

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06K 9/00 (2006.01)

G06T 7/20 (2006.01)



# [12] 发明专利申请公布说明书

[21] 申请号 200710129085.3

[43] 公开日 2007年12月26日

[11] 公开号 CN 101093541A

[22] 申请日 2002.11.29

[21] 申请号 200710129085.3

分案原申请号 02152666.4

[30] 优先权

[32] 2001.12.3 [33] US [31] 10/006,927

[71] 申请人 微软公司

地址 美国华盛顿

[72] 发明人 芮永 陈云强

[74] 专利代理机构 中国国际贸易促进委员会专利商  
标事务所  
代理人 付建军

权利要求书2页 说明书39页 附图13页

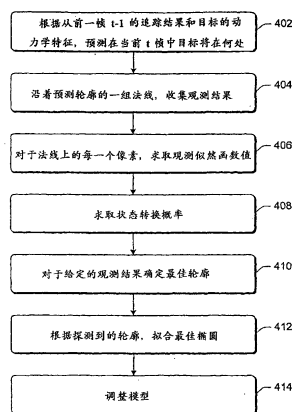
## [54] 发明名称

使用多种线索对多个人进行自动探测和追踪的方法和系统

## [57] 摘要

对多个人的自动探测和追踪包括接收视频和/或音频内容的一帧，并在该帧中为新的面孔区域确认一个候选区域。一个或多个分级验证级别用于验证人的面孔是否在候选区域中，如果一个或多个分级验证级别验证了人的面孔是在候选区域中，就作出该候选区域包括一个面孔的指示。多个音频和/或视频线索用于在视频内容中一帧一帧地追踪每个证实的面孔。

400



1. 一种沿着内容的若干帧追踪目标的方法，该方法包括：  
在一帧中预测目标将在何处；  
对惩罚粗糙的平滑约束进行编码；  
对多个可能的目标位置应用平滑约束；以及  
在该帧中选择具有最平滑轮廓的目标位置作为目标的位置。
2. 根据权利要求1的方法，其特征在于，该预测使用多个线索，包括前景颜色、背景颜色、边缘强度、运动和音频。
3. 根据权利要求1的方法，其特征在于，平滑约束既包括轮廓平滑又包括区域平滑。
4. 根据权利要求1的方法，其特征在于，对平滑约束进行编码包括产生隐藏马尔可夫模型（HMM）的状态转换概率。
5. 根据权利要求1的方法，其特征在于，对平滑约束进行编码包括产生联合概率数据关联滤波器（JPDAF）的状态转换概率。
6. 根据权利要求1的方法，其特征在于，使用多个线索追踪每个证实的面孔进一步包括对于每个面孔：  
根据该帧中观测的一个或多个线索，在后续帧中调整面孔的预测。
7. 根据权利要求1的方法，其特征在于，预测目标将在何处包括：  
存取面孔的一个或多个特性点组成的组；以及  
分析该帧以确认包括一个或多个特性点组成之组的区域。
8. 根据权利要求1的方法，其特征在于，使用多个线索追踪每个证实的面孔包括一帧一帧地同时追踪多个可能的面孔位置。
9. 根据权利要求8的方法，进一步包括使用多假设追踪技术同时追踪多个可能的位置。
10. 根据权利要求1的方法，其特征在于，该目标包括视频内容中的一个面孔。
11. 根据权利要求1的方法，其特征在于，该目标包括音频内容中的一个声源位置。

12. 一种在内容的一帧为一个未追踪的面孔探测候选区域的方法，所述方法包括：

探测在该帧中是否有运动；

如果在该帧中有运动，那么执行基于运动的初始化，以确认候选区域；

探测在该帧中是否有音频；

如果在该帧中检测到音频，那么执行基于音频的初始化，以确认候选区域；以及

如果在该帧中既没有检测到运动没有检测到音频，就使用快速面孔探测器来确认候选区域。

13. 根据权利要求 12 的方法，其特征不在于包括：

确定在跨越该帧的多条线上的多个像素处是否有运动；

对多条线中若干条的多个区段，产生帧差异之和；

对多条线中的每一条，选择具有最大和的区段；

识别选定区段的最平滑的区域；

检验该最平滑的区域是否像人的上半身；以及

提取最平滑区域中像人的头部的部分作为候选区域。

14. 根据权利要求 13 的方法，其特征不在于，确定在跨越该帧的多条线上的多个像素处是否有运动包括：

对于多个像素中的每一个，确定该帧中像素的亮度值与其它的一帧或多帧中对应像素的亮度值之间的差异是否超过阈值。

## 使用多种线索对多个人进行自动探测和追踪的方法和系统

本申请是申请号为 02152666.4、申请日为 2002 年 11 月 29 日、发明名称为“使用多种线索对多个人的自动探测和追踪”的专利申请的分案申请。

### 技术领域

本发明涉及影像和 / 或音频处理, 和 / 或计算机视觉, 尤其是涉及对多个人的自动探测和追踪。

### 背景技术

分析视频数据的系统变得越来越流行了。视频会议系统就是这种系统的实例——它们允许进行视觉互动的会议, 尽管与会者可能位于不同的地理位置。典型情况下, 视频会议的视觉方面使它比电话会议更加引人入胜, 而同时与一个或多个参加者必须旅行到会议地点才能亲自到会相比, 这也是花销更低的一种替代形式(典型情况下, 也能够以更短的通知来实现)。

某些当前的视频会议系统使用自动的基于音频的探测技术和 / 或预置来移动摄像头(例如使摄像头摇摄或俯仰)。然而, 当前的视频会议系统存在着许多问题。一个这样的问题是, 基于音频的发言者探测技术的准确度不高。另外, 典型情况下, 视频会议系统不知道会场上有多少参加者(包括参加者进入或离开会场时)、参加者位于何处(坐或立)以及哪个参加者正在发言。虽然某些系统可以手工设定参加者的信息(例如参加者的数目及其位置), 但是这就需要用户进入设定的信息范围, 它势必限制与会者在房间内移动的能力, 以及与会者加入会议的能力。

本文中介绍的多个人的自动探测和追踪, 有助于解决这些和其它

的问题。

### 发明内容

本文介绍多个人的自动探测和追踪。

本发明提供了一种方法，包括：接收内容的帧；在该帧中自动探测可能包括面孔的候选区域；使用一个或多个分级验证级别，验证人的面孔是否在该候选区域中；如果一个或多个分级验证级别验证了人的面孔在该候选区域中，就指明该候选区域包括所述面孔；以及使用多个线索在内容中一帧一帧地追踪每个证实的面孔。

本发明还提供了一种在视频内容中追踪多个人的系统，该系统包括：一个自动初始化模块，在该视频内容的一帧中为新的面孔探测候选区域；一个分级验证模块，产生候选区域的置信度；以及一个多线索追踪模块，使用多个视觉线索追踪前面带有置信度的候选区域，该置信度是由分级验证模块产生并超过阈值。

一方面，接收内容（例如音频和/或视频）的一帧，并且在该帧中为了一个新的面孔区域而确认一个或多个候选区域。然后使用分级验证法来验证某个人的面孔是否在候选区域中，如果分级验证法验证了一个人的面孔是在候选区域中，就产生该候选区域包括一张面孔的一个标记。验证了区域之后，在内容中一帧一帧地使用多个线索来追踪每个验证后的面孔。

一方面，在这种探测和追踪的框架中有三个主要的模块：自动初始化模块、分级验证模块和多线索追踪模块。由自动初始化模块来接收内容（例如音频和/或视频）的一帧，并且在该帧中为了一个新的面孔（或其它目标）区域而确认一个或多个候选区域。然后使用分级验证模块验证某个人的面孔是否在候选区域中，如果分级验证模块验证了一个人的面孔是在候选区域中，就产生该候选区域包括一张面孔的一个标记。区域经过验证之后，多线索追踪模块在内容中一帧一帧地使用多个线索来追踪每个验证后的面孔。在整个追踪过程中，被追踪的面孔由分级验证模块持续地验证。如果置信度水平高，多线索追

踪模块就保持该面孔的追踪；如果置信度变低了，就终止特定面孔的追踪。追踪模块和验证模块等待初始化模块供应更多的候选区域。

### 附图说明

本文档中自始至终使用同样的号码表示类似的组件和 / 或特性。

图 1 展示了一种示范环境，其中能够使用稳健的自动确认和追踪。

图 2 展示了另一种示范环境，其中能够使用稳健的自动确认和追踪。

图 3 展示了使用稳健的自动确认和追踪的一种示范系统。

图 4 是一个流程图，展示了一个示范过程，用于为新的面孔区域探测候选区域。

图 5 是一个流程图，展示了一个示范过程，使用基于运动的初始化，为新的面孔区域确认候选区域。

图 6 展示了视频内容的一帧的一个示范影像。

图 7 是一个流程图，展示了一个示范过程，用于执行分级验证。

图 8 展示了一个示范过程，用于基于色彩的快速验证。

图 9 是一个流程图，展示了一个示范过程，用于执行多线索追踪。

图 10 更详细地展示了一个示范的模拟和比较，用于多线索追踪。

图 11 是一个影像，展示了区域平滑的概念。

图 12 展示了从图 11 测出的亮度。

图 13 展示了图形匹配距离的一种示范计算。

图 14 展示了对某个目标从一帧到下一帧的示范追踪。

图 15 是一个流程图，展示了一个示范的无气味微粒滤波器过程。

图 16 展示了一种示范的多话筒环境。

图 17 展示了一种示范的通用计算机环境。

### 具体实施方式

本文介绍对多个人的自动探测和追踪。分析视频内容和 / 或音频内容，以自动探测内容的各帧中的个人。一旦探测到，就在后续的帧

中自动追踪这些人。如果对某个人的追踪中断了，就再次自动探测这个人，并恢复对这个人的追踪。

图 1 和图 2 展示了示范的环境，其中能够使用稳健的自动确认和追踪。在图 1 中，多 ( $n$ ) 个视频会议系统 102 能够相互对一个或多个进行音频和 / 或视频内容的通信。在视频会议系统 102 中可以使用为数众多的不同摄像头系统，比如常规的摇摄 / 俯仰 / 变焦摄像头、360 度全景摄像头（例如能够数字地而不是机械地摇摄 / 俯仰 / 变焦的摄像头）等等。一种这样的 360 度全景摄像头系统使用一个指向抛物线镜面设备的摄像头，然后使用多种标定技术消除影像的扭曲，获得若干正常的影像，由这些影像能够构造摄像头周围的 360 度全向影像。这种 360 度全景摄像头系统的一个实例，可以参见发明者 Yong Rui、Anoop Gupta、Johnathan J. Cadiz 和 Ross G. Cutler 2001 年 6 月 14 日提交的、标题为“Automated Online Broadcasting System and Method Using an Omni-Directional Camera System for Viewing Meetings Over a Computer Network”的待批准美国专利申请书 09 / 681843 号。另一种这样的 360 度全景摄像头系统使用多个摄像头（每一个都具有小于 360 度的视野），其排列方式使它们能够一起提供接近 360 度的视野。

会议系统 102 中的每一个都包括追踪模块 104，它稳健地自动探测和追踪在对应系统 102 的多个人。这种探测和追踪能够用于多种目的，比如使摄像头俯仰 / 摇摄 / 变焦、突出某个人（例如采用一个箭头指向这个人或者采用一个圆圈围绕这个人）等等。

视频会议系统 102 能够以为数众多的方式中的任何一种连接在一起。例如，可以使用一条或多条电话线路（包括数字线路，比如 ISDN），将多个系统 102 连接在一起，或者直接连接，或者通过某个中心设备或位置，可以使用常规的数据网络（例如因特网、内联网等等）将多个系统 102 连接在一起，等等。

在图 2 中，系统 112 包括追踪模块 114，它接收内容 116。典型情况下，内容 116 是音频 / 视频内容，但是也可能包括其它类型的内容

(例如共享的白板等等)以及/或者可能不包括音频内容或视频内容。追踪模块 114 分析内容 116, 并根据内容 116 中多个人的影像和/或音频, 稳健地自动探测和追踪这些人。能够以为数众多的方式中的任何一种, 使内容 116 可为系统 112 可用, 比如系统 112 的摄像头和话筒、记录着内容的记录介质(例如磁带、光盘等)、电话线路或网络输入等等。

图 3 展示了使用稳健的自动确认和追踪的一种示范系统 130。系统 130 可能是——例如——图 1 的视频会议系统 102 或者图 2 的系统 112 中的任何一种。系统 130 包括探测和追踪模块 132、通信模块 134、音频捕捉模块 136 和视频捕捉模块 138。还可以包括多种其它模块(未显示), 比如白板捕捉模块。通信模块 134 管理系统 130 与其它系统的通信, 比如其它的、图 1 的视频会议系统 102, 或者可以从其接收要分析之内容的其它设备。通信模块 134 能够支持为数众多的常规的和/或专有的协议。

在系统 130 中, 音频捕捉模块 136 管理音频内容的捕捉, 比如通过一个或多个话筒(未显示)——系统 130 的一部分。还可以进行进一步的处理(例如使用声束形成技术)以提高音频的质量。音频内容转换为数字格式(如果需要)并使探测和追踪模块 132 可用它来追踪。在系统 130 中, 视频捕捉模块 138 管理视频内容的捕捉, 比如通过一个或多个视频捕捉设备(例如模拟或数字视频摄像头(未显示))——系统 130 的一部分(可能包括——例如——固定摄像头、常规的摇摄/俯仰/变焦摄像头、360 度全景摄像头等等)。然后, 捕捉的视频内容帧转换为数字格式(如果需要)并使探测和追踪模块 132 可用它来探测和追踪若干个人。音频和视频内容相互相关(例如在捕捉的时间), 因此对于内容的任何特定部位(例如某帧), 视频和音频内容都是已知的。在另一个实施例中, 可能不包括模块 134、136 和 138 中的一个或多个。例如, 系统可能不包括或者视频捕捉模块 138, 或者音频捕捉模块 136。

探测和追踪模块 132 包括自动初始化模块 140、分级验证模块 142、



多线索追踪模块 144 和面孔 / 候选区域追踪列表 146。探测和追踪模块 132 自动探测视频内容中包括或者可能包括人面孔的若干区域，并使用多种线索来追踪探测到的区域。探测和追踪模块 132 能够探测包括面孔和面孔候选区域的多个区域，并且同时追踪这些区域。

探测和追踪模块 132 分析内容的部位，比如帧。例如，典型情况下视频内容捕捉为每秒若干帧（例如静止影像）（典型情况下在每秒 15-60 帧的量级，尽管可以使用其它速率）。这些视频帧，以及对应的音频内容（例如每  $1/15$  至  $1/60$  秒的音频数据）用作模块 132 探测和追踪所用的帧。记录音频时，典型情况下音频的采样率比视频高得多（例如尽管对于视频每秒可能捕捉 15 至 60 幅影像，却可能要捕捉数千个音频样点）。这些音频样点可能以各种不同的方式对应于视频中特定的一帧。例如，从捕捉某视频帧到捕捉下一视频帧之间的音频样点，可以是对应于该视频帧的音频帧。通过另一个实例，以视频捕捉帧为中心时间的音频样点，可以是对应于该视频帧的音频帧（例如，以每秒 30 帧捕捉视频，音频帧的范围可能是从捕捉视频帧之前的  $1/60$  秒至捕捉视频帧之后的  $1/60$  秒）。

此外，在某些情况下可能没有视频内容。在这些情况下，就能够以为数众多的方式，由采样后的音频产生音频内容的帧。例如，每  $1/30$  秒或  $1/60$  秒的音频样点可以组成音频内容的帧。

在某些情况下，音频内容可能包括并不直接对应于视频内容的数据。例如，音频内容可能是音乐的音轨，而不是视频内容中人们的语音。在这些情况下，本文中介绍的探测和追踪依赖于视频内容，而无需音频内容。

虽然本文的主要讨论是关于使用视频和音频内容，但是探测和追踪模块 132 也可以仅仅根据视频内容或者仅仅根据音频内容而操作。在没有音频内容的情况下，就不执行下面讨论的处理音频内容的过程。同样，在没有视频内容的情况下，就不执行下面讨论的处理视频内容的过程。

面孔 / 候选区域追踪列表 146 保持的信息用于每个探测的区域，

其中包括或者可能包括某个人面孔。这些可能包括某个面孔但是面孔的存在尚未验证的区域，被称为候选区域。在展示的实例中，由中心坐标 148、边界框 150、追踪持续时间 152 和上次验证后的时间 154 来描述每个区域。由一个中心坐标和一个边界框，来规定包括若干面孔或面孔候选者的视频内容区域。中心坐标 148 表示该区域的近似中心，而边界框 150 表示围绕该中心坐标的矩形区域。这个矩形区域就是包括某个面孔或面孔候选者、并被探测和追踪模块 132 追踪的区域。追踪持续时间 152 表示该区域中的面孔或面孔候选者已经被追踪了多长时间，而上次验证后的时间 154 表示多长时间以前该区域中的面孔或面孔候选者被（验证模块 142）验证过（如下面更详细的讨论）。

如列表 146 中所展示的、描述每个区域的信息仅仅是示意性的，也可以使用多种其它信息。例如，可能不包括中心坐标 148。通过另一个实例，可以使用不同于矩形的区域形状，比如圆、椭圆、三角形、五边形、六边形或者不规则的形状。

