

足部测量系统 - 基于A4纸参考

足部测量系统

本项目实现基于A4纸参考的足部测量系统，包含以下功能：

- A4纸检测和透视校正
- 足部区域分割
- 轮廓提取和测量计算
- 结果可视化和数据导出

使用方法

1. 将脚后跟紧贴A4纸边缘
2. 俯视拍照，确保A4纸完整入镜
3. 运行代码进行自动处理

```
In [ ]: # 足部测量系统 - 边缘检测方法
import cv2
import numpy as np
import matplotlib.pyplot as plt
from segment_anything import sam_model_registry, SamPredictor
from foot_report import run_shoe_recommendation
from process_foot import process_foot_measurement

# 加载模型
sam = sam_model_registry["vit_h"](checkpoint="sam_vit_h_4b8939.pth")
predictor = SamPredictor(sam)

# 设置中文显示
plt.rcParams['font.sans-serif'] = ['SimHei', 'DejaVu Sans']
plt.rcParams['axes.unicode_minus'] = False
```

```
In [8]: # 读取图像
TEST_IMAGE_PATH = "foot_with_a4.png" # 修改为你的图像路径
image = cv2.imread(TEST_IMAGE_PATH)
print(f"图像尺寸: {image.shape}")

# 显示原始图像
plt.figure(figsize=(10, 6))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title("原始图像")
plt.axis('off')
plt.show()
```

图像尺寸: (1536, 1024, 3)

原始图像



```
In [9]: # 读取图像
image = cv2.imread(TEST_IMAGE_PATH)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
predictor.set_image(image)

# 获取图像尺寸并计算中心点
height, width = image.shape[:2]
center_x = width // 2
center_y = height // 2

# 点击图像中心
input_point = np.array([[center_x, center_y]]) # [x, y] 坐标
input_label = np.array([1]) # 1表示前景点

# 生成mask
masks, scores, logits = predictor.predict(
    point_coords=input_point,
    point_labels=input_label,
    multimask_output=False
)

# 保存mask
center_mask = masks[0].astype(np.uint8) * 255
cv2.imwrite("result\center_mask.png", center_mask)
print("中心点mask已保存到 center_mask.png")
```

中心点mask已保存到 center_mask.png

```
In [10]: # 显示结果信息
mask_area = np.sum(masks[0])
```

```

print(f"分割区域面积: {mask_area} 像素")
print(f"置信度分数: {scores[0]:.3f}")
plt.figure(figsize=(10, 6))
plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title("原始图像")
plt.axis('off')
plt.subplot(1, 2, 2)
plt.imshow(masks[0], cmap='gray')
plt.title("分割Mask")
plt.axis('off')
plt.show()

```

分割区域面积: 262510 像素
置信度分数: 0.993



```

In [11]: predictor.set_image(image)

# 使用已存在的足部mask变量 center_mask
foot_mask = center_mask

# 找到足部的最顶端位置
foot_pixels = np.where(foot_mask > 0) # 找到所有白色像素
top_y = np.min(foot_pixels[0]) # 最小的y坐标就是顶部

# 在顶部找到中心的x坐标
top_row_pixels = np.where(foot_mask[top_y, :] > 0)[0] # 顶部行的所有x坐标
center_x = int(np.mean(top_row_pixels)) # 顶部的中心x坐标

# 在足部顶部往上2个像素的位置点击
paper_y = max(0, top_y - 20) # 往上20像素点, 但不能超出图像边界
paper_point = np.array([[center_x, paper_y]])

print(f"足部顶部位置: ({center_x}, {top_y})")
print(f"A4纸点击位置: ({center_x}, {paper_y})")

# 点击A4纸位置

