 、 \hat{I}_τ 中的空间索引 $\xi = x = (x_1, x_2)$ 通常是相同的。帧频转换的输出序列包括为非整数值 τ 的合成帧 \hat{I}_τ 。另外,为了合成这些干涉 (intervening) 帧,可进行插值,通过分析输入视频序列有助于检测规律趋势。为了检测合成输出帧 \hat{I}_τ 像素的规律趋势,分析至少会涉及在非整数时间索引 τ 相邻前后 (例如 t 为满足 $t < \tau < t+1$ 的整数) 的输入序列的帧 I_t 和 I_{t+1} 。

[0038] 在视频降噪的情况中,所有的目标像素值必须重新计算。根据这些转换,结合超高分辨率的缩放和降噪是超高分辨率降噪的情况。对简单的降噪而言,目标像素点网格 (ξ , τ) 与原始像素网格 (x, t) 相同: $\hat{I}_t(x)=I_t(x) - v_t(x)$, 其中 $v_t(x)$ 为由处理过程消除的噪声分量预测。对结合超高像素降噪和缩放情况而言,目标像素由不同于原始像素网格 (x, t) 的网格 (ξ , τ) 定义。该网格 (ξ , τ) 通常为如可以由原始像素网格 (x, t) 的超集定义的更精细的网格。

[0039] 本发明的另一方面涉及计算机程序产品,包括当计算机处理单元运行所述计算机程序产品时执行上述视频分析方法的指令。

[0040] 本发明的另一方面涉及视频处理方法,包括:接收输入视频序列的连续图像;应用上述方法分析输入视频序列;以及使用检测到的规律趋势产生输出视频序列。

[0041] 产生视频序列的步骤可以包括使用检测到的规律趋势在输入视频序列的连续图像中进行插值。这样的插值可以包括视频的去隔行扫描或输入视频序列的帧频转换。在另一实施例中,视频序列的处理可以包括使用检测到的规律趋势对输入视频序列进行降噪处理。

[0042] 本发明的另一方面还涉及视频处理装置,包括用于分析和处理如上所述的视频序列所设置的计算电路。

附图说明

[0043] 图 1 示出了视频处理装置实施例的框图。

[0044] 图 2 示出了适用于图 1 装置的趋势选择单元实施例的框图。

[0045] 图 3 示出了在如图 1 和图 2 中所示装置中的评估代价函数边界的典型操作过程的流程图。

[0046] 图 4 示出了在图 3 操作过程中所使用操作流程的变化例的流程图。

[0047] 图 5 和图 6 示出了如在图 1 和图 2 所述装置中的候选趋势间的判断的典型操作过程的流程图。

[0048] 图 7 和图 8 图示说明了应用于某些图像部分的帧频转换处理。

[0049] 图 9、10、11 和 12 图示说明了应用于相似图像部分的视频去隔行扫描处理。

[0050] 图 13 示出了根据本发明实施例的视频处理装置的框图。

具体实施方式

[0051] 参考图 1, 视频处理装置能够接收连续图像或者视频序列帧的数字表示式的输入。 I_t 、 I_{t+1} 表示在离散时间 t 和 $t+1$ 的帧, 而 $I_t(x)$ 、 $I_{t+1}(x)$ 表示在二维索引 $x = (x_1, x_2)$ 像素中的那些帧的像素值。所采用的时间索引 t 和空间索引 x 的管理方法可以根据各个不同的视频处理应用 (例如, 在去隔行扫描, 帧频转换和降噪之间) 而各所不同。这个问题将在下文中进行阐述。

[0052] 趋势选择单元 101 实现时间递归预测, 以便根据先前子集 D_τ 和连续输入帧, 为输出帧 \hat{I}_τ 确定候选趋势的子集 $D_{\tau'}$ 。上述“先前子集 D_τ ”确定为在输出视频序列中紧邻 \hat{I}_τ 之前的输出帧 \hat{I}_τ 。例如, $\tau' = \tau + 1$ 用于去隔行扫描或简单降噪; $\tau' = \tau + \delta$ τ 用于帧频转换或超高分辨率降噪。在时间 τ' 确定的子集 $D_{\tau'}$ 所涉及的输入帧至少包括 I_t 和 I_{t+1} 且使得 $t \leq \tau' < t+1$ 。在某些实施例中, 它们还可以包括一些先前帧 I_{t-1}, \dots, I_{t-n} ($n \geq 1$)。

[0053] 如本文提及的, “趋势” $v = (dx, dt)$ 定义为三维空间中的趋势, 在三维空间中, 二维涉及二维图像空间的像素补偿偏移 $dx = (dx_1, dx_2)$, 并且第三个趋势涉及时间偏移 dt 。有许多视频应用希望在输入视频序列中寻找规律的趋势。例如, 当进行视频插值时, 就必须根据在丢失像素临近区域中的“相似”像素来确定一些丢失像素的数值。这种临近区域可以在二维图像空间和 / 或 / 时间中扩展, 使之相关于在上述三维空间中所寻找的丢失像素。相同的, 在降噪应用中, 输入像素的数值由于噪声而损坏, 如果有可能识别出“相似”像素的一些临近区域, 就能通过平均得出噪声。此外, 这种邻近区域可以在二维图像空间和 / 或时间中扩展。以下所讨论的方法可产生图像的像素的规律趋势, 其用于确定有助于处理的“相似”像素数值。

[0054] 所述的子集 D_τ 或 $D_{\tau'}$ 用于定义稀疏几何结构。各个子集 D_τ 或 $D_{\tau'}$ 是包括所有规律的可能趋势的集 Ω 的子集。由于对每个间隙 τ 、 τ' , 由 D_τ 、 $D_{\tau'}$ 定义的几何结构可用于稀疏, 因此可以使用的不同趋势的数量可限制在相对较小的数量内。如下所述, 候选趋势 D_τ 、 $D_{\tau'}, \dots$ 的子集随着边界变化在时间中进行变化。去除在 D_τ 、 $D_{\tau'}$ 中的多余趋势, 并且不在逐个像素处理中使用。

[0055] 典型的是, Ω 可以包括 200 至 1000 个不同的趋势 ($200 \leq |\Omega| \leq 1000$, 这个范围用于表示集的大小)。子集 D_τ 、 $D_{\tau'}, \dots$ 可以具有限制在 $10 \leq |D_\tau| \leq 50$ 范围内的大小。

[0056] 然后, 趋势检测单元 102 通过检测属于由选择单元 101 确定子集 $D_{\tau'}$ 的唯一候选趋势, 根据连续帧 I_t, I_{t+1} (可能和一些先前帧 I_{t-1}, \dots, I_{t-n}) 来确定规律趋势的分布 $\{v\}$ 。大小从 Ω 减小至 $D_{\tau'}$ 就有可能在不需要高复杂性的条件下进行这样的检测。

[0057] 最终, 视频处理单元 103 使用规律 $\{v\}$ 所检测到的趋势, 来进行诸如去隔行扫描, 帧频转换或降噪等的视频处理, 以根据输入帧 I_t, I_{t+1} 获得输出视频帧。

[0058] 单元 102 和 103 可以实现任何传统的或现有技术的方法, 并给出简单的实施例将在下文中进行补充。具体的, 检测单元 102 可以使用在文献 W0 2007/115583 A1 中描述的损失函数。本发明的核心部分在单元 101 中, 且在下文中作更详细的阐述。

[0059] 当趋势选择单元 101 考虑大于趋势检测单元 102 的趋势集时, 一个有趣的可能是在单元 101 中使用比单元 102 中更简单的检测或代价函数。换言之, 局部代价函数在确定

趋势子集 $D_{\tau'}$ (选择单元 101) 步骤中的评估比在从子集 (趋势检测单元 102) 获取趋势步骤中的评估更为粗略。这在计算复杂性方面或等效的 ASIC/FPGA 的逻辑尺寸方面有着显著的节省。

[0060] 可以这样来实现:例如,使用精确度稍差的像素数值表达式。例如,以在单元 101 中的 5 或 6 比特的像素数值代替在单元 102 中的 8 至 10 比特的像素数值。另一个可能性为在趋势选择单元 101 中使用卷积窗 g (将在下文中阐述),卷积窗比在趋势检测单元 102 中所使用的计算更为简单,即窗口特性对应于简单无限脉冲响应 (IIR) 滤波器,它不需要像有限脉冲响应 (FIR) 滤波器那么多的逻辑和那么大的内存。此外,不同计算复杂性的代价函数 (在下文中进行叙述) 可以用于在单元 101 中的子集选择以及为在单元 102 中的像素趋势检测。

[0061] 选择单元 101 的目的是计算趋势子集 $D_{\tau'}$, 该子集用于提供在输出序列的间隙 τ' 中的视频序列局部规律的有效表达式。最佳的子集 D 是能最小化总代价 (或损失) 函数 $L(D)$ 中的那个子集:

$$[0062] \quad L(D) = \sum_x \min_{v \in D} [L_x(v)] \quad (1)$$

[0063] 其中,在像素 (x) 上总和覆盖了整个图像区域 (或其部分)。用于最小化 D 中的候选趋势 v 的数量 $L_x(v)$ 为局部代价 (或损失) 函数,其可以为 $v = (dx, dt)$ 具有各种不同的类型,例如:

$$[0064] \quad \text{绝对差: } L_x(v) = |I_t(x) - I_{t+dt}(x+dx)|$$

$$[0065] \quad \text{平方差: } L_x(v) = |I_t(x) - I_{t+dt}(x+dx)|^2$$

$$[0066] \quad \text{绝对差的加权和: } L_x(v) = \sum_d g(d) |I_t(x+d) - I_{t+dt}(x+d+dx)|$$

$$[0067] \quad \text{平方差的加权和: } L_x(v) = \sum_d g(d) |I_t(x+d) - I_{t+dt}(x+d+dx)|^2$$