追踪列表 146 既记录着若干面孔，又记录着若干面孔候选者，它们能够以多种方式相互区分。例如，可以保持两个字列表（一个确认面孔，另一个确认面孔候选者），或者可以增加另外的字段以便把每个字段标示为面孔或面孔候选者，或者可能是在上次验证后的时间 154 中固有的（例如，若是这个值为空格，就表示该区域尚未被验证为包括某个面孔，因此为面孔候选者）。另外，也可以包括多个列表，而不是单一的列表 146（例如，一个列表用于面孔，另一个列表用于面孔候选者）。

在运行期间，探测和追踪模块 132 一帧一帧地分析内容。对于每一帧，模块 132 激活自动初始化模块 140，它用于为新的面孔区域探测候选者。每个这样的候选者都是视频内容的某个区域，它可能包括某个新的面孔（换句话说，某个目前还没有被追踪的面孔）。一旦探测到，一个候选区域就传递到分级验证模块 142，它接着验证候选区域是否确实包括一个面孔。分级验证模块 142 为每个候选区域产生一个置信度水平，并且如果置信度水平超过某个阈值，就确定保持该候

选区域作为一个面孔区域,在追踪列表 146 中增加该区域的一个描述。如果置信度水平不超过该阈值,分级验证模块 142 就放弃该候选区域。

多线索追踪模块 144 对追踪列表 146 中已确认之区域中的每一个进行追踪。追踪模块 144 使用多种视觉线索,在内容中一帧一帧地追踪若干区域。被追踪区域中的面孔中的每一个,都是某个人的至少一个部位的影像。典型情况下,在产生该内容时人们能够移动,比如站起、坐下、走动、坐在椅子上活动等等。模块 132 不是在内容的每一帧中执行面孔探测,而是一帧一帧地追踪包括面孔(一旦探测到后)的区域,典型情况下,它的计算成本低于面孔探测。

包括追踪列表 146 中某个面孔的每个区域,除了被追踪之外,还要由分级验证模块 142 重复地再验证。多线索追踪模块 144,或者也可能是分级验证模块 142,可以确定何时一个区域要由模块 142 进行再验证。可以按规则的或不规则的间隔对区域进行再验证。再验证某个区域时,分级验证模块 142 为该区域产生一个新的置信度水平,并把该置信度水平与阈值进行比较。如果新的置信度水平超过阈值,那么使该区域的上次验证后的时间 154 复位,并且该区域留在追踪列表 146 中。然而,如果新的置信度水平不超过阈值,那么该区域便从追踪列表 146 中删除。

应当注意,可能会发生多线索追踪模块 144 丢失其追踪的情况。分级验证模块 142 通过确认对包括某个面孔之区域的追踪何时丢失(例如,该区域的置信度水平很低),来解决这些问题。这就允许自动初始化模块 140 重新探测该区域,并继续追踪重新探测到的区域。

### 自动初始化

自动初始化模块 140 使用一种或多种技术为新的面孔区域探测候选区域。这些技术包括基于运动的初始化、基于音频的声源定位以及快速的面孔探测。基于运动的初始化模块 156 使用帧间差异(视频内容的两帧或更多帧之间的差异)来探测运动,并确定其中探测到运动的区域是否包括一个面孔。基于音频的初始化模块 158 分析对应于视频内容的音频内容,探测收到声音的方向,并且搜索该方向的视频内

容区域，以确定收到声音的方向上的区域是否包括一个面孔。模块 156 和 158 都能够分析视频内容的每一帧。另外，模块 156 和 158 之一也可以仅仅对视频内容的某一特定帧进行分析，如果另一个模块 156 或 158 未能探测到任何面孔的话。

当视频内容的帧中没有运动和音频时，就使用快速的面孔探测模块 160。另外，模块 160 也可以用于虽然帧中有运动和 / 或音频，但是无论是模块 156 还是模块 158 都未能探测到面孔（也可能是不论模块 156 或 158 是否探测到面孔）时。快速的面孔探测模块 160 使用一种快速面孔探测器来分析视频内容的帧，并探测该帧中的面孔。当通知自动初始化模块 140，某个区域的再验证结果为该区域包括某个面孔的置信度消失时，就使用丢失置信度区域探测模块 162。尽管丢失了该区域包括某个面孔的置信度，某个面孔在这个区域附近仍然很有可能。丢失置信度区域探测模块 162 与模块 156、158 和 160 中的每一个进行通信，使模块 156、158 和 160 分析视频内容中围绕这个区域的部位，试图从中探测一个面孔。围绕该区域之部位的严格尺寸可以随实施方案而变化（例如，在一个示范实施方案中，该部位可能在该区域上下延伸至区域高度的一半之处，在该区域左右延伸至区域宽度的一半之处。

图 4 是一个流程图，展示了一个示范过程 200，用于为新的面孔区域探测候选区域。图 4 中的过程由图 3 中的自动初始化模块 140 实施，而且可以在软件中执行。

一开始，接收音频 / 视频内容的一帧（202）。内容的这一帧可以来自为数众多的信号源中的任何一种。例如，内容的这一帧可能是图 3 中系统 130 的一台或多台捕捉设备捕捉的，或者该内容可能是在别处捕捉并由通信传到系统 130（例如通过可拆卸的存储设备、通过网络或电话线连接等等）。一旦收到，就通过比较该帧的像素和音频 / 视频内容中前一帧的对应像素，试图探测该帧中的运动（动作 204）。如果探测到了运动，那么就执行基于运动的初始化，以便在帧中为新的面孔区域确认候选者（动作 206）。在动作 206 中使用基于运动的

初始化为新的面孔区域确认了任何的候选者之后，就试图在帧中探测音频（动作 208）。如果探测到了音频，那么就执行基于音频的初始化，以便在帧中为新的面孔区域确认候选者（动作 210）。根据基于运动的初始化和 / 或基于音频的初始化，为新的面孔区域确认的任何候选者都传递到分级验证模块 142，以便进行面孔验证（动作 212）。

返回动作 204，如果在帧中没有探测到运动，那么就试图在帧中探测音频（动作 214）。如果探测到了音频，那么就执行基于音频的初始化，以便在帧中为新的面孔区域确认候选者（动作 210），然后处理过程进入动作 212。然而，如果没有探测到音频，那么就使用一种快速的面孔探测器来为新的面孔区域确认候选者（动作 216）。根据快速的面孔探测，为新的面孔区域确认的任何候选者都传递到分级验证模块 142，以便进行面孔验证（动作 212）。

试图探测运动或音频或者使用快速面孔探测器的帧中区域可以随情况而变化。在追踪列表 146 不包括面孔和面孔候选者的情况下，那么帧中区域就是整个帧。在追踪列表 146 包括一个或多个面孔或面孔候选者的情况下，那么帧中区域包括所有当前没有被追踪的区域（换句话说，没有列在追踪列表 146 中的区域）。在丢失置信度区域探测模块 162 申请分析某个特定区域的情况下，那么帧中区域就是模块 162 确认的区域。

返回图 3，基于运动的初始化模块 156 通过把帧中的像素与前面帧和 / 或后续帧中对应的像素相比较，分析视频内容的一帧，并探测每个像素在这些帧之间是否有运动。据信一个移动的人是在视频内容的前景中，所以模块 156 试图确认这种移动前景的形状。如果该形状类似于人上半身的轮廓（一个较小的头在一个较大的肩膀之上），那么就确定该形状是一个模块候选者。

图 5 是一个流程图，展示了一个示范过程 240，使用基于运动的初始化，为新的面孔区域确认候选区域。图 5 中的过程由图 3 中的基于运动的初始化模块 156 实施，而且可以在软件中执行。

一开始，确定在每个像素处是否有运动（动作 242）。通过把像

素与前面帧中对应的像素相比较,对一帧中的每个像素作出这种确定。比较可以通过例如像素亮度(例如灰度)或颜色值来进行。在比较之前,还可以对像素应用使用多种常规的滤波器。视频内容能够使用一种常规的二维 $(x, y)$ 像素坐标系统来观察。在一帧中某个特定坐标位置处的像素,对应于另一帧中同一坐标位置处的一个像素。在被分析帧的区域中每个像素具有一种帧差异,产生如下:

$$D_t(x, y) = \begin{cases} 1, & I_t(x, y) - I_{t-1}(x, y) > d_{th} \\ 0, & \text{其它} \end{cases}$$

式中  $D_t(x, y)$  为  $t$  帧中影像的  $(x, y)$  位置处的像素与  $t-1$  帧中影像的  $(x, y)$  位置处的像素之间帧差异,  $I_t(x, y)$  为  $t$  帧中影像的  $(x, y)$  位置处的像素,  $I_{t-1}(x, y)$  为  $t-1$  帧中影像的  $(x, y)$  位置处的像素,  $d_{th}$  为决定像素是否为运动像素的阈值。 $d_{th}$  的精确数值可以随实施方案而变,比如根据帧是彩色的还是灰度的,进行过何种滤波(如果有的话)等等。作为一个特殊的实例,如果像素是 256 级灰度的,数值 20 可以用于  $d_{th}$ 。

另外,帧差异也可以不是仅仅根据两帧,而是根据三帧或更多帧来产生。在一个实施方案中,使用三帧(例如  $I_{t-1}$ 、 $I_t$ 、 $I_{t+1}$ )来探测运动的像素。只有在  $I_t(x, y) - I_{t-1}(x, y)$  和  $I_{t+1}(x, y) - I_t(x, y)$  中都有较大帧差异(例如大于  $d_{th}$ )的像素才是运动的像素。

给定了帧差异之后,就在被分析帧区域中影像的每条水平线上,对每个可能的区段产生帧差异之和(动作 244)。被分析帧区域中的影像包括多条水平线。每一水平行的像素都可能是这样一条线,或者每隔  $n$ (例如二、三、四、五等等)个水平行的像素也可以是这样一条线。每条这样的线中,存在着许多区段,在线上具有不同的起点和终点。沿着可能的区段对帧差异求和,用于在被分析的区域中试图确认最可能的前景区段。在图 6 中有更详细的说明。

图 6 展示了视频内容的一帧的一个示范影像。展示的影像 270 包括已经被追踪为包含面孔或面孔候选者的两个区域: 272 和 274, 以及为了寻求新面孔区域的候选者而分析的其余区域 276。假设该影像包括一个人 278, 两条水平线 280 和 282 将在起点  $i$  和终点  $j$  穿越影像 278。在一条特定的线  $li$  上,  $i$  点和  $j$  点之间的所有像素都应当在前景

上，两条相邻的水平线之间的边界也应当具有一种光滑约束——它们趋于具有类似的中心和类似的宽度。帧差异之和用于确认具有起点  $i$  和终点  $j$  之水平线  $l_i$  的部分。

对于每条水平线，水平线上每个可能的区段的帧差异之和  $S$  产生如下：

$$S(i, j) = \sum_{x=i}^j D(x, y) \quad 0 < i < j < N, y \in [0, M]$$

式中  $i$  为区段的起点， $j$  为区段的终点， $D(x, y)$  为沿着该区段在  $x, y$  位置处的帧差异， $N$  为水平线的长度， $M$  为水平线的数目。

对于所有可能  $i$  和  $j$ ，为了加快计算求和的速度，使用了以下过程。首先，对于  $0$  和  $N$  之间——包括端点——的每个  $i$  值，计算下式：

$$S(i, i) = D(i, y), \quad i \in [0, N]$$

然后，对于从  $k=1$  到  $k=N$ ，计算

$$S(i, i+k) = S(i, i+k-1) + S(i+k, i+k), \quad i \in [0, N-k]$$

返回图 5，一旦在水平线上对每个可能的区段都产生了帧差异之和，对于每一条水平线，具有最大和的区段就被选为该线上最可能的前景区段（动作 246）。具有最大和的区段实际上是否为新面孔区域候选者的一部分，也取决于光滑约束，正如下面的讨论。然后，就确定了最可能区段的最光滑的区域（动作 248）。对所有的水平线考虑光滑约束，就产生了最光滑的区域。实现方式如下。该过程起始于  $y=0$ （顶端的水平线）及  $E^0(i^{(0)}, j^{(0)}) = S(i^{(0)}, j^{(0)})$ ，并以下列递归函数前进到  $y=M$ （低端的水平线）：

$$E^0(i^{(y)}, j^{(y)}) = S(i^{(y)}, j^{(y)}) + \max_{i^{(y-1)}, j^{(y-1)} \in [0, N]} \left( E^0(i^{(y-1)}, j^{(y-1)}) + C \left( \begin{bmatrix} i^{(y-1)} \\ j^{(y-1)} \end{bmatrix}, \begin{bmatrix} i^{(y)} \\ j^{(y)} \end{bmatrix} \right) \right)$$

式中  $i^{(y)}$  和  $j^{(y)}$  为第  $y$  条水平线的边界， $N$  为影像的宽度。 $C(.,.)$  参数为光滑能量项。 $C(.,.)$  参数在相邻线之间对不光滑的边界给出了较大的惩罚，它规定如下：

$$C \left( \begin{bmatrix} i^{(y-1)} \\ j^{(y-1)} \end{bmatrix}, \begin{bmatrix} i^{(y)} \\ j^{(y)} \end{bmatrix} \right) = c_c \cdot \left| \frac{i^{(y)} + j^{(y)}}{2} - \frac{i^{(y-1)} + j^{(y-1)}}{2} \right| + c_w \cdot |(j^{(y)} - i^{(y)}) - (j^{(y-1)} - i^{(y-1)})|$$

式中  $c_c$  为对区段中心的不光滑罚系数，而  $c_w$  为对区段宽度的不光滑罚系数。对罚系数  $c_c$  和  $c_w$  可以使用不同的数值，在一个示范实施方

案中， $c_c$  和  $c_w$  中每一个都是 0.5。

然后，通过确定下式，可以获得最光滑的区域：

$$\max_{i^{(M)}, j^{(M)} \in [0, N]} (E^0(i^{(M)}, j^{(M)}))$$

给定了这种最光滑的区域，就能够执行向回追踪，以寻找所有水平线的边界。

给定了最光滑的区域，就检验该区域是否像人的上半身（动作 250）。在展示的实例中，人的上半身包括一个较小的头在一个较大的肩膀之上。所以，就检验这个最光滑的区域是否具有一个近似椭圆的部分（头），位于一个较宽的部分（肩膀）之上。在一个实施方案中，这种检验通过寻找相邻水平线之宽度的最大变化，首先探测颈部的位置。然后，检验颈部之上的部分（头部区域）是否具有比下部区域（肩膀区域）更小的平均宽度。也要检验头部区域的宽度和高度之比是否为近似 1: 1.2。如果所有这些检验结果都是真实的，那么被探测的区域就确定为像人的上半身轮廓。

如果该区域确实像人的上半身，那么就提取该区域中包括头部（但是不包括肩膀）的部分（动作 252），并确认为新面孔区域的候选者（动作 254）。这个提取的区域可能是人头部的近似椭圆的区域，或者是头部周围的某个区域（例如头部周围的矩形区域）。不过，如果该区域不像人的上半身，那么从该帧就没有探测到新面孔的候选者（动作 256）。

在一个实施方案中，如果在动作 254 中确认了一个新面孔区域的候选者，并且在该帧中还有任何另外的区域（不算在动作 254 中确认的候选者和任何其它的面孔和面孔候选者），就重复图 5 中的过程。这样就能在该帧中确认另外的学面孔区域的候选者。

返回图 3 和图 4，基于音频的初始化模块 158 通过使用声源定位器探测收到声音的方向，来分析一帧音频/视频内容（图 4 中的动作 210）。模块 158 假设，这个声音可能是人在发言，因此指示着视频内容的一个区域，其中可能包括一个面孔区域的候选者。能够以为数众多的不同方式来确定收到声音的方向。在一个实施方案中，一个或多



个话筒阵列捕捉声音，并使用一个或多个声源定位算法来确定声音来自哪个方向。可以使用多种不同的常规声源定位算法，比如众所周知的到达时间延迟（TDOA）技术（例如广义互相关（GCC）方法）。

在没有视频内容的情况下，通过适当地布置多个话筒，也能够完成面孔探测。使用三个或更多的话筒，其中至少两个位于不同的水平面上，而且其中至少两个位于不同的垂直面上，就能够确定声源的(x,y)坐标。例如，两个话筒可能位于垂直面中，并且两个话筒可能位于水平面中。那么可以使用多种常规的声源定位算法来确定声源的(x,y)位置——假定它是一个人的嘴巴。这个声源位置本身能够被视为探测到的面孔区域（假定发言者的嘴巴是发言者面孔的一部分），或者该位置也可以扩展（例如增大百分之二三），扩展后的位置用作探测到的面孔区域。

给定了影像中对应于收到声音之方向的区域，初始化模块 158 分析该区域，试图对该区域中的影像拟合一个肤色模型。如果拟合成功，那么拟合肤色模型的区域就确认为新面孔区域的候选者。在一个实施方案中，肤色模型为 HSV（色调饱和度值）颜色空间模型，带有用于训练模型的大量肤色训练数据。应当注意，因为音频已经指出在该区域中有一个面孔，可以使用粗略的探测过程（例如肤色模型）来使面孔定位。

