

多摄像头系统

(申请清华大学工学硕士学位论文)

培 养 单 位: 计 算 机 科 学 与 技 术 系

学 科: 计 算 机 科 学 与 技 术

研 究 生: 丁 旭

指 导 教 师: 陶 品 副 教 授

二〇一七年三月

System

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the professional degree of

Master of Engineering

by

Ding Xu

(Computer Science and Technology)

Thesis Supervisor : Associate Professor Tao Pin

March, 2017

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____

导师签名：_____

日 期：_____

日 期：_____

摘 要

论文的摘要是对论文研究内容和成果的高度概括。摘要应对论文所研究的问题及其研究目的进行描述，对研究方法和过程进行简单介绍，对研究成果和所得结论进行概括。摘要应具有独立性和自明性，其内容应包含与论文全文同等量的主要信息。使读者即使不阅读全文，通过摘要就能了解论文的总体内容和主要成果。

论文摘要的书写应力求精确、简明。切忌写成对论文书写内容进行提要的形式，尤其要避免“第 1 章……；第 2 章……；……”这种或类似的陈述方式。

本文介绍清华大学论文模板 THUTHESIS 的使用方法。本模板符合学校的本科、硕士、博士论文格式要求。

本文的创新点主要有：

- 用例子来解释模板的使用方法；
- 用废话来填充无关紧要的部分；
- 一边学习摸索一边编写新代码。

关键词是为了文献标引工作、用以表示全文主要内容信息的单词或术语。关键词不超过 5 个，每个关键词中间用分号分隔。（模板作者注：关键词分隔符不用考虑，模板会自动处理。英文关键词同理。）

关键词：TeX；L^ATeX；CJK；模板；论文

Abstract

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective, brief introduction to methodology and research process, and summarization of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “the first chapter”, “the second chapter” and the like should be avoided in the abstract.

Key words are terms used in a dissertation for indexing, reflecting core information of the dissertation. An abstract may contain a maximum of 5 key words, with semi-colons used in between to separate one another.

Key words: T_EX; L^AT_EX; CJK; template; thesis

目 录

第 1 章 引言	1
1.1 研究背景	1
1.2 主要研究内容和挑战	5
1.3 本文的主要贡献	5
第 2 章 研究现状和相关工作	6
2.1 本章引言	6
2.2 多摄像头系统	6
2.3 多摄像头系统的同步控制	9
2.3.1 硬件同步	10
2.3.2 软件同步	11
2.4 摄像头拍摄时间检测	13
2.5 本章小结	14
第 3 章 多摄像头系统的拍摄时间检测	16
3.1 本章引言	16
3.2 拍摄时间检测方法	17
3.3 LED 点阵编码方法	18
3.3.1 流水编码	18
3.3.2 二进制编码	19
3.3.3 格雷码编码	20
3.3.4 分组叠加编码	22
3.3.5 进位流水编码	23
3.3.6 编码方法比较	25
3.4 图像检测算法	25
3.5 本章小结	25
第 4 章 多摄像头系统的拍摄时间检测	26
4.1 本章引言	26
4.2 主要研究内容和挑战	26
4.3 本文的主要贡献	26

第 5 章 多摄像头系统的拍摄时间检测	27
5.1 本章引言	27
5.2 主要研究内容和挑战	27
5.3 本文的主要贡献	27
第 6 章 总结与展望	28
6.1 工作总结	28
6.2 未来展望	28
参考文献	29
致 谢	33
声 明	33
个人简历、在学期间发表的学术论文与研究成果	35

主要符号对照表

HPC	高性能计算 (High Performance Computing)
cluster	集群
Itanium	安腾
SMP	对称多处理
API	应用程序编程接口
PI	聚酰亚胺
MPI	聚酰亚胺模型化合物, N-苯基邻苯酰亚胺
PBI	聚苯并咪唑
MPBI	聚苯并咪唑模型化合物, N-苯基苯并咪唑
PY	聚吡咙
PMDA-BDA	均苯四酸二酐与联苯四胺合成的聚吡咙薄膜
ΔG	活化自由能 (Activation Free Energy)
χ	传输系数 (Transmission Coefficient)
E	能量
m	质量
c	光速
P	概率
T	时间
v	速度
劝学	<p>君子曰：学不可以已。青，取之于蓝，而青于蓝；冰，水为之，而寒于水。木直中绳。輶以为轮，其曲中规。虽有槁暴，不复挺者，輶使之然也。故木受绳则直，金就砺则利，君子博学而日参省乎己，则知明而行无过矣。吾尝终日而思矣，不如须臾之所学也；吾尝跂而望矣，不如登高之博见也。登高而招，臂非加长也，而见者远；顺风而呼，声非加疾也，而闻者彰。假舆马者，非利足也，而致千里；假舟楫者，非能水也，而绝江河，君子生非异也，善假于物也。积土成山，风雨兴焉；积水成渊，蛟龙生焉；积善成德，而神明自得，圣心备焉。故不积跬步，无以至千里；不积小流，无以成江海。骐驎一跃，不能十步；弩马十驾，功在不舍。锲而舍之，朽木不折；锲而不舍，金石可镂。蚓无爪牙之利，筋骨之强，上食埃土，下饮黄泉，用心一也。蟹六跪而二螯，非蛇鳝之穴无可寄托</p>

者，用心躁也。——荀况

第1章 引言

1.1 研究背景

随着软件与硬件技术的不断发展进步，摄像头已经成为人们日常生活中必不可少的互联网设备。摄像头的发展更加趋向于模块化，功能化，小型化，应用场景也更加广泛。在智能手机上，前置和后置摄像头成为必备设备；在安防监控领域，摄像头正在被大量布置；在自动驾驶以及其他自动控制领域，摄像头正在替代人眼成为信息输入的主要来源。同时得益于处理器计算能力地不断增强，在摄像头模块中也集成了更多的实用功能，摄像头不再仅仅作为图像采集设备。例如，利用摄像头实现人脸识别功能，替代密码输入或者进行人员搜索；利用摄像头实现运动物体检测功能，在监控过程中替代人工监视；利用摄像头实现图像处理分析，实时生成周围环境信息。摄像头越来越强大的功能给人们的日常生活提供了巨大帮助。

但是，对于单一摄像头，其成像大小、拍摄速率等方面仍存在一定的限制。因此将多个摄像头整合到一起，组成多摄像头系统，成为摄像头发展的一个重要方向。这样既可以利用不同性能参数的摄像头弥补各自的不足之处，也可以成倍扩展摄像头性能。同时，多摄像头系统可以由多个参数指标一般，较为便宜普通的摄像头组成，通过系统优化使其实现更高的性能。

例如，Stanford 大学计算机图形学实验室利用 100 个低成本的 CMOS 摄像头组成了一个大规模阵列，配合相应的编码器、FPGA、处理器和控制总线等设备，通过调整各摄像头的排列顺序和拍摄角度，实现了高像素、高帧率、多光场摄影等功能^[1]。当将各摄像头紧密排列，通过同步使其在同一时间拍摄，将拍摄到的所有图像进行拼接，可以得到一幅高像素的图像，即利用低像素多摄像头系统进行高像素拍摄。当控制各摄像头在不同时间拍摄相同图像，将各帧图像按时间顺序排列，可以突破各个摄像头拍摄帧率的限制，在每秒内得到更多图像，即利用低帧率摄像头实现高速摄影。当各摄像头之间存在一定距离，该系统即可实现多焦点拍摄，得到光场图像数据。

同样，在智能手机领域，越来越多的智能手机配备了双目摄像头系统。例如 Apple 公司的 iPhone 7 手机和华为公司的 P9 手机，均配备双目摄像头，这样可以有效优化拍摄图像的亮度、对比度和清晰度等，以获得更好的拍摄效果。诺基亚公司曾发布过一款智能手机，使用的是一颗摄像头阵列，包含 4×4 共 16 颗子镜头，为了提高图片的色彩纯净度，每个子镜头仅捕捉红、绿、蓝三种颜色当中的一种，

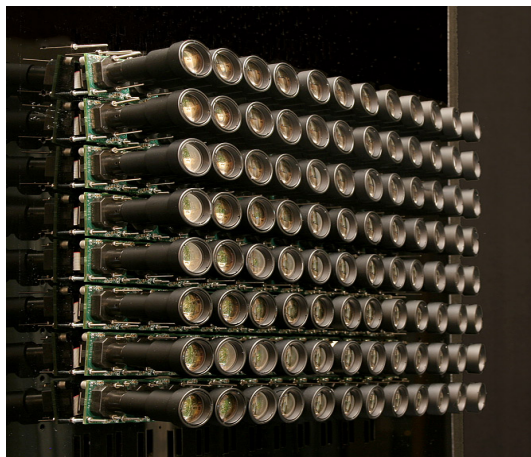


图 1.1 Stanford 大学多摄像头阵列

由处理器对所有子镜头拍摄到的图像进行叠加处理，这样还能够更好地抑制噪点。该手机使用的摄像头阵列还能够实现多点变焦功能，即在拍摄时，各个子镜头选取不同的焦点，并将所有拍摄数据保存，这样可以实现一次拍摄，得到不同焦点的图像。在拍摄后期，就可以根据需要选取合适的焦点调整图片效果。



(a) 包含 16 颗子镜头的阵列摄像头模块 (b) 选取不同焦点，在后期对图片进行变焦处理

图 1.2 诺基亚阵列摄像头手机

多摄像头系统的另一个著名应用是“子弹时间”的拍摄。“子弹时间”拍摄技巧最早出现在 1999 年的电影《黑客帝国》(The Matrix) 中，是指利用计算机辅助技术，使得拍摄到的视频呈现时间变慢或者静止，同时拍摄角度发生变化的视觉特效。实现方法即为利用一系列摄像头组成多摄像头系统，各个摄像头在同一时刻拍摄不同角度的图像，后期处理过程中将同一时刻拍摄的不同角度的图像穿插拼接，以实现时间静止的效果。