[0068] 其中 g 为卷积窗函数,即在 $(0,0)$ 附近的非零数值。

[0069] 其它的变化类型都是有可能,包括通过计算多于视频序列的两个帧的局部代价函数,即: $L_x(v) = |I_t(x) - I_{t+dt}(x+dx)| + |I_t(x) - I_{t-dt}(x-dx)|$, 以及相似的变化。

[0070] 为了简化,我们还可以定义趋势集合的局部代价 $L_x(D)$ 为该集中的所有趋势范围内的损失函数的最小值:

$$[0071] \quad L_x(D) = \min_{v \in D} [L_x(v)] \quad (2)$$

[0072] 值得注意的是,获得最小化的子集 $D(1)$ 是非常组合复杂的,因为为子集 D 增加趋势的数值取决于在该子集中已经存在着的趋势。为了解决这个难点,提出了增量方法。通过只以时间对 $D_{\tau}, D_{\tau'}, \dots$ 进行边界变化,从而使用时间递归来实现最小化。

[0073] 如在图 2 所示的趋势选择单元 101 具有评估模块 201,用于对可能的趋势集 Ω 的不同趋势 v 进行边界 $m(v)$ 评估,和判断模块 202,用于根据边界 $m(v)$ 来决定哪一个 D_{τ} 中的趋势应该从 $D_{\tau'}$ 中去除,以及哪一个 $\Omega - D_{\tau}$ 中的趋势应该包括在 $D_{\tau'}$ 中。根据公式 (1),选择加入至 D_{τ} 而获得 $D_{\tau'}$ 的趋势 v 根据它们对代价函数改善 (减小) 的边界贡献多大进行选择。相反,要从 D_{τ} 中去除的趋势 v 根据它们对减少代价函数 $L(D)$ 的边界贡献多小进行选择。

[0074] 通过对于可构成 $D_{\tau'}$ 的各种组合 D 评估 $L(D)$, 判断哪些在 $D_{\tau'}$ 中不能完成的元素。然而, 使用对应于现存趋势子集 D 的趋势 v 、标记为 $m(v|D)$ 的边界, 可以用来预测将 $\Omega - D$ 新的趋势 v 增加到 D 时的 $L(D)$ 的变化:

$$[0075] \quad m(v|D) = L(D) - L(D + \{v\}) \quad (3)$$

[0076] 式中 $D + \{v\}$ 表示集 D 和单个 $\{v\}$ 的并集。换言之, $m(v|D)$ 是测量新的趋势对降低趋势 D 子集已获得的代价函数 (1) 的边界贡献。边界 $m(v|D)$ 的计算可以使用:

$$[0077] \quad m(v|D) = \sum_x m_x(v|D) \quad (4)$$

[0078] 其中在对应于 D 的 v 的 x 位置上的局部边界 $m_x(v|D)$ 为:

[0079] • $m_x(v|D) = 0$ 当 $L_x(v) \geq L_x(D)$, 即, 当 v 在最小化像素位置 x 的代价函数时不优于在 D 中已有的趋势时;

[0080] • $m_x(v|D) = L_x(D) - L_x(v)$ 其它

[0081] 通过确定 $L_x(D)$ 和 $L_x(v)$ 的数量来计算固定 D 和在 $\Omega - D$ 中的各个 x 和候选 v 的边界 $m_x(v|D)$ 。然后, 通过更新执行 $m_x(v|D)$ 求和来计算 $m(v|D)$ 。

[0082] 让我们考虑包括新趋势 v_a , 和从 D_{τ} 中去除已选择趋势 v_r , 以便计算 $D_{\tau'}$ 的情况, 如:

$$[0083] \quad D_{\tau'} = D_{\tau} - \{v_r\} + \{v_a\}.$$

[0084] 由这样的变换引起的总代价 (1) 的降低可以表示为变换边界 $M_{\text{exch}}(v_a, v_r)$:

$$[0085] \quad M_{\text{exch}}(v_a, v_r) = L(D_{\tau'}) - L(D_{\tau} - \{v_r\} + \{v_a\})$$

$$[0086] \quad = m(v_a|D_{\tau'}) - m(v_r|D_{\tau} - \{v_r\} + \{v_a\}) \quad (5)$$

[0087] 如果 $M_{\text{exch}}(v_a, v_r) > 0$, 即, $m(v_a|D_{\tau'}) > m(v_r|D_{\tau} - \{v_r\} + \{v_a\})$, 在 D_{τ} 中用趋势 v_a 替换趋势 v_r 来减少总代价, 使得交换 v_r 和 v_a 是值得的。这些不同边界的计算是容易的, 但是其仍然可能有效的减少计算的数量。这样可以理解为下述: “如果 v_a 提供了比 v_r 所提供的更大的总代价的边界减少, 这样的交换是合理的。” 在这样的方法中, 替代在公式 (5) 中精确计算边界 $m(v_r|D_{\tau} - \{v_r\} + \{v_a\})$, 可以获得一些近似项。

[0088] 在第一类近似中, $m(v_r|D_{\tau} - \{v_r\} + \{v_a\})$ 由 $m(v_r|D_{\tau} - \{v_r\})$ 替换。通常可以验证下述不等式:

$$[0089] \quad m(v_r|D_{\tau} - \{v_r\}) \geq m(v_r|D_{\tau} - \{v_r\} + \{v_a\}) \quad (6)$$

[0090] 由该近似所获得的复杂增益是十分显著的。用于计算的边界数量现在是序列 $|D|$ 而不是 $|\Omega - D| \times |D|$ 。使用该近似, 由此可获得交换边界 $M'_{\text{exch}}(v_a, v_r)$, 如下式:

$$[0091] \quad M'_{\text{exch}}(v_a, v_r) = m(v_a|D_{\tau'}) - m(v_r|D_{\tau} - \{v_r\}) \quad (7)$$

[0092] 值得注意的是, 在公式 (7) 中的交换边界 $M'_{\text{exch}}(v_a, v_r)$ 不超过在公式 (5) 中的实际交换边界 $M_{\text{exch}}(v_a, v_r)$ 。如果近似的交换边界 $M'_{\text{exch}}(v_a, v_r)$ 为非负数, 那么实际交换边界 $M_{\text{exch}}(v_a, v_r)$ 也为非负数。因此, 根据公式 (7) 确定的交换不可能为公式 (5) 中的错误的那个。

[0093] 图 3 示出了由模块 201 对公式 (7) 中的对边界 $m(v_a|D_{\tau'})$ 和 $m(v_r|D_{\tau} - \{v_r\})$ 进行评估使用的程序流程图。在图 3 中, 假设: 对趋势选择单元 101 所接收到的各个新的输入帧 $I_{\tau+1}$ 确定候选的趋势中的一个子集 $D_{\tau'}$ 。这个假设对于视频去隔行扫描或简单的降噪 (即, $\tau = t$, $\tau' = t+1$) 或帧频倍频 ($\tau = t-1/2$, $\tau' = t+1/2$) 有效。通常, 采用不是 2 的

比率来变换帧频都是简单直接进行的（在图 3 中示出的处理类型通常是获得各个新输出帧而产生的；上述假设只为作出更明晰的阐述，因为其意味着新输出帧频与输入帧的速率相同）。根据该假设，我们可以由在操作中的时间递归而舍弃时间索引 $t-1$ 、 t 和 τ 、 τ' 。此外，如果趋势 $v (= v_r)$ 是在 $D (= D_r)$ 中并且可去除，那么以 $m(v)$ 代表 $m(v_r | D_r - \{v_r\})$ ，如果趋势 $v (= v_a)$ 在 $\Omega - D$ 中并且可加入 D 中，那么以 $m(v)$ 代表 $m(v_a | D_r)$ 。边界 $m(v)$ 对在 Ω 中的所有趋势 v 通过更新使用求和进行评估，所述求和在操作中的初始化 301 中进行零设置。

[0094] 该程序依次扫描 I_t 和 I_{t+1} 帧阵列中的像素 x ，在步骤 302 中选择第一像素。在趋势 D 中 v 范围中执行第一循环 310，以更新对应于像素 x 的趋势 $D (= D_r)$ 的求和。通过获取 D 中的第一趋势 v 并将变量 A 设置成任意大的数值（例如，其最大可能的数值），在步骤 311 中初始化第一循环 310。在循环 310 结束时，变量 A 将包括在公式 (2) 中的定义 $L_x(D)$ 的数值。

[0095] 在循环 310 的每一次递代（步骤 312）中，获得像素 x 和趋势 v 的局部代价 $L_x(v)$ 并且载入变量 L 中。在步骤 312 中，模块 201 也可计算 $L_x(v)$ ，例如根据上述可能性之一，或如果代价 $L_x(v)$ 预先计算过，则从内存中调用。测试步骤 313 执行 L 是否小于 A 的评估。如果 $L < A$ ，则在步骤 314 中将趋势索引 v 存储在变量 u 中并使变量 B 接受数值 A 。随后，在步骤 315 中将数值 L 分配给变量 A 。在循环 310 的结束时，变量 u 将包括可最小化 $L_x(v)$ 的 D 中的趋势 v 的索引，即 $u = \operatorname{argmin}_{v \in D} [L_x(v)]$ ，以及变量 B 包含适用于 D 中的趋势 v 的 $L_x(v)$ 的