在没有视频内容的情况下，模块 158 依靠声源定位而不使用肤色模型（由于没有视频内容可以应用肤色模型）。

快速模块探测模块 160 使用快速模块探测器，在帧影像的若干区域中探测面孔。探测模块 160 使用的快速面孔探测器，可能不同于分级验证模块 142 使用的面孔探测器——下面有更详细的讨论。为了在计算量和准确性之间取折衷，模块 160 使用的探测器比分级验证模块 142 使用的面孔探测器更快，但是准确性要低一些；不过，模块 160 和 142 可能基于同样的面孔探测算法，但是使用不同的参数或阈值，以便模块 160 提高探测速度（相对于模块 142 的探测速度）。另外，模块 160 和 142 也可以基于两种不同的面孔探测算法。典型情况下，

探测模块 160 使用的探测器快于分级验证模块 142 使用的探测器。

为数众多的面孔探测算法都能够用作快速面孔探测模块 160 的基础，使用什么算法所依据的主要特征是其速度。快速面孔探测模块 160 的目标是很快地探测面孔，如果需要的话，以准确性作为代价。面孔探测可能仅仅是正面，也可能是多视角（不限于正面探测）。这种算法的一个实例是 P. Viola 和 M.J. Jones 介绍的，见“Robust real-time object detection”，Technical Report Series, Compaq Cambridge Research Laboratory, CXRL 2001 / 01, Feb. 2001。这种算法的另一个实例类似于 P. Viola 和 M.J. Jones 讨论的，只是使用探测器级在开始时利用一个探测器，它覆盖宽阔的视角，发展到利用一组多个探测器，每个覆盖较窄的视角。目标从探测器的一级到另一级，每个探测器把该目标划分为或者是面孔，或者不是面孔。每个目标只要被任何一个探测器划分为不是面孔，就将它从过程中排除——仅有通过了所有的探测器级并被划分为面孔的目标才被确认为面孔。

因此，自动初始化模块 140 使用一个或多个基于运动的初始化、基于音频的声源定位和快速探测技术探测新面孔区域的候选者。然后，这些候选者传送到分级验证模块 142，以便验证这些候选者是否确实包括一个面孔。应当注意，并非所有帧都包括新面孔，因此，即使使用了所有上述的技术，自动初始化模块 140 也可能在一帧中探测不到任何新面孔的候选者。

### 分级验证

图 3 中的分级验证模块 142 验证自动初始化模块 140 确认的候选面孔区域。此外，探测和追踪模块 132 考虑了多线索追踪模块 144 在运行期间或许失去追踪的可能性。这可能是由于多种原因，比如遮挡（例如另一个与会者走到视频捕捉设备和被追踪的个人之间时）或者突然的照明变化。分级验证模块 142 以规则的或不规则的间隔，再次验证被追踪的每个目标，并适当地使面孔降级为面孔候选者。根据追踪需要的准确度（较短的间隔趋于改善准确性）、可用的计算能力规模（取决于验证的类型，追踪可能需要比再次验证少的计算能力）以

及验证模块的计算成本，间隔的长度可以变化。

在一个实施方案中，分级验证模块 142 验证目标为面孔，确认一个目标或者是面孔，或者不是面孔。另外，验证模块 142 也可以根据不同的特性（例如音频、颜色直方图距离、边界附近的边缘探测结果、面孔探测结果等等），输出概率统计的验证结果。在这种过程中，输出的概率统计的验证结果可以与微粒滤波的加权方案相结合，下面将更详细地讨论该滤波。

因为考虑计算量，附近验证模块 142 使用一种多级的分级过程来验证某个目标包括一个面孔。该验证过程是一个从粗略到精细的过程，以较快而不太准确的验证开始，如果需要，就上升到较慢而更准确的验证。在展示的实例中，该分级过程包括两个级别。此外，分级过程中也可以包括三个或更多的级别。

图 3 中的分级验证模块 142 包括基于颜色的快速验证模块 164 和多视角的面孔探测模块 166。验证模块 142 假设，典型情况下，在相邻帧期间目标并不显著改变颜色。基于颜色的验证模块 164 根据当前帧中目标的颜色直方图和前面帧中目标的估计颜色直方图之间的相似度，来验证目标。相似度高时，就假设追踪没有发生损失，不必调用多视角的面孔探测模块 166。不过，相似度低、追踪可能已经发生了损失时，因而目标从面孔降级为面孔候选者，并传送到多视角的面孔探测模块 166。如果多视角的面孔探测模块 166 验证该目标为面孔，该目标就从面孔候选者升级为面孔。不过，如果探测模块 166 不把该目标验证为面孔，该目标就从追踪列表 146 中删除。

在一个实施方案中，基于颜色的验证模块 164 对每一帧执行其验证功能，而多视角的面孔探测模块 166 以较低的频度执行其验证功能。举例来说，多视角的面孔探测模块 166 可能每数秒执行一次其验证功能，尽管根据上面讨论的多种因素，也可以使用不同的间隔。

图 7 是一个流程图，展示了一个示范过程 320，用于执行分级验证。过程 320 是由图 3 中的分级验证模块 142 执行，并且可以在软件中执行。

一开始，获得所关注区域的一个影像（动作 322）。所关注区域可能是由自动初始化模块 140 确认的一个候选区域，或者是为进行再次验证的一个区域。可以将带有要分析区域指示的整帧传送到分级验证模块 142，也可以仅仅将帧中包括要分析区域的一部分传送到分级验证模块 142。一旦收到之后，就使用基于颜色的快速验证法来验证该区域中是否有一个面孔（动作 324）。

图 8 中展示了动作 324 的、基于颜色的快速验证法的更多细节。图 8 中的过程 324 由图 3 中的基于颜色的快速验证模块 164 执行，并且可以在软件中执行。一开始，产生当前  $t$  帧中目标的颜色直方图 ( $q_t(x)$ ) (动作 362)。也产生前面帧中目标的估计颜色直方图 ( $p_{t-1}(x)$ ) (动作 364)。估计颜色直方图  $p_{t-1}(x)$  按下式产生：

$$p_{t-1}(x) = \alpha \cdot q_{t-1}(x) + (1 - \alpha) \cdot p_{t-2}(x)$$

式中  $\alpha$  表示加权值， $q_{t-1}(x)$  为前一帧  $t-1$  中目标的颜色直方图， $p_{t-2}(x)$  为对于前一帧  $t-1$  中目标的估计颜色直方图。在不同的实施方案中可以使用范围很广的  $\alpha$  值，选定的确切值是相信历史和相信当前帧之间的折衷（例如在一个示范实施方案中， $\alpha$  值的范围是从 0.25 到 0.75）。因此，根据每一帧中目标的颜色直方图，更新目标的估计颜色直方图  $p_{t-1}(x)$ 。

然后确定两个直方图的相似度（动作 366）。为了确定两个直方图  $q_t(x)$  和  $p_{t-1}(x)$  的相似度测度，使用下列众所周知的 Bhattacharyya 系数：

$$\rho(p_{t-1}(x), q_t(x)) = \int \sqrt{p_{t-1}(x) \cdot q_t(x)} dx$$

式中  $\rho$  表示统计假设试验中分类错误的概率——错误的概率越大，两种分布就越相似。 $\rho$  值的范围从 0 到 1，1 表示两个直方图相同，0 表示两个直方图完全不同。这种相似度测度在本文中也称为置信度。另外，也可以使用其它众所周知的相似度测度，比如 K-L 散度、直方图交会等等。

然后检验两个直方图之间的相似度是否超过某个阈值幅度（动作 368）。如果差异大于阈值幅度，那么该面孔被证实（动作 370）；换

句话说，该目标被证实为包括一个面孔。不过，如果差异不大于阈值幅度，那么面孔就不被证实（动作 372）；换句话说，该目标不被证实为包括一个面孔。在不同的实施方案中，可以使用不同的差异阈值。在一个示范实施方案中，阈值的范围可以从 0.90 到 0.95，在一个具体的实施方案中是 0.94。

返回图 7，根据面孔是否被证实而继续处理（动作 326）。如果面孔被证实，那么它就从面孔候选者升级为面孔（如果还不是面孔的话）（动作 328），分级验证过程完成（动作 330），这时对所关注的区域不再进行更多的验证。然而，如果面孔不被证实，那么该面孔就从面孔降级为面孔候选者（如果当前是面孔的话）（动作 332）。然后包括面孔的目标传送到图 3 的多视角面孔探测模块 166，它使用多视角面孔探测法验证该区域中是否有一个面孔（动作 334）。

多视角的面孔探测模块 166 使用了一个或多个探测过程——它们试图以不同的姿势或者从多个视角来探测人的面孔（换句话说，即使头部可能倾斜、转离影像捕捉设备等等情况下，也要探测面孔）。为数众多的面孔探测技术中的任何一种，都能够被多视角的面孔探测模块 166 所使用。

一种这样的多视角面孔探测过程是基于核心机的过程，更加详细的讨论见 S.Z. Li, Q.D. Fu, L. Gu, B. Scholkopf, Y.M. Cheng, H.J. Zhang, “Kernel Machine Based learning for Multi-View Face Detection and Pose Estimation,” *Proceedings of 8<sup>th</sup> IEEE International Conference on Computer Vision*, Vancouver, Canada, July 9-12, 2001。这种探测过程归纳如下。

令  $I_p \in \mathbb{R}^N$  为面孔的窗内灰度影像或外观。假设所有左转的面孔（视角在  $91^\circ$  和  $180^\circ$  之间）都经镜面反射成右转，所以每一个视角都在  $0^\circ$  和  $90^\circ$  之间。将姿势量化成一组  $L$  个离散值（例如选择  $L=10$ ，用于 10 个等间隔的角，从  $0^\circ$  到  $90^\circ$ ， $0^\circ$  对应于右侧面视角， $90^\circ$  对应于正面视角）。

假设为学习提供了一组训练用面孔影像。影像  $I_p$  不仅视角有变

化，照明也有变化。训练组经过视角标注——每个面孔影像都由人工标注了其尽可能接近真实值的视角值，然后按照最接近的视角值分配到  $L$  个组中的一个。这就产生了  $L$  个经过视角标注的面孔影像子集，用于学习面孔的视角子空间。另一个非面孔的训练组也用于训练面孔探测。

现在，有  $L+1$  个类，由  $l$  为索引，意义如下， $l \in \{0, 1, \dots, L-1\}$  对应于  $L$  个视角的面孔， $l=L$  对应于非面孔类。把输入  $I_p$  划分为  $L+1$  个类之一，就共同执行了两个任务——面孔探测和姿势估计。如果输入划分为  $L$  个面孔类之一，就探测到了一个面孔，对应的视角就是估计的姿势；否则，输入模式被视为非面孔模式。

使用核心机器学习面孔探测和姿势估计分为两个阶段：一个是为了核心主分量分析（KPCA）的视角子空间学习，一个是为了核心支持矢量分类器（KSVC）的分类器训练。第一阶段的训练目标是从  $L$  个面孔视角子集学习  $L$  个 KPCA 视角子空间。一组核心主分量（KPC）是从每个视角子集学到的。最重要的分量（例如最高的 50 个）用作构造视角子空间的基本矢量。这个阶段的学习产生了  $L$  个视角子空间，每一个都由一组支持矢量和对应的系数确定。每个视角信道中的 KPCA 有效地执行了从输入影像空间到输出 KPCA 特性空间（具有与最重要的分量中分量数目相同的维数）的非线性映射。

第二阶段的目标是训练  $L$  个 KSVC，以便为了面孔探测而在面孔和非面孔模式之间进行区分。这个阶段使用一个训练组，包括一个非面孔子集以及  $L$  个视角的面孔子集。一旦训练了 KSVC，就用于每个视角根据对应 KPCA 子空间的特性进行  $L+1$  个类别的划分。视角到对应的 KPCA 子空间的投影用作特性矢量。在 KSVC 中，使用众所周知的一对其余方法来解决多类问题。第二阶段给出  $L$  个 KSVC。

在试验阶段中，对于每个视角  $l$ ，向 KPCA 特性提取器提供一个试验样本，以获得该视角的特性矢量。该视角对应的 KSVC 计算出一个输出矢量  $y_l = (y_l^c | c=0, \dots, L)$ ，作为对输入的  $L+1$  个类的响应。对所有  $L$  个视角信道都这样做，所以就产生了  $L$  个这样的输出矢量

$\{y_i | i=0, \dots, L-1\}$ 。  $y_i^c$  值是判断在第  $i$  个视角 KPCA 子空间中，输入  $I_p$  的特性属于  $c$  类的证据。通过融合所有  $L$  个视角信道的证据，作出最终的分类决定。融合的一种方法是将证据求和；换句话说，对于每个类  $c=0, \dots, L$ ，计算下式：

$$y^c(I_p) = \sum_{i=0}^{L-1} y_i^c$$

这种计算给出了将  $I_p$  划分为  $c$  类的全面证据。通过使该证据最大化而作出最终的决定：如果  $c^* = \arg \max_c y^c(I_p)$ ，那么  $I_p$  属于  $c^*$ 。

继续看图 7，根据多视角面孔探测是否证实了面孔，继续进行处理（动作 336）。如果证实了面孔，那么该面孔就从面孔候选者升级为面孔（动作 328），分级验证过程完成（动作 330）。然而，如果未证实面孔，那么该候选者就从图 3 的追踪列表 146 中清除（动作 338），分级验证过程完成（动作 330）。

在分级验证中没有视频内容可分析的情况下，适当时候也可以仅仅使用音频线索来验证。例如，当被追踪面孔者正在持续发言时，或者进行众所周知的、基于音频发言者的确认时（从而允许将声源联系到发言者个人的语音，通过确定来自某个特定声源位置的语音是否符合以前从该声源位置收到的、同一发言者的确认信息，进行验证），都可以仅仅使用音频线索。

### 多线索追踪

一旦在视频内容的一帧中探测到某个面孔，图 3 中的多线索追踪模块 144 就在视频内容的后续帧中追踪该面孔。其面孔被追踪的与会者可能移动，因此在视频内容的不同帧中，面孔的位置可能会不同。不仅如此，与会者还可能转动其头部（例如这样其面孔就不再直接对着视频捕捉设备），可能发生多种遮挡（例如与会者可能在其面前挥手），照明可能会变化等等。多线索追踪模块 144 试图考虑这些每帧不同的变化。此外，因为这些变化，某些追踪的线索可能变得不可靠。多线索追踪模块 144 也试图考虑线索可靠性从一帧到另一帧中发生的这些变化。

追踪模块 144 在追踪面孔中使用多种线索。在一个实施方案中，

这些追踪线索包括面孔的形状（模拟为一个椭圆）、移动、边缘、前景颜色和背景颜色。此外，也可以不使用这些线索中的一个或多个，或者也可以使用另外的线索，比如音频线索。

有音频内容可用时，多线索追踪模块 144 可以使用音频线索协助追踪（或者作为追踪的唯一基础）。根据声源定位过程进行基于音频的追踪，并且其进行方式如同上面讨论的、图 3 中的基于音频的初始化模块 158 进行的基于音频的探测。

图 9 是一个流程图，展示了一个示范过程 400，用于执行多线索追踪。过程 400 是由图 3 中的多线索追踪模块 144 执行，并且可以在软件中执行。

一开始，根据从前一帧  $t-1$  的追踪结果和目标的动力学特征（由众所周知的 Langevin 过程模拟，下面还要更详细地讨论），预测在当前  $t$  帧中目标将在何处（动作 402）。沿着该目标预测轮廓的一组法线，收集观测结果（动作 404），并对于法线上的每一个像素，求取观测似然函数值（动作 406）。求取从  $t-1$  帧到  $t$  帧的状态转换概率（动作 408），对于给定的观测结果确定最佳轮廓（动作 410）。根据探测到的轮廓，对  $t$  帧中的影像拟合最佳椭圆（动作 412），并调整用于下一帧  $t+1$  的模型（动作 414）。

多线索追踪模块 144 包括多个模块，用于执行图 9 中的动作。在展示的实例中，追踪模块 144 包括：观察似然模块 168、平滑约束模块 170、轮廓选择模块 172 和模型调整模块 174。

多线索追踪模块 144 努力追踪人的头部——它是椭圆形的（大约 1:1.2）。对于被追踪的面孔，人的头部由一个模型来表示，它是具有多种追踪线索的一个椭圆。分析视频内容帧中的影像时，该模型与影像的多个位置进行比较，并确定哪个位置最接近该模型。这个最接近该模型的位置就被选为新的一帧中的面孔。

图 10 更详细地展示了这个示范的模拟和比较。在图 10 中，实线曲线 422 表示在特定的  $t$  帧中，根据从前一帧  $t-1$  追踪的结果，预测的人头部的轮廓。虚线曲线 424 表示  $t$  帧中人头部的真实轮廓。沿着预



测轮廓 422 的多 ( $M$ ) 条法线 426 收集一组测量值。点 428 ( $c(\varphi)$ ) 为第  $\varphi$  条法线上的真实轮廓点。点 430 ( $\rho_\varphi(N)$ ) 为第  $\varphi$  条法线上的预测轮廓点。多线索追踪模块 144 通过使预测轮廓 422 上尽可能多的轮廓点与真实轮廓线 424 上的轮廓点相同, 试图确定真实轮廓 424 的位置。