在无人驾驶领域，特斯拉第二代自动驾驶技术的摄像头系统包括了三个前置

摄像头，即一个主摄像头负责采集主要道路信息，一个配备广角镜头的副摄像头检测更多车道，一个副摄像头检测前方车辆；在四旋翼飞行器领域，大疆公司最新的精灵 4 Pro 飞行共器配备了 7 个摄像头，安装在机身周围的摄像头可以实现躲避障碍物、追踪拍摄对象等功能，安装在机身下方的摄像头可以实现精确定位、悬停等功能，同时高分辨率、高帧率的主摄像头可以实现 4K 高清视频的拍摄；在 VR、3D 视频拍摄领域，需要多摄像头实现多角度全景拍摄，或模拟人眼拍摄立体视觉效果。多摄像头系统在越来越多的领域得到了广泛应用，发挥了越来越重要的作用，也越来越受到人们的重视。



(a) 特斯拉自动驾驶技术 (b) 大疆四旋翼飞行器 (c) 诺基亚全景 VR 相机

图 1.3 多摄像头系统的应用场景

目前关于多摄像头系统的研究主要集中在软件应用和硬件优化两个方面。其中，软件应用主要是利用现有的成熟系统实现或优化特定功能。例如对多摄像头图像进行拼接校准，利用多摄像头系统进行目标跟踪识别，利用多摄像头系统进行全景拍摄三维重建等。此类应用大多在现有的多摄像头系统的基础上采集数据，或者直接利用拍摄好的视频文件进行研究，因此对于多摄像头系统的性能参数并没有较严格的要求。而关于硬件优化方面的研究，主要集中于如何搭建多摄像头系统，包括摄像头控制，数据传输，算法优化，视角调整等。此方面的研究资料较少，主要是因为对于硬件系统的研究需要耗费大量的人力物力，且属于基础研究，能够取得的突破性成果较少。

对于多摄像头系统来说，为了实现其特定的功能，需要控制系统内的各个摄像头在同一时刻拍摄或按照特定的时间间隔拍摄。如果为了将各个摄像头拍摄到的图像进行拼接，以得到更大分辨率的图像，就需要各个摄像头精确地在同一时刻进行拍摄，这样才能捕捉到拍摄对象在该时刻的状态，如果拍摄时间存在间隔，则拍摄对象的状态发生变化，就无法进行准确的拼接。如果为了将各个摄像头拍摄到的图像按时间顺序穿插排列，以得到更高的拍摄速率，就需要各个摄像头的拍摄时间存在一定的时间差。例如，对于一个双摄像头系统来说，各个摄像头的拍摄速率为 25 帧每秒 (Frames Per Second, FPS)，即各个摄像头拍摄的每两帧图

像之间的时间间隔为 40ms，如果控制两个摄像头的拍摄时间间隔为 20ms，则第一个摄像头拍摄的第一帧图像与第二个摄像头拍摄的第一帧图像之间相差 20ms，第二个摄像头的第一帧图像与第一个摄像头的第二帧图像相差 20ms。将所有拍摄图像按时间顺序交错排列，即可得到每帧图像之间相差 20ms 的视频，其拍摄速率即为 50FPS，该系统利用两个摄像头实现了帧率的提升。

而部分研究工作对各个摄像头之间的时间间隔并没有提出明确的要求，是由于时间差对其研究结果造成的影响较小。例如在进行图像拼接时，如果各个摄像头拍摄静态对象，那么在较短的时间段内，拍摄对象发生的变化不大。而如果对较快的运动对象进行拍摄时，在一定的时间段内，拍摄对象就会发生较大变化从而影响拼接结果。如图 1.4(a) 所示，待拼接的对象为静态的建筑物，两张图片的拍摄时间即使存在一定的差别仍然不影响拍摄结果。而如图 1.4(b) 所示，当拍摄对象为移动的人时，如果拍摄时间不同，则拼接后的图像中同一个人会出现两次，且动作不同^[2]。



(a) 静态拍摄对象的图像拼接



(b) 动态拍摄对象的图像拼接

图 1.4 拍摄时间存在间隔时进行图像拼接的结果

因此，在多摄像头系统的搭建过程当中，一个重要的问题在于如何实现系统内各个摄像头之间的时间同步。目前主要的同步方法有硬件同步和软件同步两种。

硬件同步指的是利用硬件设备将控制系统与各个摄像头相连，通过控制信号控制摄像头的拍摄时间。如图 1.5(a)所示 CODEX 多摄像头控制器，即可实现对多个摄像头的控制及数据传输功能。同样还可以利用外部信号，如闪光灯、声音等控制各个摄像头。如图 1.5(b)所示的由单反相机组成的阵列即可以利用闪光灯作为触发信号，实现“子弹时间”的拍摄。利用软件对多摄像头系统进行同步，主要是指由服务器控制各个摄像头，根据采集到的图像数据分析拍摄时间，从而判定系统内的同步情况，最后根据算法对摄像头进行调整。

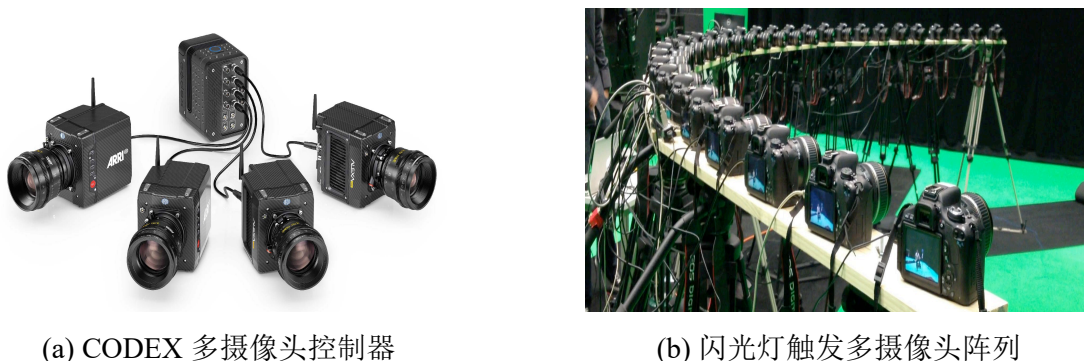


图 1.5 利用硬件进行多摄像头同步

在对多摄像头系统进行同步的过程中，大部分方法是根据摄像头或者控制系统产生的时间戳来判定摄像头拍摄时间的，即这些方法认为时间戳生成的时间与摄像头的曝光时间严格一一对应，这种方法在对同步精度要求不高的情况下是可以被认为是正确的。但是在实际操作过程当中，多摄像头系统内系统响应、信号传输、软件运行都会耗费长短不等且随机分布的时间，因此从控制单元想摄像头发送拍摄指令到生成时间戳，再到摄像头的感光器件开启曝光，这之间会存在随机变化的时间间隔，也就导致系统生成的时间戳无法精确表示摄像头拍摄时间，各个摄像头之间可能会与控制要求存在误差。

因此，对于需要精确同步的多摄像头系统来说，还需要有时间戳以外的方法来精确表示摄像头的拍摄时间，即如何检测摄像头的拍摄时间也是多摄像头系统实现过程当中的一个重要问题。目前应用较多的方法是利用灯光信号作为时钟信号，被摄像头拍摄后根据拍摄图像识别拍摄时刻的信号信息，从而确定摄像头的拍摄时刻。

1.2 主要研究内容和挑战

1.3 本文的主要贡献

第2章 研究现状和相关工作

2.1 本章引言

本章首先介绍了多摄像头系统的发展现状以及发展趋势，并列举了若干多摄像头系统的性能以及实现方法。然后介绍了多摄像头系统同步控制的几种方法，主要可以分为硬件控制和软件控制两大类。最后介绍了对摄像头拍摄时间进行检测的相关方法。

2.2 多摄像头系统

多摄像头系统一般由多个单一摄像头组合而成，包括数据采集，控制，数据存储分析等几部分。多摄像头系统的优势在于融合了各个摄像头的优点，并且利用数量上的优势成倍将这些优点放大。如果需要满足多种拍摄要求，就需要摄像头满足多种性能指标，对于单一摄像头来说大多数情况下很难满足这些要求，而能够满足要求的摄像头往往较为昂贵或难以获得，甚至有些情况下对于摄像头的多种性能要求是相互矛盾的，根本无法实现。但是多摄像头系统就能够很好地解决这一问题，多摄像头系统可以利用多个具有不同性能指标的摄像头，选取所需的功能进行融合以满足要求。同时还可以利用多个摄像头的融合使其性能指标大幅增长，从而利用廉价摄像头实现了高品质摄像头的功能。

目前多摄像头系统主要可分为两种类型，一类是大规模同类摄像头集成，一类是异种摄像头组合搭配。同类摄像头集成一般采用较多的相同性能参数摄像头组成摄像头阵列，以形成规模优势。Stanford 大学计算图形学实验室搭建了一个多摄像头阵列^[3]，就是利用了低成本摄像头的价格优势和高性能处理器的运算优势，将 100 个 CMOS 图像传感器数码摄像头组成阵列形式。每个摄像头由两部分组成，一部分负责图像采集，由互补金属氧化物半导体（CMOS）图像传感器和廉价镜头组成，一部分负责数据处理，由视频编码器，现场可编程逻辑门阵列（FPGA）和 IEEE 1394（Firewire）传输接口组成。该多摄像头系统能够对 100 个摄像头拍摄到的视频进行实时同步压缩，并将数据保存在硬盘中。

根据各个摄像头排列方式和拍摄角度的不同，该多摄像头系统能够实现多种功能。当各个摄像头紧密排列时，可以将系统视为一个单中心的合成相机，将各个摄像头拍摄到的数据进行合成处理后，就能够大幅度提升各项性能参数，如动态范围、景深、帧率和光谱灵敏度等^[4,5]。当各个摄像头排列间隔较远时，可以将该



图 2.1 Stanford 大学多摄像头阵列

系统视为多中心摄像机，即为视场相机。通过该视场相机可以应用新的三维场景重建方法以及多视角全景构建方法^[6]。当各个摄像头间隔排列时，可以将系统视为一个大合成孔径相机，使得该系统可以通过部分闭塞的环境观察拍摄对象，即实现共聚焦镜头的离散近似，使得不在选定平面上的物体变得模糊和黑暗^[7,8]。

Parziale 等人在文献^[9]中设计了一种利用多摄像头系统，在非接触条件下检测人手指纹的装置。为了避免在恶劣条件下通过按压方式提取指纹时产生的误差，该系统利用五个小型摄像头多角度对手指进行拍摄，并通过图像处理提取指纹信息，有效避免了因皮肤过于干燥或存在过多水分等因素对指纹检测造成的影响，实现了小型化高准确度的指纹检测系统。