第二最小的数值，即 $B = \min_{v \in D - \{u\}} [L_x(v)]$ 。如果在测试步骤 313 中 $L \geq A$ ，则在步骤 316 中将局部代价与 B 进行比较。如果 $A \leq L < B$ （在测试 316 中为真 (yes)），则在步骤 317 中变量 B 更新为数值 L 。如果在测试 316 中或在步骤 315 或 317 后有 $L \geq B$ ，则执行结束循环测试 318，以检测是否 D 中的所有趋势 v 都已扫描过。如果为否 (not)，则在步骤 319 选择 D 中的另一个趋势 v ，并且程序流程返回至步骤 312，以执行循环 310 的另一次递代。

[0096] 当循环 310 结束时，在像素 x 处最小化局部代价的 D 中的趋势 u 的边界 $m(u)$ 由增加数量 $B-A$ 进行更新（步骤 321）。就像素 x 而言，从 D 中去除 u 就能使代价减小那个数量，同时 D 的其它趋势的边界也不会受到影响。

[0097] 随后，由第二循环 330 对不在 D 中的可能趋势 v 的像素 x 继续进行处理，以便为涉及像素 x 的 $\Omega - D$ 趋势求和进行更新。

[0098] 在步骤 331 中，第二循环通过获取 $\Omega - D$ 中的第一趋势 v 进行初始化。在每次递代中（步骤 332），计算或调用像素 x 和趋势 v 的局部代价 $L_x(v)$ ，以载入变量 L 中。随后，进行检测 333，以评估 L 是否小于 $A = L_x(D)$ 。如果 $L < A$ ，则趋势 v 的边界 $m(v)$ 通过增加数量 $A-L$ 进行更新（步骤 334），以考虑代价函数的优化，该代价函数由把 v 加入至对应于像素 x 的 D 而获得。如果在步骤 333 中或在步骤 334 后有 $L \geq A$ ，则执行结束循环检测 335，检测是否 $\Omega - D$ 中的所有趋势 v 都已扫描过。如果为否，则在步骤 336 中选择 $\Omega - D$ 中的另一个趋势 v ，并且处理步骤返回至步骤 332，以执行循环 330 的另一次递代。

[0099] 当循环 330 结束时，在步骤 341 中确定是否相关帧阵列的所有像素 x 都已扫描过。如果为否，则在步骤 342 中选择阵列中的另一个像素 x ，并且处理流程返回至步骤 311。当检测 341 中显示所有的像素已经处理时，那么对当前帧的模块 201 操作就结束。

[0100] 对每个新的输入帧 I_{t+1} , 模块 201 因此为 Ω 所有趋势 v 输出边界 $m(v)$, 即, 为 D 趋势去除边界和为 $\Omega - D$ 趋势增加边界。

[0101] 为了在输入视频序列开始时初始处理程序, 子集 D 可以具有任意的内容, 或者它也可对前几帧以粗略的方法进行确定。正确的子集会因为选择程序流程的时间递归而快速组建。

[0102] 第二项近似可以用于进一步降低模块 201 的复杂性。在该近似中, $m(v_a | D_t)$ 由已修改的边界 $m^*(v_a | D_t)$ 替换。如公式 (4), 修改的边界 $m^*(v | D)$ 为像素之和:

$$[0103] \quad m^*(v | D) = \sum_x m^*_x(v | D) \quad (8)$$

[0104] 局部修改边界 $m^*_x(v | D)$ 定义为:

[0105] • $m^*_x(v | D) = L_x(D) - L_x(v)$ 当 $L_x(v) < L_x(\Omega - \{v\})$ 时, 即: 当 v 是从最小化在像素位置 x 上的函数的观点来看是 Ω 中的最佳趋势;

[0106] • $m^*_x(v | D) = 0$ 其它。

[0107] 根据第一和第二近似, 修改的交换边界 $M^*_{\text{exch}}(v_a, v_r)$ 可以表示为下式:

$$[0108] \quad M^*_{\text{exch}}(v_a, v_r) = m^*(v_a | D_t) - m(v_r | D_t - \{v_r\}) \quad (9)$$

[0109] 此外, 由于公式 (6) 和因为 $m^*_x(v_a | D) \leq m_x(v_a | D)$, 修改的交换边界 $M^*_{\text{exch}}(v_a, v_r)$ 不会大于实际的交换边界 $M_{\text{exch}}(v_a, v_r)$ 。所以, 从公式 (5) 的观点来看, 根据公式 (9) 所确定的交换不可能再是错误的了。

[0110] 修改的边界 $m^*_x(v_a | D)$ 可以较低代价的计算或电路进行计算, 因为, 对各个位置 x , 必须更新最多一个对应于在 $\Omega - D$ 中的单一绝对值的最佳趋势的求和; 然而, 也可采用未修改的边界 $m_x(v_a | D)$, 这样在最坏情况中 (在图 3 中, 检测 333 始终为真) 所获得的成功数量等于 $|\Omega - D|$ 。在使用硬件连线的 ASIC 或 FPGA 电路的实施例中, 在逻辑大小上影响是非常重要的。因为相同的原因, 在软件实施中的最坏情况的执行时间的的影响也是重要的。

[0111] 采用第二近似, 图 3 所示的处理流程可由图 4 所示的修改循环 430 来替换循环 330 进行修改。在步骤 431 (取代步骤 331) 中, 循环 430 通过获取 $\Omega - D$ 中的第一趋势 v 进行初始化, 并且将另一个变量 A^* 设置成数值 $A = L_x(D)$ 。在循环 430 结束时, 变量 A^* 包括在 Ω 中的所有趋势 v 的 $L_x(v)$ 最小值, 即, $L_x(\Omega)$ 。

[0112] 在每次递代中, 都在步骤 432 中计算或调用像素 x 和趋势 $v \in \Omega - D$ 的局部代价 $L_x(v)$ 并载入变量 L 。随后, 步骤 433 进行 L 是否小于 A^* 的评估。如果 $L < A^*$, 则在步骤 434 中更新上述变量 u 以包含趋势索引 v , 并将数值 L 分配给变量 A^* 。如果在步骤 433 中或在步骤 434 后有 $L \geq A^*$, 执行结束循环检测 435, 用于检测 $\Omega - D$ 中的所有趋势 v 是否都已扫描过。如果为否, 在步骤 436 中选择 $\Omega - D$ 中的另一个趋势 v , 并且处理流程返回至步骤 432, 进行循环 430 的另一次递代。

[0113] 当循环 430 结束时, 在像素 x 处最小化局部代价的 Ω 中的趋势 u 的边界 $m(u)$ 由增加数量 $A - A^*$ 进行更新 (步骤 441)。如果 $u \in D$, 步骤 441 不做任何变化。如果 $u \notin D$, 增加 u 至 D 将通过 $A - A^*$ 而减少考虑到像素 x 的代价函数, 同时 $\Omega - D$ 的其它趋势的边界不受影响。

[0114] 在循环 430 外进行更新步骤 441 使得复杂性的减少。该简化的缺点是对 $\Omega - D$ 的低于最佳优选趋势损失了一些精确性, 但是这对处理过程中的时间递归中不是严重问题,

其能表示实际关于视频序列的趋势。

[0115] 判断候选趋势 v 的不同的操作过程可以由模块 202 应用,且候选趋势 v 的边界通过模块 201 进行计算。

[0116] 在图 5 所示的简单实施例,模块 202 选择子集 $D = D_{\tau}$ 的趋势 v ,其具有由模块 201 计算的最低边界 $m(v)$,并且其因此为从 $D_{\tau'}$ 中移除的最佳的候选(步骤 501)。它还选择了具有最大边界 $m(w)$ 的 $\Omega - D$ 的趋势 w ,即加入 $D_{\tau'}$ 中的最佳候选(步骤 502)。如果 $m(w) > m(v)$ (检测 503),在步骤 504 执行替换: v 替换为在 D 中 w ,使得 $D_{\tau'} = D_{\tau} - \{v\} + \{w\}$ 。如果在步骤 503 中 $m(w) \leq m(v)$,则不替换: $D_{\tau'} = D_{\tau}$ 。

[0117] 图 6 示出另一中方法,其中模块 202 可以交换超过一对的趋势。在步骤 601 中,选择具有最低边界的子集 $D = D_{\tau}$ 中的 n 个趋势 v_1, v_2, \dots, v_n ,并且根据增加边界进行索引,即 $m(v_1) \leq m(v_2) \leq \dots \leq m(v_n)$ 。数值 n 可以为在 1 至 $|D|$ 之间的任意整数。在 $n = 1$ 的情况中,图 6 所示的处理过程与图 5 相同。在步骤 602 中,也同样选择具有最大边界的 $\Omega - D$ 的趋势 w_1, w_2, \dots, w_n ,同样被选择,并且根据减少的边界进行索引,即 $m(w_1) \geq m(w_2) \geq \dots \geq m(w_n)$ 。随后,确定可以交换的趋势对的数量。例如,在步骤 603 的初始化循环索引 i ($i = 1$) 后,模块 202 在检测 604 中比较边界 $m(w_i)$ 和 $m(v_i)$ 。如果 $\Omega - D$ 的趋势 w_i 优于 D 的趋势 v_i ,即 $m(w_i) > m(v_i)$,那么在步骤 605 中执行交换,在 D 中以 w_i 替换 v_i ,并且随后在检测 606 中比较 i 与 n 。如果 $i < n$,则不是所有的对都已检测,并且 i 在新的检测 604 中检测下一对之前,在步骤 607 中增一。当检测 604 示出对一些 $i < n$ 有 $m(w_i) \leq m(v_i)$,或当在检测 606 中 $i = n$ 时,处理过程终止。如果 n' 趋势对被交换 ($n' \leq n$),则更新趋势子集为 $D_{\tau'} = D_{\tau} - \{v_1, \dots, v_{n'}\} + \{w_1, \dots, w_{n'}\}$ 。