```

```

input_point = paper_point
input_label = np.array([1]) # 1表示前景点

# 生成A4纸mask
masks, scores, logits = predictor.predict(
    point_coords=input_point,
    point_labels=input_label,
    multimask_output=False
)

# 保存A4纸mask
a4_mask = masks[0].astype(np.uint8) * 255
cv2.imwrite("result\\a4_mask.png", a4_mask)

print("A4纸mask已保存到 a4_mask.png")
print(f"A4纸区域面积: {np.sum(masks[0])} 像素")
print(f"置信度分数: {scores[0]:.3f}")

```

足部顶部位置: (656, 467)
A4纸点击位置: (656, 447)
A4纸mask已保存到 a4_mask.png
A4纸区域面积: 348995 像素
置信度分数: 0.987

In [12]: # 可选: 可视化点击位置

```

plt.figure(figsize=(10, 6))
plt.subplot(1, 2, 1)
plt.imshow(image)
plt.scatter([center_x], [top_y], c='red', s=1, label='足部顶部')
plt.scatter([center_x], [paper_y], c='blue', s=1, label='A4纸位置')
plt.title("点击位置")
plt.axis('off')
plt.legend()
plt.subplot(1, 2, 2)
plt.imshow(masks[0], cmap='gray')
plt.title("A4纸Mask")
plt.axis('off')
plt.show()

```



```
In [13]: # 找到A4纸mask中所有白色像素的坐标
y_coords, x_coords = np.where(a4_mask > 0)

# 直接计算4个角点
top_left = [x_coords[np.argmin(x_coords + y_coords)], y_coords[np.argmin(x_coords + y_coords)]]
top_right = [x_coords[np.argmax(x_coords - y_coords)], y_coords[np.argmax(x_coords - y_coords)]]
bottom_left = [x_coords[np.argmin(x_coords - y_coords)], y_coords[np.argmin(x_coords - y_coords)]]
bottom_right = [x_coords[np.argmax(x_coords + y_coords)], y_coords[np.argmax(x_coords + y_coords)]]

# 按顺序排列: 左上、右上、右下、左下
corners = np.array([top_left, top_right, bottom_right, bottom_left], dtype=np.float32)

print("A4纸4个角点:")
corner_names = ['左上', '右上', '右下', '左下']
for corner, name in zip(corners, corner_names):
    print(f" {name}: ({corner[0]:.1f}, {corner[1]:.1f})")

# 定义新图像尺寸
new_width = 420
new_height = int(new_width * 29.7 / 21) # A4比例

# 目标角点
dst_corners = np.array([
    [0, 0],
    [new_width - 1, 0],
    [new_width - 1, new_height - 1],
    [0, new_height - 1]
], dtype=np.float32)

# 透视变换
transform_matrix = cv2.getPerspectiveTransform(corners, dst_corners)
warped_image = cv2.warpPerspective(a4_mask, transform_matrix, (new_width, new_height))

# 保存
warped_image_bgr = cv2.cvtColor(warped_image, cv2.COLOR_GRAY2BGR)
```

```

warped_a4_path="result\warped_a4.png" # 修改为你想保存的路径
cv2.imwrite(warped_a4_path, warped_image_bgr)
print("完成！保存到 warped_a4.png")

plt.figure(figsize=(10, 6))
plt.subplot(1, 2, 1)
plt.title("原图 + A4纸角点")
plt.imshow(image)
for corner, name in zip(corners, corner_names):
    plt.plot(corner[0], corner[1], 'ro')
    plt.text(corner[0] + 5, corner[1] - 5, name, color='red', fontsize=12)
plt.axis('off')
plt.legend()
plt.subplot(1, 2, 2)
plt.imshow(warped_image)
plt.title("透视变换后的mask")
plt.axis('off')
plt.show()

```

A4纸4个角点：

左上：(176.0, 333.0)

右上：(863.0, 338.0)

右下：(816.0, 1141.0)