同时，利用多摄像头系统进行视频监控也是一个广泛的应用场景。文献^[10-13]中分别介绍了利用多个摄像头在不同应用场景下组成多摄像头监控系统，通过硬件配置和软件优化，监控探测多种对象。

另一类多摄像头系统由不同性能的摄像头组合而成，这样可以充分发挥各个摄像头的性能优势，从而满足使用者对于系统的多种需求。例如在欧洲航天局（ESA）和俄罗斯联邦航天局合作推动的火星探测计划（ExoMars）中^[14,15]，为了能够使火星登陆车实现周边地形勘测，火星车自身定位，大气及周边环境监测，科学实验观测等功能，研究人员将配置在火星车上的全景摄像头（Panoramic Camera）设计成了一个多摄像头系统。该系统是由一个高分辨率摄像头（HRC）和两个广角立体摄像头（WAC）组成。其中高分辨率摄像头的有效像素为 1024×1024 ，主要负责细节图像的拍摄。而广角立体摄像头配备视角为 65° 的镜头，通过两个摄像头采集到的图像进行融合，模拟人眼实现立体成像，主要负责观测周围环境。这样，该系统通过不同性能指标的摄像头之间的配合，实现了所需的全部功能。

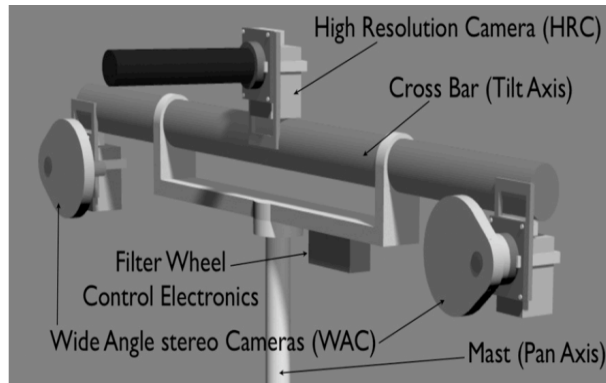


图 2.2 火星登陆上的全景摄像头

最近一段时间,多摄像头系统也被广泛应用于智能手机的摄像头模块上。Apple 公司推出的最新手机 iPhone 7 Plus 就配备了双目摄像头模块^[16],由广角和长焦两颗摄像头组成,广角摄像头主要负责近景拍摄,而长焦摄像头主要负责远景拍摄,从而相互配合实现光学变焦功能。华为公司推出的 P9 手机同样配备双摄像头系统,但是两颗摄像头分别为黑白和彩色摄像头^[17],黑白摄像头负责捕捉景物细节,使得成像更为清晰,而彩色摄像头负责捕捉颜色,使得图片色彩更为饱满,通过对不同摄像头拍摄到的图像的算法融合,得到更好的拍摄效果。



(a) iPhone 7 Plus



(b) 华为 P9 手机

图 2.3 智能手机上的多摄像头系统

除了对于多摄像头系统组成搭建方法的研究,目前还有部分研究是围绕多摄像头系统拍摄视频内容的分析展开。例如针对视频内的运动物体进行追踪,在安防监控等领域有较多应用。Ting-Hsun Chang 等人在文献^[18]中提出了一种基于贝叶斯模态融合的方法,利用多摄像头系统追踪室内环境下多个人的移动轨迹。为了连续追踪运动物体,该系统为每个新检测到的对象设定特有的模式,当对象移动或进入到其他摄像头视野内时,系统会利用贝叶斯网络组合匹配各个模式,以实现连续追踪。不同于其他单一摄像头的闭塞推理追踪方法,该方法利用多摄像

头系统进行交互追踪,能够大大提高追踪精度。同样 Kim 等人在文献^[19]中也提出了一种基于平面跟踪对应模型(TCM)的方法来对大规模多摄像头系统内的对象进行跟踪。该方法为每个运动对象计算了一个唯一且不变的追踪对应模型,使得在不同摄像头的视野内,或者在各个摄像头重叠的视野内,对应同一个物体来说有且只有一个追踪对应模型,这样就能能够准确追踪对象的运动。文献^[20-22]分别介绍了利用多摄像头系统进行对象检测或跟踪的方法。

还比如针对拍摄图像进行多摄像头校准,以实现更好的拍摄效果。Strecha 等人在文献^[23]中为了实现基于图像的三维建模技术,需要将各个摄像头进行校准并实现多角度立体视觉。Baker 等人在文献^[24]中为了校准各个摄像头,利用 LED 灯光建立点对(point correspondences),输入到大规模非线性特征值最小化程序当中,使得各个摄像头两两之间建立全息投影矩阵。Kurillo 等人在文献^[25]中提出了一种广域校准方法,利用 LED 灯光标识作为校准对象,利用矩阵分解计算摄像头初始姿态,并通过自动构建加权视觉图,同时在相机之间找到最佳的变换路径来解决全局校准。文献^[26-28]分别利用图像内的地面、基础设施等参照物对多路摄像头图像进行校准。

另外一些研究是针对多摄像头拍摄结果进行拼接。Hongming Zhang 等人在文献^[29]中为了解决前景对象在视频拼接过程中造成的割裂问题,提出了一种前景对象边缘调整的方法。该方法将各个摄像头拍摄到的图像进行前景提取,并根据前景图像的内容动态调整拼接边缘,再对于静态的背景图像进行拼接,最后将前景图像与背景图像进行融合即可得到拼接结果。文献^[30-32]中也介绍了更多的多摄像头图像拼接方法,图像拼接技术在众多领域都有着广泛而深入的研究和应用。

2.3 多摄像头系统的同步控制

多摄像头系统的同步控制,其主要的目的是为了能够精确控制系统内各个摄像头的拍摄时间,以减小拍摄时间误差对拍摄结果造成的影响。例如,当利用多个摄像头从不同角度拍摄某一对象时,如果该对象静止不动,则各个摄像头的拍摄时间即使存在误差也不会影响拍摄结果。但是如果该对象是在不断变化,则各个摄像头会在不同时刻拍摄到该对象的不同状态,如果各个摄像头之间并非精确同步,其拍摄时间存在一定波动,那么就无法拍摄到同一时刻该对象的相同状态。同样,当利用多个摄像头按照一定的时间间隔,在不同时刻连续拍摄某一对象的变化,如果各个摄像头之间并非时间同步,那么就无法控制其按照设定好的时间间隔拍摄,可能会发生拍摄时间的重叠、提前或延后。

现阶段,对多摄像头系统进行同步控制,主要分为硬件同步和软件同步两种。

利用硬件进行同步时,需要通过特殊的硬件接口,如 IEEE-1394 接口、Camera Link 接口等,将系统内各个摄像头连接到一起,通过控制器发送控制信号给摄像头。而利用软件进行同步时,则是根据拍摄到的图像数据分析拍摄时从而对各个摄像头进行调整。

2.3.1 硬件同步

硬件同步控制主要可分为两类:一类是利用多路图像数据采集卡实现,一类是利用外部触发信号源实现。

图像数据采集卡是一种对摄像头图像数据进行获取、处理、存储等功能的硬件设备。通过硬件接口同摄像头进行连接,接口的种类包括模拟接口,如 AV 接口、S 端子等,和数字接口,如 PCI 总线、USB 接口等。图像数据采集卡一般都集成了对摄像头的控制模块、编解码模块、存储模块等功能模块,因此具有多个输出接口的采集卡能够对多路摄像头进行控制。在图像数据采集卡的使用过程当中,一般对于摄像头的要求较高,需要摄像头具有专门的通信接口能够同采集卡进行连接,并且能够对采集卡的控制信号做出响应,同时也要保证采集到的数据能够被采集卡所识别。虽然采用图像数据采集卡能够在硬件层面对多路摄像头进行比较精确的同步控制,但是由于此类采集卡多为成熟产品,开放性较低,难于对其进行更底层地控制,同时对于摄像头的要求限制也较多,其应用场景并不广泛。

目前市面上可以购买的图像数据采集卡种类很多,其主要差别在于摄像头接口种类,输出接口数量以及其他一些附加功能。同样也有很多关于设计并实现多路图像数据采集卡的研究。如朱海宽等人在文献^[33]中设计并实现了一种基于 PCI 总线的多路图像采集卡;刘晓军等人在文献^[34]中实现了一种具有 HD-SDI 高清串行数字接口的 FPGA 视频采集卡;王剑飞等人在文献^[35]中实现了在 Linux 平台下的视频采集设备的驱动程序。

为了消除图像数据采集卡在使用过程中的种种限制,更多的多摄像头系统采用外部触发信号源实现同步控制功能。这类系统内的各个摄像头都具有触发信号的收发接口,通过接口与控制器相连接,控制器根据系统需求在特定的时刻发送触发信号给各个摄像头。姜广文等人在文献^[36]中提出了一种基于外触发的多摄像头同步系统的设计方法。该方法采用 USB 接口的数字输出模块作为同步触发信号发生器,将高精度的触发信号发送给各个摄像头,实现同步功能。国蓉等人在文献^[37]中利用同步信号发生器和锁相环电路实现同步功能。当摄像头接收到同步触发信号后,开始进行曝光拍摄。经试验验证,触发信号与摄像头拍摄周期之间的相位差为 10 度,远小于摄像头的曝光时间间隔。在文献^[38]中,多摄像头系统中的

所有摄像头通过 IEEE-1394 总线节点连接, 并通过控制器自动对摄像头进行控制, 该系统的最小控制误差为 $125\mu\text{s}$ 。但是由于 IEEE-1394 总线的性能限制, 每个节点能够控制的摄像头有一定的数量限制, 并且不同总线之间的同步需要额外的同步单元, 实现较为复杂^[39]。Zitnick 等人在文献^[40]中利用两个集线器 (concentrators) 来控制 8 个摄像头, 并利用 FireWire 火线接口将两个集线器进行同步, 系统实现更为复杂。同时, 还有一些多摄像头系统利用 GPS 等通用信号进行同步, 这样同步信号全局分发, 可以有效消除误差。如黄芳等人在文献^[41]和刘团结等人在文献^[42]中所介绍的方法都利用的 GPS 信号进行同步。但是这些方法往往对于系统和摄像头的性能、接口要求较多, 且要能够保证实时接收同步信号, 实现难度较大。

