

(12) 发明专利申请

(10) 申请公布号 CN 102256065 A

(43) 申请公布日 2011. 11. 23

(21) 申请号 201110208090. X

(22) 申请日 2011. 07. 25

(71) 申请人 中国科学院自动化研究所

地址 100190 北京市海淀区中关村东路 95 号

(72) 发明人 卢汉清 刘静 王金桥 祝晓斌

(74) 专利代理机构 中科专利商标代理有限责任公司 11021

代理人 梁爱荣

(51) Int. Cl.

H04N 5/265(2006. 01)

H04N 7/18(2006. 01)

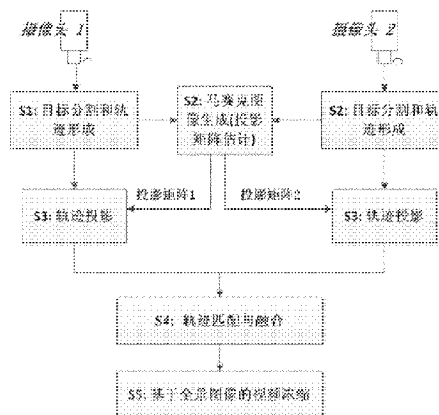
权利要求书 1 页 说明书 8 页 附图 2 页

(54) 发明名称

基于视频监控网络的视频自动浓缩方法

(57) 摘要

一种基于视频监控网络的视频自动浓缩方法,其是从两个具有重叠区域的摄像机获得第一视频源和第二视频源并对运动目标分割和跟踪,提取背景图像和视频监控网络运动轨迹;根据背景图像生成基于视频监控网络的全景图像,得到两个摄像机各自对应于全景图像的投影矩阵并实时更新基于视频监控网络背景的全景图像;利用两个摄像机全景图像的投影矩阵,把轨迹投影到全景图像空间得到投影轨迹;采用图匹配,基于随机游走思想对两个摄像机的视频源的投影轨迹匹配,得到匹配轨迹对,对重叠区域的匹配轨迹进行选择 and 融合,得到完整的大场景轨迹;对大场景轨迹按时间重新排列,在全景图像上对完整的大场景轨迹进行展现,得到基于全景图像的视频浓缩。



1. 一种基于视频监控网络的视频自动浓缩方法,其特征在于包括步骤如下:

步骤 S1:从两个具有重叠区域的摄像机分别获得第一视频源和第二视频源,并对运动目标进行分割和跟踪,提取背景图像、提取视频监控网络运动轨迹;

步骤 S2:根据提取的背景图像生成基于视频监控网络的全景图像,得到两个摄像机各自对应于全景图像的投影矩阵,并随时间推移实时更新基于视频监控网络背景的全景图像;

步骤 S3:利用两个摄像机全景图像的投影矩阵,把轨迹投影到全景图像空间,得到投影轨迹;

步骤 S4:采用图匹配的方法,基于随机游走思想,对两个摄像机的的视频源的投影轨迹进行匹配,得到匹配轨迹对,对重叠区域的匹配轨迹进行选择 and 融合,得到完整的大场景轨迹;

步骤 S5:对大场景轨迹按时间重新排列,然后在全景图像上对完整的大场景轨迹进行展现,得到基于全景图像的视频浓缩。

2. 根据权利要求 1 所述的基于视频监控网络的视频自动浓缩方法,其特征在于,对视频监控网络运动轨迹进行匹配的步骤包括:把视频监控网络的每条轨迹看作一个点,每个摄像机代表一个图,然后把两个相邻摄像机之间的运动轨迹匹配问题当作求两个图像点集之间的最大匹配问题来解决;把两个图像的点与点的关系度量,转换成点对之间的图像度量,并把两个图像转换为一个点对图,在点对图上采用快速加权随机游走的思想,当随机游走收敛到稳定状态,从稳定状态中根据图像点的得分,求得最大匹配集。

3. 根据权利要求 1 所述的基于视频监控网络的视频自动浓缩方法,其特征在于,还包括视频监控网络运动轨迹的特征表述是采用多个特征同时来表述运动轨迹,所述多个特征是 32 维梯度方向直方图、RGB3 种颜色分量的直方图、运动轨迹的平均加速度、运动轨迹的平均速度和轨迹长度。

4. 根据权利要求 1 所述的基于视频监控网络的视频自动浓缩方法,其特征在于,所述重叠区域匹配轨迹的选择与融合是采用自适应权重的方法对长度较长和观测点较多的轨迹,采用更大的权重。另外设定阈值,如果权重低于阈值,则舍弃该的轨迹,如果等于或大于阈值,则选择该轨迹进行融合,最后获得两条轨迹的像素值乘以权重后叠加轨迹。