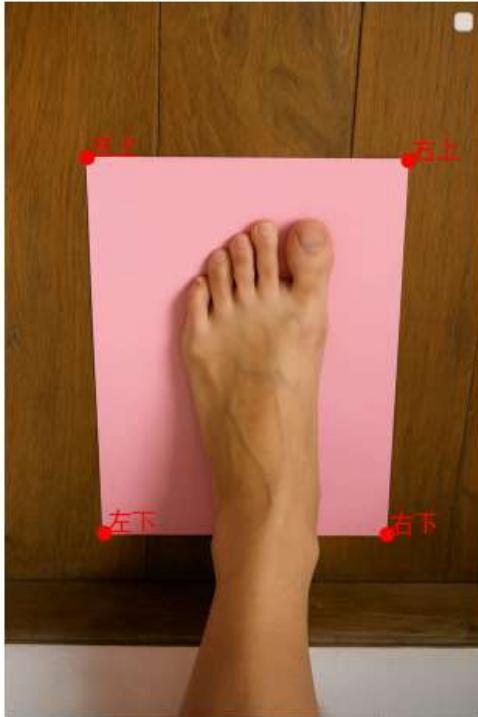
左下：(212.0, 1137.0)

完成！保存到 warped_a4.png

C:\Users\21202\AppData\Local\Temp\ipykernel_44816\2424128049.py:48: UserWarning:
No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.

plt.legend()

原图 + A4纸角点



透视变换后的mask



In [14]: # 运行测量和报告生成
results,process_foot_measurement_path = process_foot_measurement(image_path=warped_a4_path)
foot_length_mm = results['foot_length_mm']
foot_width_mm = results['max_width_mm']
print(f"测量结果 - 脚长: {foot_length_mm:.1f} mm, 脚宽: {foot_width_mm:.1f} mm")

- 步骤1: 检测足部...
- 检测到足部, 足长: 254.0 mm
- 步骤2: 椭圆修正足后跟区域...
- 椭圆修正完成 (足后跟起始位置: 501px)

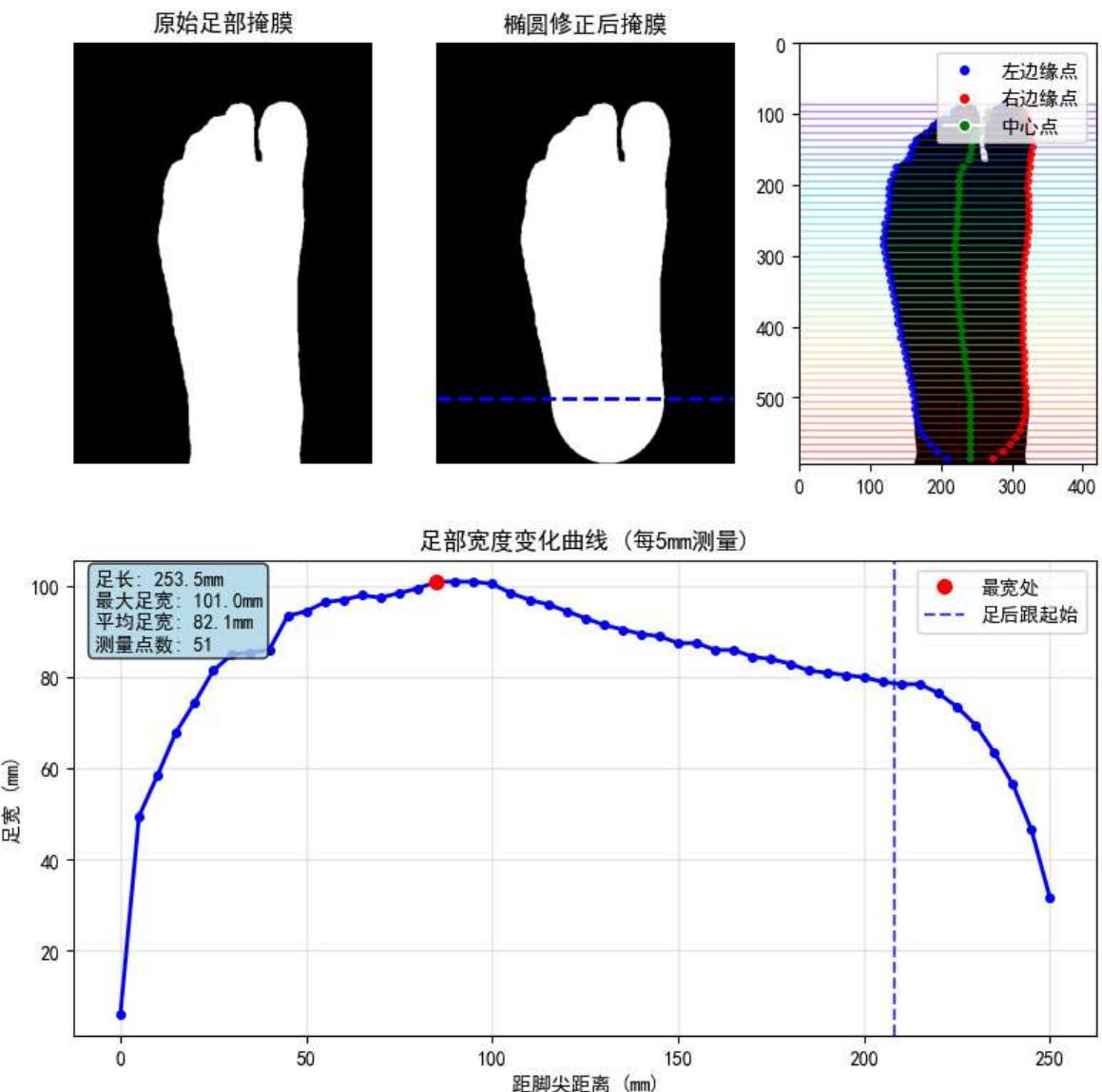
步骤3: 每5mm测量足宽...