2.3.2 软件同步

利用硬件进行多摄像头同步时, 往往要求摄像头具有特定的硬件接口以实现同步信号的接收, 这就使得大多数日常生活中接触到的普通摄像头无法应用这种方法进行同步, 给多摄像头系统搭建提出了更高的硬件要求。而软件同步是利用各个摄像头拍摄到的数据, 通过分析视频内容确定拍摄时间或者各个摄像头之间的时间间隔, 从而实现同步控制功能, 因此对于摄像头并无更多的硬件接口要求, 适用的范围更加广泛, 应用场景更多。

软件同步也可以主要分为两类, 一类是利用系统内部信号进行同步, 一类是利用系统外部信号进行同步。利用内部信号进行同步是指各个摄像头连接到系统的同步服务器上, 通过服务器运行的软件计算各个摄像头所需的拍摄时刻, 然后发送拍摄信号触发摄像头进行拍摄, 并通过拍摄到的图像分析拍摄时间的准确性, 反馈获得触发精度, 进一步调整优化各个摄像头的拍摄时刻。利用外部信号进行软件同步, 可以在多摄像头系统内部署一些通用的时间同步协议, 如网络时间协议 (Network Time Protocol, NTP)、精确时间同步协议 (Precision Time Protocol, PTP) 等。NTP 协议利用时间服务器向各个客户端分发时间报文, 以控制整个协议网络内时间同步。其精度在局域网中可以达到 0.1ms , 而在互联网中由于网络传输速率的影响精度会下降至 $1\text{-}50\text{ms}$ ^[43]。PTP 协议是一种同步精度更高的时间协议, 又称 IEEE-1588 协议。通过建立主从同步系统来实现频率同步与时间同步, 其最高的同步精度能够达到亚毫秒级别^[44]。

Svoboda 等人在文献^[45]中设计了一个多摄像头视频拍摄系统, 其中各个摄像头同 Linux 服务器相连, 各个服务器又同一台同步控制服务器通过局域网络连接。在拍摄过程当中, 同步控制服务器向各个摄像头服务器发送拍摄信号, 控制各个摄像头进行拍摄。该系统的同步精度取决于服务器互联网络的信号传输速率、各

个服务器对拍摄信号的响应时间和摄像头的硬件响应时间等，因此同步精度有较大的不确定性，精度不高。

Rai 等人在文献^[46]中利用四个摄像头同服务器相连，其中三台服务器设为客户端，一台设为时间服务器。时间服务器定时向各客户端发送时间日志，用来同步各个服务器的系统时间，当时间服务器发送命令要求各个客户端在某一时刻拍摄时，因为系统内各个服务器的系统时间相同，就可以保证拍摄时间的同步。在低帧率的情况下，系统的同步误差可以减小到 5ms 左右，但是在高帧率的条件下，由于系统能软件运行消耗更多的内存、CPU 等系统资源，同步误差会增大。

Ahrenberg 等人在文献^[47]中首先设置所有摄像头的系统时间相同，在开始拍摄时同步控制服务器仅向各个摄像头发送一个起始脉冲和一个将来某时刻的拍摄时间。等各个摄像头的系统时间达到这个拍摄时间时，各个摄像头开始拍摄。该方法并没有考虑到各个摄像头的系统时间会发生漂移，一次时间校准之后，经过一段时间的运行各个摄像头之间的同步时间会发生变化，导致无法精确同步。

Hyuntae Cho 在文献^[48]中介绍了一种利用 ZigBee 无线通信网络同步多摄像头系统的方法。如图 2.4 所示，该方法利用两两摄像头之间互相发送同步报文计算网络延迟时间 d 和摄像头时间差 δ 。首先摄像头 A 在 $T1$ 时刻发送同步报文，摄像头 B 在 $T2$ 时刻接收到报文，则 $T2 = T1 + d + \delta$ 。然后摄像头 B 在 $T3$ 时刻发送确认报文给摄像头 A，摄像头 A 在 $T4$ 时刻接收到报文，则 $T4 = T3 + d - \delta$ 。因此可以计算得到网络延迟时间 d 和摄像头时间差 δ 的取值：

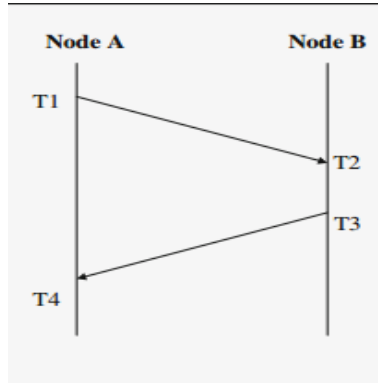


图 2.4 ZigBee 无线通信网络同步方法

$$\begin{aligned} d &= \frac{(T2 - T1) + (T4 - T3)}{2} \\ \delta &= \frac{(T2 - T1) - T3}{2} \end{aligned} \quad (2-1)$$

得到各个摄像头之间两两时间差后,即可根据时间差计算同步偏移量,实现系统同步。该方法实现较为简单,但应用前提是保持系统内网络传输延时固定,当网络状况发生变化时,摄像头之间的时间差计算就会出现误差。

第二种软件同步的方法是利用系统外部信号进行同步。这里的外部信号可以是计时器、灯光或者声音等多种可以被摄像头检测到的形式,当摄像头检测到这些信号后,以其作为基准点触发各个摄像头在预设好的时刻开始进行拍摄。

Shrestha 等人在文献^[49]介绍了一种外部信号同步方法,该方法利用闪光灯作为外部信号,在各个摄像头拍摄到的视频当中检测拍摄到闪光的那一帧,由于对于一次闪光,各个摄像头会在同一时刻捕捉到,因此根据闪光进行匹配即可同步时间。该方法主要分为两步:第一步是检测闪光,当某一帧图像拍摄到闪光时,图像的亮度直方图偏明亮的一部分会出现峰值,并且其相邻图像的亮度直方图中会出现低谷。因此在视频序列内按照此规律即可找到闪光灯所在帧。第二步是根据闪光对各个视频序列进行校准。如果将是否是闪光灯出现帧进行二值化表示,则各个视频序列可视为一个二进制向量,其互相之间的匹配问题就可以利用非精确字符串匹配实现。这个方法的同步精度同视频拍摄的帧率有关,误差为正负一帧。同时 Shrestha 等人在文献^[50]中还介绍了以声音为同步依据的同步方法,匹配各个视频序列内的声纹信息,用以同步各个摄像头。

Jingyu Yan 等人在文献^[51]中以时空兴趣点作为同步依据对摄像头进行同步。时空兴趣点是一种在视频图像中出现的,在时间和空间维度下都具有唯一性的特征点,并且在不同的拍摄角度下,这些特征点仍然不发生变化。这些特征点可以代表图像内物体的消失或者出现,物体的分裂或者合并以及物体运动速度的变化等等不同含义,通过设定不同的参数值选取图像当中最合适的特征点,根据其分布情况进行匹配。

在文献^[52,53]中介绍了多摄像头系统同步在无人驾驶汽车上的应用,利用软件同步方法将汽车上装配的摄像头同步,实时采集车辆周围信息。

2.4 摄像头拍摄时间检测

对于一个同步的多摄像头系统而言,其中各个摄像头的拍摄时间是由系统内部时间确定的,系统会给每张图像打上时间戳用以确定系统同步精度。但是由于在多摄像头系统内可能存在的网络传输延时,硬件信号处理延时,程序运行延时,系统时间漂移等因素,会导致图像时间戳与摄像头实际的曝光时间存在一定的差异,这个差异就是系统的同步误差。因此在利用时间戳作为摄像头拍摄时间检测标准的基础之上,还需要更加准确的方法来检测摄像头的拍摄时间。

戴琼海等人在文献^[54]中利用多个LED灯显示串行时间码IRIG-B信号。外部触发控制器控制LED灯开始显示,并在同一时刻向摄像机发送拍摄信号。则摄像机的曝光开始时间与LED灯的显示时间相同,拍摄到的图像中LED灯显示的时间即为摄像机的拍摄时间。此方法的不足之处在于需要外部控制其进行控制,要求摄像机具有接收控制信号的硬件接口。同时由于系统延时无法保证摄像机曝光开始时间与显示时间精确相同。

赵莉等人在文献^[55]中利用LED点阵显示二进制整数表示时间,当摄像机拍摄到点阵后即可确定时间。由于点阵显示不断变化,为了避免摄像机拍摄到多个显示的时间产生重叠,该方法需要设定点阵中LED灯的亮灭变化周期不大于要求的检测精度,同时摄像机曝光时间不大于此变化周期,但是当曝光周期处于两个显示时间之间时,仍有可能拍摄到时间叠加。而且当要求检测精度较高时,曝光时间也相应变短,从而导致拍摄到的图像曝光不充分或者曝光时间过短摄像机性能无法满足需求。

Qi Zhao等人在文献^[56]中利用单一LED灯作为信号源,控制其亮灭变化满足独立同分布。使用多个摄像机对其进行拍摄并将拍摄结果根据亮灭分布进行匹配,从而计算各摄像机之间的相对时间偏差。该方法需要多个摄像机长时间拍摄才能够达到较高检测精度,检测灵活性较低。

2.5 本章小结

本章介绍了多摄像头系统的相关研究现状,主要包括多摄像头系统的研究实例、同步控制方法和同步精度检测方法。目前多摄像头系统主要有两类实现方式,一类是大规模同类摄像头集成,可以利用较为低廉,性能一般的摄像头实现高性能摄像头的功能;一类是由异种摄像头组合搭配实现,针对系统的不同需求选取不同性能的摄像头组合搭配实现。多摄像头系统还有其他的实现方法,但其根本目的还是为了利用更多的易于获取的摄像头实现更高性能。