5. 根据权利要求 1 所述的基于视频监控网络的视频自动浓缩方法,其特征在于,基于全景图像的视频浓缩是用能量损耗函数来表示视频浓缩中,对监控网络中已经匹配和融合的大场景轨迹重新排列带来的能量损耗,获得能量损耗后,用优化的方法求出一个轨迹重新排列的最优解,使能量损耗函数最小,最后把重新排列的轨迹,在全景图上叠加展现。

6. 根据权利要求 1 所述的基于视频监控网络的视频自动浓缩方法,其特征在于,所述能量损耗包括轨迹丢失运动损耗、轨迹交叉遮挡损耗、轨迹时间错乱损耗、重叠区域轨迹选择损耗和背景更新损耗。

基于视频监控网络的视频自动浓缩方法

技术领域

[0001] 本发明属于图像处理技术领域,具体涉及基于视频监控网络的视频自动浓缩方法。

背景技术

[0002] 随着视频监控技术的迅速发展,单摄像机的视频监控设备越来越普及,视频监控设备每天都记录了海量的视频数据。已有的视频监控技术主要针对单个摄像机,对监控场景中的感兴趣目标进行自动检测、跟踪和分类,以及相应的行为分析。每路摄像机有各自的监控任务,监控系统中摄像机和摄像机的监控任务之间没有关联。然而单个摄像机的视野范围有限,为了满足大范围监控的需要,就需要多个摄像机协同工作。随着摄像机数目的急剧增加,传统的基于人工的被动监控已经无法满足监控任务的需要。如何自动分析和归纳多个摄像机所拍摄的视频中的内容,真正意义上实现大范围监控仍然处在研究的初期阶段。

[0003] 国内外有很多大学和研究机构在进行视频摘要研究,卡内基梅隆、哥伦比亚大学、IBM 研究院等,国内如微软亚洲研究院、清华大学等研究机构都曾在进行该方面研究。视频摘要技术从不同的角度可以进行不同的分类:传统的分类将视频摘要技术分为视频略览和视频梗概两大类。传统的视频摘要技术引入了用户定义的感兴趣的帧做为关键帧的概念,可以分为动态(video browsing)和静态(video abstraction)两种技术。基于关键帧的视频总结技术以“帧”作为不可再分的最小表示单元,总结视频中不可避免存在大量的冗余信息。基于目标运动信息的视频总结技术于2006年提出,最后的输出视频不再是由输入视频中的完整帧组成,来自不同帧的运动目标可能被放入同一帧中输出。微软亚洲研究院提出视频蒙太奇(Video Montage)技术,同时分析时间和空间上的信息分布,抽取时空上的运动信息用立体块表示,进而运用首次拟合以及图割方法完成立体块的压缩任务,产生摘要视频。以色列希伯来大学的 A. Rav-Acha 等人系统地提出了一种基于目标运动信息的视频总结技术框架(video synopsis)。在此工作的基础上,该研究组又进一步深入研究了在线监控情况下的视频摘要技术。

发明内容

[0004] 为了解决如何在大范围内对运动目标进行分析和监控的技术问题,本发明的目的是提出面向监控网络的自动视频浓缩方法。

[0005] 为实现上述目的,本发明提供基于视频监控网络的视频自动浓缩方法包括步骤如下:

[0006] 步骤 S1:从两个具有重叠区域的摄像机分别获得第一视频源和第二视频源,并对运动目标进行分割和跟踪,提取背景图像、提取视频监控网络运动轨迹;

[0007] 步骤 S2:根据提取的背景图像生成基于视频监控网络的全景图像,得到两个摄像机各自对应于全景图像的投影矩阵,并随时间推移实时更新基于视频监控网络背景的全景

图像；

[0008] 步骤 S3 :利用两个摄像机全景图像的投影矩阵,把轨迹投影到全景图像空间,得到投影轨迹；

[0009] 步骤 S4 :采用图匹配的方法,基于随机游走思想,对两个摄像机的的视频源的投影轨迹进行匹配,得到匹配轨迹对,对重叠区域的匹配轨迹进行选择 and 融合,得到完整的大场景轨迹；

[0010] 步骤 S5 :对大场景轨迹按时间重新排列,然后在全景图像上对完整的大场景轨迹进行展现,得到基于全景图像的视频浓缩。

[0011] 优选实施例,对视频监控网络运动轨迹进行匹配的步骤包括:把视频监控网络的每条轨迹看作一个点,每个摄像机代表一个图,然后把两个相邻摄像机之间的运动轨迹匹配问题当作求两个图像点集之间的最大匹配问题来解决;把两个图像的点与点的关系度量,转换成点对之间的图像度量,并把两个图像转换为一个点对图,在点对图上采用快速加权随机游走的思想,当随机游走收敛到稳定状态,从稳定状态中根据图像点的得分,求得最大匹配集。