距脚尖距离(mm) | 足宽(mm) | 足宽(cm)

0.0	6.0	0.60
5.0	49.5	4.95
10.0	58.5	5.85
15.0	68.0	6.80
20.0	74.5	7.45
25.0	81.5	8.15
30.0	85.0	8.50
35.0	85.5	8.55
40.0	86.0	8.60
45.0	93.5	9.35
50.0	94.5	9.45
55.0	96.5	9.65
60.0	97.0	9.70
65.0	98.0	9.80
70.0	97.5	9.75
75.0	98.5	9.85
80.0	99.5	9.95
85.0	101.0	10.10
90.0	101.0	10.10
95.0	101.0	10.10
100.0	100.5	10.05
105.0	98.5	9.85
110.0	97.0	9.70
115.0	96.0	9.60
120.0	94.5	9.45
125.0	93.0	9.30
130.0	91.5	9.15
135.0	90.5	9.05
140.0	89.5	8.95
145.0	89.0	8.90
150.0	87.5	8.75
155.0	87.5	8.75
160.0	86.0	8.60
165.0	86.0	8.60
170.0	84.5	8.45
175.0	84.0	8.40
180.0	83.0	8.30
185.0	81.5	8.15
190.0	81.0	8.10
195.0	80.5	8.05
200.0	80.0	8.00
205.0	79.0	7.90
210.0	78.5	7.85
215.0	78.5	7.85
220.0	76.5	7.65
225.0	73.5	7.35
230.0	69.5	6.95
235.0	63.5	6.35
240.0	56.5	5.65
245.0	46.5	4.65
250.0	31.5	3.15

⌚ 最宽位置：距脚尖 85.0mm 处，宽度 101.0mm

📊 步骤4：生成可视化...