在搭建多摄像头系统的过程中,一个关键问题是如何解决同步问题。由于多摄像头系统需要各个摄像头之间在拍摄时间上受到精确控制,将摄像头进行同步即成为必不可少的环境。目前较为常见的同步控制方法可分为硬件同步和软件同步。硬件同步是指将各个摄像头通过硬件接口连接,由同步控制器发送同步信号控制各个摄像头的拍摄。这种方法虽然同步精度相对较高,但需要摄像头具有相应的硬件接口,对于摄像头的要求限制较多。另一种同步控制方法为软件同步,利用系统内的软件程序分析各个摄像头的拍摄时间并相应地进行调整,不断优化同步精度。这种方法适用于大多数摄像头,对于摄像头没有硬件限制,但是由于软

件运行过程当中可能存在的网络延时、信号传输延时、程序运行延时、系统时间漂移等原因，会导致系统的同步精度降低，且这些误差随机存在，并不可控。

因此为了检测多摄像头系统的同步情况，需要对其进行拍摄时间检测。由于由系统为每帧拍摄图像确定的时间戳也可能存在误差，因此需要通过外部信号确定摄像头的拍摄时间。可以利用灯光、声音等信息作为检测信号，对摄像头的拍摄结果进行分析，确定真实的拍摄时间，最后进行匹配同步。这种方法能够有效排除系统内部误差干扰，检测系统的同步精度。

第3章 多摄像头系统的拍摄时间检测

3.1 本章引言

对于多摄像头系统来说，控制各个摄像头在特定时刻进行拍摄是其主要的同步方法。在实际的系统应用过程当中，服务器会通过网络发送信号，控制各个摄像头进行拍摄，但是在系统内可能存在多种因素导致摄像头的真实拍摄时间与服务器设定的拍摄时间不符，而这就导致了多摄像头系统内存在的同步误差。比如服务器发送命令控制各个摄像头在 t_1 时间后进行拍摄，但是服务器程序运行会消耗一定的时间 t_2 ，控制命令经网络由服务器传输给各个摄像头，传输过程会消耗一定时间 t_3 。摄像头接收到命令设定自身参数在 t_1 时间后拍摄，而在 t_1 时刻摄像头开始拍摄时，又会消耗掉一定的硬件响应时间 t_4 ，因此摄像头真实的拍摄时间并非服务器设定的 t_1 ，而是 $T = t_2 + t_3 + t_1 + t_4$ 。并且系统中可能存在的各种延时随机出现，对于各个摄像头来说其延时长短也同样不确定，无法准确进行预测。

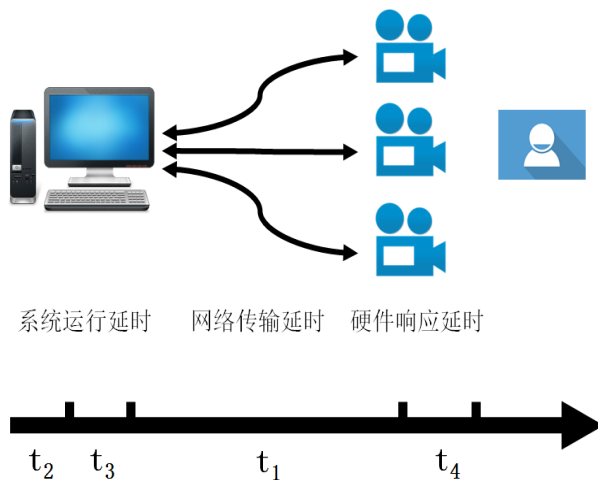


图 3.1 多摄像头系统拍摄过程中可能存在延时

因此，对于多摄像头系统来说，为了提高同步精度，并且检验同步效果，就不能仅仅依靠系统内部时钟显示的时间进行判断，需要借助外界参考物来检测各个摄像头的拍摄时间。本章提出了一种基于 LED 点阵的拍摄时间检测方法。该方法利用 FPGA 控制 LED 点阵作为检测器，通过设计不同编码方法使点阵显示不同的状态序列。当摄像头拍摄到检测器图像后，对图像进行分析，从而检测出摄像头真实的拍摄时间。该方法通过图像进行检测，排除了系统内部各项误差因素的干扰，并且检测器与摄像头相互分离无需数据通信，同时还适用于多种帧率、曝光

时间的摄像头，保证了该方法的普适性。通过对检测器参数的调整可以实现不同精度的检测，且无需调整摄像头参数，使得该方法能够在摄像头正常工作过程中实时进行检测。

3.2 拍摄时间检测方法

采用本方法对摄像头拍摄时间进行检测，主要可分为如下3个步骤：

(1) 检测器初始化：该方法的检测器由FPGA控制器和LED点阵两部分组成，分别实现了系统控制和检测信号显示的功能。在初始化阶段，编写控制程序利用FPGA控制LED点阵中各个LED灯的亮灭变化，显示出不同的LED灯组合状态。当每个状态持续一段时间，若干不同状态即可组成状态序列。因此，通过调整FPGA的控制策略，即可控制LED点阵服从一定的编码规律。而根据此规律又可以确定每个LED点阵状态在状态序列当中所处的位置，即可确定每个状态出现的时间。

(2) 获取检测图像：设置好检测器后，在需要对拍摄时间进行检测时，将LED点阵置于多摄像头系统的拍摄范围内，控制各个摄像头对其进行拍摄，并将图像保存供分析处理，只需保证拍摄到的图像清晰、完整即可。由于检测器独立于多摄像头系统，相互之间无需进行通信，避免了信号连接、传输、识别等众多操作，有效扩大了该检测方法的适用范围。

(3) 分析图像计算拍摄时间：当获得各个摄像头拍摄到的LED点阵图像后，利用图像处理服务器对各副图像进行处理分析。在每幅图像当中提取LED点阵所在位置，并分析点阵当中各个LED灯的亮灭情况，即LED点阵在摄像头拍摄时刻的状态，从而可以根据LED点阵的编码规律确定此状态在序列当中所处的位置，即可以推断摄像头拍摄的时间。多个摄像头之间也可以根据拍摄到的状态的不同确定拍摄时间间隔。

根据在初始化时选择的编码方式的不同，检测出的拍摄时间精度也不尽相同。为了提高该方法的检测精度，要求每个LED点阵状态的持续时间较短，因此要求LED点阵的变化频率较高，对于控制信号的响应速度较快。同时，为了使状态序列足够长，要求LED点阵包含足够多的LED灯，使其能够组合成足够多的不同状态。另外，由FPGA控制LED点阵中各个LED灯期进行变化，因此要求FPGA具有特定接口能够向LED点阵发送控制信号。并且为了在点阵高频率变化时保证足够的时间精度，要求FPGA处理信号和传输信号的时间延迟要尽可能小。对于图像识别服务器来说，为了保证检测的准确性和高精度要求，其对于LED点阵的识别和对于LED灯亮灭状态的判断需要具有更高的准确性。

3.3 LED 点阵编码方法

对 LED 点阵中各个 LED 灯的亮灭变化情况进行编码，目的是为了使每个由 LED 灯组成的点阵状态具有唯一性，并且形成特定的状态序列，以便能够对摄像头拍到的状态进行识别，确定该状态在序列中所处的位置，同时还要尽可能提高识别正确率。因此，对于点阵编码方法，要求其具有唯一性、有序性、易识别性。针对以上要求，本文提出了以下 4 种编码方法，并对其进行了比较。不失一般性，以下采用如图 3.2 所示的 4×4 的 LED 点阵对编码方法进行描述。

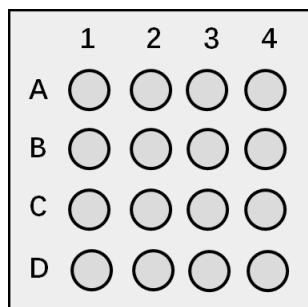


图 3.2 4×4 的 LED 点阵

该点阵共有 16 个 LED 灯，依次编号 A1、A2、...、B1、B2、...、D4。各个 LED 灯的亮灭状态相互独立。各个 LED 灯周期性变化，每个控制周期的持续时间 (t_{dur}) 相同。每一个控制周期内，各个 LED 灯保持亮灭不变，全部 LED 灯的亮灭状态组成 LED 点阵在该周期内的状态，全部 LED 点阵状态组成该编码方法生成的状态序列。各 LED 灯的亮灭状态和控制周期的时间长短由 FPGA 控制器进行控制，且控制周期持续时间远远大于 LED 灯亮灭响应时间和控制信号生成、传输时间。

3.3.1 流水编码

流水编码方法控制 LED 点阵在一个控制周期内仅有一个 LED 灯点亮，下一个周期时下一个 LED 灯点亮，上一周期点亮的 LED 灯熄灭。各 LED 灯按照由左至右、由上至下的顺序依次点亮。当最后一个 LED 灯，即 D4 灯点亮后，下一周期时 D4 灯熄灭、A1 灯点亮，进行下一循环，如图 3.3 所示：

该编码方法实现较为简单，当 LED 点阵共包含 n 个 LED 灯，控制周期时间长度为 t_{dur} 时，从第一个 LED 灯点亮到最后一个 LED 灯点亮，即一个总的循环周期时间长度 (T) 为：

$$T = n * t_{dur} \quad (3-1)$$

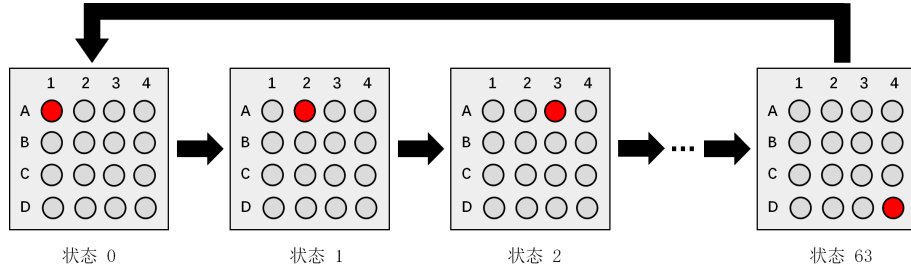


图 3.3 流水编码的点阵状态序列

当摄像头拍摄到第 k 个状态时，即可以判定其拍摄时刻与 LED 点阵开始运行时刻之间的时间差为 $k * t_{dur}$ 。对于两个摄像头分别拍摄到第 k_1 和 k_2 个状态，则这两个摄像头之间的时间差为 $(k_1 - k_2) * t_{dur}$ 。