[0012] 优选实施例,还包括视频监控网络运动轨迹的特征表述:采用多个特征同时来表述运动轨迹,所述多个特征是 32 维梯度方向直方图 (32D HOG feature)、RGB3 种颜色分量的直方图、运动轨迹的平均加速度、运动轨迹的平均速度和轨迹长度。

[0013] 优选实施例,所述重叠区域匹配轨迹的选择与融合是采用自适应权重的方法对长度较长和观测点较多的轨迹,采用更大的权重。另外设定阈值,如果权重低于阈值,则舍弃该的轨迹,如果等于或大于阈值,则选择该轨迹进行融合,最后获得两条轨迹的像素值乘以权重后叠加轨迹。

[0014] 优选实施例,基于全景图像的视频浓缩是用能量损耗函数来表示视频浓缩中,对监控网络中已经匹配和融合的大场景轨迹重新排列带来的能量损耗,获得能量损耗后,用优化的方法求出一个轨迹重新排列的最优解,使能量损耗函数最小,最后把重新排列的轨迹,在全景图上叠加展现。

[0015] 优选实施例,所述能量损耗包括轨迹丢失运动损耗、轨迹交叉遮挡损耗、轨迹时间错乱损耗、重叠区域轨迹选择损耗和背景更新损耗。

[0016] 本发明的有益效果:本发明基于图匹配方法,采用加权随机游走思想,解决了具有重叠区域的轨迹匹配问题。对每个视频监控单独提取背景,然后根据摄像机的拓扑结构,用全景图生成的算法,生成一幅视频监控网络背景的全景图,由于时间的变化,视频背景会变化,采用基于根据能量损耗在线更新的算法,对背景图像进行更新。另外本发明用单独视频的背景来生成全景图像,解决了视频网络全局运动展现问题。另外本发明用能量函数优化的思想,对视频网络的轨迹在时间和空间上重新排列,极大的压缩了视频长度,同单摄像机视频监控视频摘要相比,本发明有如下优点:多角度信息的提供,视觉范围的增大。与传统的视频摘要方法相比,我们的方法可以极大的压缩视频长度,提供大场景的运动目标整体行为的分析和监控。视频监控网络运动轨迹的特征表述,用多个特征同时来表述运动轨迹,增强了轨迹匹配的准确度和提高视频浓缩的效果。

附图说明

- [0017] 图 1 是本发明的结构图；
[0018] 图 2 是本发明的点对图构成示例图；
[0019] 图 3 是本发明的加权随机游走示例图；
[0020] 图 4 是本发明的视频自动浓缩技术示例图；

具体实施方式

[0021] 为使本发明的目的、技术方案和优点更加清楚明白，以下结合具体实施例，并参照附图，对本发明进一步详细说明。

[0022] 本发明的结构图在图 1 中显示，主要包括五个部分：步骤 S1. 运动目标的分割与轨迹形成；步骤 S2. 视频监控网络全景图像的生成和更新；步骤 S3. 轨迹投影；步骤 S4. 重叠区域运动轨迹的匹配和融合；步骤 S5. 基于视频监控网络全景图像的视频浓缩。

[0023] 步骤 S1：运动目标的分割与轨迹形成

[0024] 从两个摄像机分别获得第一视频源 1 和第二视频源 2，我们先对第一视频源 1 和第二视频源 2 用混合高斯模型进行背景建模，背景会随着时间动态更新，提取出更新的背景，然后用更新的背景减除法，提取视频的前景。在提取前景后，为了提取有用的轨迹，我们用中值滤波算法，滤除噪点。每个前景用一个外接矩形 (bound-box) 和掩码表示，外接矩形 (bound-box) 为前景区域经过 3 次膨胀后的外接矩形，掩码代表矩形内前景部分。我们对相邻两帧之间的前景，提取其速度，位置，大小信息，然后用在线聚类算法，获取完整的视频运动轨迹。