图 3 中的观测似然模块 168 产生  $\rho_\varphi(\lambda)$  值, 它表明在线  $\varphi$  上像素  $\lambda$  处的影像亮度, 如下式:

$$\rho_\varphi(\lambda) = I(x_{\lambda\varphi}, y_{\lambda\varphi})$$

式中  $\varphi$  的范围从 1 到  $M$  (法线 246 的总数目),  $\lambda$  的范围沿着法线从  $-N$  到  $N$  (每条法线有  $2N+1$  个像素),  $x_{\lambda\varphi}, y_{\lambda\varphi}$  为第  $\varphi$  条法线上像素  $\lambda$  处的对应影像坐标,  $I(x_{\lambda\varphi}, y_{\lambda\varphi})$  为  $(x_{\lambda\varphi}, y_{\lambda\varphi})$  点处的影像亮度。

为了探测轮廓点, 通过使用隐藏马尔可夫模型 (HMM), 可以结合不同的线索 (例如边缘强度、前景和背景的颜色模型) 和先验约束 (例如轮廓平滑约束)。隐藏马尔可夫模型对于本领域的技术人员是众所周知的, 因此除了它们适于本文所介绍的多个人的自动追踪以外, 不再进一步讨论。HMM 的隐藏状态为每条法线上的真实轮廓点 (表示为  $s = \{s_1, \dots, s_\varphi, \dots, s_M\}$ )。HMM 的观测结果  $O = \{O_1, \dots, O_\varphi, \dots, O_M\}$  是沿着每条法线  $\varphi$  收集的。一个 HMM 是由状态的数目 (在我们的情况下是  $2N+1$ )、观测模型  $P(O_\varphi | s_\varphi)$  和转换概率  $p(s_\varphi | s_{\varphi-1})$  指定的。

观测似然模块 168 继续产生多线索观测似然函数如下。在线  $\varphi$  上的观测结果 (表示为  $O_\varphi$ ) 可以包括多个线索, 例如沿着法线的像素亮度 (如  $\rho_\varphi(\lambda), \lambda \in [-N, N]$ ) 和边缘强度 (如  $z_\varphi$ )。通过使用多种常规边缘探测过程中的任何一种, 比如众所周知的 Sobel 边缘探测器或 Canny 边缘探测器, 都可以得到边缘探测结果  $z_\varphi$  的观测似然模型。由于噪音和影像干扰, 沿着每条法线  $\varphi$  可能有多个边缘。使用  $J$  值来表示探测到的边缘的数目 ( $z_\varphi = (z_1, z_2, \dots, z_J)$ )。在  $J$  个探测到的边缘中, 最多只有一个在图 10 中的真实轮廓线 424 上。所以我们可以规定  $J+1$  个假设:

$$H_0 = \{e_j = F : j = 1, \dots, J\}$$

$$H_1 = \{e_j = T, e_k = F : k = 1, \dots, J, k \neq j\}$$

式中  $e_j = T$  表明第  $j$  条边缘与真实轮廓线相关联, 而  $e_j = F$  表明第  $j$

条边缘不与真实轮廓线相关联。所以假设  $H_0$  表明没有边缘与真实轮廓线相关联。

假设沿着法线的影像干扰为众所周知的、空间密度为  $\gamma$  的泊松过程，真实的目标测量结果是标准差为  $\sigma_z$  的正态分布，边缘似然模型按下式获得：

$$p(z_\phi | s_\phi = \lambda_\phi) \propto 1 + \frac{1}{\sqrt{2\pi\sigma_z q \gamma}} \sum_{m=1}^J \exp\left(-\frac{(z_m - \lambda_\phi)^2}{2\sigma_z^2}\right)$$

式中  $q$  为假设  $H_0$  的先验概率。

除了边缘似然模型以外，HMM 框架中也结合了有关前景和背景的区域性质的其它线索，例如混合色彩模型。令  $p(v|FG)$  和  $p(v|BG)$  分别表示前景 (FG) 和背景 (BG) 的颜色分布。后验概率  $p(BG|v)$  和  $p(FG|v)$  能够按下式得到：

$$P(BG|v) = \frac{p(v|BG)}{p(v|BG) + p(v|FG)} \quad (1)$$

$$P(FG|v) = \frac{p(v|FG)}{p(v|BG) + p(v|FG)}$$

如果  $s_\phi = \lambda_\phi$  为法线  $\phi$  上的轮廓点，那么区间  $[-N, s_\phi]$  就在前景上，区间  $[s_\phi + 1, N]$  就在背景上。结合边缘似然模型和颜色后验概率，对于 HMM 就会得出下列多线索观测似然函数：

$$P(O_\phi | s_\phi) = p(z | s_\phi) \cdot \prod_{i=-N}^{s_\phi} P(BG | v = \rho_\phi(i)) \cdot \prod_{i=s_\phi+1}^N P(FG | v = \rho_\phi(i)) \quad (2)$$

在类似方式下，也可以结合其它线索，比如音频线索（如根据声源位置和声音来自特定位置的可能性）。在没有视频内容可供分析的情况下，就仅仅使用音频线索。此外，为了补充或取代这种音频队列，音频也可以用作无气味微粒滤波的建议函数，下面还要更详细地讨论。

HMM 中的另一个分量是转换概率，它确定在时间  $t-1$  的状态如何转换为在时间  $t$  的另一种状态。图 3 中的平滑约束模块 170 得出了转换概率。

为了获得平滑的轮廓，使用转换概率对平滑约束进行编码，并惩罚粗糙。参考图 10，可见法线 426 密集（例如在 30 条法线的数量级）时，在相邻的法线 426 上真实轮廓线 424 的点，与预测轮廓线 422 趋

于具有相同的位移（在每条法线上指数为 0）。这种相关性有助于获得平滑的轮廓。

在 HMM 中，给定了当前的状态  $s_\phi$ ，当前观测结果  $O_\phi$  就独立于前一个状态  $s_{\phi-1}$  和前一个观测结果  $O_{\phi-1}$ 。此外，由于马尔可夫模型的性质，我们有  $p(s_\phi | s_1, s_2, \dots, s_{\phi-1}) = p(s_\phi | s_{\phi-1})$ 。

那么，可以通过状态转换  $p(s_\phi | s_{\phi-1})$  捕捉轮廓平滑约束如下：

$$p(s_\phi | s_{\phi-1}) = c \cdot \exp(-(s_\phi - s_{\phi-1})^2 / \sigma_s^2) \quad (3)$$

式中  $c$  为归一化常数， $\sigma_s$  为预先规定的调节轮廓平滑程度的常数。这种转换概率惩罚相邻法线之间轮廓点的突然变化，因此产生平滑的轮廓。然后，可以由轮廓选择模块 172 获得最佳的轮廓。

平滑约束模块 170 根据上面计算式 (3) 产生的转换概率，考虑了轮廓点，而不管法线上的其它像素。此外，平滑约束模块 170 也可以使用基于 JPDAF（联合概率数据关联滤波器）的方法，不仅对轮廓平滑约束编码，而且也对在法线上多个（例如所有）像素上观测到的区域平滑约束编码。在展示的实例中，使用基于动态规划的 JPDAF 过程来改善实时性能。

在典型条件下，人体部分（例如面部或头部）的像素亮度值在其区域中平缓地变化。所以，在人们的追踪中，前景和背景具有平滑的区域是一个合理的假设，因而两条相邻法线上的观测结果类似。令  $s_\phi$  和  $s_{\phi+1}$  分别为线  $\phi$  和线  $\phi+1$  上的轮廓点。这两个轮廓点将这两条线划分为前景区段和背景区段。根据区域平滑假设，不仅  $s_\phi$  和  $s_{\phi+1}$  相互接近，而且这两条法线上所有其它的像素也都匹配得很好。为了获得区域平滑约束，使用一种联合概率数据关联滤波器来处理法线匹配。换句话说，它不是单点对单点的匹配问题，而是  $(2N+1)$  个点对  $(2N+1)$  个点的匹配问题。通过一起考虑沿着这些法线上的所有像素，可以获得更稳健的匹配结果。所以，典型情况下，基于 JPDAF 过程的转换概率更准确。令  $D^F(i,j)$  和  $D^B(i,j)$  分别为前景（在线  $\phi$  上的  $[-N, i]$ ，在线  $\phi+1$  上的  $[-N, j]$ ）和背景（在线  $\phi$  上的  $[i+1, N]$ ，在线  $\phi+1$  上的  $[j+1, N]$ ）的匹配距离。那么，可以规定转换概率如下，以取代上面参考计算式

### (3) 讨论的转换概率:

$$\log(p(s_2 | s_1)) = D^F(s_1, s_2) + D^B(s_1, s_2) + (s_2 - s_1)^2 / \sigma_s^2 \quad (4)$$

区域平滑概念可以由图 11 中展示的合成影像来说明。图中展示了两个区域: 表示背景干扰的矩形区域 460 和表示目标的近似圆形区域 462。也展示了两条相邻的法线 464 和 466。点 a 和 b 为法线 464 上探测到的边缘点, 而点 c 和 d 为法线 466 上探测到的边缘点。目的是在这两条法线 464 和 466 上寻找轮廓线在何处。除了某些变形以外, 测量结果 482 和 484 相互类似。仅仅根据轮廓平滑约束, 从 a 到 c 的轮廓和从 b 到 c 的轮廓几乎具有相同幅度的平滑能量, 因为  $|a-c| \cong |b-c|$ 。不过, 如果我们还考虑区域平滑约束, 可能的轮廓可以是 ad 或 bc, 而不是 ac 或 bd。HMM 可以进一步根据所有的观测线区分轮廓候选者 ad 和 bc。

为了获得新的转换概率, 要计算所有可能的  $((2N+1)^2$  个) 状态对之间的匹配。图 13 以图形的方式说明了匹配距离的计算。给定了线 464 和 466, 匹配距离的计算可以由以下递归方程说明, 并且可以参见图 13:

$$D^F(i, j) = \min \begin{cases} D^F(i-1, j) + d(\rho_1(i), \rho_2(i)) \\ D^F(i, j-1) + d(\rho_1(i), \rho_2(i)) \\ D^F(i-1, j-1) + d(\rho_1(i), \rho_2(i)) \end{cases}$$

式中  $d(.,.)$  为匹配两个像素的成本。 $D^F(i, j)$  为 464 线上  $[-N, i]$  区段和 466 线上  $[-N, j]$  区段之间的最佳匹配距离。起始于  $D^F(0, j) = D^F(i, 0) = 0$ , 其中  $i, j \in [-N, N]$ , 从  $i = -N$  到  $N$  和  $j = -N$  到  $N$ , 使用以上递归式获得匹配距离  $D^F(i, j)$ 。计算  $D^B(i, j)$  是通过类似的过程, 不过起始于  $D^B(N, N) = 0$ , 到  $D^B(-N, -N)$ 。获得了所有的匹配距离之后, 可以计算出状态转换概率, 并可以通过图 3 中的轮廓选择模块 172 完成轮廓追踪, 下面还要更详细地讨论。

给定了观测结果序列  $O = \{O_\phi, \phi \in [1, M]\}$  和转换概率  $a_{i,j} = p(s_{\phi+1} = j | s_\phi = i)$  之后, 轮廓选择模块 172 通过使用以下众所周知的 Viterbi 算法, 寻找最可能的状态序列  $s^*$ , 确定所发现的最佳轮廓:

$$s^* = \arg \max_s P(s | O) = \arg \max_s P(s, O)$$

值  $V(\phi, \lambda)$  定义如下:

$$V(\phi, \lambda) = \max_{s_{\phi-1}} P(O_{\phi-1}, s_{\phi-1}, s_{\phi} = \lambda)$$

使用马尔可夫条件独立假设, 对于初始值  $V(1, \lambda) = \max_{s_1} P(O_1 | s_1)P(s_1)$ ,  $V(\phi, \lambda)$  能够递归计算如下:

$$V(\phi, \lambda) = P(O_{\phi} | s_{\phi} = \lambda) \cdot \max_j P(s_{\phi} = \lambda | s_{\phi-1} = j) V(j, \phi - 1)$$

$$j^*(\phi, \lambda) = P(O_{\phi} | s_{\phi} = \lambda) \cdot \arg \max_j P(s_{\phi} = \lambda | s_{\phi-1} = j) V(j, \phi - 1)$$

式中初始状态概率  $P(s_1) = 1/(2N+1)$ ,  $s_1 \in [-N, N]$ 。  $j^*(\phi, \lambda)$  项记录了  $\phi$  线上来自  $\lambda$  状态的“最佳以前状态”。所以, 在序列的终点, 就获得了  $\max_s P(O, s) = \max_{\lambda} V(M, \lambda)$ 。从  $s_M^* = \arg \max_{\lambda} V(M, \lambda)$  开始, 以  $s_{\phi-1}^* = j^*(s_{\phi}^*, \phi)$  向后追踪  $j^*$ , 可以获得最优状态序列  $s^*$ 。

给定了最佳状态序列  $s^* = \{s_1^*, \dots, s_M^*\}$ ,  $\phi$  线上最佳轮廓点  $s_{\phi}^*$  对应的影像坐标记为  $[x_{\phi}, y_{\phi}]$ 。因为使用椭圆作为参数轮廓模型, 对于每个轮廓点  $[x_{\phi}, y_{\phi}]$ , 下式成立:

$$ax_{\phi}^2 + by_{\phi}^2 + cx_{\phi}y_{\phi} + dx_{\phi} + ey_{\phi} - 1 = 0$$

这些方程的矩阵表达式为:

$$A \cdot f = b$$

式中

$$A = \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_M^2 & y_M^2 & x_M y_M & x_M & y_M \end{bmatrix}$$

且  $b = [1, 1, \dots, 1]^T$ 。通过最小均方 (LMS) 解可以获得最佳拟合椭圆的参数  $f^* = [a, b, c, d, e]^T$ :

$$f^* = (A^T A)^{-1} A^T b \quad (5)$$

上面的椭圆表达式  $f = [a, b, c, d, e]^T$  在数学上是方便的。但是这 5 个参数却没有清楚的物理解释。在追踪中, 往往使用不同的 5 元椭圆表达式:

$$\theta = [x, y, \alpha, \beta, \Phi]$$

式中  $(x, y)$  为椭圆的中心,  $\alpha$  和  $\beta$  为椭圆长短轴的长度,  $\Phi$  为椭圆的方向。因为  $f$  和  $\theta$  是同一椭圆的两个表达式, 它们在本文中交换地使用。

在动态环境中，被追踪的目标和背景都可能逐渐改变外观。因此，模型调整模块 174 动态地调整观测似然模型。调整观测似然模型的一种方法是，完全相信 Viterbi 算法在  $t-1$  帧返回的轮廓，对该轮廓内外的所有像素求平均，以获得在  $t$  帧新的前景 / 背景颜色模型。不过，如果在  $t-1$  帧发生了错误，这个过程就可能以错误的方式调整模型。因此，模型调整模块 174 以概率统计的方式训练观测模型。

不是完全相信在  $t-1$  帧获得的轮廓，而是使用一种向前 - 向后的算法，来决定如何更新观测模型。“向前概率分布”定义如下：

$$\alpha_\phi(s) = p(O_1, O_2, \dots, O_\phi, s_\phi = s)$$

它能够使用如下递归来计算：

$$\alpha_1(s) = p(s_1 = s)p(O_1 | s_1 = s)$$

$$\alpha_{\phi+1}(s) = \left[ \sum_u \alpha_\phi(u) a_{u,s} \right] p(O_{\phi+1} | s_{\phi+1} = s)$$

同样，“向后概率分布”定义如下：

$$\beta_\phi(s) = p(O_{\phi+1}, O_{\phi+2}, \dots, O_M, s_\phi = s)$$

它能够使用如下递归来计算：

$$\beta_M(s) = 1$$

$$\beta_\phi(s) = \sum_u a_{s,u} p(O_{\phi+1} | s_{\phi+1} = u) \beta_{\phi+1}(u)$$

计算了向前和向后概率之后，我们可以计算  $\phi$  线上每种状态的概率如下：

$$P(s_\phi = s | \mathbf{O}) = \frac{\alpha_\phi(s)\beta_\phi(s)}{\sum_u \alpha_\phi(u)\beta_\phi(u)}, \quad s \in [-N, N]$$

它表示测量线  $\phi$  上  $s$  处具有轮廓点的概率。

根据这些概率，沿着法线结合  $P(s_\phi = s | \mathbf{O})$ ，可以计算像素  $\lambda_\phi$  在前景（或背景）中的概率如下：

$$P(\lambda_\phi \in \text{BG}) = 1 - P(\lambda_\phi \in \text{FG}) = \prod_{s=-N}^{\lambda_\phi} p(s_\phi = s | \mathbf{O})$$

这个概率给我们提供了一种稳健的方法，在调整观测模型期间对不同的像素加权。分类置信度越高的像素对颜色模型的贡献越大，分类置信度越低的像素贡献越小：

$$\begin{aligned}
 p(v | BG) &= \frac{\sum_{s=-N}^N P(s \in BG) \cdot O_\phi(s)}{\sum_{s=-N}^N P(s \in BG)} \\
 p(v | FG) &= \frac{\sum_{s=-N}^N P(s \in FG) \cdot O_\phi(s)}{\sum_{s=-N}^N P(s \in FG)}
 \end{aligned} \tag{6}$$

