

# **视觉伪装系统算法设计**

**2012 年 8 月**

## 目 录

<b>1 研究背景（即之前写的研究方案） .....</b>	<b>1</b>
1.1 研究目标 .....	1
1.2 研究意义及应用方向 .....	1
1.3 国内外研究现状 .....	1
1.4 本文的组织结构 .....	3
<b>2 设计方案.....</b>	<b>4</b>
2.1 总体方案 .....	4
2.2 硬件平台 .....	4
2.3 软件方案（需根据第二版程序修改） .....	5
2.4 视觉伪装算法 .....	6
<b>3 目标抠图.....</b>	<b>8</b>
3.1 寻找目标区域 .....	8
3.1.1 单屏情况 .....	8
3.1.2 多屏情况 .....	10
3.2 拟合目标轮廓 .....	10
3.2.1 多边形拟合 .....	10
3.2.2 多边形削减 .....	12
3.3 计算透视变换参数 .....	12
<b>4 纹理合成.....</b>	<b>14</b>
4.1 基于示例的图像修补 .....	14
4.1.1 算法思想 .....	15
4.1.2 算法流程 .....	17
4.1.3 纹理合成结果 .....	19
4.2 透视反变换 .....	21
<b>5 颜色校正.....</b>	<b>23</b>
5.1 颜色空间 .....	24
5.1.1 CIE1931-RGB 颜色空间 .....	24
5.1.2 CIE1931-XYZ 颜色空间 .....	25
5.1.3 Lab 颜色空间 .....	26
5.1.4 RGB 颜色空间 .....	28

5.2	基于 Lab 颜色空间的颜色校正 .....	30
<b>6</b>	<b>相似性评估 .....</b>	<b>31</b>
6.1	Lab 相似性度量 .....	31
6.2	（如果 206 提出还需纹理评估或其它评估方法，则增加此章内容） .....	31
<b>7</b>	<b>存在问题（内部人士参考） .....</b>	<b>32</b>



# 1 研究背景（即之前写的研究方案）

## 1.1 研究目标

通过视觉伪装技术削弱目标的光学特征，缩小目标与背景的颜色差别，达到降低目标显著性和改变目标外形视觉的目的，使敌方的可见光相机、电视摄像机等光学探测、跟踪、瞄准设备和系统不能或不易发现目标的可见光信号。

## 1.2 研究意义及应用方向

可见光侦察是军用装备面临的主要战场侦察威胁之一。目标的可见光暴露特征主要体现在其尺寸大小、外形特征以及与背景的光谱反射差别等方面。传统的可见光隐身手段有：在物体的表面涂上与背景颜色一致的迷彩和在武器表面罩上伪装网等。这两种手段虽然实施起来较简单，但隐身效果只在两种情况下较为明显——目标静止或目标在变化不大的背景下运动，因而影响了武器装备的机动性能<sup>[1, 2]</sup>。

课题研究如何利用视觉伪装技术使目标外壳与周围环境融为一体，使外壳的光学特征随周围环境的变化而发生改变，从而适应复杂多变的背景。研究成果可应用于坦克、军用车辆、弹药库等地面武器装备和设施，使其躲避来自空中的可见光侦察、实现动态、主动的智能可见光隐身。

## 1.3 国内外研究现状

隐身技术是战术思想的革命，是战争和技术发展的必然结果。由于它针对的是现代电子传感器，人们也称之为信息化战争时代的“电子迷彩服”。所谓隐身技术，是指降低飞机、导弹、舰艇、坦克等目标的可探测特征，使敌方探测设备难以发现的综合性技术。采用该技术，可避免被探测和跟踪，从而避免被摧毁，成为作战平台提高防护力、加强作战突然性的主要技术途径<sup>[3]</sup>。

隐身技术包括雷达隐身技术、红外隐身技术、可见光隐身技术和声波隐身技术等。其中发展最快、应用最广的是雷达隐身技术<sup>[3]</sup>。随着军事技术的发展，可见光成像制导系统发现、跟踪、拦截目标的能力越来越强，飞机、坦克等武器系统所面临的威胁日趋严重。1960年美国的U-2侦察机在苏联的领空被击落，使得以美国为首的军事强国重新意识到可见光隐身的重要性，开始了系统性的研究。20世纪70年代，美国在可见光隐身技术方面取得重大进展，并将其运用在越南战场。越战中美国空军战机上表面采用丛

---

林绿、中间绿色和棕褐色三种色调，下表面采用较明快的“非常浅的灰色”，较好地隐藏在热带雨林的环境下，有效降低了被发现的几率。

现有可见光隐身技术可分为迷彩隐身、伪装网隐身和动态可见光隐身，下面进行具体介绍<sup>[1]</sup>。

### （1）迷彩隐身

迷彩隐身就是将涂料、颜料或其它材料直接喷涂或粘贴在目标表面，是用以减少、改变目标和背景之间波谱反射和辐射特性差异而实施的一种隐身技术。其目的是要缩小目标与背景之间的亮度差别以及色度差别、降低目标的显著性和改变目标的视觉外形。迷彩隐身技术的关键是选取斑块颜色和确定斑块的形状大小。

迷彩隐身是最基本的隐身技术，目前已形成一套完整的技术系统：迷彩涂料的规格化、迷彩作业的机械化和应用计算机技术设计建彩图案以及隐身效果的评估。国内外的研究人员正在积极研制宽频谱迷彩隐身涂料，使其不仅在可见光和近红外波段有较好的隐身效果，而且在中远红外和雷达波段也具有一定的隐身作用。美国研制的一种多功能隐身涂层，其可见光的光谱特性与背景基本一致，中红外  $3\mu\text{m}$ - $5\mu\text{m}$  的发射率为 0.5-0.9，远红外  $8\mu\text{m}$ - $14\mu\text{m}$  的发射率为 0.6-0.95，在毫米波 1mm-10mm 处的吸收率为 10dB-15dB。

迷彩隐身虽然实施比较简单、收效快，但要得到较好的隐身效果，则需武器装备静止或运动在相对变化不大的背景中，使武器装备的机动性能受到了限制。

### （2）伪装网隐身

伪装网是用来遮盖车辆、坦克等军事武器装备或目标以达到低可探测率的隐身织物。历经几十年的发展研究，伪装网已由防护单一波段、模拟单一背景发展到具备防紫外线、可见光、近红外、中远红外和雷达波段、适用于不同背景的产品。

美国和瑞典在伪装网的研制方面始终处于世界领先地位。美国特里达因公司研制的超轻型伪装网，主要用于陆军直升机和其它大型装备的伪装，其有效防护波段包括可见光、近红外、热红外和雷达波段。瑞典 Barracuda 公司开发的多谱超轻型伪装系统 (BMS—ULCAS)，其网面是双层结构，即将切花成三维 U 形的装饰布固定在不挂钩的基布上。可见光、近红外的伪装是通过装饰布实现的，雷达和热红外伪装是由装饰布和基布共同实现的，每平方米的质量为 210g。另外，结合不同装备的伪装要求，该公司研制生产出多种专用伪装系统，如机动伪装系统(MCS)、高机动车载系统(HMBS)等。

伪装网的使用简单，可以迅速达到遮蔽目标的效果，但它也有和迷彩隐身相类似的缺陷，即限制了武器装备的机动性能。

### （3）动态可见光隐身

迷彩隐身和伪装网隐身虽然实施起来较简单，但隐身效果只在两种情况下较为明显：目标静止或目标在变化不大的背景下运动。因而影响了武器装备的机动性能。为了克服传统隐身技术的缺陷，军事专家积极探索和研究动态隐身技术，以适应复杂多变的

背景。动态隐身技术的发展依赖于“智能”变色材料的合成开发。目前，研究较多的有热致变色、光致变色和电致变色三种智能变色材料，它们都有可能应用于武器装备的智能蒙皮，其亮度和颜色会随着背景光的改变而相应地改变，从而实现动态隐身。动态可见光隐身技术因为有着巨大的军事应用价值，所以一些军事强国的研究人员在该技术上投入大量的精力，但按照目前公开报道的资料来看，他们的研究仍处于初级阶段。从对智能材料研究进展的分析可以看出：电致变色材料因性能优越于热致变色材料和光致变色材料，有可能率先成为动态隐身的智能外衣。这种隐身技术克服了迷彩和伪装网隐身的固有的缺陷，提高了目标的机动性、作战范围以及全天候作战的能力。

## 1.4 本文的组织结构

第 1 章介绍视觉伪装系统的研究背景、研究意义及应用方向、国内外研究现状。

第 2 章综述视觉伪装系统的总体设计方案，包括硬件平台和软件方案。

第 3 章至第 6 章分别阐述视觉伪装系统中涉及到的图像处理算法：目标抠图、纹理合成、颜色校正和相似性评估。

第 7 章指明可改进的方向（内部人士参考）。

---

## 2 设计方案

### 2.1 总体方案

拟选用一台工控机、一台笔记本、一台平板电脑、三块液晶屏、一个 USB 摄像头、一个摄像头支架、一个无线 AP 构成视觉伪装系统。其中，工控机配备有多屏卡，物理实现伪装图像的分屏显示。在此硬件平台的基础上，开发相应的系统控制软件和应用软件，使得三块液晶屏能够分屏显示与周围环境相似的伪装图像，笔记本/平板电脑可通过无线网络控制程序运行、调节程序参数。

整个处理过程如下：（1）工控机控制液晶屏显示特定初始图像（如红色）；（2）**目标抠图**：USB 摄像头捕捉当前目标图像，传送至工控机，工控机采用抠图技术将目标（即液晶屏）所在图像区域抠除；（3）**纹理合成**：利用纹理合成技术插补目标区域所在的图像空洞，并通过透视变换将合成图像调整至实际目标尺寸，得到伪装图像；（4）**颜色校正**：液晶屏分屏显示伪装图像，根据 USB 摄像头拍摄的伪装结果进行颜色校正；（5）**相似性评估**：对伪装效果（USB 摄像头拍摄图像中目标和背景的相似性）进行评估。其中，前 4 步由工控机完成，第 5 步由笔记本通过 WiFi 获得工控机保存的 USB 摄像头拍摄图像，对图像进行相似性评估。

本章简单介绍系统的硬件平台和软件方案，第 3 章至第 6 章对系统涉及到的图像处理算法（目标抠图、纹理合成、颜色校正和相似性评估）进行详细介绍。**算法介绍时以单屏显示为例，多屏显示只在目标抠图模块部分略有不同，见 3.2.2 节。**