[0025] 步骤 S2：视频监控网络全景图像的生成和更新

[0026] 根据提取的背景图像生成基于视频监控网络的全景图像，得到两个摄像机各自对应于全景图像的投影矩阵，并随时间推移实时更新基于视频监控网络背景的全景图像；在运动目标的分割与轨迹形成中，我们得到背景图像，我们提取图像的尺度不变 (SIFT) 特征，再树方法 (K-d 树) 对每个特征提取 N 个最近邻。针对每个图像之间的特征匹配，我们用随机抽样一致性算法 (RANSAC) 去除噪点，最后用概率模型去验证图像间的匹配关系。在得到图像之间对应的匹配点后，我们用光束法 (bundle adjustment) 得到两个图像之间的投影矩阵 (Homography)，然后用融合的办法得到全景图。投影矩阵 (Homography)，在视频浓缩时，用来对轨迹进行空间转换使用。具体的算法流程如下：

[0027]

输入：两幅背景图像

- 1) 提取两幅图像的 SIFT 特征点;
- 2) 对每个特征点, 采用 K-D 树方法找到它的 N 个最近邻特征;
- 3) 对每幅图像, 我们进行如下操作:

1、用 RANSAC 的方法去除噪声, 选择几何上匹配的特征点, 估计两幅图像之间的投影矩阵(Homography);

2、找到匹配图像对的特征重合部分区域;

3、针对每个重合区域:

1) 用光束法(bundle adjustment)的去寻找摄像机的旋转参数和焦距;

2) 然后用多条带融合(mutli-band blending)的方法得到全景图;

输出：全景图像

[0028] 由于视频监控是长时间的行为, 所以背景图像也需要一个更新的过程, 我们设定一个阈值, 当新背景图像和视频浓缩中正在使用的背景图像差别高于阈值, 我们就使用新的背景图像, 计算方法如下:

$$[0029] \quad \sum_{(x,y) \in B} \|B_{new}(x,y) - B(x,y)\| > C$$

[0030] x, y 为图像的坐标, $B_{new}(x, y)$ 为新背景图像的亮度像素值, $B(x, y)$ 为视频浓缩中正在使用的背景图像的亮度像素值, C 为人为设定的阈值, 在我们的实验中, 我们定义 C 为图像长宽乘积的 $1/3$ 。

[0031] 步骤 S3 : 轨迹投影

[0032] 在得到单摄像机视频运动轨迹后, 我们利用视频监控网络全景图像的生成和更新中得到的投影矩阵, 把轨迹投影到全景图平面。在得到投影后的轨迹后, 为了增强摄像机之间的轨迹匹配的准确度, 提高视频浓缩的效果, 我们采用了如下一些特征同时来表述运动轨迹: 梯度方向直方图 (HOG feature), RGB 颜色直方图, 运动轨迹的平均加速度, 运动轨迹的平均速度, 轨迹长度。HOG 特征可以有效地描述运动目标的形状信息, 我们采用了 32 维梯度方向直方图, 然后对轨迹上所有观测点 HOG 做平均。我们采用 RGB 3 个颜色分量的颜色直方图, 每个分量占 8 个柱 (bin), 然后对轨迹上所有观测点的直方图做平均。运动轨迹的平均加速度和平均速度, 对轨迹摄像机之间轨迹匹配非常有用, 而轨迹长度可以用来度量轨迹的重要性。最后我们得到特征向量如下:

$$[0033] \quad (f_{RGB}^{n,i}, f_{HOG}^{n,i}, V^{n,i}, A^{n,i}, L^{n,i})$$

[0034] 式中上标 n, i 分别代表第 n 个摄像机的第 i 条轨迹, $f_{RGB}^{n,i}, f_{HOG}^{n,i}$ 为第 n 摄像机中第 i 条轨迹颜色直方图和 HOG 直方图, $V^{n,i}$ 为第 n 摄像机中第 i 条轨迹平均速度, $A^{n,i}$ 为第 n 摄像机中第 i 条轨迹平均加速度, $L^{n,i}$ 为第 n 摄像机中第 i 条轨迹长度。

[0035] 步骤 S4 : 重叠区域运动轨迹匹配和融合

[0036] 采用图匹配的方法,基于随机游走思想,对两个摄像机的视频源的投影轨迹进行匹配,得到匹配轨迹对,对重叠区域的匹配轨迹进行选择 and 融合,得到完整的大场景轨迹;