调整后的新模型反映了追踪期间颜色分布的变化。然后，在下一帧的轮廓搜索期间，把调整后的新模型代回(1)式中。在展示的实例中没有训练转换概率，因为在典型情况下，它们趋于在追踪过程期间保持相对不变。此外，也能够以类似于训练颜色分布的方式训练转换概率。

返回图9，多线索追踪过程400可以进一步参考图14。图14展示了从时间  $t-1$  的一帧522到时间  $t$  的下一帧524追踪一个目标的过程。根据前一帧  $t-1$  中的追踪结果和该目标的动态，预测当前帧  $t$  中目标将在何处(动作402)。沿着预测轮廓的一组法线收集观测结果(动作404)。使用众所周知的 Langevin 过程来模拟人运动的动态：

$$\begin{bmatrix} \theta_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ 0 & a \end{bmatrix} \begin{bmatrix} \theta_{t-1} \\ \dot{\theta}_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} m_t$$

式中  $\theta = [x, y, \alpha, \beta, \phi]$  为参数椭圆， $a = \exp(-\beta_0 \tau)$ ， $b = \bar{v} \sqrt{1-a^2}$ 。  $\beta_0$  为比率常数， $m$  为从高斯分布  $N(0, Q)$  导出的热激励过程， $\tau$  为离散时间步长， $\bar{v}$  为稳态均方根速度。

对于法线  $\phi$  上的每一个像素，使用上面的(2)式，根据法线上每个像素的边缘探测结果和颜色值，求取观测似然函数(动作406)：

$$p(O_\phi | s_\phi = \lambda_\phi), \lambda_\phi \in [-N, N], \phi \in [1, M]$$

同时，也根据 JPDAF 求取状态转换概率(动作408)，如上面(4)式所示。

利用前面计算出的观测似然函数和转换概率矩阵，由 Viterbi 算法寻找对于给定观测结果的最佳轮廓(动作410)，并且使用上面的(6)式，根据探测的轮廓拟合最佳椭圆(动作412)。

然后，使用向前-向后算法估计法线上每个像素(向前和向后)的软分类，根据上面的(6)式，更新前景和背景的颜色模型(动作414)。

对于视频内容中的每一帧，重复图 9 中的过程 400。

多线索追踪模块 144 试图考虑线索的可靠性和线索可靠性的变化。例如，前景和背景的性质都进行模拟（参见上面的（1）式），在上面的（2）式中使用该模型来探测边界（例如，假若前景的颜色与背景类似，它将不会对边界探测作出很大贡献，该过程将更多地依靠更容易判别的其它线索，比如运动）。在追踪期间，也要调制由上面（6）式表达的背景和前景的模型。

对上面讨论的多线索追踪过程，还可以进行多种修改。根据一种修改方案，被追踪面孔的一个或多个特性点组成一组，得到保持，分析新的每一帧以确定组内特性点的位置。一旦组内的特性点定位之后，根据已定位的组内各点，就能够粗略估计面孔的位置，然后这种粗略的估计结果会用作上面讨论的参数轮廓追踪过程的初始猜测。换句话说，分析新的一帧以便为参数轮廓追踪过程定位一个初始猜测，而不是依靠上面讨论的预测位置。在相邻帧之间目标移动距离大（大到上面讨论的预测位置可能与后续帧中的实际轮廓位置不够接近）的情况下，这种修改尤其有用。

可以追踪多种不同的特性点，比如眼角、嘴角、鼻孔等等。还可以追踪音频中的声源作为特性点，以补充或取代视觉特性。能够使用多种特性的追踪过程，比如众所周知的 Lucas-Kanade 特性追踪器。有关 Lucas-Kanade 特性追踪器的其它信息，可以参见 J. Shi and C.Tomasi, "Good Features to Track," IEEE Conf. On Computer Vision and Pattern Recognition, pp.593-600, 1994。

对上面讨论的多线索追踪过程可以进行的另一种修改是，执行概率统计的采样时，从特性点（探测的轮廓点）采样，而不是从状态空间采样。例如，从探测的所有轮廓点中可以采取几个轮廓点，在采用的轮廓点上拟合参数形状。

对多线索追踪过程可以进行的另一种修改是，对面孔追踪多个可能的位置——换句话说，追踪多个假设而不是一个假设。可以使用微粒滤波技术来保持多个技术，所以勉强的假设不被立即排除。相反，



保持勉强的假设，使它们有时间来证明是正确的选择。下一步介绍一种这样的微粒滤波器技术，称为无气味的微粒滤波器。

使用无气味卡尔曼滤波器 (UKF) 的无气味微粒滤波器 (UPF)，被多线索追踪模块 144 用于追踪多个假设。无气味的变换 (UT) 用于计算  $g()$  的泰勒级数展开式中高至第二阶 (第三阶为高斯先验项) 的均值和协方差。令  $n_x$  为  $x$  的维数， $\bar{x}$  为  $x$  的均值， $P_x$  为  $x$  的协方差，UT 计算  $y=g(x)$  的均值和协方差如下：

首先，确定性地产生  $2n_x + 1$  个求和点  $S_i=\{X_i, W_i\}$ ：

$$X_0 = \bar{x}$$

$$X_i = \bar{x} + \left(\sqrt{(n_x + \lambda)P_x}\right)_i \quad i = 1, \dots, n_x$$

$$X_i = \bar{x} - \left(\sqrt{(n_x + \lambda)P_x}\right)_i \quad i = n_x + 1, \dots, 2n_x \quad (7)$$

$$W_0^{(m)} = \lambda / (n_x + \lambda), \quad W_0^{(c)} = W_0^{(m)} + (1 - \alpha^2 + \beta)$$

$$W_i^{(m)} = W_i^{(c)} = 1 / (2 \cdot (n_x + \lambda)) \quad i = 1, \dots, 2n_x$$

$$\lambda = \alpha^2(n_x + \kappa) - n_x$$

式中  $\kappa$  为比例参数，它控制着求和点之间的距离和均值  $\bar{x}$ ， $\alpha$  为比例参数，它控制着非线性函数  $g()$  产生的高阶效应， $\beta$  为一个参数，控制着第 0 个求和点的加权， $(\sqrt{(n_x + \lambda)P_x})_i$  为矩阵第  $i$  列的平方根。在一个实施方案中，对于标量情况， $\alpha=1$ ， $\beta=2$ ， $\kappa=2$ 。注意，计算均值和协方差时，第 0 个求和点的加权不同。

然后，求和点通过非线性变换传播：

$$Y_i = g(X_i) \quad i = 0, \dots, 2n_x \quad (8)$$

$y$  的均值和协方差按下式计算：

$$\bar{y} = \sum_{i=0}^{2n_x} W_i^{(m)} Y_i, \quad P_y = \sum_{i=0}^{2n_x} W_i^{(c)} (Y_i - \bar{y})(Y_i - \bar{y})^T \quad (9)$$

$y$  的均值和协方差准确至泰勒级数展开的第二阶。

使用 UT，将状态空间扩展到包括噪音分量： $x_t^a = [x_t^T \quad m_t^T \quad n_t^T]^T$ ，可以实现无气味的卡尔曼滤波器。令  $N_a = N_x + N_m + N_n$  为扩展后状态空间的维数，其中  $N_m$  和  $N_n$  为噪音  $m_t$  和  $n_t$  的维数， $Q$  和  $R$  为噪音  $m_t$  和  $n_t$  的协方差，UKF 可以归纳如下：

初始化：

$$\bar{x}_0^a = [\bar{x}_0^T \quad 0 \quad 0]^T, \quad P_0^a = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix} \quad (10)$$

对于每个时间事件，迭代下列步骤：

a) 使用上面的 (7) 式中的过程计算求和点：

$$X_{t-1}^a = [\bar{x}_{t-1}^a \quad \bar{x}_{t-1}^a \pm \sqrt{(n_a + \lambda)P_{t-1}^a}] \quad (11)$$

b) 时间更新：

$$X_{t|t-1}^x = f(X_{t-1}^x, X_{t-1}^v), \quad \bar{x}_{t|t-1} = \sum_{i=0}^{2n_a} W_i^{(m)} X_{i,t|t-1}^x \quad (12)$$

$$Y_{t|t-1} = h(X_{t|t-1}^x, X_{t-1}^n), \quad \bar{y}_{t|t-1} = \sum_{i=0}^{2n_a} W_i^{(m)} Y_{i,t|t-1}^x \quad (13)$$

$$P_{t|t-1} = \sum_{i=0}^{2n_a} W_i^{(c)} [X_{i,t|t-1}^x - \bar{x}_{t|t-1}] [X_{i,t|t-1}^x - \bar{x}_{t|t-1}]^T \quad (14)$$

c) 测量更新：

$$P_{y_t y_t} = \sum_{i=0}^{2n_a} W_i^{(c)} [Y_{i,t|t-1} - \bar{y}_{t|t-1}] [Y_{i,t|t-1} - \bar{y}_{t|t-1}]^T \quad (15)$$

$$P_{x_t y_t} = \sum_{i=0}^{2n_a} W_i^{(c)} [X_{i,t|t-1}^x - \bar{x}_{t|t-1}] [Y_{i,t|t-1} - \bar{y}_{t|t-1}]^T \quad (16)$$

$$K_t = P_{x_t y_t} P_{y_t y_t}^{-1} \quad (17)$$

$$\bar{x}_t = \bar{x}_{t|t-1} + K_t (y_t - \bar{y}_{t|t-1}), \quad P_t = P_{t|t-1} - K_t P_{y_t y_t} K_t^T \quad (18)$$

利用 UKF，最新的观测结果能够方便地加入状态估计中（例如上面的测量更新 c））；不过，它对状态分布进行了高斯假设。相反，微粒滤波器能够模拟任意的分布，但是却难以将新的观测结果  $y_t$  加入建议的分布。UKF 用于为微粒滤波器产生建议的分布，得出混合的 UPF。确切地说，对于每个微粒，建议的分布如下：

$$q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t}) = N(\bar{x}_t^{(i)}, P_t^{(i)}), \quad i = 1, \dots, N \quad (19)$$

式中  $\bar{x}_t$  和  $P_t$  为  $x$  的均值和协方差，是使用 UKF ((10)-(18)式) 计算出的。应当注意，尽管高斯假设并非真实地近似后验分布  $p(x_t | x_{t-1}, y_{0:t})$ ，对于明确的  $\bar{x}_t$  和  $P_t$ ，产生各个微粒也不是一个大问题。不仅如此，因为 UKF 近似后验的均值和协方差高至第二阶，所以很好地保存了系统的非线性性质。通过将 UKF 步骤和 (19) 式插入一般的微粒滤波器算法，很容易获得 UPF 过程。

图 15 是一个流程图，展示了一个示范的 UPF 过程 550。图 15 中

的过程是由图 3 中的多线索追踪模块 144 执行，并且可以在软件中执行。

一开始，使用(11)-(18)式利用 UKF 更新微粒  $x_t^{(i)}$ ， $i=1, \dots, N$ ，获得  $\bar{x}_t^{(i)}$  和  $P_t^{(i)}$ （动作 552）。然后，从建议的分布  $q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t}) = N(\bar{x}_t^{(i)}, P_t^{(i)})$  中，对微粒  $x_t^{(i)}$ ， $i=1, \dots, N$  进行采样（动作 554）。然后使用如下的（20）式计算微粒加权（动作 556）：

$$\begin{aligned} \tilde{w}_t^{(i)} &= \frac{p(y_{1:t} | x_{0:t}^{(i)})p(x_{0:t}^{(i)})}{q(x_{0:t-1}^{(i)} | y_{1:t-1})q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})} \\ &= \tilde{w}_{t-1}^{(i)} \frac{p(y_{1:t} | x_{0:t}^{(i)})p(x_{0:t}^{(i)})}{p(y_{1:t-1} | x_{0:t-1}^{(i)})p(x_{0:t-1}^{(i)})q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})} \quad (20) \\ &= \tilde{w}_{t-1}^{(i)} \frac{p(y_t | x_t^{(i)})p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})} \end{aligned}$$

然后，使用如下的（21）式对重要性加权进行归一化（动作 558）：

$$w_t(x_{0:t}^{(i)}) = \tilde{w}_t(x_{0:t}^{(i)}) / \sum_{i=1}^N \tilde{w}_t(x_{0:t}^{(i)}) \quad (21)$$

式中微粒  $\{x_{0:t}^{(i)}, w_t(x_{0:t}^{(i)})\}$  是取自已知分布  $q$ ， $\tilde{w}_t(x_{0:t}^{(i)})$  和  $w_t(x_{0:t}^{(i)})$  是未归一化的和归一化的重要性加权。

然后，使用如下的（22）式确定有效微粒尺寸  $S$ （动作 560）：

$$\tilde{w}_t^{(i)} = \tilde{w}_{t-1}^{(i)} \frac{p(y_t | x_t^{(i)})p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})} = \tilde{w}_{t-1}^{(i)} p(y_t | x_t^{(i)}) \quad (22)$$

如果  $S < S_T$ ，那么乘（或者压制）加权后的微粒，以产生  $N$  个等权的微粒（动作 562）。然后，使用如下的（23）式计算  $g()$  的期望（动作 564）：

$$E_p(g(x_{0:t})) = \lim_{N \rightarrow \infty} \sum_{i=1}^N g(x_{0:t}^{(i)}) w_t(x_{0:t}^{(i)}) \quad (23)$$

利用  $g_t(x_t) = x_t$  可以计算  $x_t$  的条件均值，利用  $g_t(x_t) = x_t x_t^T$  可以计算  $x_t$  的条件协方差。

现在讨论根据音频，使用图 15 中的 UPF 过程 550 来追踪与会者。典型情况下，两个话筒就足以估计水平摇摄角。本文中讨论根据水平摇摄角的追踪，也可以进行类似的操作来根据发言者的垂直俯仰角进行追踪。图 16 展示了一种示范的多话筒环境。在图 16 中，假设两个话筒处在位置 A 和 B，声源处在位置 C。当声源的距离（即  $|OC|$ ）远大于话筒对基线  $|AB|$  时，摇摄角  $\theta = \angle COX$  可以估计如下：

$$\theta = \angle COX \approx \angle BAE = \arcsin \frac{|BE|}{|AB|} = \arcsin \frac{D \times v}{|AB|} \quad (24)$$

式中  $D$  为两个话筒之间的时间延迟,  $v=342 \text{ m/s}$  为声音在空气中传播的速度。

为了在追踪应用中使用 UPF 框架, 首先要建立 4 个实体: 在 (12) 式中使用的系统动态模型  $x_t=f(x_{t-1}, m_{t-1})$ , 在 (13) 式中使用的系统观测模型  $y_t=h(x_t, n_t)$ , 在 (22) 式中使用的似然模型  $p(y_t|x_t)$ , 以及在 (18) 式中使用的发明的模型  $y_t - \bar{y}_{t-1}$ 。一旦这 4 个实体建立之后, 就直截了当地使用图 15 中的 UPF 过程 550 进行追踪。

系统动态模型  $x_t=f(x_{t-1}, m_{t-1})$  确定如下。令  $x = [\theta, \dot{\theta}]^T$  为状态空间, 式中它们分别是摇摄角和摇摄角的速度。为了模拟一个发言者的运动动态, 使用众所周知的 Langevin 过程  $d^2\theta/dt^2 + \beta_\theta \cdot d\theta/dt = m$ , 其离散形式为:

$$\begin{bmatrix} \theta_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} 1 & \tau \\ 0 & a \end{bmatrix} \begin{bmatrix} \theta_{t-1} \\ \dot{\theta}_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix} m_t \quad (25)$$

$$a = \exp(-\beta_\theta \tau), \quad b = \bar{v} \sqrt{1-a^2}$$

式中  $\beta_\theta$  为比率常数,  $m$  为从  $N(0, Q)$  导出的热激励过程,  $\tau$  为离散时间步长,  $\bar{v}$  为稳态均方根速度。

系统观测模型  $y_t=h(x_t, n_t)$  确定如下。系统观测结果  $y_t$  为时间延迟  $D_t$ 。根据上面的 (24) 式, 观测结果与状态的关系为

$$y_t = D_t = h(\theta_t, n_t) = |AB| v \sin \theta_t + n_t \quad (26)$$

式中  $n_t$  为测量噪音, 服从  $N(0, R)$  的高斯分布。

似然模型  $p(y_t|x_t)$  确定如下。令  $J$  为 GCCF (广义互相关函数) 中峰值的数目。在  $J$  个峰值位置中, 最多只有一个是来自真正的声源。所以, 可以定义  $J+1$  个假设:

$$\begin{aligned} H_0 &= \{c_j = C : j = 1, \dots, J\} \\ H_j &= \{c_j = T, c_k = C : k = 1, \dots, J, k \neq j\} \end{aligned} \quad (27)$$