### 2.2 硬件平台

硬件平台如图 2.1 所示，下面分别介绍各部分功能。

多屏卡：对于一幅完整图像，硬件实现 3 块显示屏分屏显示功能。

工控机：系统核心。实现目标抠图、纹理合成、颜色校正等算法。

液晶屏 1/2/3：通过 VGA 接口与工控机相连，显示初始图像和伪装图像。

摄像头：通过 USB 接口偶与工控机相连，负责图像采集。

摄像头支架：将摄像头固定于平板电脑上方约 1.5 米处，可进行高度、伸出度调节。

笔记本/平板电脑：通过 WiFi 控制工控机程序运行、调节程序参数。

无线 AP：提供 WiFi 通信。



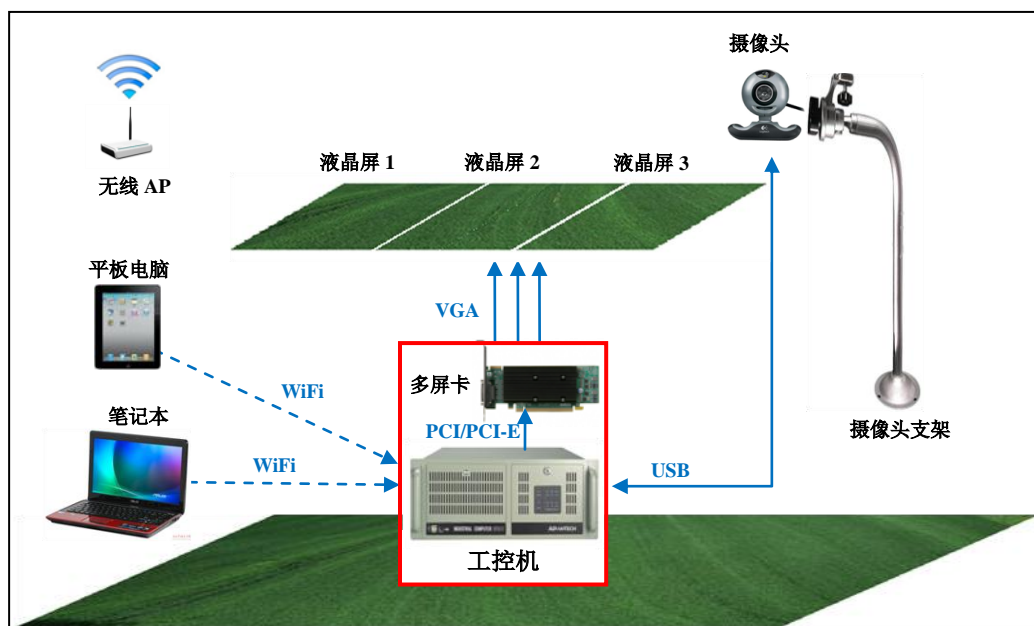


图 2.1 硬件平台

## 2.3 软件方案（需根据第二版程序修改）

软件方案如图 2.2 所示。工控机为系统核心，包含图像采集、抠图、纹理合成、图像显示、接收参数、接收指令等模块。初始时，图像显示模块输出待显示的初始图像（如红色），图像采集模块通过 USB 读取一帧摄像头拍摄图像，然后调用抠图模块提取待伪装目标区域，纹理合成模块利用纹理合成技术对伪装目标区域生成伪装图像，由图像显示模块将其显示在液晶屏上，注意，由于采用多屏卡硬件实现图像分屏显示，故图像分块模块可省。接收参数模块通过 WiFi 接收由笔记本发送参数模块传送过来的参数，接收指令模块接收由遥控的发送指令模块通过 WiFi 发送来的指令，将其传送给图像显示模块，调节伪装图像的颜色。上述通信均采用 socket 编程。

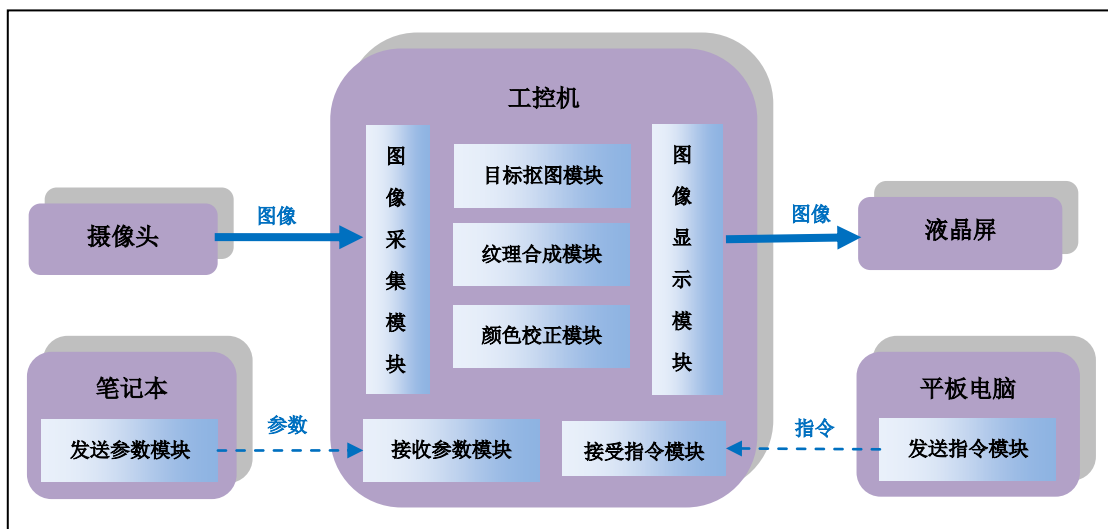


图 2.2 软件方案

## 2.4 视觉伪装算法

视觉伪装系统涉及到目标抠图、纹理合成、颜色校正和相似性评估四个图像处理算法模块。目标抠图、纹理合成和颜色校正模块在工控机（服务器）上运行，是视觉伪装算法的核心部分，相似性评估模块在笔记本（客户端）运行。图 2.3 给出了各模块之间的关系图。目标抠图模块对输入的初始图像  $I_{init}$  提取目标所在区域，得到目标模板  $M_T$ ，并计算透视变换参数；纹理合成模块根据初始图像  $I_{init}$  和目标模板  $M_T$  对目标区域进行纹理合成，并将合成结果通过透视反变换调整至实际目标尺寸，得到伪装图像  $C$ ；颜色校正模块根据当前拍摄图像  $I$ 、目标背景模板  $M_{B\&T}$  对伪装图像  $C$  进行颜色校正，目标背景模板  $M_{B\&T}$  根据目标模板  $M_T$  生成；相似性评估模块对当前拍摄图像  $I$  中的目标区域和背景区域的相似性进行评估。各模块详细介绍见本文第 3 章至第 6 章。

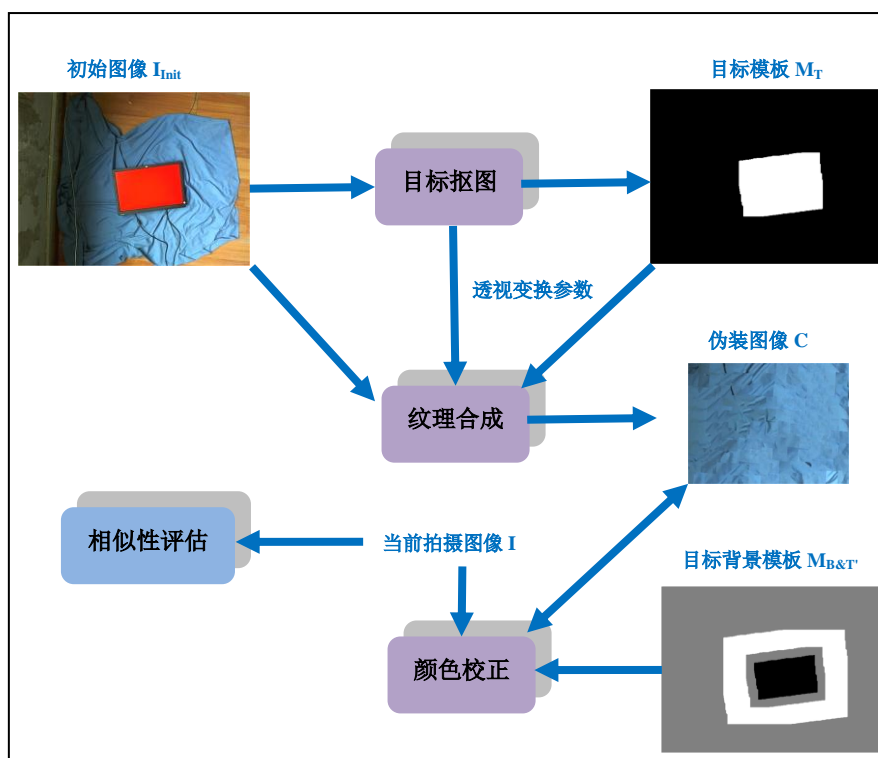


图 2.3 视觉伪装系统算法模块

### 3 目标抠图

液晶屏显示初始图像，摄像头采集到图像后，需利用抠图技术确定目标所在区域和透视变换参数，以便后续用纹理合成技术填补目标区域，并通过透视变换将合成图像调整至实际目标尺寸，得到伪装图像。

目标抠图模块输入初始图像  $I_{init}$ ，输出目标模板  $M_T$  和透视变换参数，共包含三个步骤，如图 3.1 所示：（1）寻找目标区域：根据初始图像  $I_{init}$ ，确定目标所在区域，生成目标模板  $M_T$ ，后续纹理合成算法对目标区域合成纹理；（2）拟合目标轮廓：用四边形拟合目标区域，得到拍摄目标四边形顶点坐标  $\{(x_i, y_i)\}_{i=1}^4$ ；（3）计算透视变换参数：根据拍摄目标和实际目标四对顶点坐标，计算透视变换参数，拍摄目标纹理合成后，经过透视反变换得到适合显示的伪装图像。

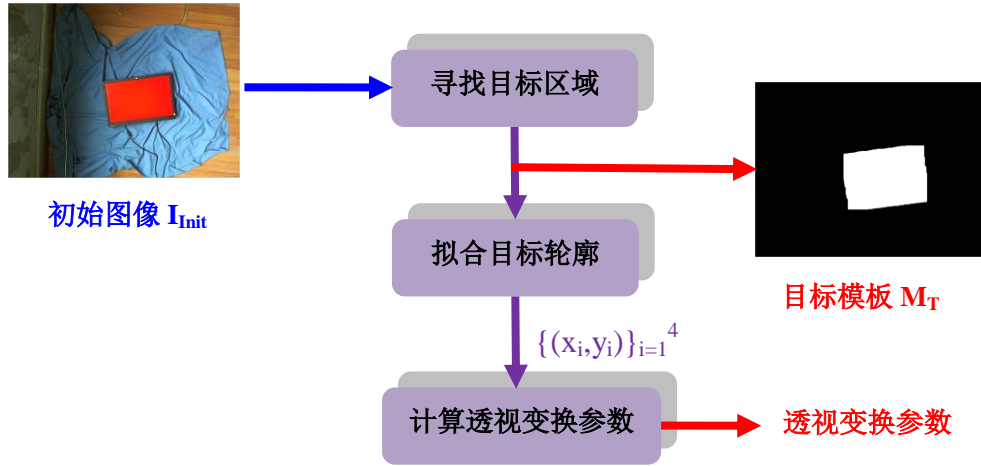


图 3.1 目标抠图

#### 3.1 寻找目标区域

##### 3.1.1 单屏情况

利用区域生长法寻找目标区域。区域生长（Region Growing）是指将成组的像素或区域发展成更大区域的过程。从种子点的集合开始，将与每个种子点有某种相似属性的相邻像素合并到此区域，它是一个迭代的过程，每个种子像素点都迭代生长，直到不再有满足条件的新像素点为止<sup>[4]</sup>。