[0037] 4.1 运动轨迹匹配

[0038] 对于相邻的两个摄像机,我们把每个摄像机看做一幅图像,图像里面的节点就代表一条轨迹。这样对于相邻两幅图像的轨迹匹配问题,我们可以看做图像之间点的最大匹配问题来解决。假设 G^a 和 G^b 是摄像机 a 和摄像机 b 代表的轨迹图, G^a 和 G^b 的图匹配可以用随机游走 (random walk) 思想在轨迹点对相似图 G^{rw} 来求解。 G^a 和 G^b 到 G^{rw} 的转换可以参考图 2。在图 2 中, G^a 中的 1 代表图像点 1, 2 代表图像点 2, G^b 中的 p 代表图像点 p, q 代表图像点 q, G^{rw} 上标 rw 为 reweighted 的缩写,代表由 G^a 和 G^b 构成的新的图, G^{rw} 中点 1p 代表由 G^a 的点 1 和 G^b 的点 p 构成的新点, G^{rw} 中点 1q 代表由 G^a 的点 1 和 G^b 的点 q 构成的新点, G^{rw} 中点 2p 代表由 G^a 的点 2 和 G^b 的点 p 构成的新点, G^{rw} 中点 2q 代表由 G^a 的点 2 和 G^b 的点 q 构成的新点。图 G^{rw} 点之间的双向箭头,代表着点的关系,箭头越粗代表关系越近,距离越小。

[0039] 轨迹点对相似图 G^{rw} 考虑了轨迹点对之间的相似性, G^a 和 G^b 图匹配问题就转换为用随机游走 (random walk) 的思想在轨迹点对相似图 G^{rw} 上对点进行排序 and 选择。我们选择一个指示向量 x , 使下面的函数最大:

[0040] $x^* = \arg \max (x^T W x)$

[0041] 其中 $x \in \{0, 1\}^{n^a n^b}$, $\forall i \sum_{a=1}^{n^a} x_{ia} \leq 1, \forall i \sum_{i=1}^{n^b} x_{ia} \leq 1$

[0042] 关系度量矩阵 W 考虑了两个摄像机 ($node_i^n, node_j^m$) and ($node_k^n, node_l^m$) 点对之间的关系, T 代表对向量转置、 $node_i^n$ 代表摄像机 n 中的第 i 条轨迹、 $node_j^m$ 代表摄像机 m 中的第 j 条轨迹、 $node_k^n$ 代表摄像机 n 中的第 k 条轨迹、 $node_l^m$ 代表摄像机 m 中的第 l 条轨迹。在我们的实验中,关系度量矩阵由如下式子获得:

[0043] $W = \exp(-|D_{i,j}^a - D_{k,l}^b|^2 / \sigma^2)$

[0044] D 为两点间特征的欧式距离, σ 为自己定义的方差。为了匹配在随机游走 after 更加相似, 非匹配轨迹更加远离, 我们在随机游走的过程中, 加入一个吸收因子, 如图 3 所示。图 3 中 abs 代表吸引子, G^{rw} 上标 rw 为 reweighted 的缩写, 图上中 G^{rw} 是在图 2 中由 G^a 和 G^b 构成的图 G^{rw} 中加入一个吸引子所得。图 2 中示出 G^{rw} 的点 1p、1q、2p、2q, 可以参考图 2 解释。

[0045] 加权随机游走 (Reweighted random walk) 的详细算法流程如下:

[0046]

- 1 计算相似性矩阵 W ，给定权重(reweighted)因子 α 和膨胀因子 β ；
- 2 对冲突的匹配对相似性矩阵 $W_{ia,jb}$ 置 0，即 $W_{ia,jb} = 0$ ；
- 3 计算相似性矩阵的最大值 $d_{\max} = \max_{ia} \sum W_{ia,jb}$ ； ia 和 jb 为相似性矩阵的下标；
- 4 计算过渡矩阵 $P = W / d_{\max}$ ，并初始化向量 x ；
- 5 循环：
- 6 计算 $\bar{x}^T = x^T P$ ， P 为过渡矩阵， T 表示转置， \bar{x} 为所有新矩阵；
- 7 计算 $y = \exp(\beta \bar{x} / \max \bar{x})$ ；

[0047]

- 8 循环：
- 归一化行向量 $y_{ai} = y_{ai} / \sum_{i=1}^I y_{ai}$ ， y_{ai} 代表 y 矩阵第 a 行第 i 列个元素
- 归一化列向量 $y_{ai} = y_{ai} / \sum_{i=1}^A y_{ai}$
- 9 直到向量 y 收敛；
- 10 $y = y / \sum y_{ai}$ ；
- 11 求出 $x^T = \alpha \bar{x}^T + (1 - \alpha) y^T$ ， x_{ai} 代表 x 矩阵第 a 行第 i 列个元素

$$x = x / \sum x_{ai}$$
- 12 直到向量 x 收敛；
- 13 按照向量 x 的向量值的从大到小排列，得到最佳匹配对；

[0048] 算法 2

[0049] 4.2 轨迹的选择和融合

[0050] 在求得摄像机之间的轨迹匹配后，需要对重叠区域的匹配轨迹进行选择 and 融合，以得到完整的大场景轨迹。假设 $T_n(t)$ 和 $T_m(t)$ 代表 t 时刻两个摄像机的一对匹配轨迹，我们用自适应权重的方法对 $T_n(t)$ 和 $T_m(t)$ 进行融合的方法如下：