[0118] 在一个实施例中,当单元 102 检测规律趋势时,只使用具有大于给定阈值 T 的边界 $m(v)$ 的趋势 v 。一旦 $D_{\tau'}$ 已由块 202 确定时,其可以简单的完成,通过在趋势检测单元 102 中忽视 $D_{\tau'}$ 的趋势 v ,使得 $m(v) < T$ 。

[0119] 或者,当 $m(w)$ 小于阈值时,可以避免在 $D_{\tau'}$ 中包含 $\Omega - D_{\tau}$ 的新趋势 w 。有不同的方法达到这个目的。例如,如果使用图 6 的操作过程,将数值 n 设置为 $\{1, 2, \dots, |D|\}$ 中的最大的整数,使得对所有的索引都有 $m(w_i) > T$, i 使得 $1 \leq i \leq n$ 。

[0120] 使用阈值 T 有助于消减候选趋势集,并且有助于选择适用于视频几何复杂性的候选趋势的数量,即,以选择适用于视频趋势的离散集。

[0121] 图 7 和 8 示出了由视频处理单元 103 执行插值以及尤其是在输出和输入视频序列的帧频之间比率为 2 的帧频转换情况下实施本发明实施例所获得的结果。

[0122] 在该实施例中,视频序列是水平滚动的字幕文字“Sweet”。701 和 801 表示在时间 t 的图像,703 和 803 表示在时间 $t+1$ 的图像,以及 702 和 802 表示在时间 $\tau' = t+1/2$ 的合成图像,其中图 7 中有不匹配而图 8 有正确的插值。在图 701/801 和 703/803(时间 t 和 $t+1$) 之间,整个文本“Sweet”向左滚动了 10 个像素。有可能导致不匹配的是文本中多次包含具有周期为 8 个像素的字符“e”,并且趋势检测单元 102 将在时间 t 的第一个“e”误视为在时间 $t+1$ 的下一个输入图像中的另一个相似的“e”,导致图 702 所示的伪像。

[0123] 在图 7 和 8 所示实施例中,单元 101 所使用的代价函数居中,并且 Ω 只包括 $dt = 1/2$ 的趋势 $v = (dx, dt)$ 。那么,在位置 x 和时间 $\tau' = t+1/2$ 的趋势 $v = (dx, dt)$ 的代价为,例如,

[0124] $L_x(v) = |I_{\tau', -dt}(x-dx) - I_{\tau', +dt}(x+dx)| = |I_t(x-dx) - I_{t+1}(x+dx)|$, 或者优选为该代价与非负空间窗函数 g 进行卷积的窗计算版本。两个规律的趋势可以根据在该序列中的局部测量获得:

$$[0125] \quad v^{(1)} = (dx_1^{(1)}, dx_2^{(1)}, dt^{(1)}) = (-5, 0, \frac{1}{2}); \text{以及}$$

$$[0126] \quad v^{(2)} = (dx_1^{(2)}, dx_2^{(2)}, dt^{(2)}) = (-1, 0, \frac{1}{2})。$$

[0127] 一旦单元 102 检测到在时间 $\tau' = t+1/2$ 的像素 x 的趋势 $v = (dx, 1/2)$, 则在单元 103 中完成的帧频变换的插值可以包括计算 $\hat{I}_{\tau'}(x) = \hat{I}_{t+1/2}(x) = [I_t(x-dx) + I_{t+1}(x+dx)]/2$ 。

[0128] 在图 7 中, 我们假设没有使用稀疏几何, 使得在检测单元 102 中考虑在 Ω 中所有的趋势。对文本中的第一个和第三个“e”之间的一些像素, 检测到的趋势可能把在时间 t 的第一“e”与在时间 $t+1$ 的第二“e”归组 (参考图 7 中的方框), 以及把在时间 t 的第二“e”和在时间 $t+1$ 的第三“e”归组, 导致不正确的时间插值。标示 702 示出由该不正确插值导致具有伪像的不正确图像。一个包括对应于两个检测到的趋势 $v^{(1)}$ 、 $v^{(2)}$ 的混合插值数值的简单工作区同样不能解决该问题。

[0129] 在单元 101 中使用稀疏几何 $D_{\tau'}$ 有助于克服这样的问题。事实上, 如果子集 $D_{\tau'}$ 不包括趋势 $v^{(1)} = (-5, 0, \frac{1}{2})$, 对应于 $D_{\tau'}$ 的 $v^{(1)}$ 的边界将会很高, 因为只有 $v^{(1)}$ 用于字母“S” “w” 和 “t” 的滚动的计算。那么, $v^{(1)}$ 将在时间 τ' 进入 $D_{\tau'}$ 。完成了这个步骤, 因为 $v^{(1)}$ 为在包括所有“e”字符的视频的可能趋势, 那么 $v^{(2)} = (-1, 0, \frac{1}{2})$ 的边界将变得非常的低或甚至为零, 因为其中没有规律的可能趋势的视频范围, 而 $v^{(1)}$ 不是这样的。作为结果, 趋势 $v^{(2)}$ 将不计入集 $D_{\tau'}$ 中, 这样它将不在检测单元 102 中考虑, 或因为其边界低于阈值 T 而被忽略。正确的插值将如 802 所阐述的那样进行计算。

[0130] 值得注意的是, 时间插值可以在除了两原始帧之间半程之外的时间完成。例如, 50Hz 和 60Hz 的帧频标准之间转换的应用可以在时间 $\tau' = t+h/6$ 完成插值, 其中 h 是 1, 2, 3, 4 或 5 中的一个。在单元 101 和 102 中使用的损失函数然后可以进行相应的调整。

[0131] 图 9 至 12 与图 7 至 8 相似, 示出了本发明应用于超高分辨率视频去隔行扫描的实施例。

[0132] 图 9 至 10 示出了在设备输入中以隔行扫描视频格式滚动的相同文字“Sweet”。标示 901、1001、1101 和 1201 示出在时间 $t-1$ 的偶输入场, 标示 903、1003、1103 和 1203 示出了在时间 $t+1$ 的下一个偶输入场, 以及标示 902 和 1002 示出在时间 t 干涉奇输入场。去隔行扫描的目的是计算在时间 t 的偶数行, 以便合成在时间 $\tau' = t$ 包括奇数和偶数行的完整渐进帧 (progressive frame)。

[0133] 在图 9 至 12 所示实施例中, 在单元 101 中使用的代价函数居中, 并且 Ω 可以仅包括趋势 $v = (dx_1, dx_2, dt)$, 使得 $dt = 1$ 和 dx_2 为偶数的。此后, 在位置 $\xi = x = (x_1, x_2)$ 和时间 $\tau' = t$ 的 Ω 的趋势 $v = (dx, dt)$ 为例如 $L_x(v) = |I_{t-dt}(x-dx) - I_{t+dt}(x+dx)|$ 或该代价窗计算的版本。几个规律趋势可以在该序列以前获得, 包括 $v^{(1)} = (dx_1^{(1)}, dx_2^{(1)}, dt^{(1)}) = (-5, 0, 1)$

和 $v^{(2)} = (dx_1^{(2)}, dx_2^{(2)}, dt^{(2)}) = (-1, 0, 1)$ 。

[0134] 一旦单元 102 检测到时间 $\tau' = t$ 时的像素 x 趋势 $v = (dx, 1)$, 在处理单元 103 为去隔行扫描实施插值, 可能包括计算 $\hat{I}_{\tau'}(\xi) = \hat{I}_t(x) = [I_{t-1}(x-dx) + I_{t+1}(x+dx)]/2$ 。

[0135] 在图 11 中, 我们再次假设选择单元 101 将 Ω 的所有趋势提供给检测单元 102, 且不使用稀疏几何。检测单元 102 就不能正确区分趋势 $v^{(1)} = (-5, 0, 1)$ 和 $v^{(2)} = (-1, 0, 1)$, 并且输出只能再次显示如 1102 中所示的错位类型的伪像。

[0136] 图 12 示出了较好的去隔行扫描的结果, 只有当趋势 $v^{(1)} = (-5, 0, 1)$ 由选择单元 101 保留在稀疏几何中时, 而多余的趋势 $v^{(2)} = (-1, 0, 1)$ 在分析的选择步骤中去除。

[0137] 此外, 在一个去隔行扫描的应用中, 当计算时间 $\tau' = t$ 的像素时, 可以使用 Ω 的趋势中 $dt = 2$ 的数值, 在 $t-2$ 和 $t+2$ 之间计算趋势, 以用于计算具有更高清晰度的趋势。这样意味着在插值中可以相同的方法使用趋势 $v = (dx, 1)$ 和 $2v = (2dx, 2)$ 。因为隔行扫描源的奇偶限制, 可以计算对应的损失函数 $|I_{t-2}(x-2dx) - I_{t+2}(x+2dx)|$ 。如果由单元 102 检测到趋势 $2v = (2dx, 2dt) = (2dx_1, 2dx_2, 2dt)$, 对半趋势 v 而言纵坐标 dx_2 可以为奇数。这样允许适当去隔行扫描的视频序列包括半像素垂直速度。如果这样趋势表示式涉及趋势选择和检测单元 101 和 102, 则处理单元 103 可以插值 $\hat{I}_{\tau'}(\xi)$ 为:

[0138] $\hat{I}_{\tau'}(\xi) = \hat{I}_t(x) = [I_{t-2}(x-2dx) + I_{t+2}(x+2dx)]/2$

[0139] 使用的趋势检测可以包括 $dt = 1$ 或 $dt = 2$ 的时间步骤。其对应于比较各种不同的趋势以及不同的时间偏移 (1 或 2, 或者甚至更多)。

[0140] 去隔行扫描应用中的另一可能性是计算趋势的代价, 其中除了场可以均匀间隔时间拍摄之外, 场也可以非规则间隔时间拍摄。例如, 视频内容原始源是使用“电影电视 (telecine)”将电影转化为视频的情况。例如, 在欧洲使用的 2:2 电影电视, 当 25fps (帧每秒) 的电影转换为 50fps 的视频, 各个电影帧用于产生两个视频场, 因此视频源内容的场 I_0, I_1, I_2, I_3 是分别在时间 $0s, 0s, 2/50s, 2/50s$ 拍摄, 而不是在时间 $0/50s, 1/50s, 2/50s$ 和 $3/50s$ 拍摄的。此外, 视频信号可以包括电影源内容和视频源内容的组合, 使得检测必须是以像素来进行的。可以选择特定的局部代价函数, 以检测是否为确定像素, 视频是否为电影源, 以及该场是否在相同的电影帧的之前或之后产生。那么, 各个像素的趋势的设置是下述之一:

[0141] (电影前) (film-before)

[0142] (电影后) (film-after)

[0143] (视频, v) (video, v)

[0144] 其中“电影-前”表示在给定像素位置, 内容为电影源, 并且先前场来自于相同电影帧, 使得遗失的像素可以从先前场中的相同位置中获得, 其中“电影-后”表示在给定像素位置, 内容是电影源, 并且其后场来自于相同的电影帧, 以及其中 (视频, v) 表示在当前像素位置, 内容是视频源, 并且趋势向量为 v 。这样的说明阐述了可以由局部表示符号将“趋势”定义比单一 3D 向量 v 更为复杂的另一种情况。在这样情况中, “趋势”为 (电影-前)、(电影-后)、(视频, v) 之一的表示符号, 其中 v 为向量。

[0145] 在超高分辨率视频降噪的情况中, 图 1 中的处理单元 103 通过使用在 ξ 的定向平均函数 K_v 为每个目标像素 ξ 、 τ 计算它的新数值, 通过公式:

$$[0146] \quad \hat{I}_\tau(\xi) = \sum_{x,t} K_v(\xi-x, \tau-t) \cdot I_t(x)$$

[0147] 其中,在 (ξ, τ) 的周边区域中输入图像的所有像素 (x, t) 范围求和,如果输入网格的一些点 (x, t) 有 $\xi = x, \tau = t$ 并且 K_v 取决于局部趋势 $v = (dx, dt)$, 则包括像素 (ξ, τ) 本身。在一个典型实施例中,平均函数 K_v 为沿着趋势 $v = (dx, dt)$ 的定向平均函数。该函数的示例为:

$$[0148] \quad K_v(x, t) = K_1(t) \times K_2(x-t, dx/dt)$$

[0149] 其中 K_1 和 K_2 为 1D 和 2D 的平均核函数,例如高斯核函数。

[0150] 在另一个实施例中,在处理单元 103 中执行的视频处理从趋势检测单元 102 处接收不同数量趋势。这些趋势中的每一个都可以对应于相关的检测。在趋势数量为 0 的情况下,回退 (fallback) 插入函数或平均函数可以被使用。在趋势数量大于 1 的情况下,目标像素数量可以通过结合像素数值进行计算,该像素值通过对每个趋势的插值或平均函数计算。这种组合可以是平均的,使用相关测量的权重平均的,或中值的,或权重中值的,或用于结合这些像素数值的其它的方法。

[0151] 在另一个示例实施例中,沿着趋势 $v = (dx, dt)$ 处理的降噪可以是任何类型的已知定向滤波,包括无限脉冲响应 (IIR) 滤波。

[0152] 在另一个示例实施例中,当处理的信号是视频信号时,稀疏几何用于改善在文献 WO 2007/059795 A1 阐述类型的处理。因而,趋势 (dx, dt) 可以被限制为 $dt = 1$ 的数值以及 dx 的整数数值。它们可以用于建立在时间 t 的帧的像素与帧 $t+1$ 的像素之间的映像: $(x, t) \mapsto (x + dx, t + 1)$, 并且提供了适用于文献 WO 2007/059795 A1 中使用的第一归组预测的实施例。

[0153] 在趋势选择器 101 的实施例中,候选趋势的子集 Ω 被划分为多个子集 $\Omega_1, \dots, \Omega_J (J > 1)$, 并且子集 $\Omega_1, \dots, \Omega_J$ 中只有一个在每个时间 τ' 中由趋势选择单元 101 考虑,以提供给纳入已选趋势 $D \tau'$ 的候选子集。当子集 Ω 对于在每个周期 τ' 中候选进行完整扫描太大时,将会非常有趣。例如,在子集 Ω_j 被考虑时 ($1 \leq j \leq J$), 在图 3 中所示的循环 330 或在图 4 中所示的 430 为在 Ω_j 中的趋势 v 执行而不是在 D 中。

[0154] 在某些情况中,除了对整个图像区域的整体子集 $D \tau'$ 的选择以外,把图像支持 (image support) 拆分至不同的像素窗 $W_{p,q}$ 可能变得有趣,例如定义为矩形区域:

$$[0155] \quad W_{p,q} = \{(x_1, x_2) : w \times (p-1) < x_1 \leq w \times p \text{ and } h \times (q-1) < x_2 \leq h \times q\}$$

[0156] 式中 h 和 w 分别为这些窗的高和宽 (在像素中表示), 并且窗索引 p, q 分别在 $1 \leq p \leq P, 1 \leq q \leq Q$ 的范围中。窗的总数为 $P \times Q$ 。当 $P = Q = 1$ 时,仅有一个如上述包括整个图像的窗。对在每个窗 $W_{p,q}$ 内的各个趋势 v , 边界 $m_{p,q}(v|D)$ 可以使用与 (4) 相似的方程式进行计算,但具有限制该窗 $W_{p,q}$ 的交换图像区域的总和:

$$[0157] \quad m_{p,q}(v|D) = \sum_{x \in W_{p,q}} m_x(v|D) \quad (10)$$

[0158] 趋势的局部子集 $D_{\tau',p,q} \subset D_{\tau'}$ 可以使用这些边界进行计算。因此,候选趋势的第三子集 $D_{\tau',p,q}$ 定义为 I_{t+1} 的整个区域确定的第二子集 $D_{\tau'}$ 的子集,根据在输入图像 I_t 和 I_{t+1} 中为窗 $W_{p,q}$ 的像素计算的代价边界 $m_{p,q}(v|D)$ 。当趋势检测单元 102 在一个窗 $W_{p,q}$ 内的像素 $\xi = x$ 上检测趋势时,则考虑从 $D_{\tau',p,q}$ 中的唯一候选趋势。这有助于增加用于避

免不良的趋势的检测的鲁棒性。再参考图 7 至 12 所阐述的实施例,选择允许去除不良趋势 $(-1, 0, 1/2)$ [或 $(-2, 0, 1)$], 并且只使用正确的趋势 $(-5, 0, 1/2)$ [或者 $(-10, 0, 1)$]。如果视景更为复杂并且在图片其余部分的对象正好表现出规律的趋势 $(-1, 0, 1/2)$, 这个向量 $(-1, 0, 1/2)$ 将在 $D_{\tau'}$ 中表示, 并且在单元 101 中获得选择的趋势可能在适当处理滚动文字中丧失。如果选择边界在更小的窗 $W_{p, q}$ 中重新计算, 窗 $W_{p, q}$ 同时包括滚动文本和具有单一规律趋势 $(-1, 0, 1/2)$ 的对象的可能性会更低。

[0159] 当使用过于小的窗 $W_{p, q}$ 时 (例如在图 7 至 12 中在只包括有一个或两个“e”的区域的情况中), 选择将由于过小的窗而变得困难, 其再不可能在两个不同规律的趋势之间进行判断。一个多尺度选择方案可以用于避免这个问题, 通过递归拆分图像支持至窗, 以及各个窗至子窗。对各个窗, 趋势子集选定为母区域 (整个图像或更高层级的窗) 选择的趋势子集的子集。在多尺度选择的方案中, 窗 $W_{p, q}$ 的一个或多个进一步被拆分至多个子窗 $W_{p, q, r, s}$, 并且对每个子窗而言, 候选趋势的第四子集 $D_{\tau', p, q, r, s}$ 定义为窗 $W_{p, q}$ 所确定的第三子集 $D_{\tau', p, q}$ 的子集, 并根据在输入图像 I_t 和 I_{t+1} 中的子窗 $W_{p, q, r, s}$ 的像素计算的代价边界 $m_{p, q, r, s}(v|D)$ 为:

$$[0160] \quad m_{p, q, r, s}(v|D) = \sum_{x \in W_{p, q, r, s}} m_x(v|D) \quad (11)$$