可视化完成并保存到 result\foot_measurement_summary.png

💾 修正后的掩膜已保存到 result\modified_foot_mask.png

💾 测量数据已保存到 foot_measurements.json

测量结果 - 脚长: 253.5 mm, 脚宽: 101.0 mm

<Figure size 640x480 with 0 Axes>

```
In [18]: report=run_shoe_recommendation(foot_length_mm, foot_width_mm)
```

智能鞋码推荐报告

测量数据：

脚长: 253.5 mm (25.4 cm)
脚宽: 101.0 mm (10.1 cm)

宽长比: 0.398

国际尺码推荐表：

类别	国家	推荐尺码	宽度类型	特别建议
男鞋	中国	40.5	D	建议考虑大半码
	欧洲	40.5		
	美国	7.5		
	英国	7.0		
	日本	25.5cm		
女鞋	中国	41.0	EE	建议考虑大半码
	欧洲	41.0		
	美国	9.0		
	英国	7.0		
	日本	26.5cm		

脚型分析详情：

【男性脚型】

- 类型：宽(W)
- 特征：宽脚/脚背高
- 建议：建议选择宽版鞋款或考虑大半码

【女性脚型】

- 类型：加宽(XW)
- 特征：特宽脚
- 建议：建议选择特宽版鞋款或专门的宽脚鞋款

温馨提示：

1. 不同品牌可能存在尺码差异，建议购买前试穿
2. 运动鞋建议预留5-10mm活动空间
3. 皮鞋和正装鞋建议选择贴合的尺码
4. 脚部会因时间和温度略有变化，建议下午试鞋