[0051]

$$T'_{n,m} = \begin{cases} w_1 T_n(t) + w_2 T_m(t), & \text{重叠区域,} \\ T_n(t), & \text{在摄像头n区域} \\ T_m(t), & \text{在摄像头m区域} \end{cases}, \quad T'_{n,m} \text{ 为 } T_n(t) \text{ 和 } T_m(t) \text{ 融合成的新} \\ \text{轨迹, 为大场景轨迹}$$

[0052] 其中 w_1 代表轨迹 T_n 的权重， w_2 代表轨迹 T_m 的权重。权重的获取是根据轨迹的重要程度而来，我们对观测点多的轨迹和较长的轨迹，赋予更高的权重，方法如下：

$$[0053] \quad w_1 = \frac{|T_n| + L_{T_n}}{|T_n| + |T_m| + L_{T_n} + L_{T_m}}, w_2 = \frac{|T_m| + L_{T_m}}{|T_n| + |T_m| + L_{T_n} + L_{T_m}}$$

[0054] 其中 $|T_m|$ 和 $|T_n|$ 为轨迹的观测点数, L_{T_n} 和 L_{T_m} 为轨迹的长度。

[0055] 步骤 S5: 视频监控网络的视频浓缩

[0056] 对大场景轨迹按时间重新排列, 然后在全景图像上对完整的大场景轨迹进行展现, 得到基于全景图像的视频浓缩。

[0057] 如图 4 示出本发明的视频自动浓缩技术示例, 假设 $T_i(x, y, t)$ 为最终映射后得到第 i 条轨迹, $T_j(x, y, t)$ 为第 j 条轨迹。在视频浓缩中, 我们要打乱原始视频源中的轨迹的时间顺序, 但尽量不破坏轨迹的空间位置, 对所有轨迹进行重新排列, 达到压缩视频长度的目的。

[0058] 重新排列轨迹, 会带来很多能量损耗, 如图 4, 原始视频中有 5 条轨迹, 分别为轨迹 1、轨迹 2、轨迹 3、轨迹 4、轨迹 5, 其中轨迹 1 和轨迹 2 在原始视频中为两条不交叉的轨迹, 但在浓缩视频中, 可能存在交叉, 这就带入一个能量损耗, 因为交叉处, 一条轨迹会遮挡另外一条轨迹。我们总共定义了如下一些损耗 E_a 、 E_c 、 E_t :

$$[0059] \quad 1) \quad E_a = \sum_{x,y,t}^{i \in \text{Lost}} T_i(x, y, t)$$

[0060] Lost 代表轨迹丢失。此项为轨迹丢失带来的损耗, 在浓缩视频中, 原始的轨迹没有出现, 其损耗为轨迹所有亮度分量的和。

$$[0061] \quad 2) \quad E_c = \sum_{x,y,t \in T_i \cap T_j} T_i(x, y, t) T_j(x, y, t)$$

[0062] 为轨迹发交叉的损耗, 对于交叉, 我们用交叉处两条轨迹的亮度分量乘积总和作为惩罚, 交叉会影响最后效果, 所以惩罚比较大, 这种损耗应该尽量避免。

[0063]

$$3) \quad E_t = \begin{cases} 0, & \text{两条轨迹时间顺序没变} \\ c \times \exp(-(t_{T_i} - t_{T_j}) / \sigma) & \end{cases}$$

[0064] 为两条轨迹时间先后顺序发生改变的损耗, 如图 4, 轨迹 2 和轨迹 3 的时间顺序就发生了改变。 t_{T_i} 和 t_{T_j} 为轨迹的结束时间点。另外 c 为自定义的加权系数, σ 为自己设定的高斯模型的方差。

$$[0065] \quad 4) \quad E_{\text{overlapping}} = \sum_{x,y,t \in T_i \cap T_j} T_i(x, y, t) + T_j(x, y, t) - T_{ij}(x, y, t)$$

[0066] $E_{\text{overlapping}}$ 为重叠区域轨迹选择带来的损耗。 $T_{ij}(x, y, t)$ 为 $T_i(x, y, t)$ 和 $T_j(x, y, t)$ 合成的轨迹, 代表在运动区域使用合成轨迹带来的损耗。