对初始图像  $I_{init}$  进行区域生长，具体算法如下：

1）定义生长条件（相似性）：原始图像位于 RGB 颜色空间，由于 Lab 颜色空间具有视觉感知均匀性（见 5.1 节），将像素点之间的相似性定义为像素点在 Lab 颜色空间中

的欧氏距离，即

$$S((x_0, y_0), (x_1, y_1)) = \sqrt{(L_{(x_0, y_0)} - L_{(x_1, y_1)})^2 + (a_{(x_0, y_0)} - a_{(x_1, y_1)})^2 + (b_{(x_0, y_0)} - b_{(x_1, y_1)})^2}$$

如果相似性小于等于某个阈值  $T$ ，则认为两个像素点相似，属于同一集合。;

2) 在显示屏内部选择 1 个种子点  $(x_{seed}, y_{seed})$ ，目标初始集合为种子点：

$$T_{Init} = \{(x_{seed}, y_{seed})\}。$$

3) 从  $T_{Init}$  开始向外扩张，不断将与集合中各个像素八连通（属于集合中某个点的八邻域，如图 3.2 所示）、且满足生长条件的像素点加入集合。

N4	N3	N2
N5	(i, j)	N1
N6	N7	N8

图 3.2 点(i,j)的八邻域

4) 过程 3) 进行到不再有满足条件的新像素点加入集合为止。

图 3.3 显示了区域生长的不同阶段。

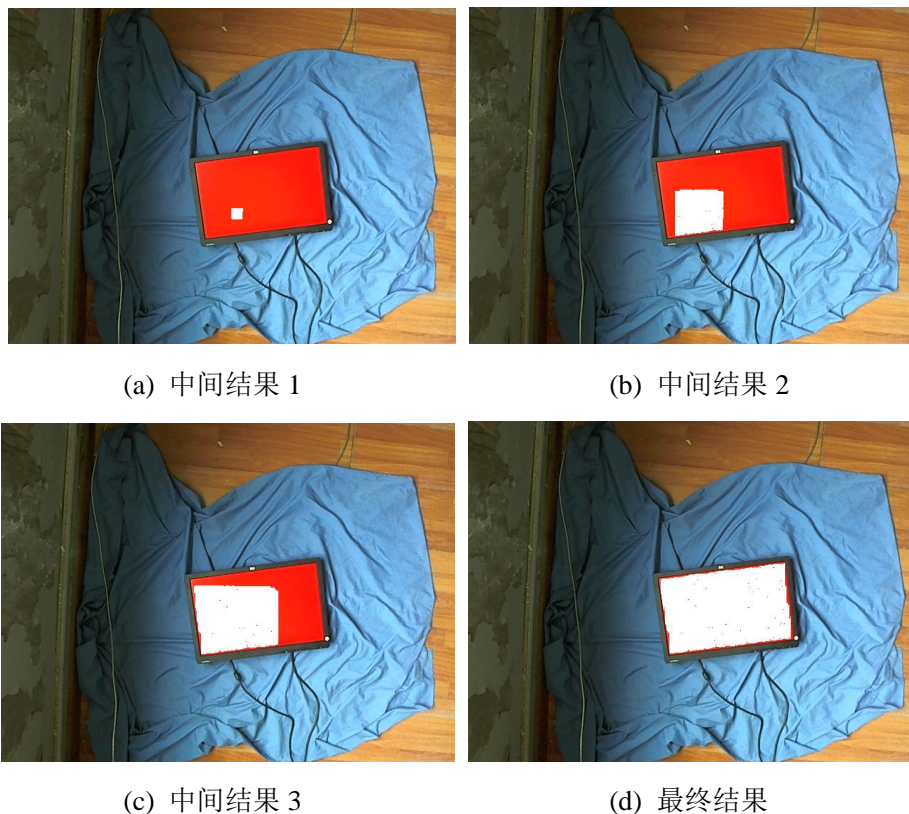


图 3.3 区域生长

用**空洞填充**算法填充目标区域中可能存在的空洞，并通过**闭操作**平滑目标区域，就得到了目标区域，由于显示屏存在边框，所以，目前得到的目标区域是显示屏的显示区

---

域，而在纹理合成中需要对整个显示屏区域生成伪装纹理，所以，对上述结果进行**膨胀操作**（膨胀元素的大小为显示屏边框大小），就可得到真正的目标区域（即整个显示屏所在区域），如图 3.4 所示。



图 3.4 目标区域

### 3.1.2 多屏情况

多屏显示下，对每个屏均选取一个种子点，分别进行区域生长，在膨胀操作中，各个屏的显示区域会自动连接在一起，形成一个完整的目标区域。

## 3.2 拟合目标轮廓

目标区域反映在摄像头拍摄图像上为一四边形，在得到目标区域模板  $M_T$  后，用四边形对目标轮廓进行拟合，输出四边形顶点  $\{(x_i, y_i)\}_{i=1}^4$ 。

记录目标的轮廓点坐标，拟合分两步进行：（1）对目标轮廓分段线性拟合，得到一个多边形；（2）将多边形边数削减为 4，得到四边形。

### 3.2.1 多边形拟合

将目标的闭合轮廓分为长度相等的两个轮廓段（分别用红色、绿色表示），分别进行分段线性拟合<sup>[5]</sup>，如图 3.5 所示。

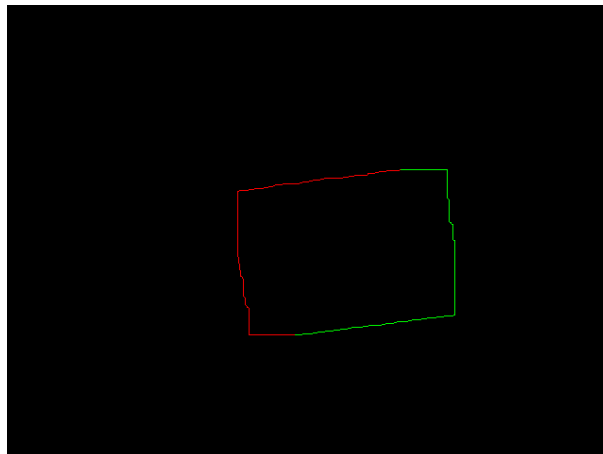


图 3.5 目标轮廓

轮廓段的分段线性拟合是一个迭代过程。对图 3.6 中的轮廓段 AB 进行拟合，步骤如下：

- 1) 计算轮廓段 AB 上所有点到直线 AB 的距离，选取距离最大的点，记为 C，C 即为新的顶点，再分别对轮廓段 AC 和 CB 重复上述步骤；
- 2) 对于某段轮廓 XY，如果距离最大点小于等于某个阈值，则认为轮廓 XY 能被直线 XY 拟合，不必再继续对其分段拟合。

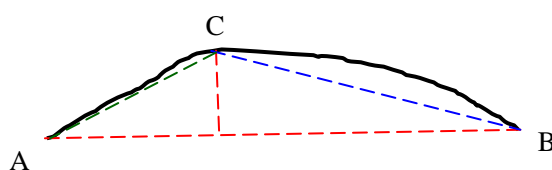


图 3.6 轮廓段拟合

目标轮廓的多边形拟合结果如图 3.7 所示。

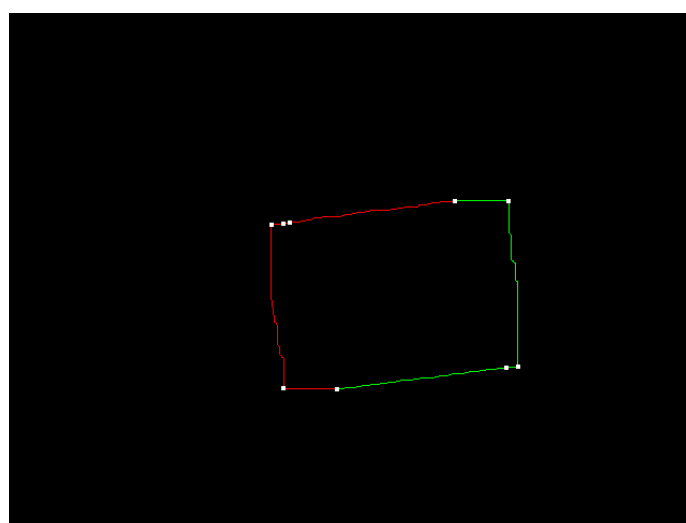


图 3.7 多边形拟合结果



### 3.2.2 多边形削减

已知实际目标为矩形，反映在拍摄图像上为一四边形，故需将拟合多边形削减为四边形。削减方法为：对当前多边形的每个顶点，计算其相邻两边所在直线的夹角（0-90度之间），将夹角最小的点去除，如此反复进行到只剩4个顶点。图 3.8 给出了图 3.7 所示多边形的削减结果（顶点用白色小方块表示）。至此，得到反映目标区域的四边形顶点坐标 $\{(x_i, y_i)\}_{i=1}^4$ 。

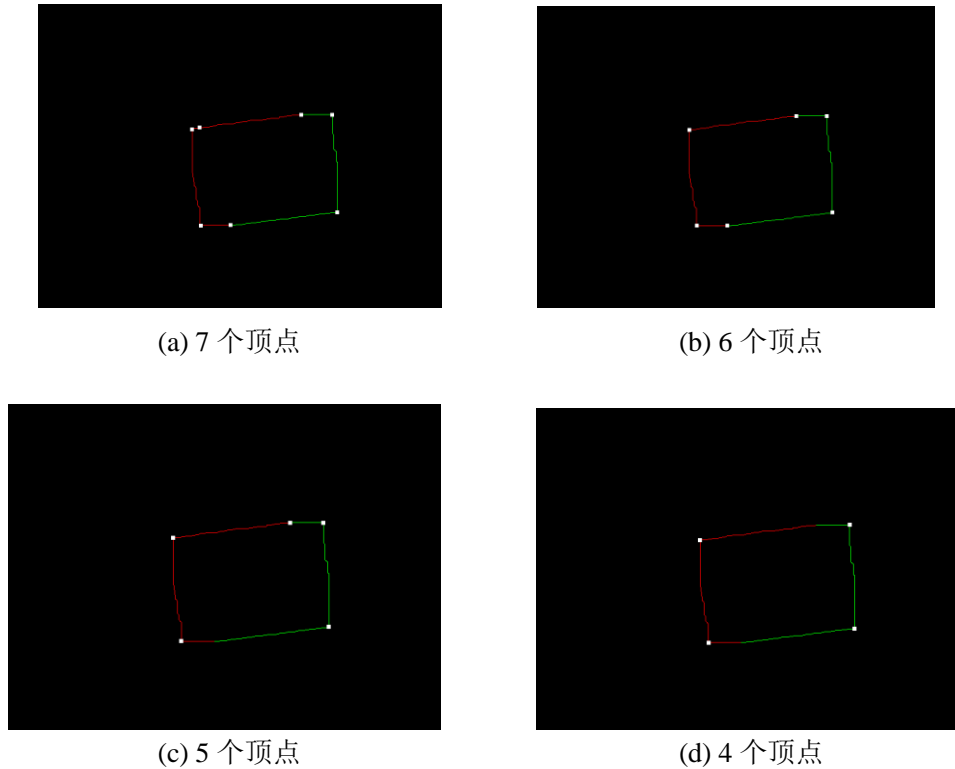


图 3.8 多边形削减

### 3.3 计算透视变换参数

摄像头拍摄到的目标与实际目标（显示屏）之前存在形状和大小的差异，对图像中目标区域合成纹理后，需将其映射到实际目标（见 2.4 节）。

实际目标与拍摄目标之间存在透视变换（Perspective Transformation），实际目标中位置 $(x, y)$ 处的像素点，经过透视变换后对应拍摄目标中位置 $(x', y')$ 处的像素点，透视变换公式如下<sup>[6]</sup>：

$$\begin{aligned} x' &= \frac{Ax + By + C}{Hx + Gy + 1} \\ y' &= \frac{Dx + Ey + F}{Hx + Gy + 1} \end{aligned}$$