式中  $c_j=T$  表明第  $j$  个峰值与真正的声源相关联,  $c_j=C$  表明其它情况。所以, 假设  $H_0$  表明, 没有一个峰值与真正的声源相关联。所以, 结合的似然模型为:

$$\begin{aligned} p(y_t | x_t) &= \pi_0 p(y_t | H_0) + \sum_{j=1}^J \pi_j p(y_t | H_j) \\ &= \pi_0 U + N_m \sum_{j=1}^J \pi_j N(D_j, \sigma_D) \end{aligned} \quad (28)$$

$$\text{s.t. } \pi_0 + \sum_{j=1}^J \pi_j = 1$$

式中  $\pi_0$  为假设  $H_0$  的先验概率, 从第  $j$  个峰值的相对高度可以获得  $\pi_j$ ,  $j=1, \dots, J$ ,  $N_m$  为归一化因子,  $D_j$  为第  $j$  个峰值相应的时间延迟,  $U$  表示均匀分布,  $N()$  表示高斯分布。

新发明的模型  $y_t - \bar{y}_{\text{qt-1}}$  确定如下。与似然模型相同, 新发明的模型也需要考虑多峰的事实:

$$y_t - \bar{y}_{\text{qt-1}} = \sum_{j=1}^J \pi_j (D_j - \bar{y}_{\text{qt-1}}) \quad (29)$$

式中  $\bar{y}_{\text{qt-1}}$  为预测的侧 UKF 获得的测量结果 (见上面的 (18) 式)。

根据视频, 使用图 15 中的 UPF 过程 550 来追踪与会者, 类似于根据音频来追踪与会者。为了在追踪应用中使用 UPF 框架, 首先要建立 4 个实体: 系统动态模型  $x_t = f(x_{t-1}, m_{t-1})$ , 系统观测模型  $y_t = h(x_t, n_t)$ , 似然模型  $p(y_t | x_t)$ , 以及新发明的模型  $y_t - \bar{y}_{\text{qt-1}}$ 。一旦这 4 个实体建立之后, 就直截了当地使用图 15 中的 UPF 过程 550 进行追踪。

系统动态模型  $x_t = f(x_{t-1}, m_{t-1})$  确定如下。令  $(r, s)$  表示影像坐标。在基于轮廓的追踪中, 系统状态为椭圆中心的位置及其水平和垂直速度, 即  $x_t = [r_t, s_t, \dot{r}_t, \dot{s}_t]^T$ 。类似于音频数据的系统动态模型, 使用众所周知的 Langevin 过程来模拟人运动的动态:

$$\begin{bmatrix} r_t \\ s_t \\ \dot{r}_t \\ \dot{s}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & \tau & 0 \\ 0 & 1 & 0 & \tau \\ 0 & 0 & a_r & 0 \\ 0 & 0 & 0 & a_s \end{bmatrix} \begin{bmatrix} r_{t-1} \\ s_{t-1} \\ \dot{r}_{t-1} \\ \dot{s}_{t-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ b_r \\ b_s \end{bmatrix} m_t \quad (30)$$

系统观测模型  $y_t = h(x_t, n_t)$  确定如下。椭圆的中心处在当前状态位置  $(r_t, s_t)$ 。从椭圆的中心产生  $K$  条射线, 与椭圆边界相交。椭圆中心用作局部坐标系的原点, 所以要获得交点  $(u_k, v_k)$ ,  $k=1, 2, \dots, K$ ,

$$\begin{aligned} u_k &= \sqrt{\tan^2 \phi_k / (1.44 \tan^2 \phi_k + 1)} \\ v_k &= \sqrt{1 / (1.44 \tan^2 \phi_k + 1)} \end{aligned} \quad (31)$$

可以通过联合求解椭圆方程和射线方程:

$$\begin{cases} \frac{u_k^2}{1} + \frac{v_k^2}{1.2^2} = 1 \\ u_k = v_k \tan(\phi_k) \end{cases} \quad (32)$$

将局部 $(u, v)$ 坐标变换回影像坐标就获得了以下观测结果:

$$\begin{aligned} y_t &= h(x_t, n_t) \\ &= [(u_k + r_t, v_k + s_t)] + n_t, \quad k=1,2,\dots,K \end{aligned} \quad (33)$$

式中  $n_t$  为测量噪音, 服从  $N(0, R)$  的高斯分布。应当注意, 观测模型是高度非线性的。

似然模型  $p(y_t|x_t)$  确定如下。沿着  $K$  条射线中的每一条, 使用众所周知的 Canny 边缘探测器来计算边缘强度。结果函数是一个多峰函数, 正如音频数据似然模型中的 GCCF。多个峰值表示沿着这条射线有多个边缘候选者。令峰值的数目为  $J$ , 我们可以使用与为了音频数据开发的似然模型同样的似然模型, 来模拟沿着射线  $k$  的边缘似然:

$$\begin{aligned} p^{(k)}(y_t|x_t) &= \pi_{k0} p^{(k)}(y_t|H_0) + \sum_{j=1}^J \pi_{kj} p^{(k)}(y_t|H_j) \\ &= \pi_{k0} U + N_m \sum_{j=1}^J \pi_{kj} N((u_k, v_k)_j, \sigma_{kj}) \end{aligned}$$

考虑到所有的  $K$  条射线, 所以整体似然为:

$$p(y_t|x_t) = \prod_{k=1}^K p^{(k)}(y_t|x_t) \quad (34)$$

新发明的模型  $y_t - \bar{y}_{t-1}$  确定如下。与似然模型相同, 新发明的模型也需要考虑多峰的事实:

$$y_t^{(k)} - \bar{y}_{t-1}^{(k)} = \sum_{j=1}^J \pi_{kj} ((u_k, v_k)_{t,j} - (u_k, v_k)_{t-1})$$

式中  $k=1,2,\dots,K$ ,  $\pi_{kj}$  为沿着射线  $k$  对第  $j$  个峰值的混合加权, 可以从对应的边缘强度获得。

### 通用计算机环境

图 17 展示了一种通用计算机环境 600, 它可以用于实施本文中介绍的、多个人的自动探测和追踪。计算机环境 600 仅仅是计算环境的一个实例, 并非试图建议对计算机和网络体系结构之使用范畴或功能的任何限制。计算机环境 600 也不应当被解释为对示范计算机环境 600 中展示的组件中任何一个或组合具有任何的依赖或需要。

计算机环境 600 包括计算机 602 形式的通用计算设备。例如, 计算机 602 可以是图 1 中的系统 102、图 2 中的系统 112、图 3 中的系统 130 等等。计算机 602 的组件可以包括但是不限于一个或多个处理器

或者说处理单元 604、一个系统存储器 606 和一条系统总线 608——它将多种系统组件，包括处理器 604 连接到系统存储器 606。

系统总线 608 表示几种总线结构中任何一个或多个，包括存储器总线或者说存储器控制器、外围总线、加速图形端口以及处理器或者使用多种总线体系结构中任何一种的局部总线。例如，这种体系结构可以包括工业标准结构 (ISA) 总线、微通道结构 (MCA) 总线、增强的 ISA (EISA) 总线、视频电子标准协会 (VESA) 局部总线以及外围部件互连 (PCI) 总线——也被称为中楼总线。

典型情况下，计算机 602 包括多种计算机可读的介质。这种介质可以是任何可用的介质——计算机 602 可存取，并包括易失性的和非易失性的、可拆卸的和不可拆卸的介质。

系统存储器 606 包括易失性存储器形式的计算机可读的介质，比如随机存取存储器 (RAM) 610，和 / 或非易失性存储器，比如只读存储器 (ROM) 612。在 ROM 612 中存放着基本输入 / 输出系统 (BIOS) 614，包含着基本的例程，有助于在计算机 602 之内的元件之间传递信息，比如在启动期间。典型情况下，RAM 610 包含着处理单元 604 可立即存取和 / 或正在操作的数据和 / 或程序模块。

计算机 602 也可以包括其它可拆卸的 / 不可拆卸的、易失性的 / 非易失性的计算机存储介质。例如，图 17 展示了硬盘驱动器 616——用于在不可拆卸的、非易失性的磁介质 (未显示) 上读取和写入，磁盘驱动器 618——用于在可拆卸的、非易失性的磁盘 620 (如“软盘”) 上读取和写入，以及光盘驱动器 622——用于在可拆卸的、非易失性的光盘 624 比如 CD-ROM、DVD-ROM 或其它光存储介质上读取和 / 或写入。硬盘驱动器 616、磁盘驱动器 618 和光盘驱动器 622 都通过一个或多个数据介质接口 626，连接到系统总线 608。此外，硬盘驱动器 616、磁盘驱动器 618 和光盘驱动器 622 也可以通过一个或多个接口 (未显示) 连接到系统总线 608。

这些盘片驱动器和它们相关联的计算机可读介质，为计算机 602 所用的计算机可读的指令、数据结构、程序模块和其它数据提供了非

易失性的存储。尽管该实例展示了硬盘 616、可拆卸的磁盘 620 和可拆卸的光盘 624，应当承认，也可以采用能够存放计算机可存取之数据的其它类型的计算机可读介质，来实现示范的计算系统和环境，比如磁带或其它磁存储设备、闪存卡、CD-ROM、数字万能盘（DVD）或其它光存储介质、随机存取存储器（RAM）、只读存储器（ROM）、可电擦除的可编程只读存储器（EEPROM）等等。

在硬盘 616、磁盘 620、光盘 624、ROM 612 和 / 或 RAM 610 中，可用存放任何数目的程序模块，包括例如操作系统 626、一个或多个应用程序 628、其它程序模块 630 和程序数据 632。这种操作系统 626、一个或多个应用程序 628、其它程序模块 630 和程序数据 632 中的任何一种（或者它们的某些组合），都可以成为支持分布式文件系统的全部或部分驻留组件。

用户可以通过输入设备比如键盘 634 和定点设备 636 如“鼠标”，向计算机 602 输入命令和信息。其它输入设备 638（未确切显示）可能包括话筒、操纵杆、游戏垫板、卫星天线、串口、扫描仪等等。这些和其它输入设备通过连接到系统总线 608 的输入 / 输出接口 640，连接到处理单元 604，但是也可以通过其它接口和总线结构连接，比如并口、游戏端口或者通用串行总线（USB）。

监视器 642 或其它类型的显示设备也可以通过接口——比如视频适配器 644——连接到系统总线 608。除了监视器 642，其它的外围输出设备可以包括能够通过输入 / 输出接口 640 连接到计算机 602 的部件，比如扬声器（未显示）和打印机 646。

计算机 602 可以运行在网络化环境中，它使用一台或多台远程计算机的逻辑连接，比如远程计算设备 648。例如，远程计算设备 648 可能是一台个人计算机、便携计算机、服务器、路由器、网络计算机、对等设备或者其它普通网络节点等等。远程计算设备 648 展示为便携计算机，相对于计算机 602，它可以包括本文中介绍的许多或全部要素。

在计算机 602 和远程计算机 648 之间的逻辑连接，描述为局域网



(LAN) 650 和一般的广域网 (WAN) 652。这种联网环境在办公室、企业计算机网络、内联网和因特网中是司空见惯的。

在 LAN 联网环境中实现时,计算机 602 通过网络接口或者说适配器 654, 连接到局域网 650。在 WAN 联网环境中实现时, 典型情况下计算机 602 包括调制解调器 656 或其它装置, 以便建立通过广域网的通信调制解调器 656——它可以在计算机 602 的内部或外部——可以通过输入/输出接口 640 或其它适当的机制, 连接到系统总线 608。应当承认, 展示的网络连接是示范性的, 在计算机 602 和 648 之间, 也可以采用其它方式建立通信连接。

在网络化的环境中, 比如计算环境 600 展示的环境, 相对于计算机 602 或者其中部分所描述的程序模块, 可能存放在远程存储器的存储设备中。例如, 远程应用程序 658 驻留在远程计算机 648 的存储器设备中。为了便于展示, 应用程序和其它可执行程序组件比如操纵系统, 在本文中展示为离散的块, 尽管应当理解这些程序和组件在不同的时间驻留在计算设备 602 的不同的存储部件中, 并被该计算机的数据处理器执行。

分布式文件系统 150 的实施方案, 可以在计算机可执行指令——比如程序模块, 由一台或多台计算机或其它设备执行——的一般环境中介绍。一般说来, 程序模块包括例程、程序、对象、组件、数据结构等等, 它们执行特定的任务或者实现特定的抽象数据类型。典型情况下, 程序模块的功能可以结合或分配, 如多个实施例中的介绍。

对于加密的文件, 其文件格式的实施方案可以存放在或者传送到某种形式的计算机可读的介质上。计算机可读的介质可以是计算机可以存取的、任何可用的介质。计算机可读的介质可能包括例如但是不限于“计算机存储介质”和“通信媒介”。

“计算机存储介质”包括以任何方法或技术实现的、用于信息——比如计算机可读的指令、数据结构、程序模块或其它数据——存储的易失性的和非易失性的、可拆卸的和不可拆卸的介质。计算机存储介质包括但是不限于 RAM、ROM、EEPROM、闪存或其它存储器技术、

CD-ROM、数字万能盘(DVD)或其它光存储介质、盒式磁带、磁带、磁盘存储或其它磁存储设备,或者能够用于存放所需信息并且可由计算机存取的任何其它介质。

“通信媒介”典型情况下使计算机可读的指令、数据结构、程序模块或其它数据具体化为调制后的数据信号,比如载波或其它传输机制。通信媒介也包括任何信息传递媒介。术语“调制后的数据信号”表明的信号使其特征的一个或多个,以在信号中编码信息的方式设置或改变。通信媒介包括例如但是不限于有线媒介比如有线网络或直接有线连接,以及无须媒介比如声波、RF、红外线和其它无须媒介。在计算机可读媒介的范畴之内,也包括以上类型的任何组合。

尽管本文主要讨论涉及人的面孔,但是类似于本文中讨论的人的面孔,其它目标也可以自动探测和/或追踪。

### 结论

虽然上面的介绍使用的语言,对结构特性和/或方法动作才是具体的,应当理解,在附带的权利要求书中规定的本发明不限于介绍的具体特性和动作。相反,具体的特性和动作是公开为本发明实施的示范形式。