[0161] 适用于输出图像 $\hat{I}_{\tau'}$ 的子窗 $W_{p, q, r, s}$ 的像素的规律趋势在子集 $D_{\tau', p, q, r, s}$ 中进行检测, 可能在拆分窗的一次或多次递归循环之后。

[0162] 在一些实施例中, 已选趋势的子集 $D_{\tau'}$ 可进行限制, 以满足不同安全标准。例如:

[0163] - 一些具体的趋势 (典型的例如 $(0, 0, 1)$) 可以限定在 $D_{\tau'}$ 中, 而不管有关这些趋势的边界;

[0164] - 趋势集 Ω 还可以拆分成 R 个组 $\Omega^{(1)}, \dots, \Omega^{(R)}$, 并且可对每个组 $\Omega^{(r)}$ ($1 \leq r \leq R$, $R > 1$) 进行限制, 只有一个或有限数量的趋势选择, 以列入在子集 $D_{\tau'}$ 中。

[0165] 上述实施例可以通过由通用的微处理器或者数字信号处理器执行的软件来执行, 在这样的情况中, 涉及图 1 至 6 的所示上述模块可以被认为理解为是或是组成软件模块或例行程序或其中部分。其还可以采用如图 13 所示的硬件组件来执行, 在视频插值模块 1303 之前和 / 或之后, 除了其它的视频处理模块 1302、1304 以外, 例如为视频流插值的采用专用集成电路 ASIC, 或现场可编程门阵列 (FPGA) 用于插值视频流。此外, 视频处理模块 1303 可以执行如上述的降噪方法。在一个示例性实施例中, 视频处理块 1302、1303、1034 可由单一芯片 1301 中执行。该芯片还具有视频输入和输出接口, 以及外部 RAM (随机访问存储器) 装置 1305 和 1306, 以作为在 1302、1303 和 1304 中执行不同的视频处理步骤所需的暂时存储。该实施例的其它变化例可同样作为本发明的一部分, 采用更完善的视频处理芯片或甚至包括其它功能的片上系统芯片。而且, 硬件器件可以合并在不同类型的视频装置中。

[0166] 在上文详细阐述本发明示例性实施例的同时, 各种不同的变化实施例、改进实施例以及相似实施例都可以被本领域的技术人员所悉知。此外, 上文的阐述不用于限制由本发明的权利要求所定义的本发明的目的。