通过四对一一对应的坐标点就能求取透视变换八个参数 A、B、C、D、E、F、H、



G。在 3.3 节中，已经得到拍摄目标的四个顶点坐标 $\{(x_i, y_i)\}_{i=1}^4$ ，它们对应实际目标的四个顶点，实际目标的大小即显示屏分辨率。图 3.9 给出了纹理合成后的拍摄目标与实际目标。有了透视变换参数，就可通过透视反变换将实际目标每个像素点的颜色计算出来，得到用于显示的伪装图像，具体见 4.2 节。

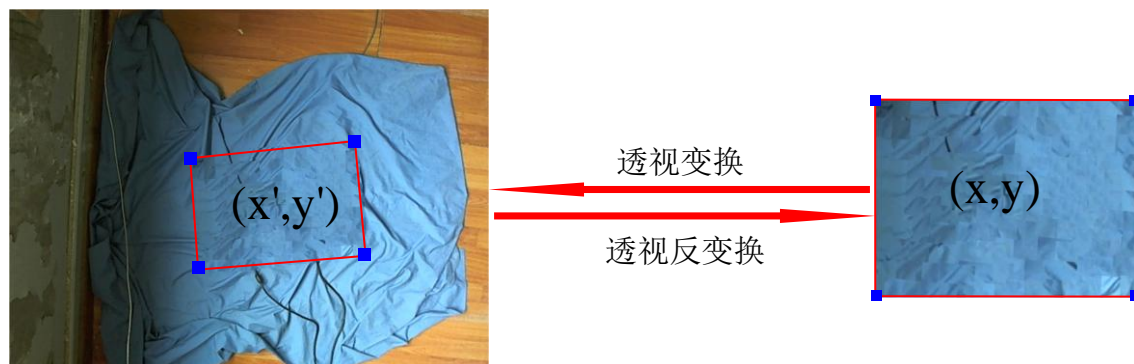


图 3.9 透视变换

## 4 纹理合成

要想削弱目标的光学特征、缩小目标与背景的颜色差别，需要用到纹理合成技术。纹理合成技术利用背景图像生成目标区域伪装图像，使得摄像头无法区分目标和其周围环境。

纹理合成模块输入初始图像  $I_{\text{Init}}$ 、目标模板  $M_T$  和透视变换参数，输出伪装图像  $C$ ，共包含两个步骤，如图 4.1 所示：（1）基于示例的图像修补：根据初始图像  $I_{\text{Init}}$  和目标模板  $M_T$ ，采用基于示例的图像修补算法对目标区域合成纹理，生成结果图  $R$ ；（2）透视反变换：根据纹理合成结果图  $R$  和透视变换参数，得到适合显示的伪装图像  $C$ 。

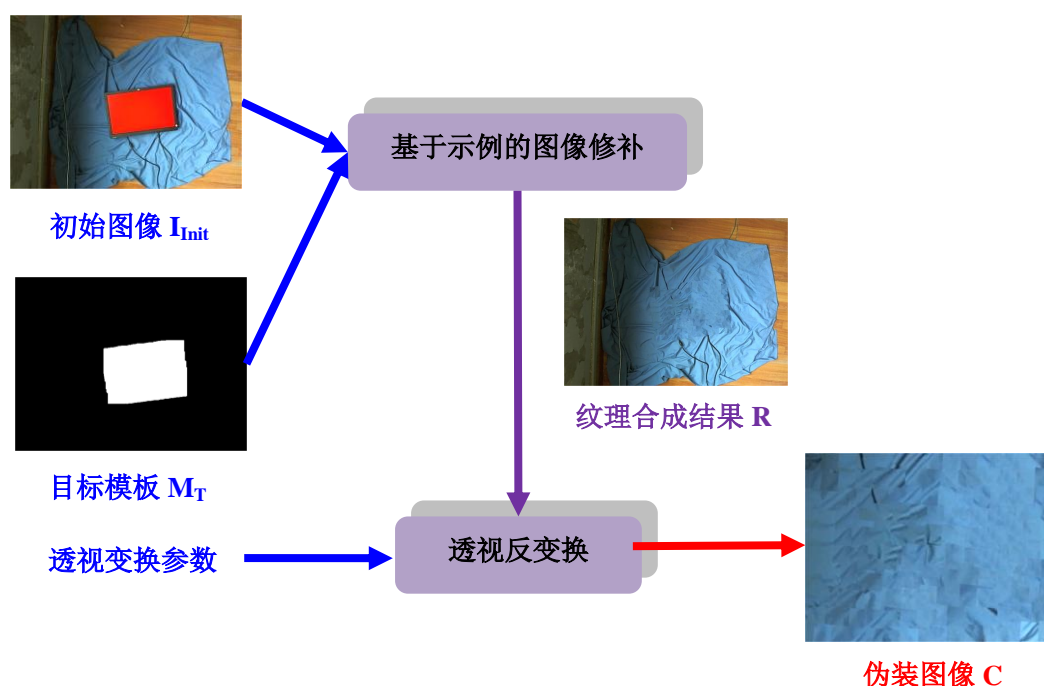


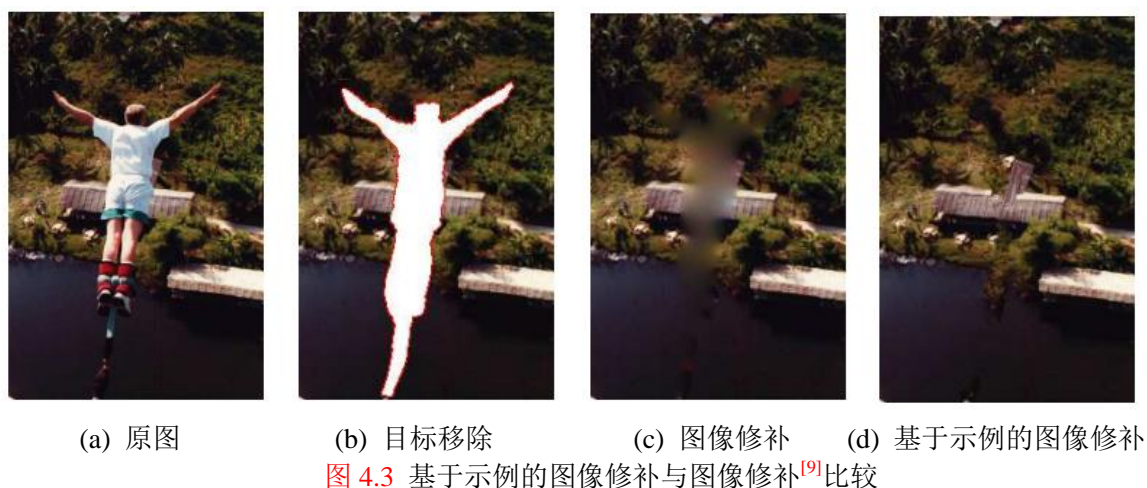
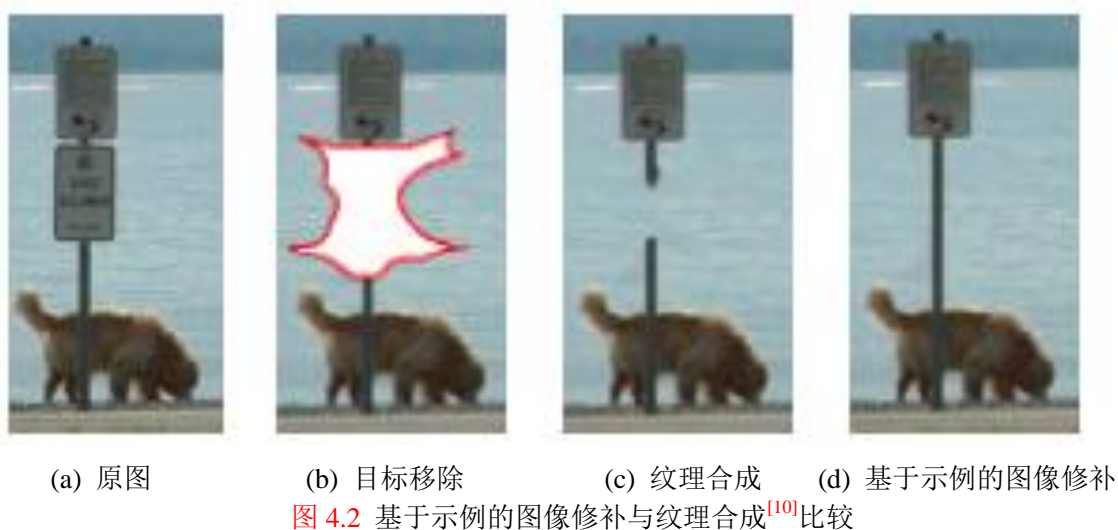
图 4.1 纹理合成

### 4.1 基于示例的图像修补

本节解决这样一个问题：将目标从图像中移除，填上人工生成的图像，使得人眼无法察觉目标移除的痕迹。现有解决方案大致分为两种<sup>[7]</sup>：（1）纹理合成（texture synthesis）：用“纯纹理”填充目标区域，这里纹理指带有适度随机性的重复二维纹理图案。其中，基于示例（exemplar-based）的纹理合成技术受到广泛关注，这种技术从源区域中采样并拷贝颜色值，从而简单有效地生成新纹理<sup>[8]</sup>；（2）图像修补（image inpainting）：通过扩散（diffusion）向目标区域传播线性结构（linear structure），这种方法受物理热流量的偏微

分方程启发，在图像恢复中得到成功应用<sup>[9]</sup>。

纹理合成技术不能有效处理真实世界场景包含的大量线性结构和复合纹理（composite texture），而图像修补技术在扩散过程中会产生显著的模糊（blur）现象。本节采用的基于示例的图像修补（exemplar-based image inpainting）算法能结合两种方法的优势，用线性结构影响基于示例的纹理合成的填充顺序，使得算法具有基于示例的纹理合成的有效性，同时能够反映目标区域周围线性结构<sup>[7]</sup>。图 4.2、4.3 分别给出了基于示例的图像修补与纹理合成<sup>[10]</sup>、图像修补<sup>[9]</sup>的比较结果。



#### 4.1.1 算法思想

算法的核心是一个等照度线（isophote）驱动的图片采样过程。等照度线垂直于梯度方向，反映线性结构。最终的图像合成质量和（1）结构与纹理的传播及（2）填充顺序有关，下面分别进行介绍<sup>[7]</sup>。