当摄像头拍摄到某一 LED 点阵的状态时，可能是在 LED 点阵保持状态不变的整个控制周期 t_{dur} 中的任意时刻进行拍摄的，因此其检测精度与控制周期的大小有关，最大误差为控制周期时间长度。因此，为了提高检测精度，往往需要控制周期 t_{dur} 较小。同时当 LED 灯数量较少时，根据公式 3-3 可知，该编码算法的总循环周期 T 较短。这可能导致在测量过程中使用摄像头拍摄到的不同状态处于不同循环周期内，且无法确定其所属的循环周期，因此最终的测量结果为 $x * T + k * t_{dur}$ ，其中 x 可以取 $0, 1, 2, \dots$ ，总循环周期过小会导致无法确定 x 取值，使得测量结果存在误差。

3.3.2 二进制编码

此编码方法中，将每一个 LED 灯视为一个二进制数位，LED 灯点亮代表 1、熄灭代表 0，整个 LED 点阵视为一个 16 位的二进制数。在每个控制周期内，LED 点阵显示一个二进制数，下一个周期时发生变化，按照二进制数变化规律显示下一个二进制数。状态序列如图 3.4 所示：

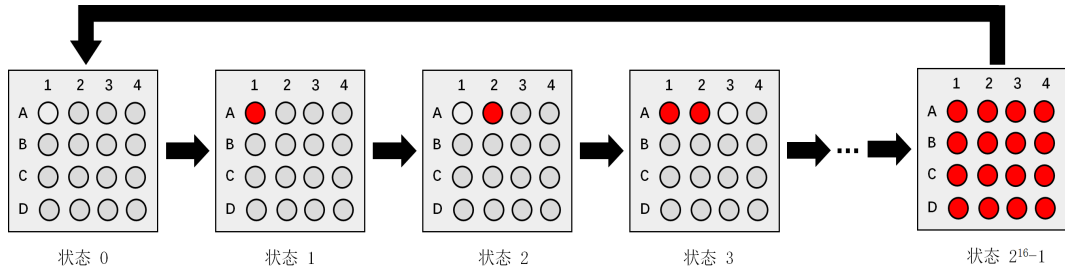


图 3.4 二进制编码的点阵状态序列

使用该编码方法，当 LED 点阵共包含 n 个 LED 灯时，一个总的循环周期内共有 2^n 个 LED 点阵状态，则一个总的循环周期的时间长度为：

$$T = 2^n * t_{dur} \quad (3-2)$$

循环周期的时间长度随着 LED 灯的数量指数增长, 因此可以利用较少的 LED 灯实现较长的循环周期, 使得在检测过程中不会出现不同状态属于不同循环周期的情况。二进制编码方法的检测精度同流水编码一样, 等于 LED 点阵的控制周期, 即 t_{dur} 。

此方法存在的问题是当摄像机的曝光过程跨越两个或两个以上控制周期时, 在一个曝光周期内会拍摄到多个 LED 点阵状态叠加的图像。由于相邻的若干个状态叠加后的图像不唯一, 因此也就无法识别出摄像机拍摄到的图像是由哪几个状态叠加得到的, 也就无法判断正确的拍摄时间。例如状态 1、状态 2 叠加和状态 2、状态 3 叠加后得到的图像相同, 无法根据叠加图像识别出正确的拍摄时间。

3.3.3 格雷码编码

二进制格雷码 (binary gray code) 在信号传输领域应用较为广泛, 其特点是相邻两个码字中仅有一位发生变化, 使得相邻码字的变化最小, 能够有效防止数据传输过程当中出现的错误^[57,58], 对于四位格雷码和二进制的对应关系如表 3.1 所示。

表 3.1 四位格雷码和二进制

十进制	格雷码	二进制	十进制	格雷码	二进制
0	0000	0000	8	1100	1000
1	0001	0001	9	1101	1001
2	0011	0010	10	1111	1010
3	0010	0011	11	1110	1011
4	0110	0100	12	1010	1100
5	0111	0101	13	1011	1101
6	0101	0110	14	1001	1110
7	0100	0111	15	1000	1111

为了计算两个格雷码之间的差值, 可以利用格雷码与二进制码之间转换公式, 将格雷码转换为二进制码, 然后计算两个二进制码之间的差, 即为两个格雷码的差值。对于某个格雷码 ($g_{n-1}g_{n-2}...g_2g_1g_0$) 其对应的二进制数为 ($b_{n-1}b_{n-2}...b_2b_1b_0$), 转换方法如下:

$$\begin{aligned} b_{n-1} &= g_{n-1} \\ b_{i-1} &= g_{i-1} \oplus b_i \quad i = 1, 2, \dots, n-1 \end{aligned} \quad (3-3)$$

格雷码编码方法同二进制编码方法类似，也是将整个 LED 点阵视为一个 16 位的二进制数，按照格雷码的编码规律，在每个控制周期内利用 LED 灯显示一个格雷码码字。第一个状态为 0000 0000 0000 0000，最后一个状态为 1000 0000 0000 0000，即 D4 灯点亮。最后一个状态显示结束后，返回至第一个状态，进入下一个循环周期。状态序列如图 3.5 所示：

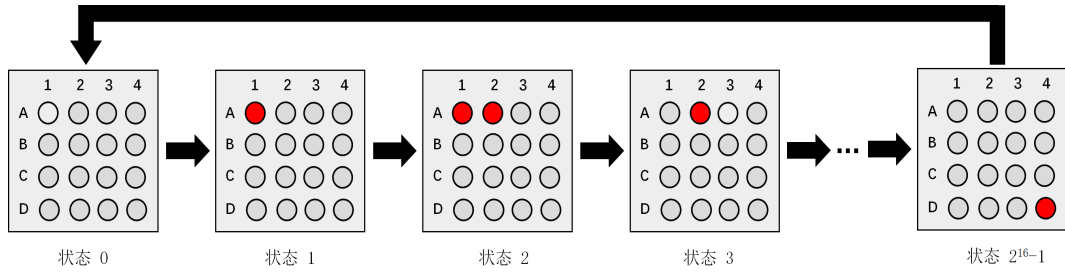


图 3.5 格雷码编码的点阵状态序列

使用该编码方法，当 LED 点阵共包含 n 个 LED 灯时，一个总的循环周期内共有 2^n 个 LED 点阵状态，则一个总的循环周期的时间长度为：

$$T = 2^n * t_{dur} \quad (3-4)$$

同二进制编码方法一样，格雷码编码方法也存在状态叠加后的图像不唯一问题。例如状态 1、状态 2 叠加和状态 2、状态 3 叠加后得到的图像相同，无法根据叠加图像识别出正确的拍摄时间。但是，当采用格雷码编码时，如果摄像头曝光时间与控制周期时间相同或小于控制周期时，摄像头将拍摄到一个状态的图像或者是两个状态的叠加图像。根据格雷码特点可知，当一个状态与其相邻状态叠后，只有一位发生变化，因此识别误差相对于二进制编码方法较小。例如，对于 4 位二进制编码，第 3 个状态 0011 与第 4 个状态 0100 叠加得到第 7 个状态 0111，与真实状态相差 3 或 4 个控制周期。而对于 4 位格雷码编码，第 3 个状态 0010 与第 4 个状态 0110 叠加得到第 4 个状态 0110。但是当调整控制周期，使得拍摄图像中有多个状态叠加时，并不能保证只有一位发生变化，仍然会产生较大识别误差。

3.3.4 分组叠加编码

为了解决二进制编码和格雷码编码存在的问题，在格雷码编码的基础上提出分组叠加编码方法。由于在检测过程中拍摄到 LED 点阵状态叠加是不可避免的，而且状态叠加造成的识别误差是由于叠加状态后得到的码字不唯一所导致，因此可以设计一种编码方式，使得在两种状态叠加时，得到的叠加码字唯一且按规律排列，即使得叠加码字为格雷码。以 4 位编码为例，状态 1、2、3 显示的码字分别为 0000、0001、0010，则相邻状态叠加得到的叠加码字为 0001、0011，满足格雷码编码要求。全部编码序列如图 3.6 所示：

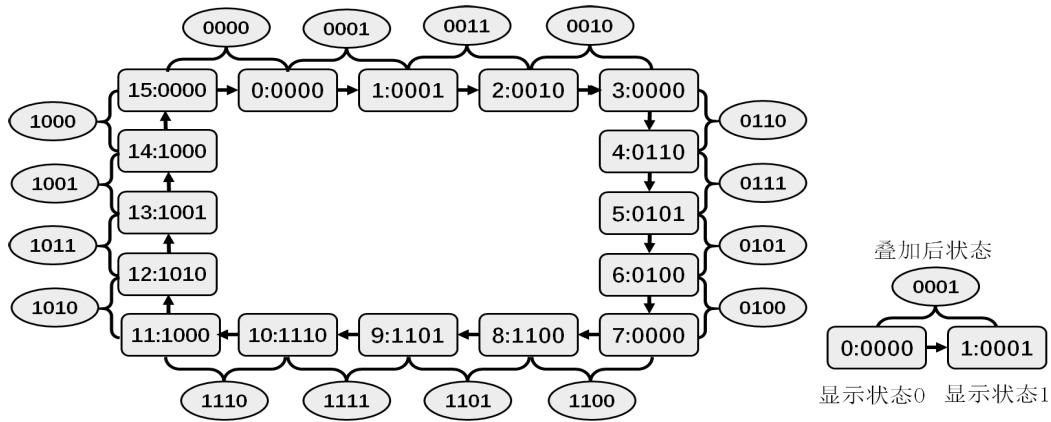


图 3.6 分组叠加编码的码字序列和叠加结果

设定摄像头曝光时间为 k 倍 (k 为正整数) 控制周期时间长度，则大部分情况下在一个曝光周期内，摄像拍摄到的图像是 $k + 1$ 个 LED 点阵状态的叠加。仅当曝光开始时间与某一控制周期开始时间相同时，可拍摄到 k 个状态叠加，但此情况出现概率较低。对于一个 4×4 LED 点阵，取 $k = 4$ ，将每行 4 个 LED 灯视为一组，共分为 4 组。每一组分别按照上述编码方法进行编码，显示编号 0 到 15 共 16 个码字。每个码字显示的持续时间为 4 个控制周期，但是 4 组 LED 灯码字分别相差一个控制周期发生变化，这样即可以保证每个控制周期都有一组码字发生变化，又可以保证 LED 点阵的状态在各个控制周期内不相同。当摄像头对 LED 点阵进行拍摄时，能够拍摄到 $k + 1 = 5$ 个控制周期的 LED 点阵状态的叠加图像，此叠加图像内，各组 LED 灯所显示的码字都发生且仅发生了一次变化，因此 4 组 LED 灯的叠加结果分别满足格雷码序列，且拍摄到的叠加图像是唯一的。