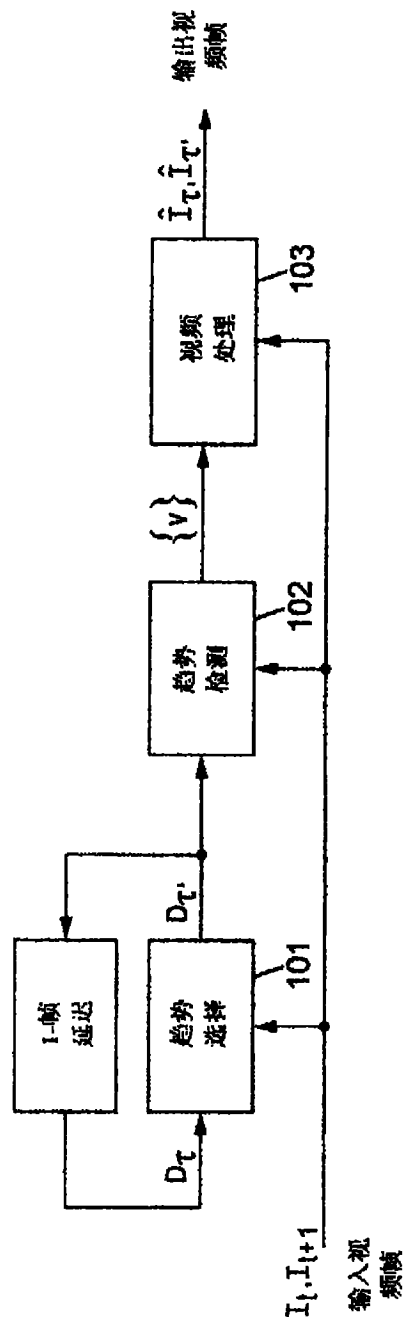


FIG. 1

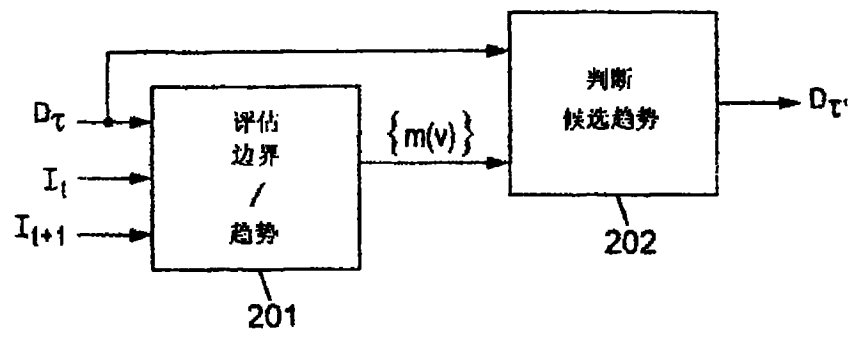


FIG. 2

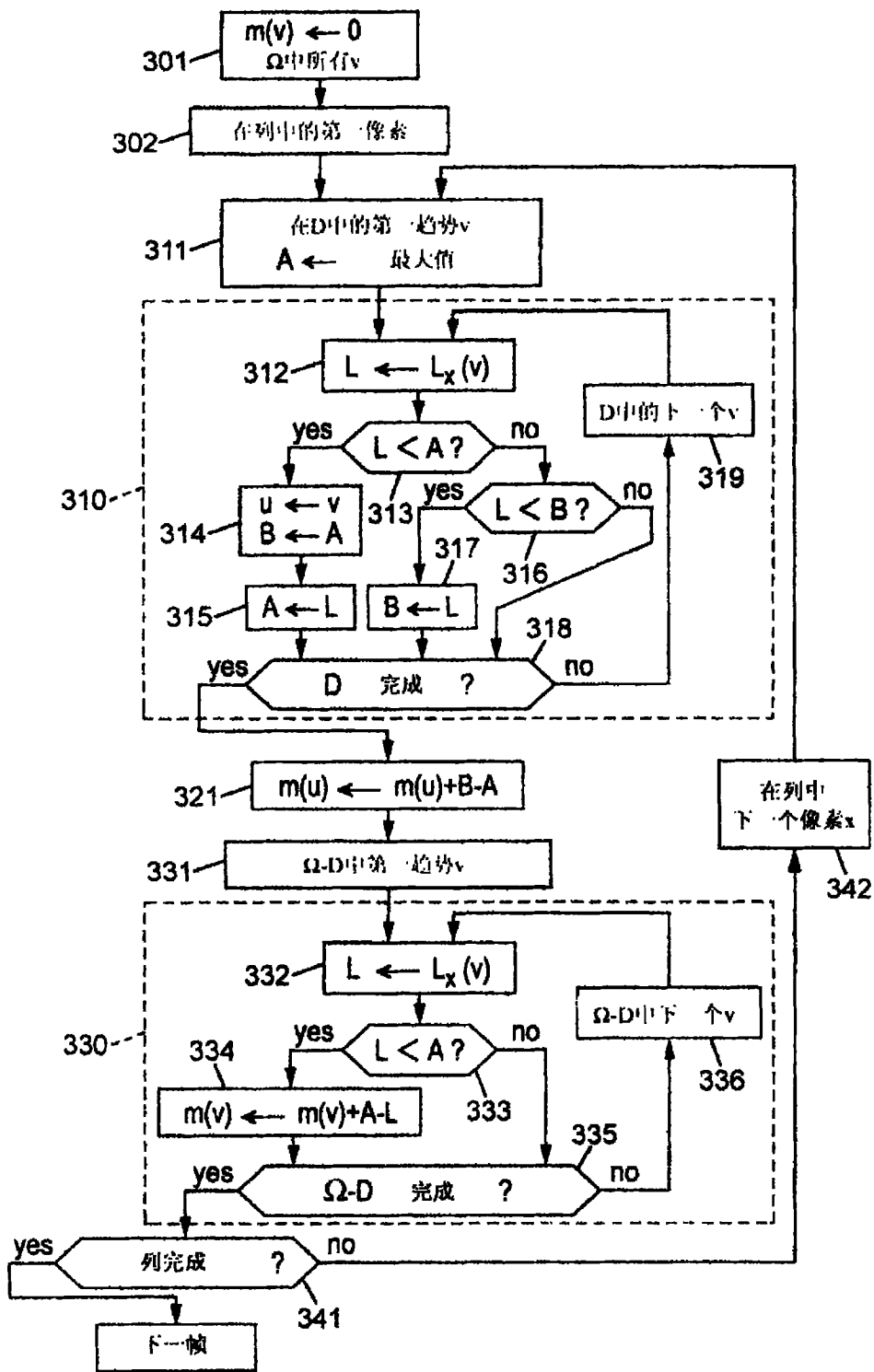


FIG. 3

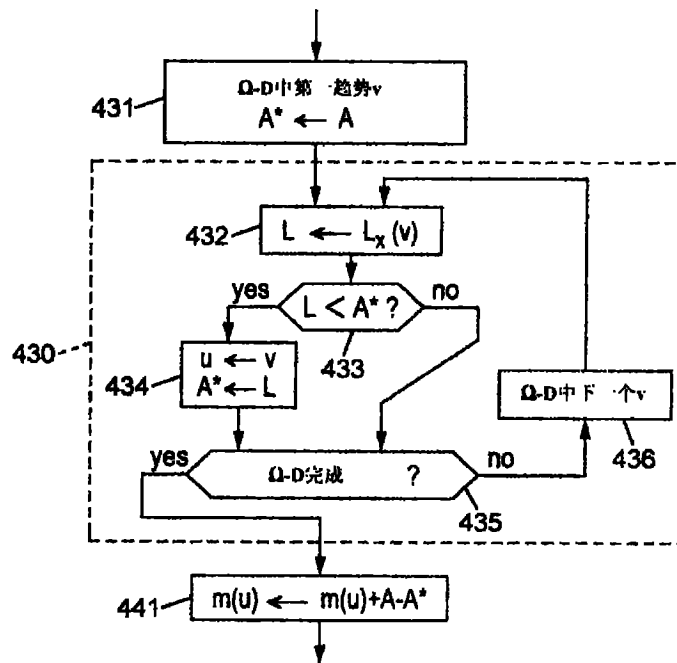


FIG. 4

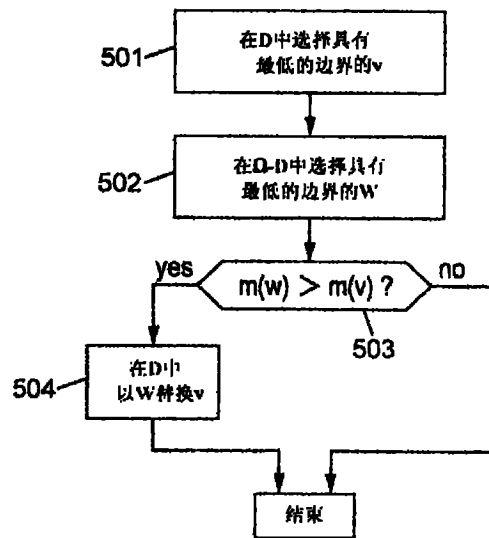


FIG. 5

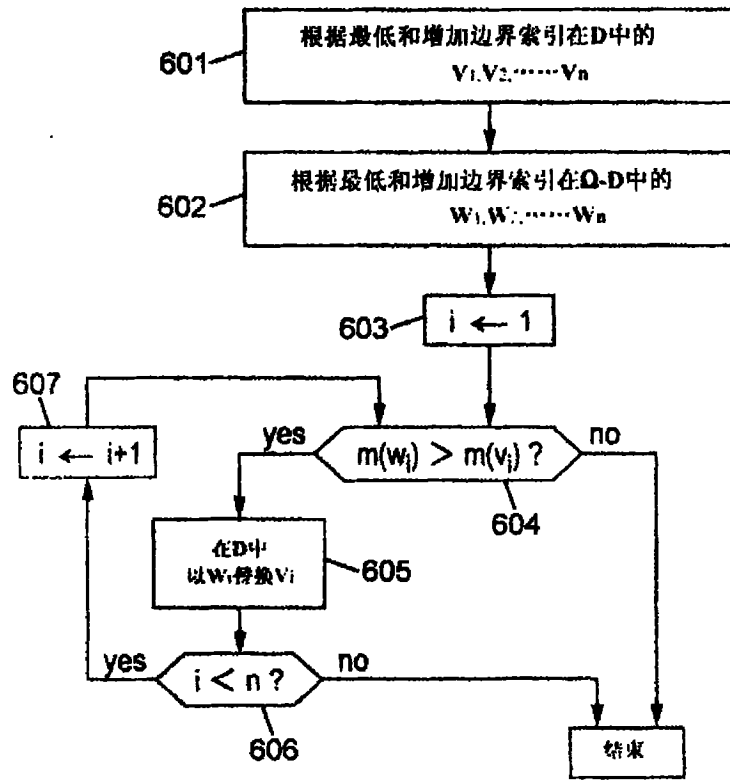


FIG. 6

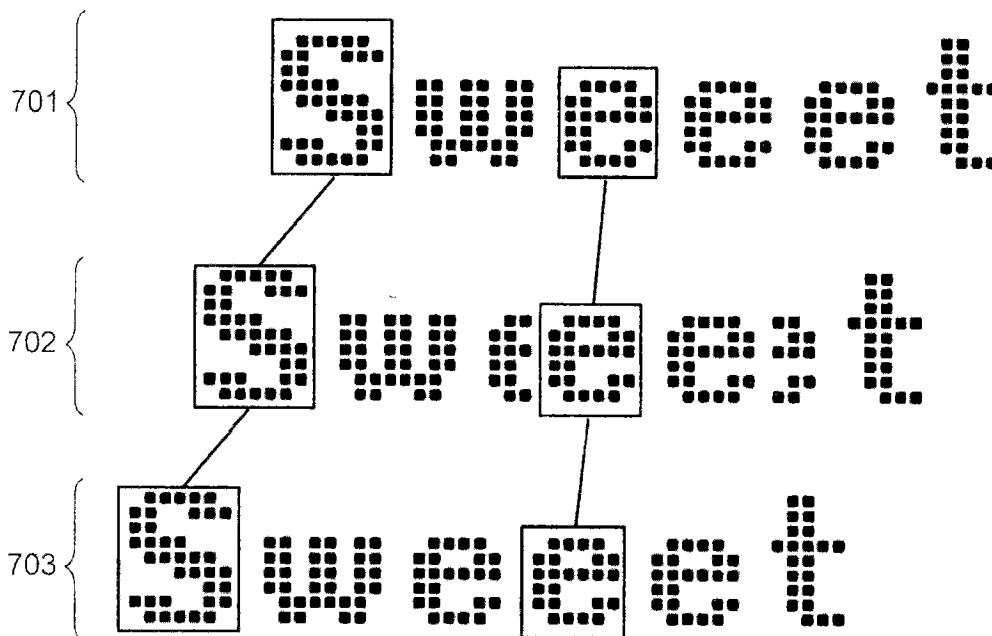