## 结构与纹理的传播

图 4.4 给出了算法结构纹理传播示意图，从中可看到基于示例的图像修补算法是如何处理结构和纹理信息的。待填补的区域，即目标（target）区域标记为  $\Omega$ ，目标轮廓标记为  $\delta\Omega$ ，在填充过程中  $\delta\Omega$  不断向目标内部推进，所以  $\delta\Omega$  又称为“填充前沿（fill front）”。供采样的源（source）区域在算法中是固定的，标记为  $\Phi$ 。假设当前需要填补的是以点  $p$  为中心的方块  $\Psi_p \in \Omega$ ，如图 4.4-(b)所示，在源区域中找到与  $\Psi_p$  已填补部分最匹配的块  $\Psi_q \in \Phi$ 。从图 4.4-(b)中可看到， $\Psi_p$  位于一个图像边缘的延长线上，所以它的相似块也位于同样的（或相似的）边缘上，如图 4.4-(c)的  $\Psi_{q'}$ 、 $\Psi_{q''}$ 。将最匹配块  $\Psi_q$  对应位置的颜色值拷贝到  $\Psi_p$  的待填补区域，就可以将等照度线自动传播到目标内部，如图 4.4-(d)所示。尽管此时边缘并不垂直于目标轮廓  $\delta\Omega$ ，被传播的结构仍然维持与源区域一致的方向。

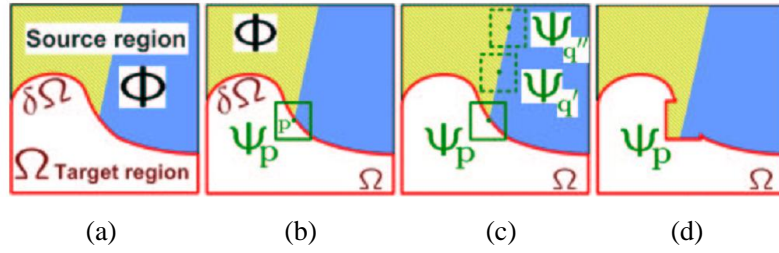


图 4.4 结构纹理传播示意图

## 填充顺序

图 4.5 给出了同心层（concentric layer）填充和理想填充顺序示意图，其中，同心层填充方式又称为洋葱剥离（onion-peel）。图 4.5-(a)为待填充的图像，目标区域为一个凹形。图 4.5-(b)、(c)、(d)算法给出了逆时针洋葱剥离的填充过程，填充结果并不理想，目标区域包含一个曲线。理想的填充算法应该给位于图像结构延长线上的那些目标区域较高的填充优先权，如图 4.5-(b')、(c')、(d')所示，显然，这种填充顺序对于目标区域的形状变化更为鲁棒。

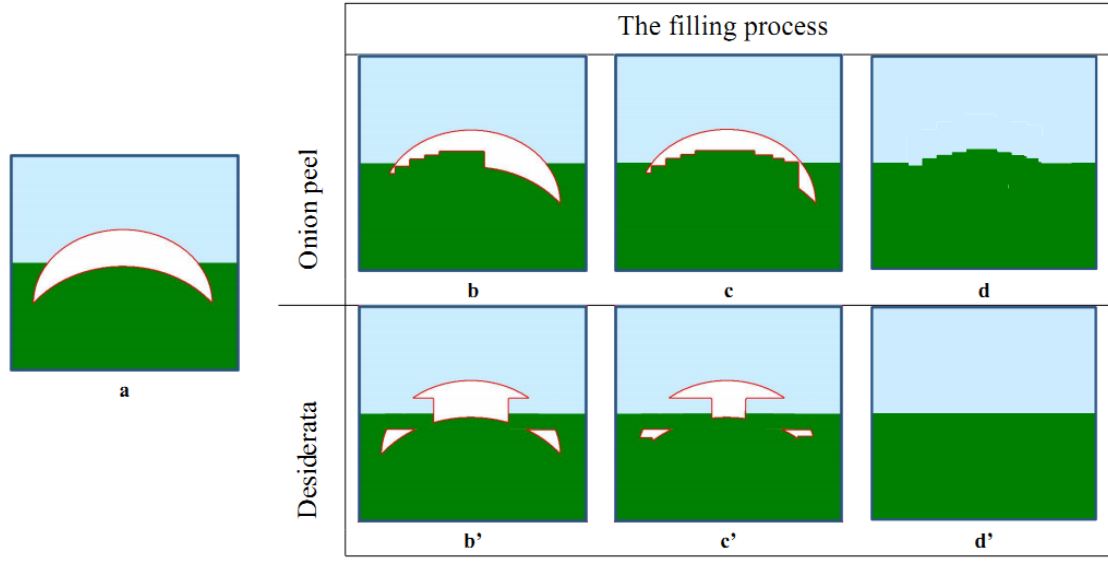


图 4.5 填充顺序示意图

基于示例的图像修补算法能克服同心层（concentric layer）填充的缺点，实现（1）线性结构的正确传播；（2）对目标区域的形状变化鲁棒；（3）平衡结构和纹理传播。

#### 4.1.2 算法流程

用  $I$  表示整个图像区域。算法需指定目标区域  $\Omega$ 、源区域  $\Phi$ 、采样块  $\Psi$  大小，其中，采样块通常比源区域的纹理元（texel）略大。每个已被填充的像素都关联一个颜色值和一个置信度（confidence）值，置信度值反映该像素颜色值的置信度。算法运行过程中，填充前沿  $\partial\Omega$  上的像素还关联一个临时的优先权（priority）值，优先权决定了不同像素的块填充顺序，优先权最高的像素最先被填充。然后，算法进入迭代，直到所有像素都被填充。迭代包含三个步骤：计算优先权、传播纹理和结构信息、更新置信度值<sup>[7]</sup>。

##### 1. 计算优先权

图 4.6 给出了优先权计算示意图。对于填充前沿  $\partial\Omega$  上某一点  $p$ ，关联的图像块是  $\Psi_p$ ，它的优先权  $P(p)$  为两项的乘积：

$$P(p) = C(p)D(p),$$

其中， $C(p)$  称为置信度项（confidence term）， $D(p)$  称为数据项（data term），具体定义如下：

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}, \quad D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha},$$

其中， $|\Psi_p|$  为块  $\Psi_p$  的面积， $\alpha$  为归一化因子，确保数据项在 [0,1] 之间， $n_p$  是填充前沿



$\delta\Omega$ 在点  $p$  处的单位法向量,  $\nabla I_p^\perp$  为点  $p$  处梯度的正交向量。初始时,

$$C(p) = \begin{cases} 0, & \forall p \in \Omega, \\ 1, & \text{otherwise.} \end{cases}$$

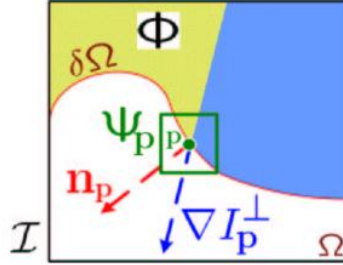


图 4.6 优先权计算示意图

置信度项  $C(p)$  依赖于  $\Psi_p$  中已填充区域的置信度, 因此, 目标区域的外围比内部具有更高的置信度, 所以置信度项可看作是同心层填充的一个近似。数据项  $D(p)$  反映等照度线击中填充前沿  $\delta\Omega$  的强度, 数据项确保线性结构能被优先合成。

从图 4.7 中可直观看出置信度项和数据项的影响, 绿色代表值较高, 红色代表值较低。图 4.7-(a) 中, 毛刺部分被较多的已填充区域包围, 具有较高的置信度, 而内陷部分被较少的已填充区域包围, 具有较低的置信度, 因此, 置信度项平滑目标轮廓使其趋向于一个圆。图 4.7-(b) 中, 在线性结构延长线上的点具有较高的数据项, 因此, 数据项确保线性结构能传播到目标区域。

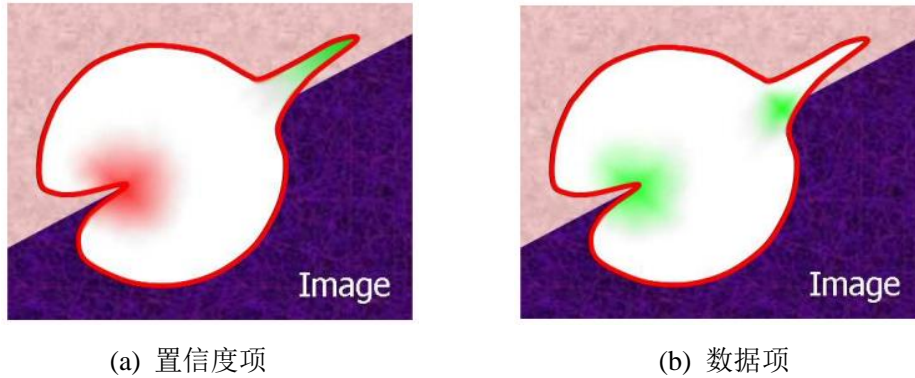


图 4.7 置信度项和数据项的影响

## 2. 传播纹理和结构信息

计算完填充前沿  $\delta\Omega$  所有点的优先权后, 找出优先权最大的块进行填充, 即  $\Psi_{\bar{p}}$ ,  $\bar{p} = \arg \max_{p \in \delta\Omega} P(p)$ 。传统图像修复算法通过扩展进行基于像素的传播, 这种方式容易引起图像平滑, 从而导致显著的模糊现象, 如图 4.3 所示。基于示例的图像修补算法通过从源区域  $\Phi$  采样直接传播纹理, 即在源区域  $\Phi$  寻找与  $\Psi_{\bar{p}}$  最相似的块,

$$\Psi_{\bar{q}} = \arg \min_{\Psi_q \in \Phi} d(\Psi_{\bar{p}}, \Psi_q),$$

其中， $d(\Psi_a, \Psi_b)$  定义为两个块的像素颜色值的平方差的总和（Sum of Squared Differences, SSD）。注意  $\Psi_{\bar{p}}$  只有部分像素有颜色值，只需计算这部分区域的 SSD 即可。由于 Lab 颜色空间具有视觉感知均匀性（见 5.1 节），颜色值用 Lab 颜色空间表示。寻找到最相似块  $\Psi_{\bar{q}}$  后，用  $\Psi_{\bar{q}}$  对应颜色值填充  $\Psi_{\bar{p}}$  中待填充的区域，实现纹理和结构信息的传播。