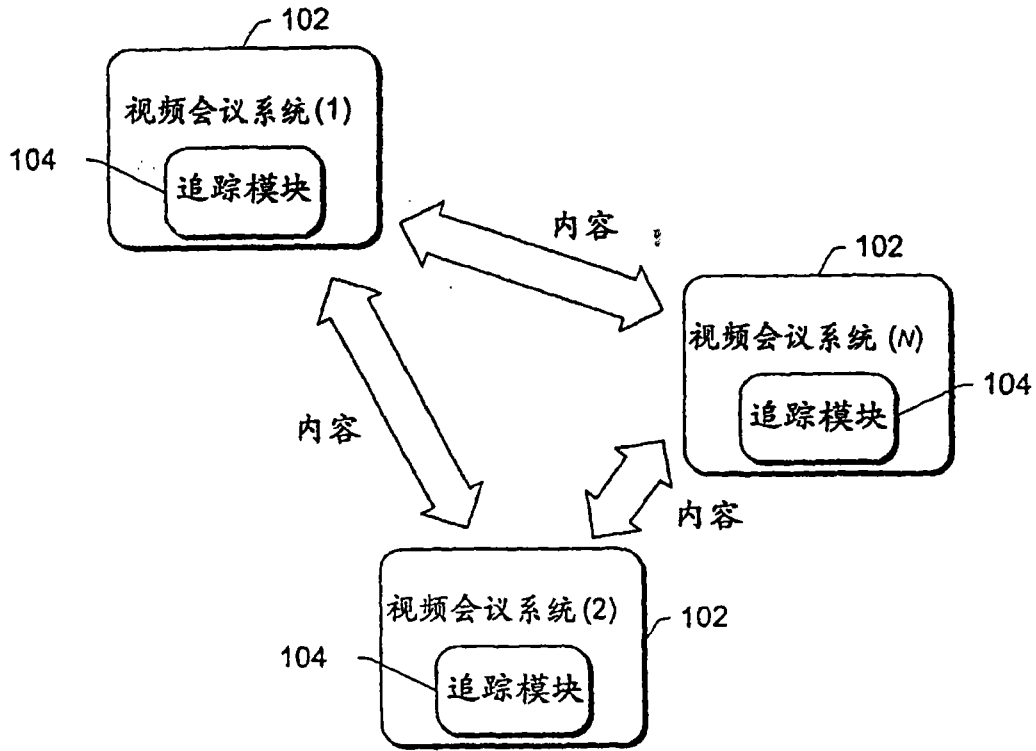


图1

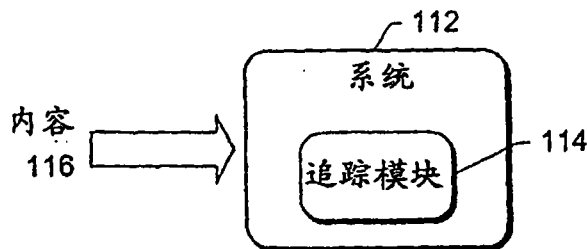


图2

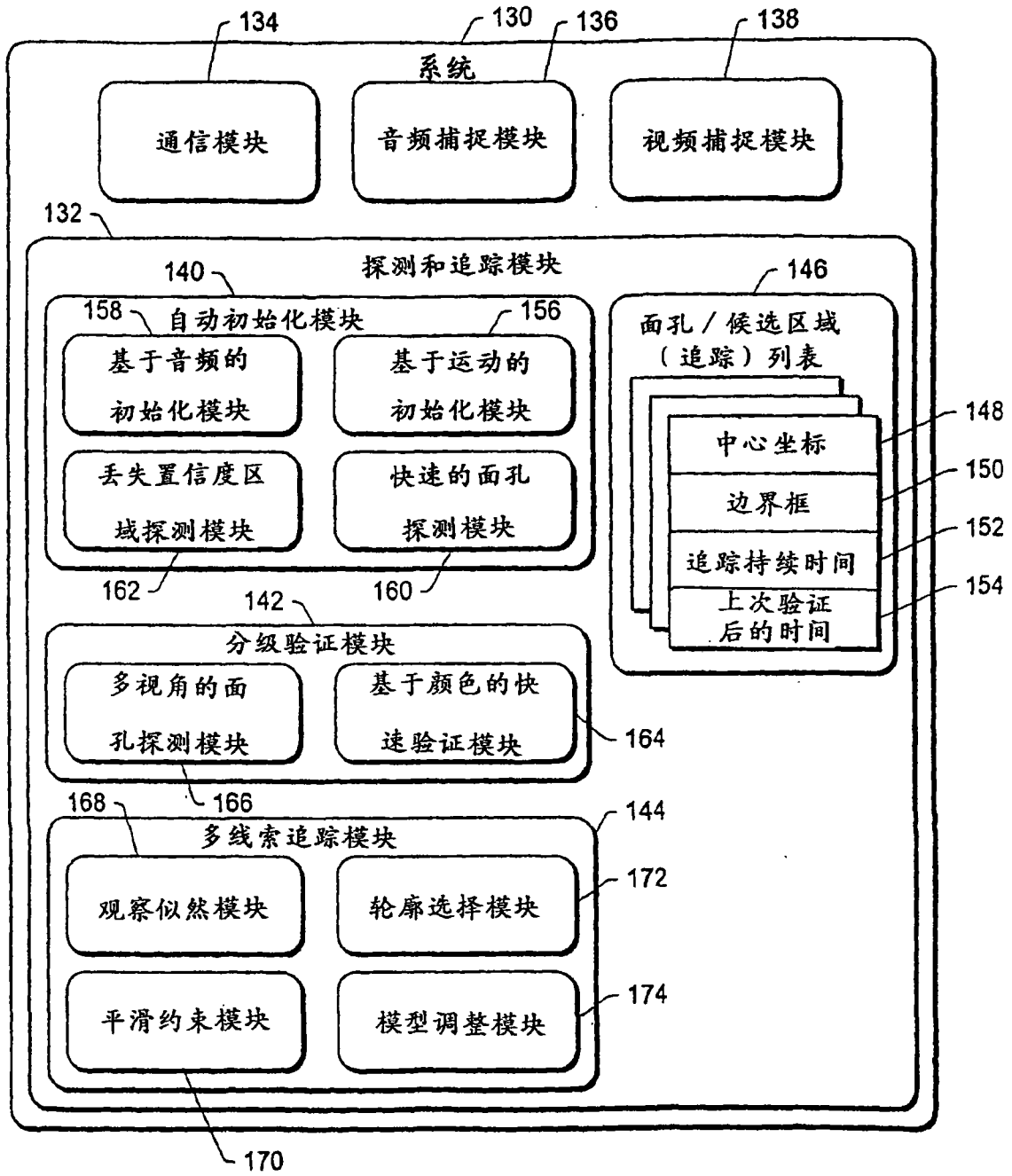


图3

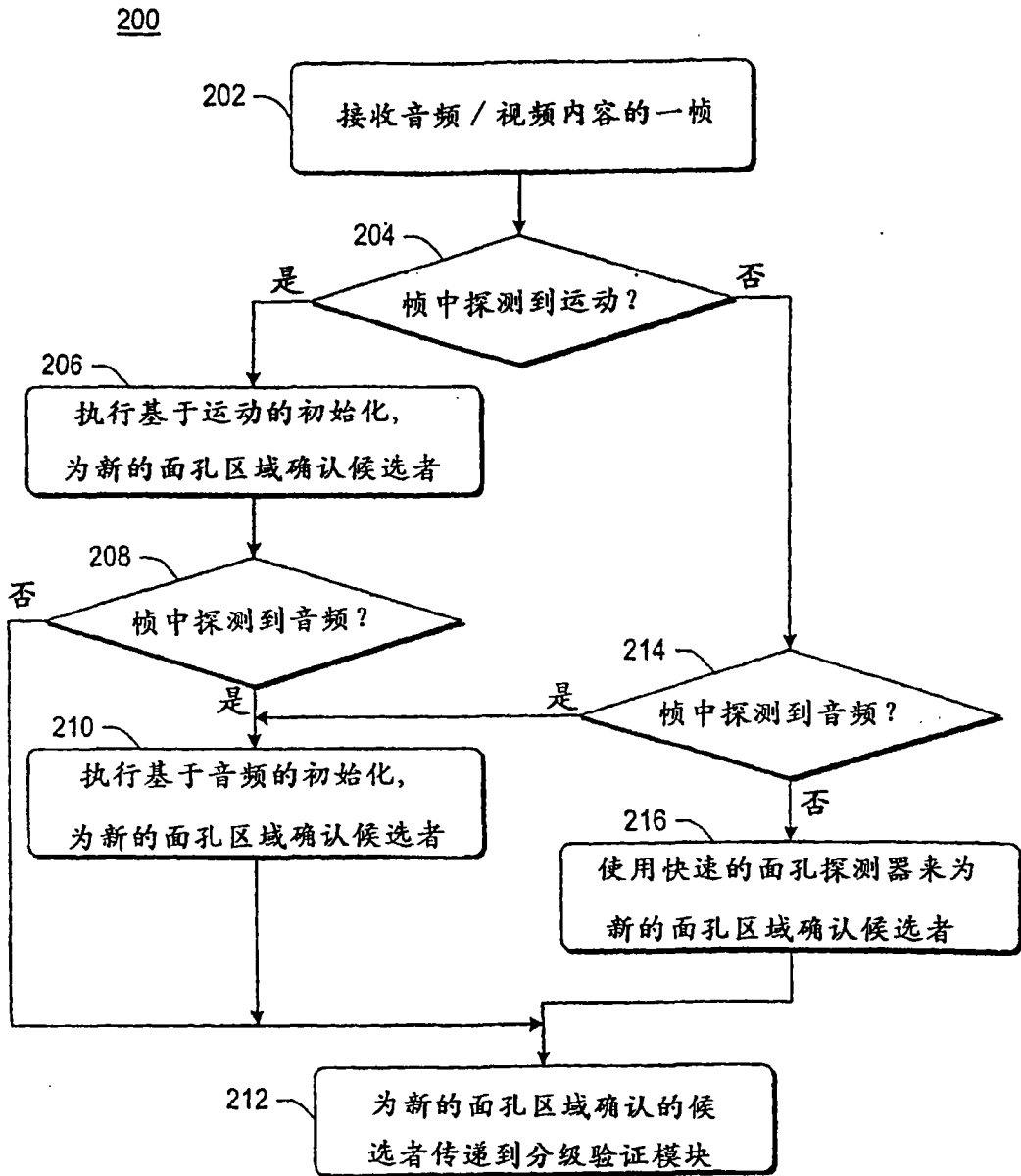


图 4

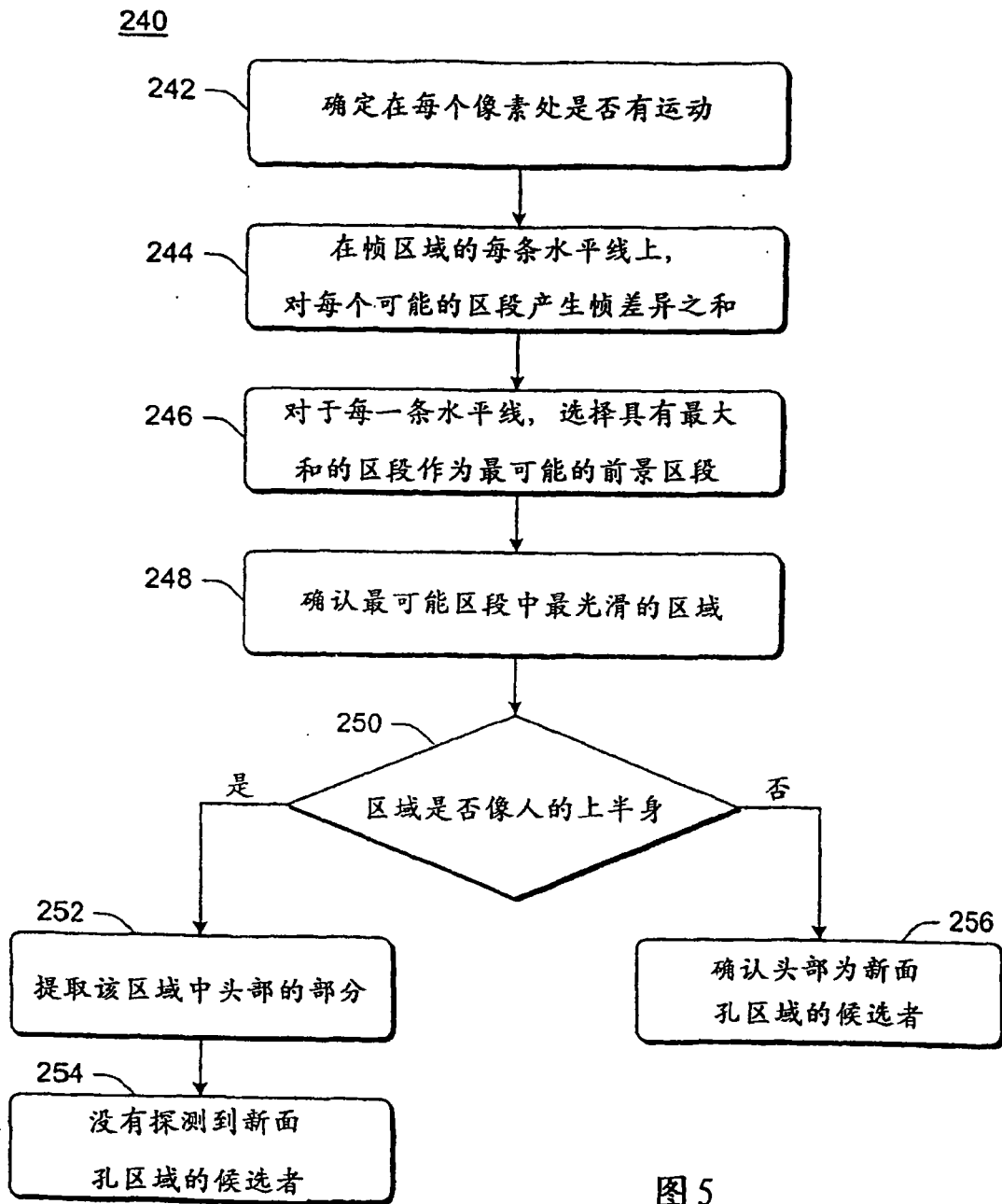


图5

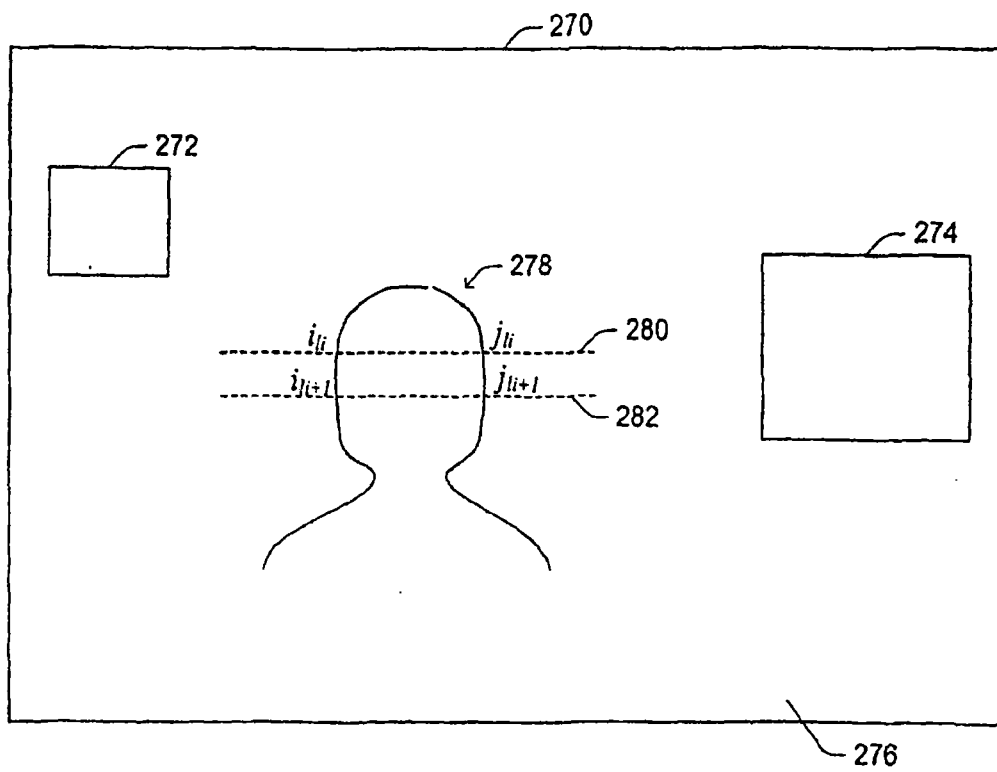


图 6

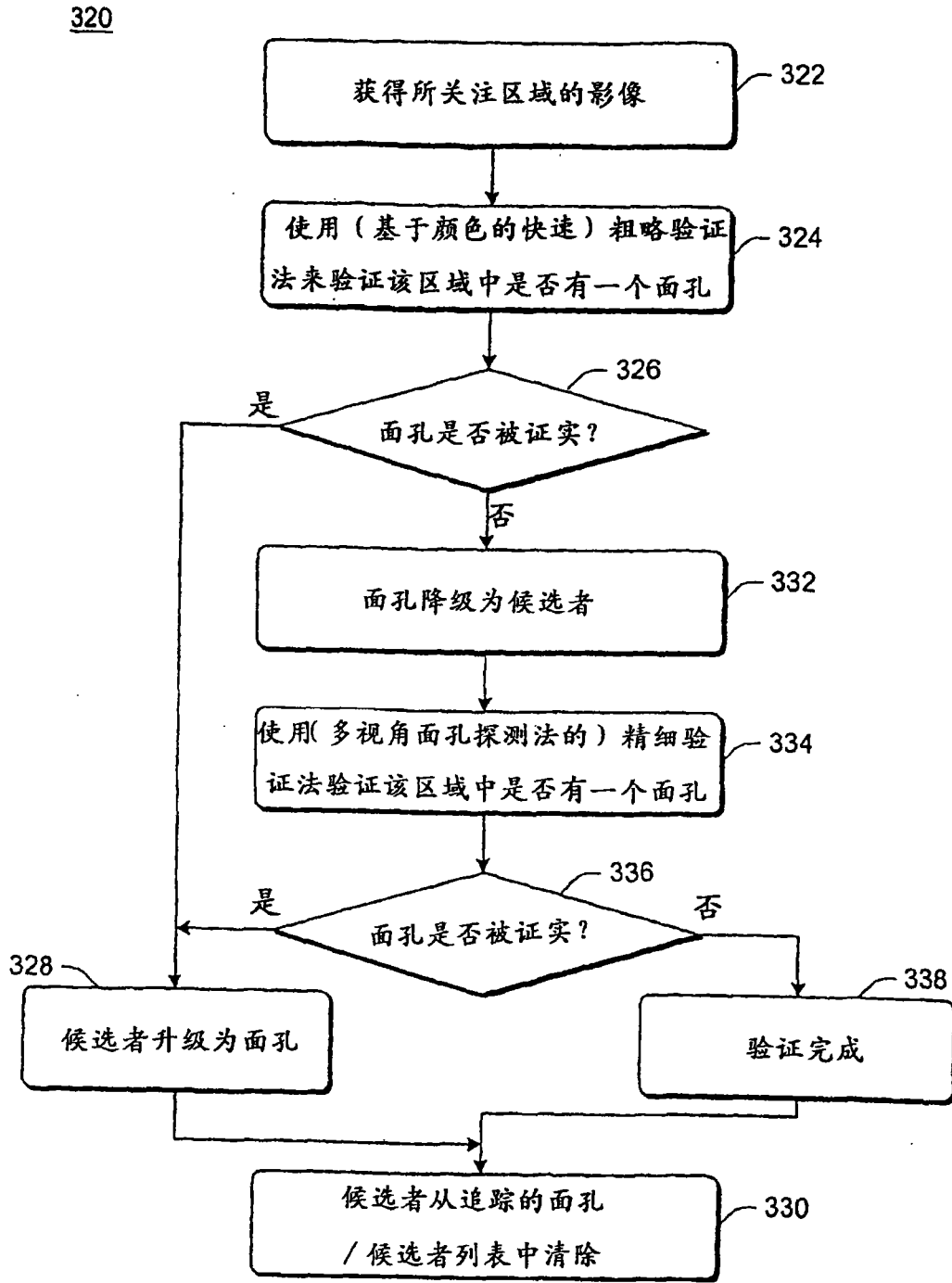


图7



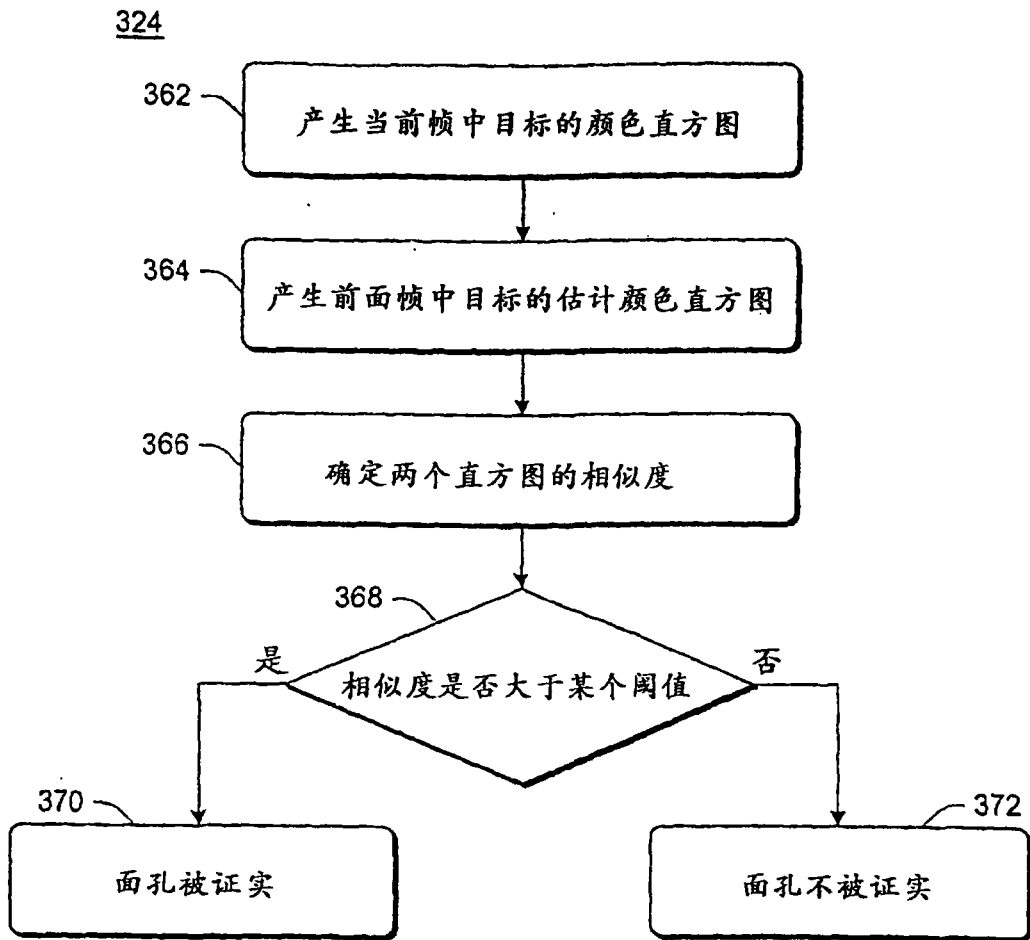


图 8

400

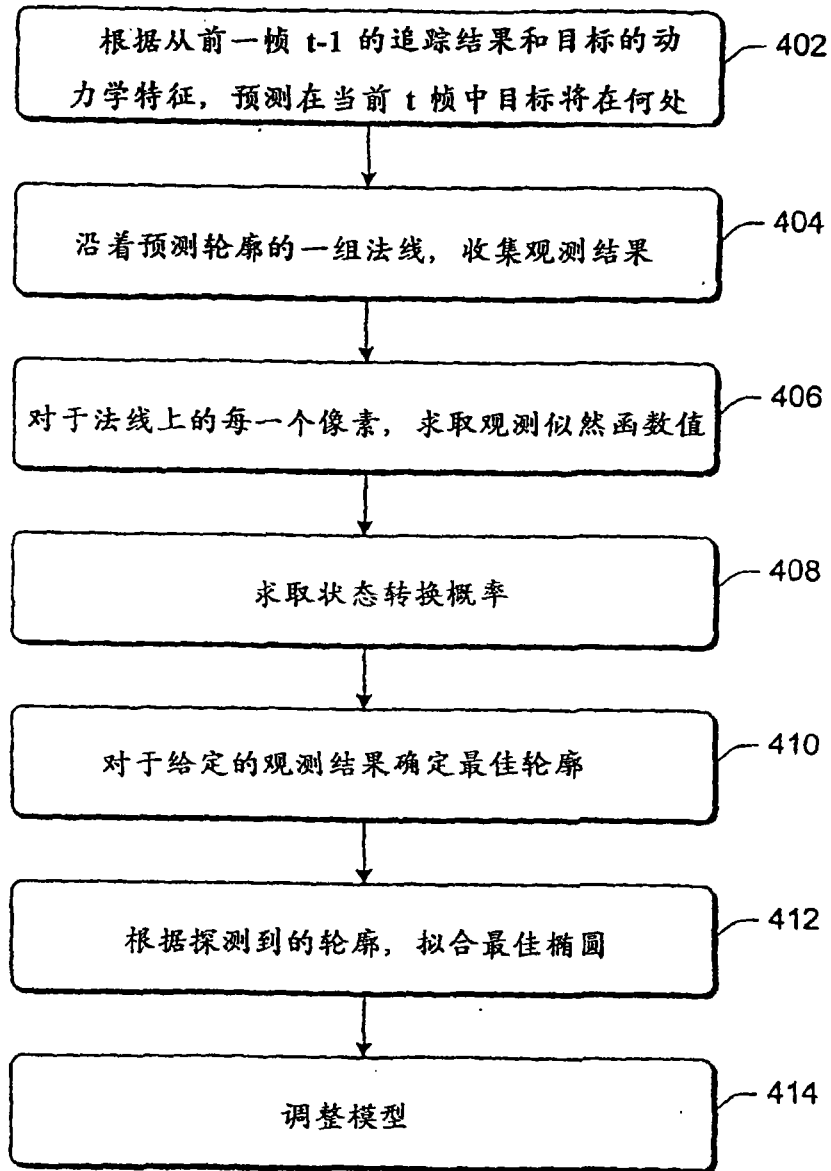