当第一组 LED 灯显示完显示序列内所有 16 个码字后，如图 3.7 所示状态 63，为避免与状态 0 相同进入小循环，使第一组 LED 灯即将显示的码字序号减 1，即显示第 14 个码字，然后继续按照上述规律进行变化。当第一组 LED 灯全部循环结束，即显示完状态 511 后，将第二组显示的码字序号减 1 继续进行循环。显示序

列如图 3.7所示：

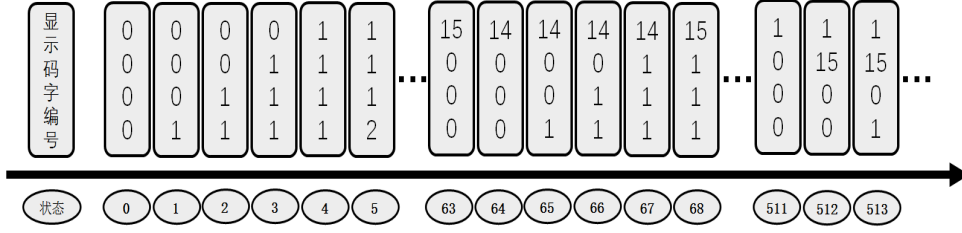


图 3.7 分组叠加编码的显示序列

具体而言，当摄像头拍摄到状态 0 至 4 的叠加图像后，4 个组均为第 0 和第 1 个码字叠加，因此得到的叠加码字分别为 0001、0001、0001、0001。当拍摄到状态 1 值 5 时，得到的叠加码字分别为 0001、0001、0001、0011。每次拍摄都能得到不同的叠加状态。通过叠加码字可以确定是哪两个显示码字叠加得到，就可以确定摄像头拍摄到的是哪五个状态叠加，从而可以判断摄像头拍摄的开始时刻位于哪个状态内，但是无法判断开始时刻在这个状态内的准确位置。因此检测精度为该状态的持续时间，即控制周期时间长度 t_{dur} ，即曝光时间的四分之一。该编码方法的检测精度（A）与摄像头曝光时间 t_{exp} 满足如下关系：

$$A = t_{dur} = t_{exp}/k \quad (3-5)$$

则为了提高检测精度，可以降低曝光时间或者增大分组数量。但是一般情况下曝光时间不能过低，因此需要分组数量 k 较大，也就需要 LED 点阵中 LED 灯数量较多。

该方法存在的最大问题在于可操作性欠缺。由于要求各组内相邻两个显示状态叠加而成的叠加状态满足格雷码编码要求，显示状态序列的选取并不存在一定规律。上文示例当中每组包含 4 个 LED 灯，而当每组内 LED 灯数量发生变化时，显示序列同样需要发生变化，且不能保证一定存在满足条件的序列。因此在实际应用当中，该编码方法可操作性较低。

3.3.5 进位流水编码

此方法综合流水编码和分组叠加编码。对于 4×4 LED 点阵，将点阵分为 4 组，每行 1 组，每组等级依次升高。每组内各 LED 灯按照流水编码方法依次点亮。对于等级最低组，即第一组，每个控制周期熄灭当前点亮的 LED 灯，点亮下一个 LED 灯。组内最后一个 LED 灯熄灭后，第一个 LED 灯点亮，依次循环。对于其

他等级组，只有当其低等级组的最后一个 LED 灯熄灭，发送进位信号给该等级组，该组内才会发生 LED 灯亮灭变化，熄灭当前点亮的 LED 灯，点亮下一个 LED 灯，如图 3.8 所示：

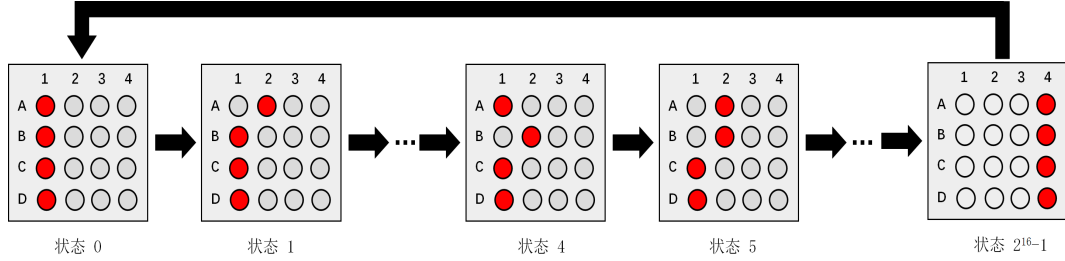


图 3.8 进位流水编码的点阵状态序列

假设控制周期的时间长度为 t_{dur} ，将 LED 点阵分为 k 组，每组包含 $n_i (i = 0, 1, 2, \dots, k-1)$ 个 LED 灯，每组内各个 LED 灯亮灭变化的周期长度为 t_i 。最低等级组的变化周期与控制周期时间相同，即 $t_1 = t_{dur}$ 。因为只有当第一组内所有 LED 灯循环变化一次后才会向第二组发送进位信号，所以第二等级组的变化周期为 $n_1 * t_1$ 。同理则第 i 组的变化周期为：

$$t_i = t_{dur} * \prod_{j=1}^{i-1} n_j \quad (3-6)$$

通常情况下，摄像头的曝光时间 t_{exp} 要长于控制周期，因此在摄像头拍摄到的图像内有一个以上的 LED 灯被点亮。同时， t_{exp} 要短于最高等级组内所有 LED 灯变化周期之和 $n_k * t_k$ ，因此在摄像头拍摄到的图像内有一个以上的 LED 灯是熄灭的。一定可以找到第 i 组，使其满足

$$t_i \leq t_{exp} \leq n_i * t_i \quad (3-7)$$

当拍摄到的图像内出现状态叠加时，等级低于 i 的组内所有 LED 灯均被点亮；第 i 组内部分 LED 灯被点亮，部分熄灭；等级高于 i 的组内有一个或两个 LED 灯被点亮。对于等级高于 i 的任一组，当与其相邻的低等级组的最后一个和第一个 LED 灯被点亮时，说明有进位信号产生，因此该组内有两个灯点亮，否则仅有一个灯点亮。当拍摄到一张满足以上描述的 LED 点阵图片后，首先找到第一个 LED 灯非全部点亮的等级组 i ；然后判断第 i 组及等级高于 i 的组内第一个被点亮的 LED

灯，标记其位置分别为 $P_i, P_i + 1, \dots, P_k$ 。则摄像头的拍摄时间为：□ □ □ □ □

$$T = \sum_{j=i}^k p_j * t_j \quad (3-8)$$

根据公式 3-8 可知，拍摄时间的检测精度为第 i 组内各个 LED 灯的变化周期 t_i 。根据公式 3-6 可知为了提高检测精度，可以减小控制周期和低等级组内 LED 灯的数量。同时公式 3-7 要求 $t_{exp} \leq n_i * t_i$ ，需要同时增加第 i 组内 LED 灯的数目。因此采用进位流水编码方法进行拍摄时间检测时，可以通过减少控制周期长度和低等级组内 LED 灯的数量，增加高等级组内 LED 灯的数量来提高检测精度。

3.3.6 编码方法比较

针对以上 4 种编码方法，利用包含 n 个 LED 灯的矩阵，在控制周期为 t_{dur} 的条件下进行比较，比较结果如表 3.2 所示：

表 3.2 编码方法比较结果

十进制	格雷码	二进制	十进制	格雷码	二进制
0	0000	0000	8	1100	1000
1	0001	0001	9	1101	1001
2	0011	0010	10	1111	1010
3	0010	0011	11	1110	1011
4	0110	0100	12	1010	1100
5	0111	0101	13	1011	1101
6	0101	0110	14	1001	1110
7	0100	0111	15	1000	1111

通过比较可以看出，流水编码总循环周期过短，需要检测器包含较多 LED 灯才能满足检测需求；格雷码编码在状态叠加的情况下存在识别误差，无法准确检测拍摄时间，且需要摄像头曝光时间较短；分组格雷码在进行高精度检测时需要摄像头较短的曝光时间或者检测器包含较多 LED 灯；进位流水编码克服了以上三种方法存在的问题，能够进行高精度检测。