[0067] 5). E_L 为压缩长度的损耗

[0068] 这个为限制参数, 为了定义压缩率, 压缩后视频越长, 我们定义这个值越大, 具体值我们定义为压缩后的帧数乘以 100。

[0069] 最后的能量损失函数 E 公式为

$$[0070] \quad E = E_a + E_c + E_t + E_{\text{overlapping}} + \lambda * E_L, \lambda \text{ 为自己定义的参数, } E_L \text{ 为压缩长度的损耗。}$$

[0071] 通过优化算法, 比如模拟退火法, 解这个能量损失函数, 使它的值最小, 就得到最后的视频浓缩需要的轨迹排列, 然后把排列好的轨迹, 合成到全景图上。

[0072] 以上所述,仅为本发明中的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉该技术的人在本发明所揭露的技术范围内,可理解想到的变换或替换,都应涵盖在本发明的包含范围之内。

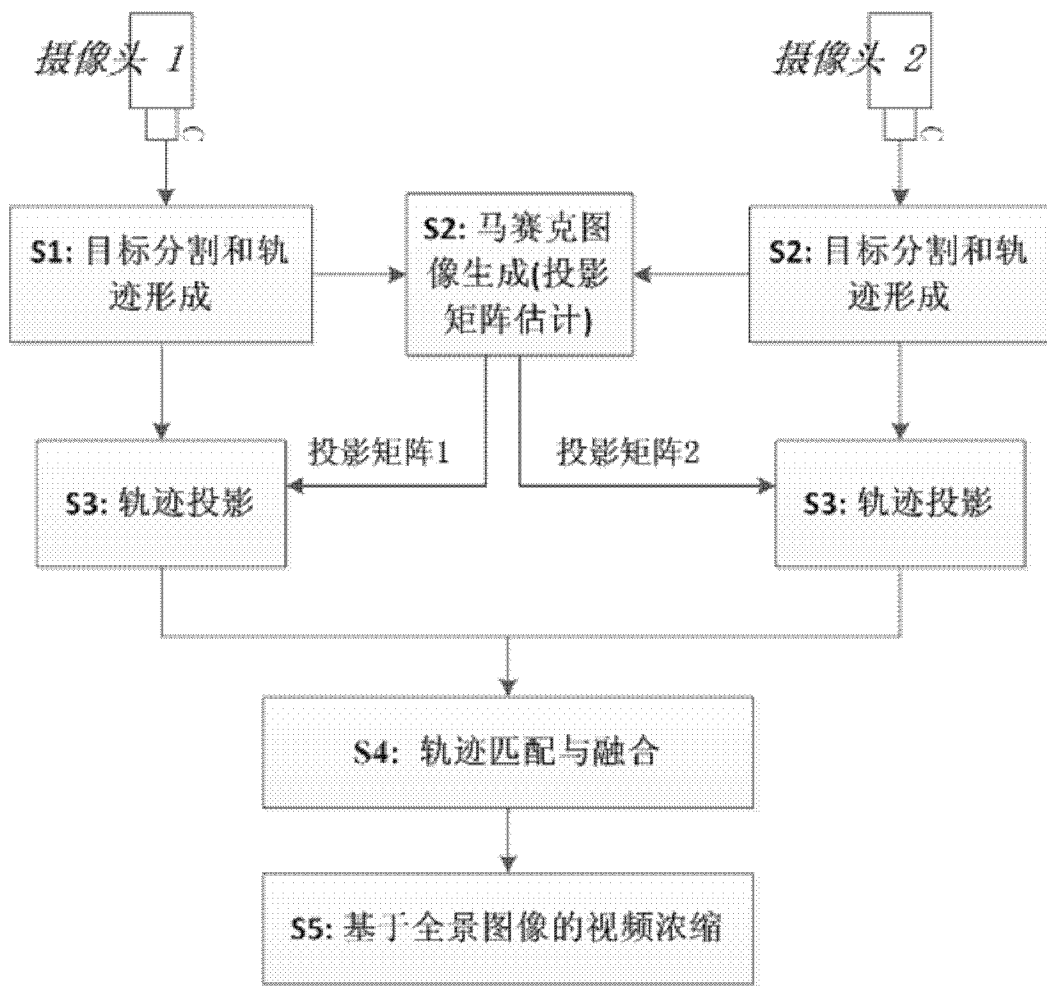


图 1

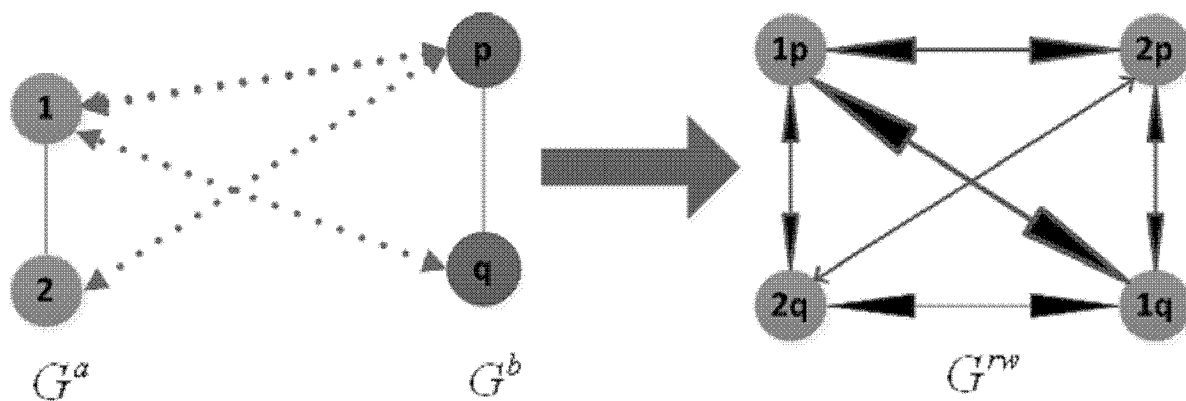


图 2

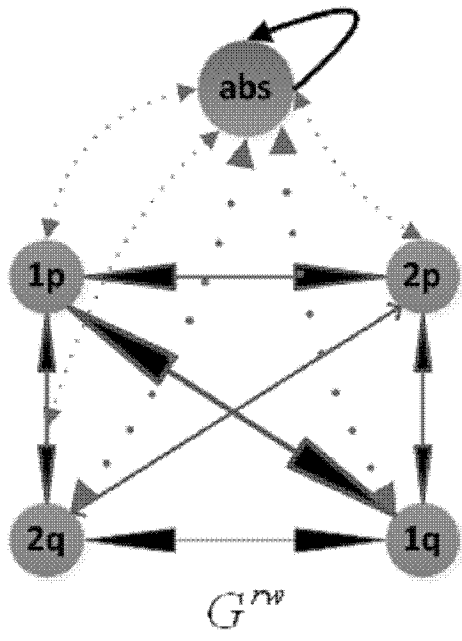


图 3

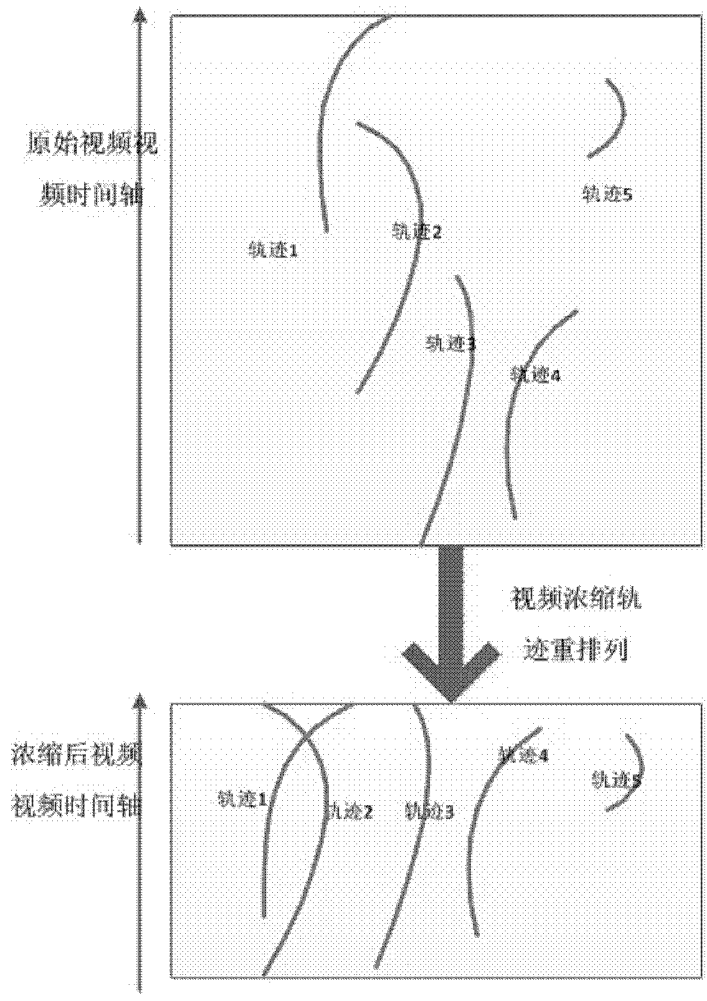


图 4