图9

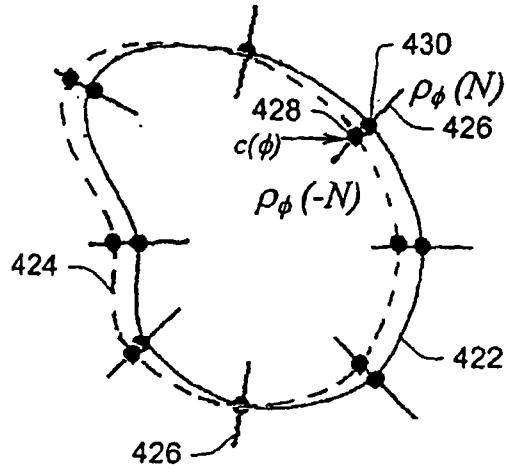


图10

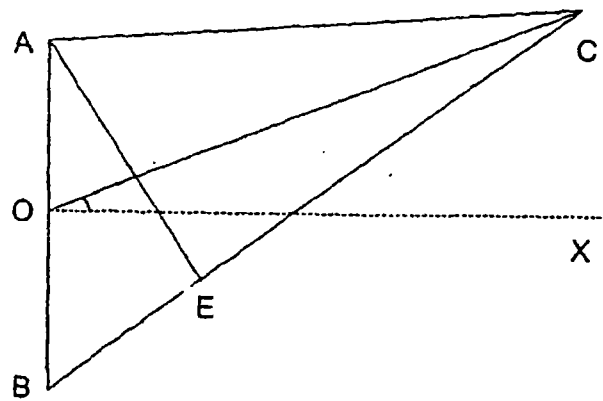


图16

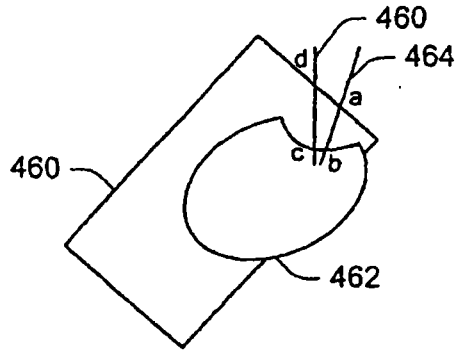


图 11

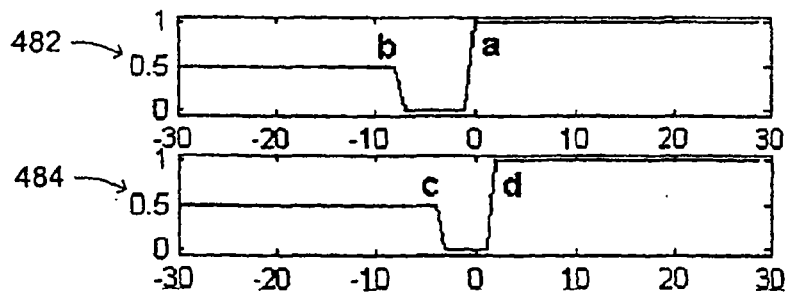


图 12

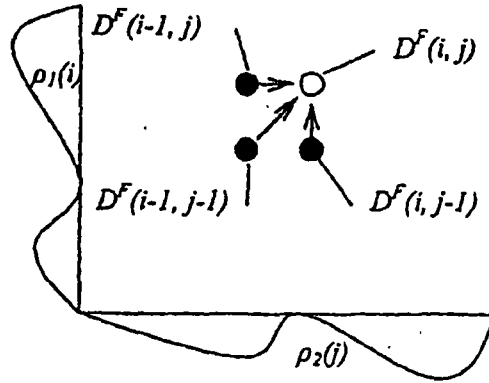


图13

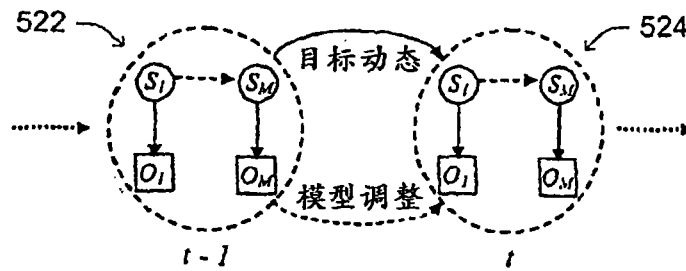


图14

550

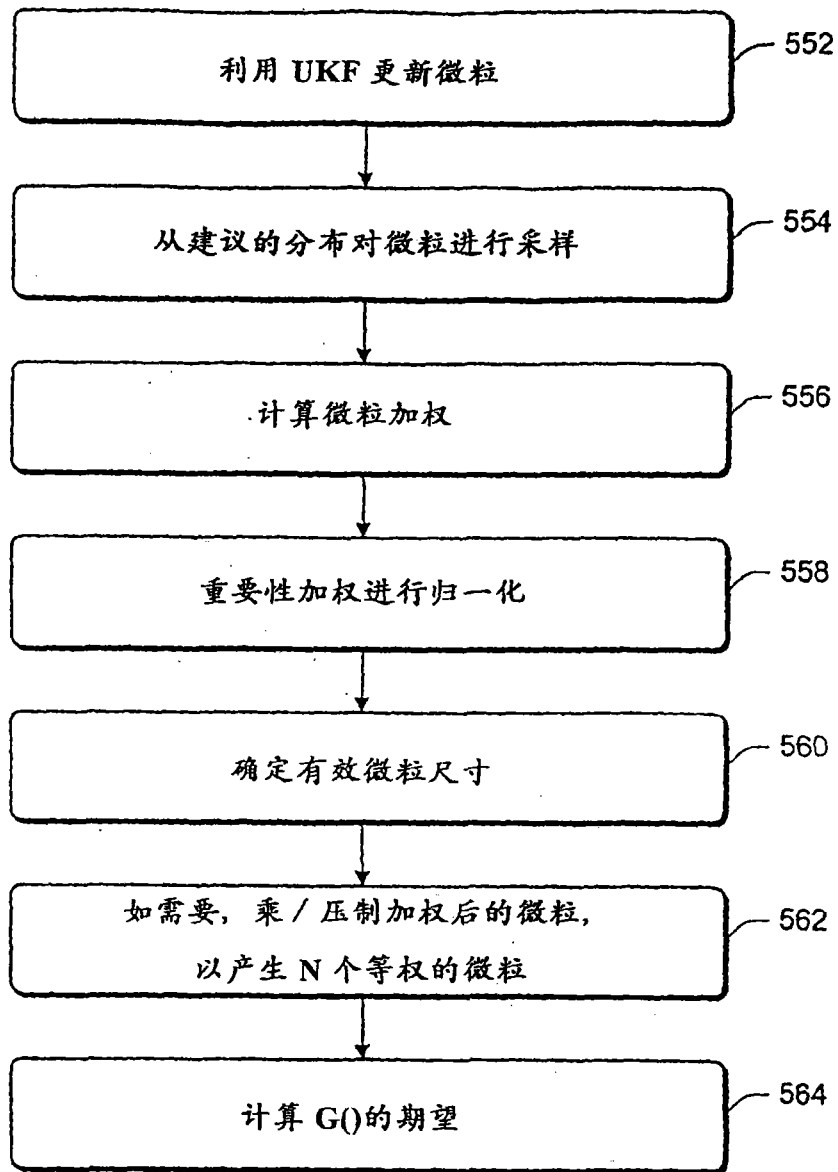


图 15

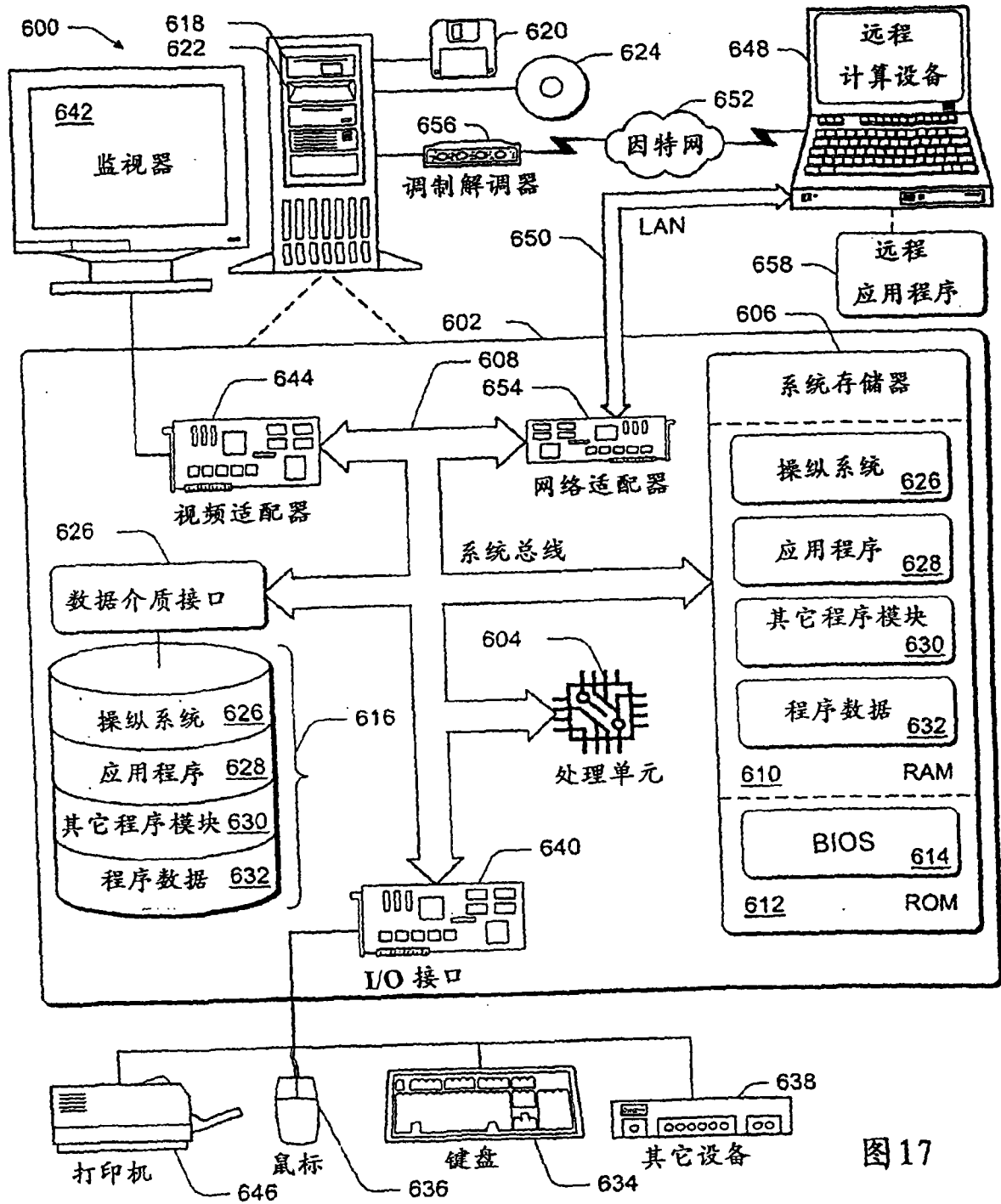


图 17