### 3. 更新置信度值

$\Psi_{\bar{p}}$  中的待填充区域被填充后，这部分像素点的置信度值与点  $\bar{p}$  处相同，即

$$C(p) = C(\bar{p}) \quad \forall p \in \Psi_{\bar{p}} \cap \Omega。$$

随着填充的进行，置信度值逐渐下降，越靠近目标中心的颜色值越不可靠。基于示例的图像修补算法伪代码如图 4.8 所示，其中，上标  $t$  代表当前迭代。

-输入：待填充图像  $I$ ，其中源区域为  $\Phi$ ，目标区域（待填充区域）为  $\Omega$ ；块大小。

-初始化：提取初始边界  $\partial\Omega^0$ 。

-重复：

- 1a: 识别当前边界  $\partial\Omega^t$ ，如果  $\partial\Omega^t = \emptyset$ ，填充结束，退出。
- 1b: 计算边界点优先权  $P(p)$ ， $\forall p \in \partial\Omega^t$ 。
- 2a: 寻找优先权最大的块  $\Psi_{\bar{p}}$ ， $\bar{p} = \arg \max_{p \in \partial\Omega^t} P(p)$ 。
- 2b: 在源区域寻找与  $\Psi_{\bar{p}}$  最相似的示例块  $\Psi_{\bar{q}}$ ， $\Psi_{\bar{q}} = \arg \min_{\Psi_q \in \Phi} d(\Psi_{\bar{p}}, \Psi_q)$ 。
- 2c: 将图像数据从  $\Psi_{\bar{q}}$  拷贝到  $\Psi_{\bar{p}}$ ， $\forall p' \in \Psi_{\bar{p}} \cap \Omega$ 。
- 3: 更新置信度  $C(p)$ ， $C(p) = C(\bar{p})$ ， $\forall p \in \Psi_{\bar{p}} \cap \Omega$ 。

-输出：已填充的图像。

图 4.8 基于示例的图像修补算法

### 4.1.3 纹理合成结果

根据目标模板  $M_T$ ，生成纹理合成模板，如图 4.9 所示。其中，白色区域为待填充目

---

标区域 $\Omega$ ，灰色区域为源区域 $\Phi$ 。

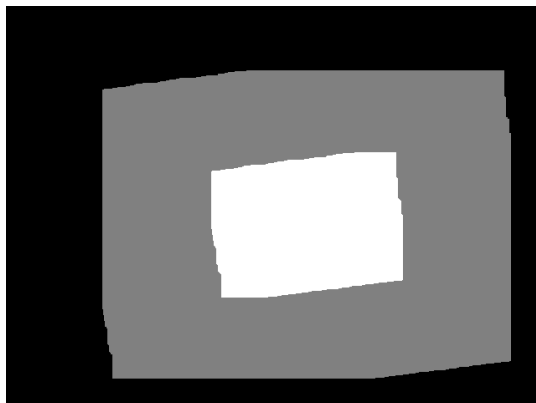


图 4.9 纹理合成模板

为提高运算速度，每次迭代从源区域中随机挑选  $N$  个块与待填充块进行比较，而不遍历整个源区域。令块大小为 14，图 4.9 所示源区域共包含 104888 个示例块，图 4.10 标记了某次迭代随机选取 2000 个示例块的位置。



图 4.10 某次迭代随机示例块位置

图 4.11 给出了纹理合成结果图  $R$ ，其中，红色线表示目标轮廓。线性结构（布褶皱、电线）被有效传播进目标区域，呈现出一个好的伪装效果。





图 4.11 纹理合成结果 R

## 4.2 透视反变换

根据纹理合成结果图 R 和目标抠图模块得到的透视变换参数，就可得到适合显示的伪装图像 C。对于伪装图像上某个像素点  $(x, y)$ ，根据透视变换公式（见 3.3 节）可计算出它在纹理合成结果图上对应的像素点坐标  $(x', y')$ 。 $(x', y')$  可能不是整数，而图像只能在整数坐标上取值，所以，采用双线性插值（Bilinear Interpolation）<sup>[11]</sup> 的方法。如图 4.12 所示，取与  $(x', y')$  相邻的四个像素点  $(x'_0, y'_0)$ 、 $(x'_1, y'_0)$ 、 $(x'_0, y'_1)$  和  $(x'_1, y'_1)$ ，其中， $x'_0$  和  $y'_0$  分别是  $x'$  和  $y'$  的整数部分， $x'_1 = x'_0 + 1$ ， $y'_1 = y'_0 + 1$ ，令  $dx = x' - x'_0$ ， $dy = y' - y'_0$ ，则伪装图像上像素点  $(x, y)$  某通道的颜色值  $L(x, y)$  为：

$$L(x, y) = (1 - dy) * ((1 - dx) * L(x'_0, y'_0) + dx * L(x'_1, y'_0)) + dy * ((1 - dx) * L(x'_0, y'_1) + dx * L(x'_1, y'_1))。$$

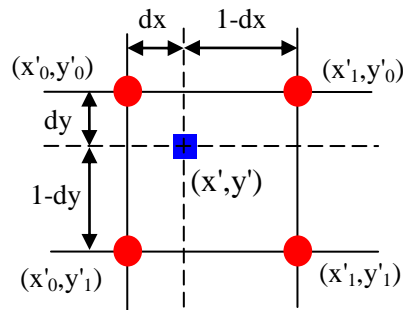


图 4.12 双线性插值

最终生成的伪装图像如图 4.13 所示，用于显示屏全屏显示。



图 4.13 伪装图像 C

## 5 颜色校正

纹理合成结果图  $R$  中，目标颜色与背景颜色相似，在数字图像层面能得到良好的伪装效果。然而，当伪装图像  $C$  在显示屏上显示时，伪装图像与周围环境确存在着不可消除的颜色偏差，使得伪装效果较差。下面分析这种偏差产生的原因。

图 5.1 给出了视觉伪装系统的图像变化过程。摄像头对真实场景进行拍摄，通过光-电变换将其转化为数字图像，本文提到的图像处理算法都是在数字图像层面处理的，这一步产生的数字图像称为“数字图像 1”，如初始图像  $I_{init}$ 、纹理合成结果图  $R$ 、伪装图像  $C$ 。显示屏通过电-光变换全屏显示伪装图像  $C$ ，然后，摄像头对此时的显示屏进行拍摄，通过光-电变换得到数字图像，称为“数字图像 2”，我们关心的正是数字图像 2 中目标（显示屏）和背景的相似性。在数字图像 1 中，通过纹理合成，使得目标与背景有较好的相似性，然而，伪装图像经过了一次摄像头电-光变换和显示屏光-电变换后，在数字图像 2 中不可能达到与数字图像 1 一样的效果。

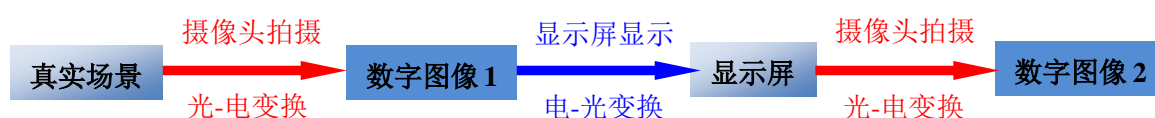


图 5.1 视觉伪装系统图像变化过

图 5.2 给出了某个场景下纹理合成结果图（数字图像 1）和摄像头拍摄的伪装效果（数字图像 2），显然，进一步地颜色校正十分必要。



图 5.2 纹理合成结果图和摄像头拍摄的伪装效果图

由于摄像头电-光变换和显示屏光-电变换不好模拟，我们设计一个负反馈控制系统，不考虑颜色偏差产生的原因，直接在 Lab 颜色空间对其进行色校正。

## 5.1 颜色空间

本节介绍几种常用的颜色空间：CIE1931-RGB、CIE1931-XYZ、Lab 和 RGB<sup>[4,12]</sup>。

### 5.1.1 CIE1931-RGB 颜色空间

颜色是人眼对可见光的一种感觉。人的视觉系统所能看到的可见光是一种电磁波信号，其波长在  $400\mu\text{m} \sim 700\mu\text{m}$  之间，如图 5.3 所示。关于颜色视觉的形成理论有多种说法，比较有影响的是扬-赫姆霍尔兹的三色学说（Three-Component Theory）。三色学说的基本思想是：人眼视网膜中有 3 种对颜色敏感的锥体细胞，它们分别对红、绿、蓝 3 种颜色最敏感，三者共同作用，使人们产生了不同的颜色感觉。



图 5.3 可见光

由三色学说的原理可知，任何一种颜色都可以通过红、绿、蓝三原色按照不同比例混合来得到。1931 年，国际照明委员会（Commission Internationale d'Eclairage, CIE）规定红、绿、蓝三种光的波长分别为：红光（R）700nm、绿光（G）546.1nm、蓝光（B）435.8nm。自然界中各种颜色都能由这 3 种原色光按一定的比例混合而成：

$$C = rR + gG + bB,$$

其中，C 为试验光的颜色，r、g、b 为颜色匹配中所需要的三原色光的相对量。以 R、G、B 为坐标单位向量的空间称为 CIE1931-RGB 颜色空间，图 5.4 给出了 CIE1931-RGB 空间的三刺激值。

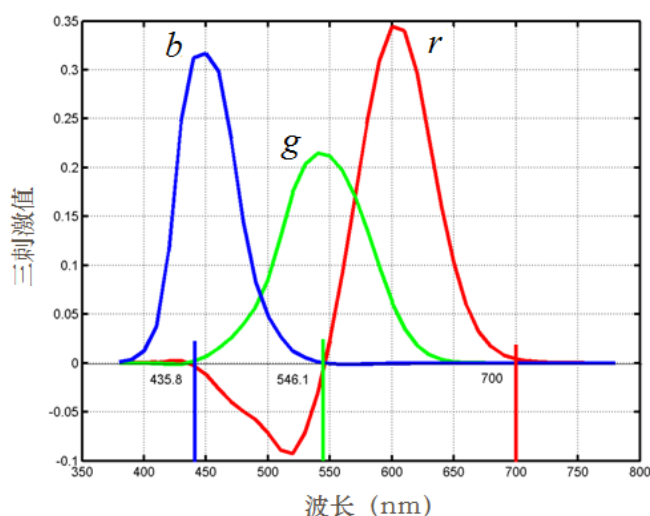


图 5.4 CIE1931-RGB 空间的三刺激值

### 5.1.2 CIE1931-XYZ 颜色空间

从图 5.4 可看出，500nm 附近的光对应的  $r$  是负数，而实际并不存在负的光强。为此，CIE 规定了三种假想的标准原色  $X$ （红）、 $Y$ （绿）、 $Z$ （蓝），构造了 CIE1931-XYZ 颜色空间，以便使颜色匹配函数的三刺激值都是正值。CIE1931-XYZ 空间的光颜色匹配函数定义如下：

$$C = xX + yY + zZ。$$

CIE1931-XYZ 空间的颜色匹配曲线如图 5.5 所示。

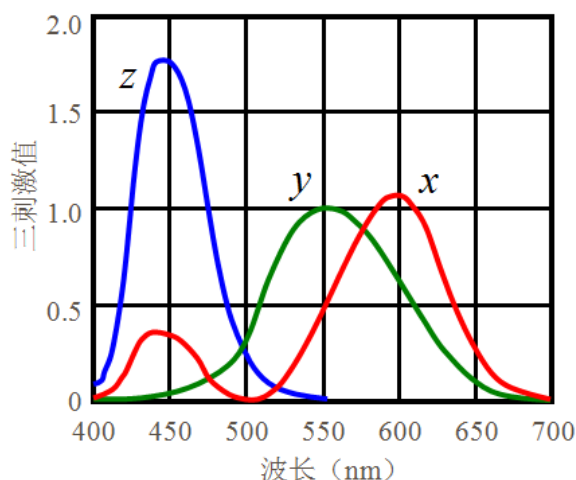


图 5.5 CIE-XYZ 空间的颜色匹配曲线

$$\text{令 } \bar{x} = \frac{x}{x+y+z}, \quad \bar{y} = \frac{y}{x+y+z}, \quad \bar{z} = \frac{z}{x+y+z}。 \quad \bar{x} + \bar{y} + \bar{z} = 1, \text{ 此时, 仅需 } \bar{x}、\bar{y} \text{ 就可}$$