3.4 图像检测算法

3.5 本章小结

第 4 章 多摄像头系统的拍摄时间检测

- 4.1 本章引言
- 4.2 主要研究内容和挑战
- 4.3 本文的主要贡献

第 5 章 多摄像头系统的拍摄时间检测

- 5.1 本章引言
- 5.2 主要研究内容和挑战
- 5.3 本文的主要贡献

第 6 章 总结与展望

6.1 工作总结

6.2 未来展望

参考文献

- [1] Wilburn B, Joshi N, Vaish V, et al. High performance imaging using large camera arrays. *ACM Transactions on Graphics (TOG)*, volume 24. ACM, 2005. 765–776.
- [2] 顾西存, 黄颖, et al. 基于代数多重网格的图像拼接中动态目标的检测 dynamic object detection in image mosaic based on algebraic multigrid. *Computer Science and Application*, 2016, 6(02):94.
- [3] Stanford Computer Graphics Laboratory. The stanford multi-camera array[EB/OL]. [2017-03-23]. <http://graphics.stanford.edu/projects/array/>.
- [4] Wilburn B, Joshi N, Vaish V, et al. High-speed videography using a dense camera array. *Computer Vision and Pattern Recognition*, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2. IEEE, 2004. II–II.
- [5] Wilburn B, Joshi N, Vaish V, et al. High performance imaging using large camera arrays. *ACM Transactions on Graphics (TOG)*, volume 24. ACM, 2005. 765–776.
- [6] Wilburn B, Smulski M, Lee H H K, et al. The light field video camera. *Media Processors 2002*, 2002, 4674:29–36.
- [7] Vaish V, Wilburn B, Joshi N, et al. Using plane+ parallax for calibrating dense camera arrays. *Computer Vision and Pattern Recognition*, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 1. IEEE, 2004. I–I.
- [8] Vaish V, Garg G, Talvala E V, et al. Synthetic aperture focusing using a shear-warp factorization of the viewing transform. *Computer Vision and Pattern Recognition-Workshops*, 2005. CVPR Workshops. IEEE Computer Society Conference on. IEEE, 2005. 129–129.
- [9] Parziale G, Diaz-Santana E, Hauke R. The surround imagertm: A multi-camera touchless device to acquire 3d rolled-equivalent fingerprints. *International Conference on Biometrics*. Springer, 2006. 244–250.
- [10] Taj M, Cavallaro A. Multi-camera track-before-detect. *Distributed Smart Cameras*, 2009. ICDSC 2009. Third ACM/IEEE International Conference on. IEEE, 2009. 1–6.
- [11] Bellotto N, Sommerlade E, Benfold B, et al. A distributed camera system for multi-resolution surveillance. *Distributed Smart Cameras*, 2009. ICDSC 2009. Third ACM/IEEE International Conference on. IEEE, 2009. 1–8.
- [12] Fiore L, Fehr D, Bodor R, et al. Multi-camera human activity monitoring. *Journal of Intelligent and Robotic Systems*, 2008, 52(1):5–43.
- [13] Kettner V, Zabih R. Bayesian multi-camera surveillance. *Computer Vision and Pattern Recognition*, 1999. IEEE Computer Society Conference on., volume 2. IEEE, 1999. 253–259.
- [14] Griffiths A D, Coates A J, Jaumann R, et al. Context for the esa exomars rover: the panoramic camera (pancam) instrument. *International Journal of Astrobiology*, 2006, 5(03):269–275.
- [15] Bell J, Squyres S, Herkenhoff K, et al. Mars exploration rover athena panoramic camera (pancam) investigation. *Journal of Geophysical Research: Planets*, 2003, 108(E12).

- [16] Apple Inc. iphone 7 - apple[EB/OL]. [2017-03-23]. <http://www.apple.com/iphone-7/>.
- [17] 华为技术有限公司. Huawei p9[EB/OL]. [2017-03-23]. <http://consumer.huawei.com/cn/mobile-phones/p9/index.html>.
- [18] Chang T H, Gong S. Tracking multiple people with a multi-camera system. Multi-Object Tracking, 2001. Proceedings. 2001 IEEE Workshop on. IEEE, 2001. 19–26.
- [19] Kim K, Davis L S. Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. European Conference on Computer Vision. Springer, 2006. 98–109.
- [20] Zha Z J, Zhang H, Wang M, et al. Detecting group activities with multi-camera context. IEEE Transactions on Circuits and Systems for Video Technology, 2013, 23(5):856–869.
- [21] Saini M, Atrey P K, Mehrotra S, et al. W3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video. Multimedia Tools and Applications, 2014, 68(1):135–158.
- [22] Wang X. Intelligent multi-camera video surveillance: A review. Pattern recognition letters, 2013, 34(1):3–19.
- [23] Strecha C, Von Hansen W, Van Gool L, et al. On benchmarking camera calibration and multi-view stereo for high resolution imagery. Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. Ieee, 2008. 1–8.
- [24] Baker P, Aloimonos Y. Complete calibration of a multi-camera network. Omnidirectional Vision, 2000. Proceedings. IEEE Workshop on. IEEE, 2000. 134–141.
- [25] Kurillo G, Li Z, Bajcsy R. Wide-area external multi-camera calibration using vision graphs and virtual calibration object. Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on. IEEE, 2008. 1–9.
- [26] Theriault D H, Fuller N W, Jackson B E, et al. A protocol and calibration method for accurate multi-camera field videography. Journal of Experimental Biology, 2014. jeb–100529.
- [27] Knorr M, Niehsen W, Stiller C. Online extrinsic multi-camera calibration using ground plane induced homographies. Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013. 236–241.
- [28] Heng L, Bürki M, Lee G H, et al. Infrastructure-based calibration of a multi-camera rig. Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014. 4912–4919.
- [29] Zhang H, Zeng W, Chen X. Foreground based borderline adjusting for real time multi-camera video stitching. Image and Graphics, 2009. ICIG'09. Fifth International Conference on. IEEE, 2009. 355–360.
- [30] Malesa M, Malowany K, Kujawska M. Multi-camera dic system with spatial data stitching procedure for measurements of engineering objects. Photonics Letters of Poland, 2014, 6(4):157–159.
- [31] Zhong Z, Peng Y, Li X, et al. Color balance and seam-line removal for multi-camera stitching image. Journal of Next Generation Information Technology, 2014, 5(2):47.

- [32] Lu Y, Wang K, Fan G. Photometric calibration and image stitching for a large field of view multi-camera system. *Sensors*, 2016, 16(4):516.
- [33] 朱海宽, 万丽雯. 基于 pci 总线多路图像采集卡的研究. *自动化技术与应用*, 2008, 27(2):93–96.
- [34] 刘晓军. 采用 hd-sdi 的高清视频采集卡的设计与实现. *电视技术*, 2009, 33(1):91–93.
- [35] 王剑非, 孟彦杰, 赵勇. 基于 linux 操作系统的视频采集卡驱动程序设计. *微计算机信息*, 2007, (01Z):94–96.
- [36] 姜广文, 晁志超, 江和平, et al. 基于外触发和软件控制的多摄像机同步采集处理系统. *应用光学*, 2009, (5):756–760.
- [37] 国蓉, 焦旸, 高明, et al. 具有外同步功能的 ccd 视频监控摄像机硬件设计. *西安工业大学学报*, 2014, 34(4):268–273.
- [38] Prochazka A, Kubicek M, Pavelka A. Multicamera systems in the moving body recognition. *Multimedia Signal Processing and Communications, 48th International Symposium ELMAR-2006 focused on. IEEE*, 2006. 45–48.
- [39] Litos G, Zabulis X, Triantafyllidis G. Synchronous image acquisition based on network synchronization. *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on. IEEE*, 2006. 167–167.
- [40] Zitnick C L, Kang S B, Uyttendaele M, et al. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics (TOG)*, volume 23. ACM, 2004. 600–608.
- [41] 黄芳, 崔希民, 何建设. 基于 gps 时间同步的分布式数据采集与监测系统研究. *矿山测量*, 2004. 15–17.
- [42] 刘团结, 童庆禧, 郑兰芬. 一种航空多光谱数字相机系统的设计与实现. *测绘通报*, 2003. 7–10.
- [43] Mills D, Martin J, Burbank J, et al. Network time protocol version 4: Protocol and algorithms specification. Technical report, 2010.
- [44] Correll K, Barendt N, Branicky M. Design considerations for software only implementations of the ieee 1588 precision time protocol. *Conference on IEEE*, volume 1588, 2005. 11–15.
- [45] Svoboda T, Hug H, Van Gool L. Viroom—low cost synchronized multicamera system and its self-calibration. *Joint Pattern Recognition Symposium. Springer*, 2002. 515–522.
- [46] Rai P K, Tiwari K, Guha P, et al. A cost-effective multiple camera vision system using firewire cameras and software synchronization. *Proceedings of the 10th International Conference on High Performance Computing, Hyderabad, India*, volume 1720, 2003.
- [47] Ahrenberg L, Ihrke I, Magnor M. A mobile system for multi-video recording. *1st European Conference on Visual Media Production (CVMP)*, 2004. 127–132.
- [48] Cho H. Time synchronization for multi-hop surveillance camera systems. *Theory and Applications of Smart Cameras. Springer*, 2016: 283–305.
- [49] Shrestha P, Weda H, Barbieri M, et al. Synchronization of multiple video recordings based on still camera flashes. *Proceedings of the 14th ACM international conference on Multimedia. ACM*, 2006. 137–140.

- [50] Shrsttha P, Barbieri M, Weda H. Synchronization of multi-camera video recordings based on audio. Proceedings of the 15th ACM international conference on Multimedia. ACM, 2007. 545–548.
- [51] Yan J, Pollefeys M. Video synchronization via space-time interest point distribution. Advanced Concepts for Intelligent Vision Systems, volume 1, 2004. 12–21.
- [52] Macmillan T, Campbell S P, Newman D A, et al. Auto-alignment of image sensors in a multi-camera system, March 24, 2015. US Patent 8,988,509.
- [53] Schaffner M. Vehicle vision system utilizing camera synchronization, November 1, 2016. US Patent 9,481,301.
- [54] 戴琼海, 林星. 检验多视角采集系统的同步性的方法和装置. 中国, 2011, CN102082957B.
- [55] 赵莉, 查冠华, 田洁. 高速摄影机数据记录系统研究. 光子学报, 1992, 21(2):176–180.
- [56] Zhao Q, Chen Y Q. High-precision synchronization of video cameras using a single binary light source. Journal of Electronic Imaging, 2009, 18(4):040501–040501.
- [57] Mehta H, Owens R M, Irwin M J. Some issues in gray code addressing. VLSI, 1996. Proceedings., Sixth Great Lakes Symposium on. IEEE, 1996. 178–181.
- [58] Bitner J R, Ehrlich G, Reingold E M. Efficient generation of the binary reflected gray code and its applications. Communications of the ACM, 1976, 19(9):517–521.

致 谢

衷心感谢导师 xxx 教授和物理系 xxx 副教授对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 xxx 教授热心指导与帮助，不胜感激。感谢 xx 实验室主任 xx 教授，以及实验室全体老师和同学们的热情帮助和支持！本课题承蒙国家自然科学基金资助，特此致谢。

感谢 THUTHESIS，它的存在让我的论文写作轻松自在了许多，让我的论文格式规整漂亮了许多。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1989 年 8 月 12 日出生于辽宁省阜新市。

2008 年 9 月考入清华大学自动化系，2012 年 7 月本科毕业并获得工学学士学位。

2014 年 9 月免试进入清华大学计算机科学与技术系攻读硕士学位至今。

发表的学术论文

- [1] Ding Xu, Tao Pin. Synchronization Detection of Multicamera System based on LED Matrix. International Conference on Embedded Software and Systems, 2016
- [2] 丁旭, 陶品. 基于 LED 点阵的摄像机拍摄时间检测方法. Conference on Pervasive Computing of China, 2016