FIG. 7

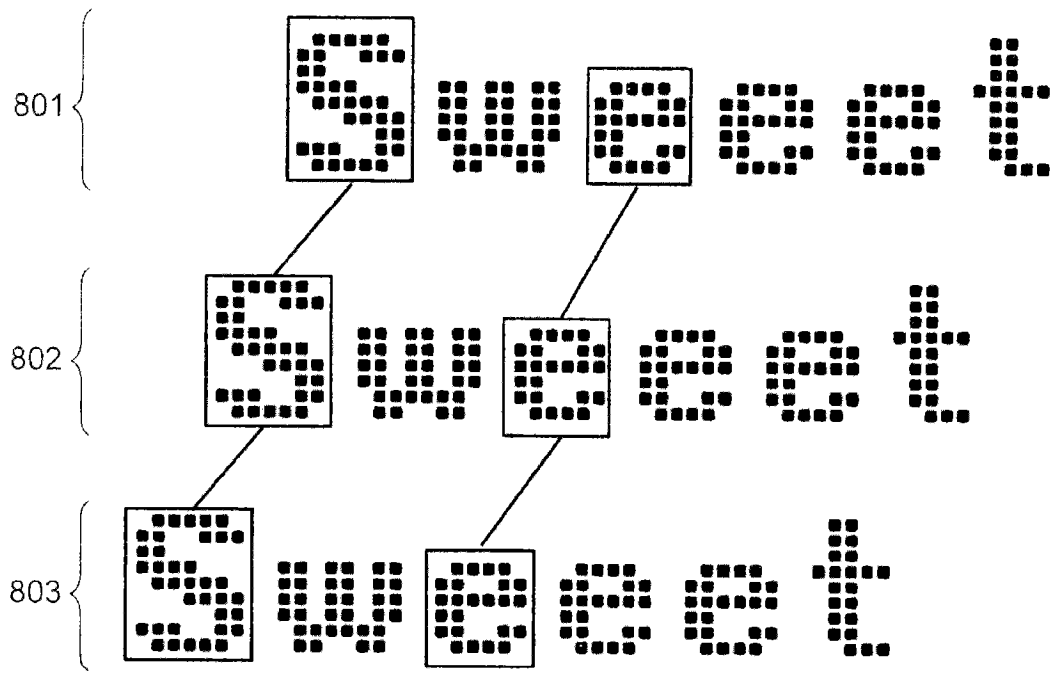


FIG. 8

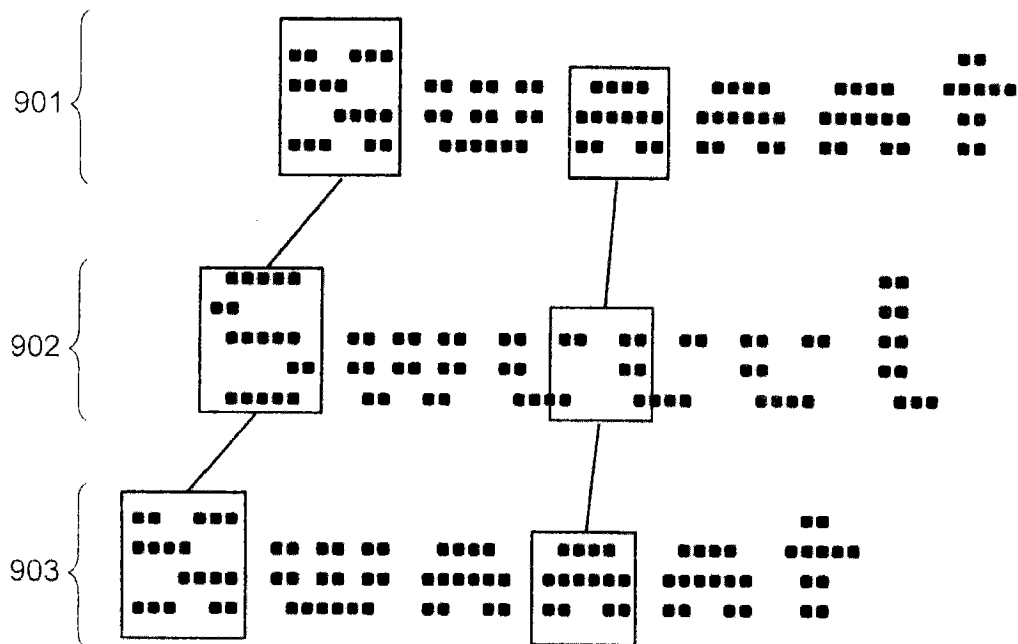


FIG. 9

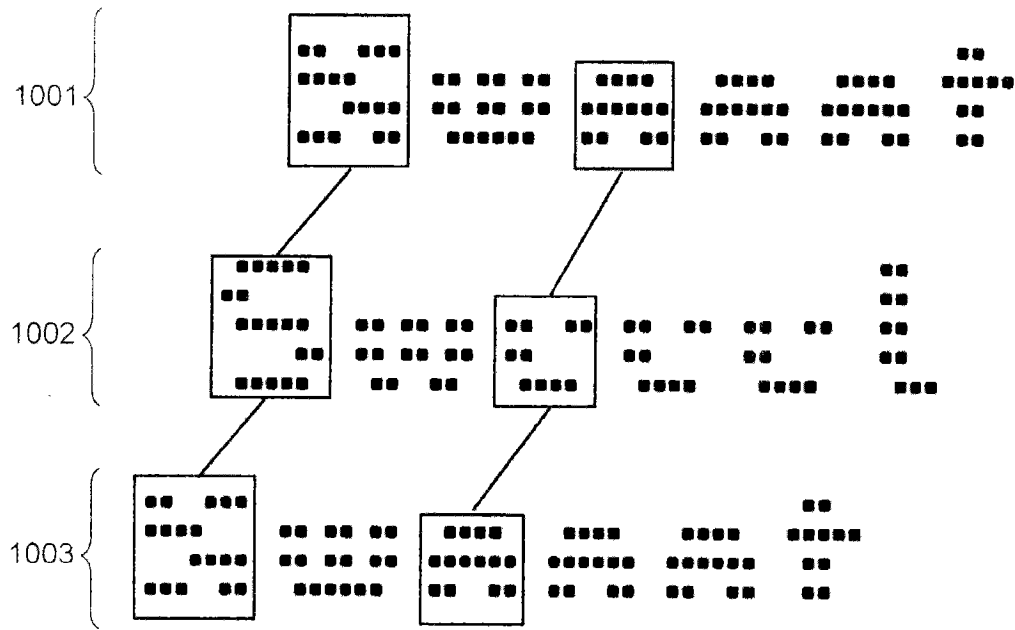


FIG. 10

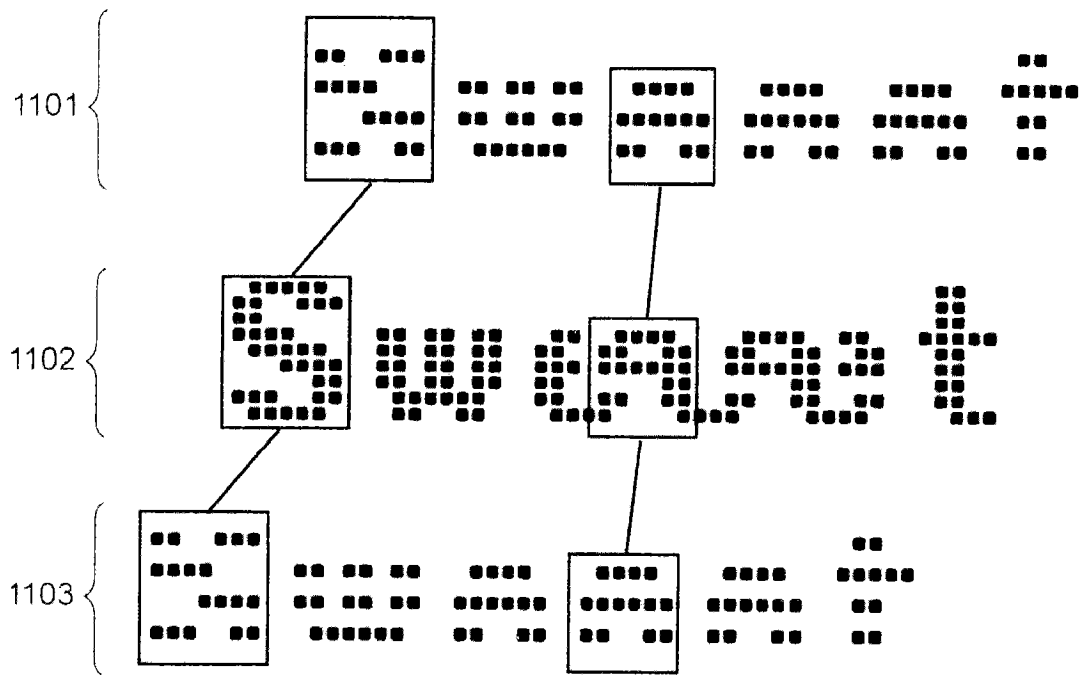


FIG. 11

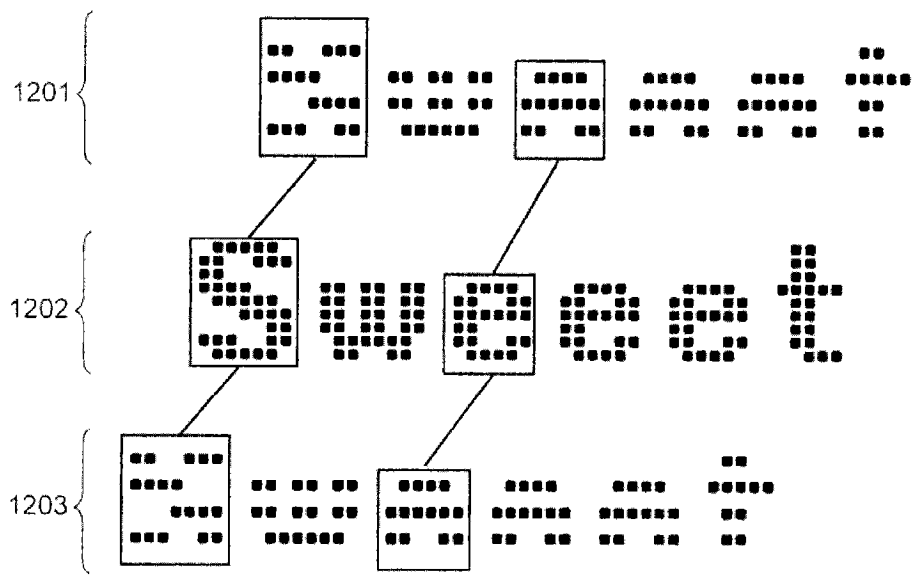


FIG. 12

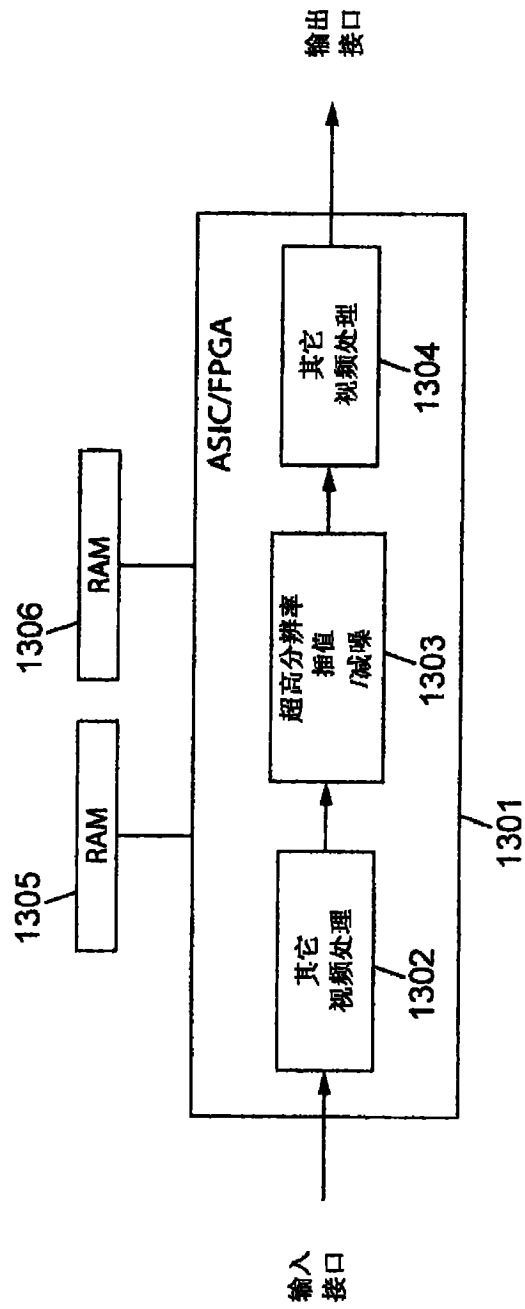


FIG. 13