确定颜色，用  $\bar{x}$ 、 $\bar{y}$  表示的颜色空间称为翼形 CIE 色度图，如图 5.6 所示。色度图的轮廓线表示所有可见光波长的轨迹，沿线的数字代表该位置的可见光的主波长。色度图的形状说明，没有一个 3 个顶点均在可见光翼形区的三角形可以完全覆盖该区域，这与前面所说的不能通过 CIE1931-RGB 颜色空间的红光、绿光、蓝光的加法混合来匹配所有颜色相吻合。

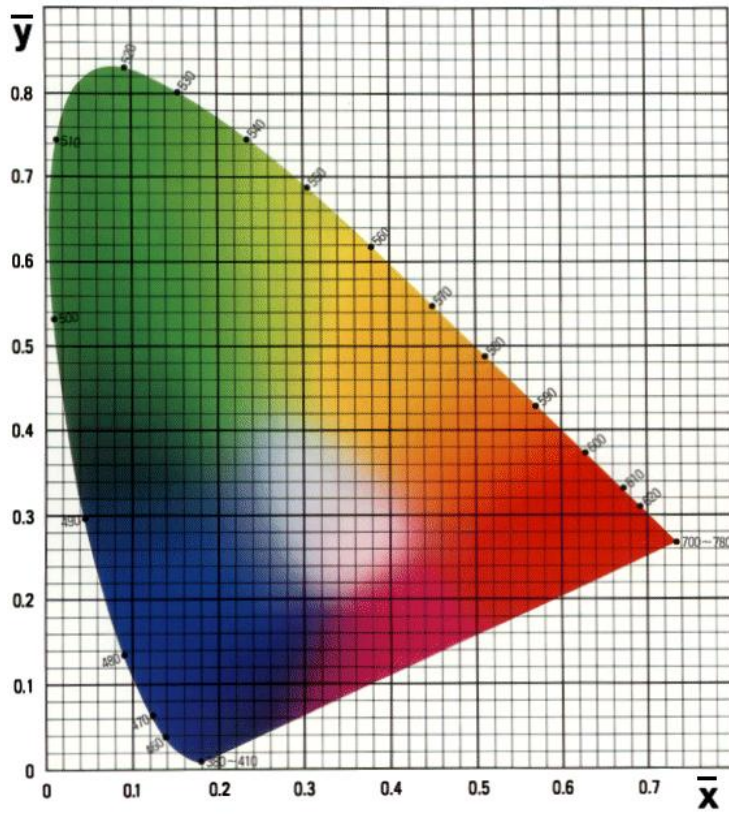


图 5.6 翼形 CIE 色度图

CIE1931-RGB 与 CIE1931-XYZ 颜色空间的转换关系如下：

$$\begin{aligned}\bar{x} &= \frac{0.49000\bar{r} + 0.31000\bar{g} + 0.20000\bar{b}}{0.66697\bar{r} + 1.13240\bar{g} + 1.20063\bar{b}}, \\ \bar{y} &= \frac{0.17697\bar{r} + 0.81240\bar{g} + 0.01063\bar{b}}{0.66697\bar{r} + 1.13240\bar{g} + 1.20063\bar{b}}, \\ \bar{z} &= \frac{0.00000\bar{r} + 0.01000\bar{g} + 0.99000\bar{b}}{0.66697\bar{r} + 1.13240\bar{g} + 1.20063\bar{b}}\end{aligned}$$

其中， $\bar{r} = \frac{r}{r+g+b}$ ， $\bar{g} = \frac{g}{r+g+b}$ ， $\bar{b} = \frac{b}{r+g+b}$ 。

### 5.1.3 Lab 颜色空间

CIE1931-XYZ 颜色空间是感知不均匀的，两种不同颜色之间的欧氏距离并不能正确反映人们色彩感觉差别的大小，这使得在 CIE1931-XYZ 空间不能直观地评价颜色。Lab 颜色空间是 CIE1976- $L^*a^*b^*$  的简称，Lab 颜色空间模仿人的视觉特性，建立线性化的颜色差异感知。在 CIE1976- $L^*a^*b^*$  颜色空间中，不同颜色之间的差异可以用它们的欧式距离来表示。图 5.7 给出了 Lab 颜色空间示意图，L 为亮度分量，表示从纯黑到纯白变化，a、b 为颜色分量，a 表示从红色到绿色的变化，b 表示从黄色到蓝色的变化。



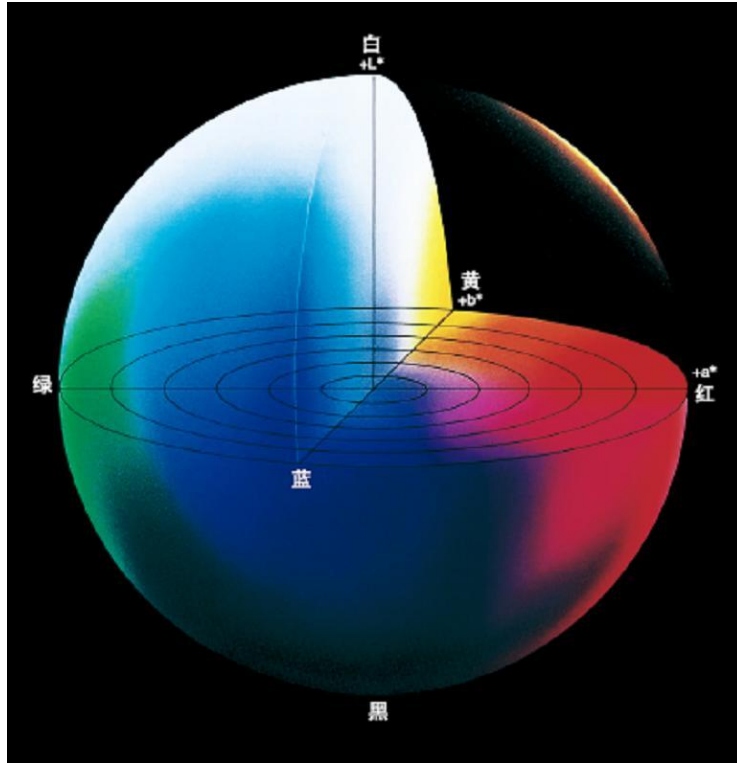


图 5.7 Lab 颜色空间

Lab 颜色空间基于了非线性压缩的 CIE1931-XYZ 颜色空间坐标。从 CIE1931-XYZ 到 Lab 颜色空间的变换公式如下：

$$\begin{aligned} L &= 116f(y/Y_n) - 16 \\ a &= 500[f(x/X_n) - f(y/Y_n)], \\ b &= 200[f(y/Y_n) - f(z/Z_n)] \end{aligned}$$

其中， $\delta = 6/29$ ， $f(t) = \begin{cases} t^{1/3}, & t > \delta^3, \\ \frac{1}{3\delta^2}t + \frac{2\delta}{3}, & \text{otherwise.} \end{cases}$ ， $X_n, Y_n, Z_n$  为 D65 标准白光源（色温大约

为 6504K）的 CIE1931-XYZ 三色刺激值， $X_n = 0.950456$ ， $Y_n = 1$ ， $Z_n = 1.088754$ 。

从 Lab 到 CIE1931-XYZ 的反变换公式如下：

$$\begin{aligned} x &= \begin{cases} X_n f_x^3, & f_x > \delta, \\ (f_x - 2\delta/3)3\delta^2 X_n, & \text{otherwise.} \end{cases} \\ y &= \begin{cases} Y_n f_y^3, & f_y > \delta, \\ (f_y - 2\delta/3)3\delta^2 Y_n, & \text{otherwise.} \end{cases} \\ z &= \begin{cases} Z_n f_z^3, & f_z > \delta, \\ (f_z - 2\delta/3)3\delta^2 Z_n, & \text{otherwise.} \end{cases} \end{aligned}$$

其中,  $f_y = (L+16)/116$ ,  $f_x = f_y + a/500$ ,  $f_z = f_y - b/200$ 。

#### 5.1.4 RGB 颜色空间

RGB 颜色空间是我们使用最多, 最熟悉的设备相关的颜色空间模型。前面所说的颜色空间都是设备无关的, 如 CIE1931-XYZ 是用真实的三原色光表示的。RGB 颜色空间是设备相关的, 不同设备对 RGB 空间内的同一点有不同的显示效果。RGB 颜色空间常用于彩色阴极射线管 (Cathode Ray Tube, CRT) 等彩色光栅图形显示设备中: 显示器使用 R、G、B 数值来驱动电子枪发射电子, 分别激发荧光屏上的红、绿、蓝三种颜色的荧光粉发出不同亮度的光线, 并通过相加混合产生各种颜色, 如图 5.8 所示。

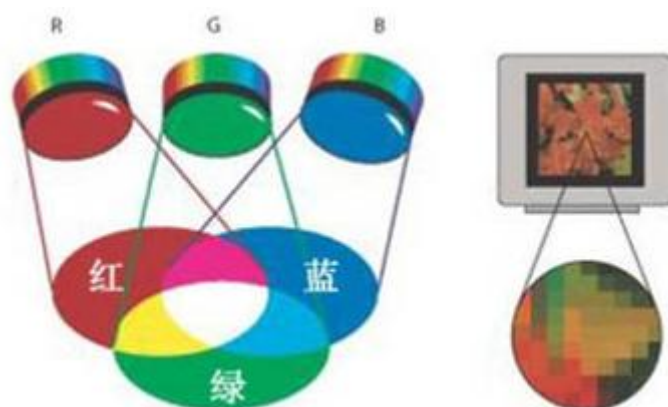


图 5.8 彩色阴极射线管显示器

RGB 颜色空间可用图 5.9 所示的单位立方体来表示。在立方体的主对角线上, 各原色的强度相等, 产生由暗到明的白色, 也就是不同的灰度值, 其中,  $(0, 0, 0)$  为黑色,  $(1, 1, 1)$  为白色。正方体的其他六个角点分别为红、黄、绿、青、蓝和品红。

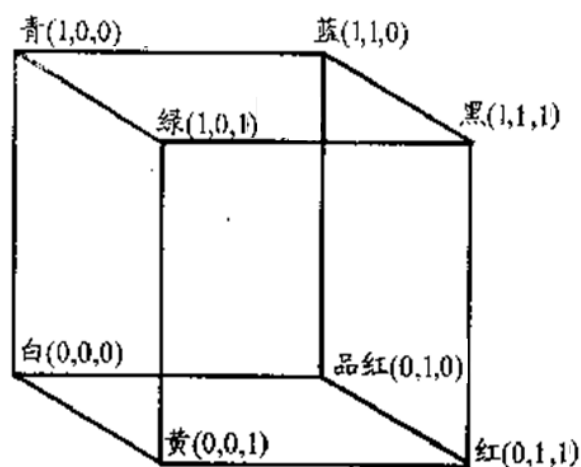


图 5.9 RGB 单位立方体



下面以 sRGB 颜色空间为例，讲述 RGB 与 CIE1931-XYZ 颜色空间的变换。sRGB 颜色空间是惠普与微软一起开发的用于显示器、打印机以及因特网的一种标准 RGB 色彩空间。图 5.10 给出了 CIE 色度图表示的 sRGB 颜色空间的色域以及原色的位置，白点 D65 显示在中间，红色位于[0.6400, 0.3300]、绿色位于[0.3000, 0.6000]、蓝色位于[0.1500, 0.0600]。

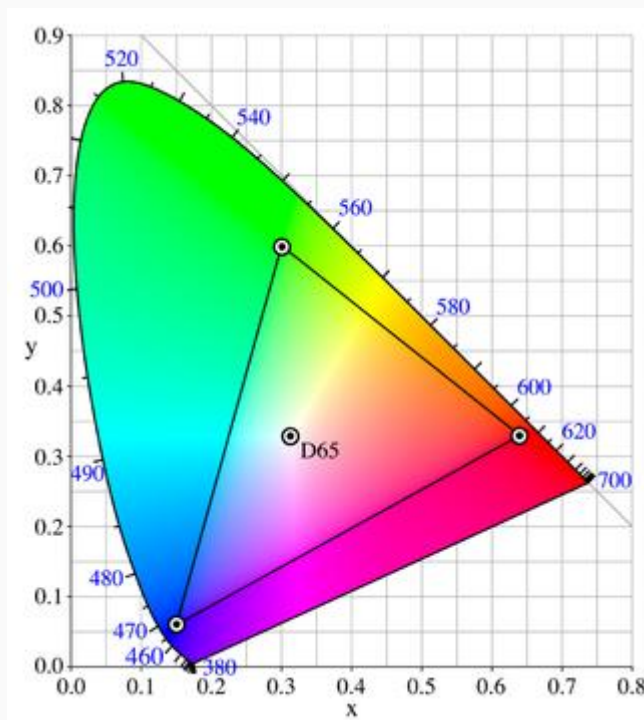


图 5.10 CIE 色度图表示的 sRGB 颜色空间的色域

从 CIE1931-XYZ 到 sRGB 的变换公式为：

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & 0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

R、G、B 的取值范围为[0, 1]。从 sRGB 到 CIE1931-XYZ 的变换公式为：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

事实上，sRGB 还考虑了 CRT 显示器 gamma 变换，在 CIE1931-XYZ 到 sRGB 变换后和 sRGB 到 CIE1931-XYZ 变换前分别进行了 gamma 校正和 gamma 变换，gamma 值设为 2.2。

## 5.2 基于 Lab 颜色空间的颜色校正

由于 Lab 颜色空间的感知均匀性，常被用来评估颜色差异（见 6.1 节）。颜色校正模块采用目标与背景区域的  $L\ a\ b$  通道均值差作为评估准则，并在 Lab 空间进行颜色校正。

设计一个负反馈系统，根据摄像头拍摄图像  $I$ （属于图 5.1 中的“数字图像 2”），对伪装图像  $C$  进行实时校正，直至图像  $I$  中目标与背景的  $L\ a\ b$  通道均值差  $\Delta L, \Delta a, \Delta b$  小于设定范围。单次校正的具体流程如图 5.11 所示。将摄像头捕捉的当前图像  $I$  变换到 Lab 空间，根据目标背景模板  $M_{B\&T}$  给出的目标和背景区域，计算目标背景的  $L\ a\ b$  通道均值，得到均值差  $\Delta L, \Delta a, \Delta b$ ，用均值差调整变换到 Lab 空间的伪装图像  $C$ ，然后将其变换回 RGB 空间，用于全屏显示。RGB 与 Lab 颜色空间之间的变换通过以 CIE1931-XYZ 颜色空间作为中介完成，具体参考 5.1 节内容。

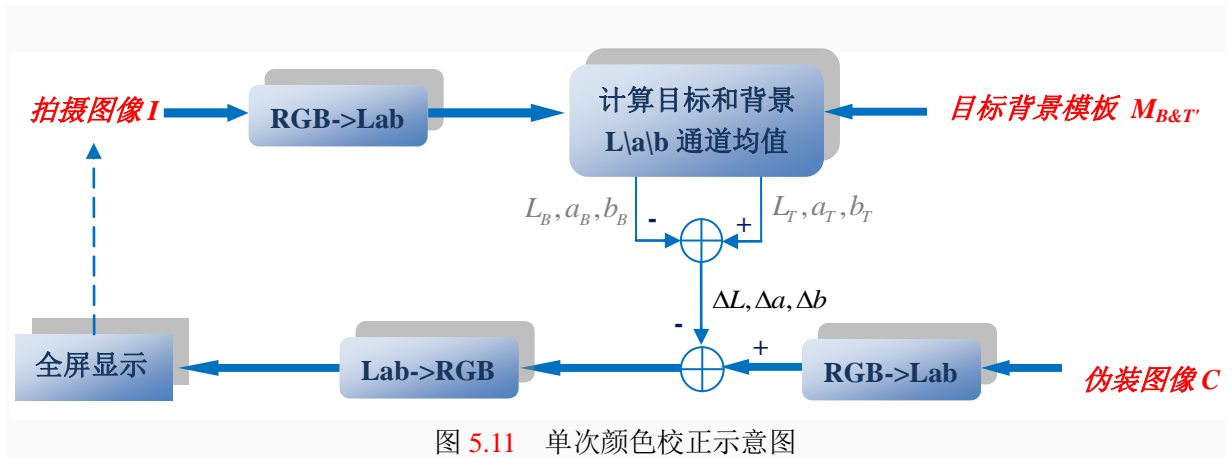


图 5.11 单次颜色校正示意图

图 5.12 给出了目标背景模板  $M_{B\&T}$ ，其中，黑色为目标区域，白色为背景区域。需要注意的是，为了避免显示屏边框的干扰，此时的目标指去除了显示屏边框的显示区域，与 3.1 节求取的目标不同。背景区域由显示屏区域，即 3.1 节得到的目标模板  $M_T$  膨胀生成，为围绕着显示屏的回字形区域，膨胀元素大小可人为设定。

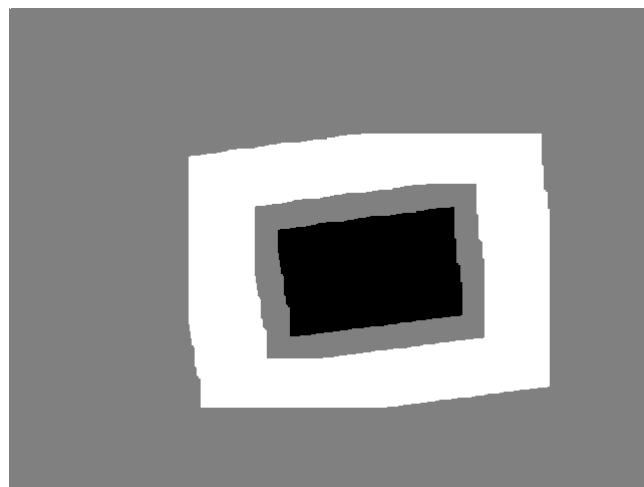


图 5.12 目标背景模板  $M_{B\&T}$

## 6 相似性评估

不同的视觉捕获设备捕捉到的图像颜色不同，如人眼和摄像头之间、不同的型号的摄像头之间，由于光照影响，同一捕获设备在不同位置上捕捉到的图像颜色也可能不同，所以，伪装算法不可能同时对多个设备（或不同位置上的同一设备）产生好的伪装效果，相似性评估仅在系统使用的摄像头上进行。

在客户端读取摄像头拍摄的当前图像，从中选定要比较的目标区域和背景区域，采用相似性度量进行相似性评估。

### 6.1 Lab 相似性度量

如 5.1 节所说，Lab 颜色空间具有感知均匀性，常被用于测定颜色差异。Lab 相似性度量比较目标区域和背景区域的 L\ab 通道均值差 $\Delta L$ 、 $\Delta a$ 、 $\Delta b$ ，并计算总色差 $\Delta E = \sqrt{\Delta L^2 + \Delta b^2 + \Delta a^2}$ 。

### 6.2 （如果 206 提出还需纹理评估或其它评估方法，则增加此章内容）

## 7 存在问题（内部人士参考）

算法存在以下问题，可讨论是否有必要改进。

- （1）伪装图像近看有块效应。
- （2）由于摄像头分辨率低，拍摄到的目标区域过小，导致无法生成高分辨率的伪装图像，伪装图像近看不够细腻。
- （3）由于显示屏具有边框，摄像头拍摄到的显示区域边缘处线性结构伪装效果不太好，有个错位，参考图 5.2 电线部分。一个解决方案是令伪装图像仅包含目标区域（显示区域+显示屏边框）中的显示区域。方法如下：在 3.2 节拟合目标轮廓时，对 3.1 节膨胀前的目标区域轮廓进行拟合，然后利用目标轮廓的四对点计算透视变换参数，这样在 4.2 节透视反变换时，就能对不包括显示屏边框的显示区域生成伪装图像。

**如需增加篇幅，可**

- （1）在附录部分增加涉及到的一些基本的图像处理算法，如膨胀、闭操作、空洞填充、求取图像梯度、计算两个块的 SSD 等。
- （2）在附录部分和各章节贴一些不同背景下的结果图。

## 参考文献

- [1]. 金伟, 路远, 同武勤, 杨丽. 可见光隐身技术的现状与研究动态. 飞航导弹, 2007, (8): 12-15
- [2]. 何小溪, 吕振华. 军用车辆的可见光隐身迷彩设计研究. 《新技术新工艺》· 数字技术与机械加工工艺装备, 2009, (8): 28-31
- [3]. 军事科学院军事百科研究所. 信息化战争时代的“电子迷彩服”——隐身技术. 国防, 2005, (5): 71-72
- [4]. R. C. Gonzalez, R. E. Woods. Digital Image Processing 2nd Edition. New Jersey: Prentice Hall, 2002
- [5]. 卢智勇, 梁光明. 基于多边形拟合的细胞图像特征提取算法. 计算机仿真, 2009, 26(11): 268-271
- [6]. 没找到合适的透视变换的参考文献
- [7]. A. Criminisi, P. Pérez, K. Toyama. Region filling and object removal by exemplar based image inpainting. IEEE Transactions on Image Processing, 2004, 13(9): 1200-1212
- [8]. A. Efros, W. T. Freeman. Image quilting for texture synthesis and transfer. In Proc. ACM Conf. Comp. Graphics (SIGGRAPH), 2001: 341-346
- [9]. M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester. Image inpainting. In Proc. ACM Conf. Comp. Graphics (SIGGRAPH), 2000: 417-424
- [10]. P. Harrison. A non-hierarchical procedure for re-synthesis of complex texture. In Proc. Int. Conf. Central Europe Comp. Graphics, Visua. and Comp. Vision, 2001
- [11]. Bilinear interpolation. [http://en.wikipedia.org/wiki/Bilinear\\_interpolation](http://en.wikipedia.org/wiki/Bilinear_interpolation)
- [12]. 何援军. 计算机图形学. 北京: 机械工业出版社, 2007