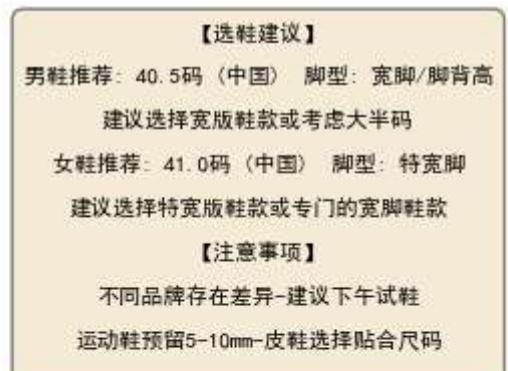
报告已保存至: result\shoe_size_report.png

智能鞋码推荐报告

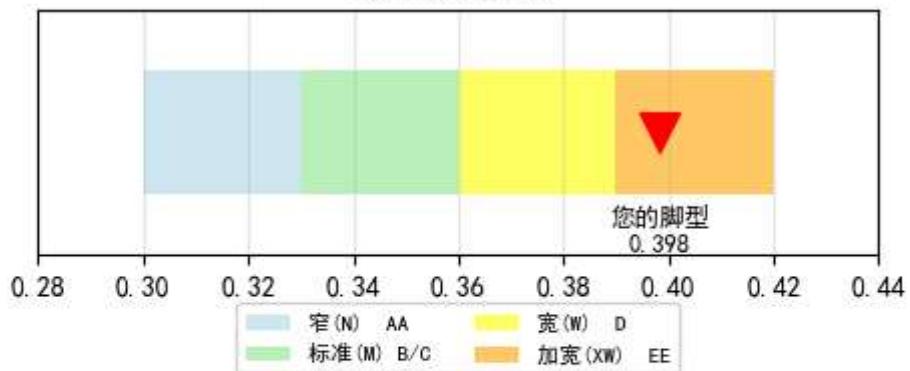
脚长: 253.5mm | 脚宽: 101.0mm | 宽长比: 0.398

国际尺码对照表 & 选鞋建议

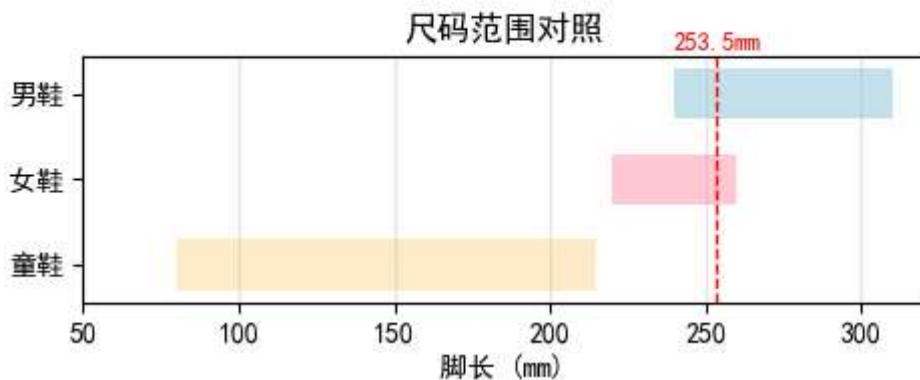
类别	中国码	欧洲码	美国码	英国码	日本码	宽度
男鞋	40.5	40.5	7.5	7.0	25.5	D
女鞋	41.0	41.0	9.0	7.0	26.5	EE



脚型宽度分析



脚宽/脚长 比例



```
In [19]: from openai import OpenAI
import json

client = OpenAI(api_key="sk-198343e3213d42e8bc17c259a4398c56", base_url="https://api.openai.com")

# 构建结构化的数据
foot_data = {
    "measurement_results": results,
    "analysis_report": report,
    "processing_path": process_foot_measurement_path
}

user_content = f"""
基于以下足部测试数据，请提供专业的：
1. 3D打印鞋晶格设计建议（包括材料、密度分布、结构类型）
2. 个性化的运动健康建议
3. 鞋具选择指导

数据详情：
{json.dumps(foot_data, indent=2, ensure_ascii=False)}
"""

response = client.chat.completions.create(
    model="deepseek-chat",
    messages=[
        {"role": "system", "content": "你是一名足部矫形师和运动康复专家，擅长3D打印鞋晶格设计。"},
        {"role": "user", "content": user_content},
    ],
    stream=False,
    temperature=0.7 # 控制创造性，较低值更事实性
)

print("=" * 50)
print("专业建议：")
print("=" * 50)
print(response.choices[0].message.content)
```

=====

专业建议：

=====

根据您提供的详细足部测量数据，我将为您提供专业的3D打印鞋具设计建议、运动健康指导和鞋具选择建议。

一、3D打印鞋晶格设计建议

材料选择：

- **主体材料**：TPU（热塑性聚氨酯），硬度建议在85A-95A之间
- **关键支撑区域**：可考虑使用尼龙复合材料增强特定区域

晶格密度分布设计：

1. **前掌区域**（0-85mm位置）：

- 中等密度晶格（60-70%填充率）
- 采用菱形或六边形结构提供良好的横向支撑

2. **中足区域**（85-150mm位置）：

- 高密度支撑区域（75-85%填充率）
- 渐变密度设计，在足弓处提供额外支撑
- 采用梯度蜂窝结构

3. **后跟区域**（150-250mm位置）：

- 最高密度设计（85-90%填充率）
- 采用同心圆放射状结构提供稳定性
- 边缘区域密度略低以提供缓冲

结构类型建议：

- **主结构**：Gyroid曲面晶格结构，提供各向同性力学性能
- **足弓支撑**：定制化拱形支撑结构，基于您的足弓高度数据
- **边缘加固**：在宽度变化剧烈区域增加边界加固设计

二、个性化运动健康建议

日常训练建议：

1. **足部强化训练**：

- 毛巾抓取练习：每天3组，每组15次
- 足底筋膜球按摩：每次运动前后各5分钟
- 小腿拉伸：保持每次30秒，重复3次

2. **步态改善训练**：

- 单腿站立平衡练习：每天2组，每组30秒
- 脚跟-脚尖行走：每次10米，重复3次
- proprioceptive训练：在不平坦表面行走

3. **运动选择**：

- 推荐：游泳、cycling、椭圆机训练
- 谨慎选择：高冲击运动如跑步、篮球
- 如需跑步，建议在柔软表面并进行间歇训练

康复建议：

- 每日进行足底筋膜放松
- 穿着支撑性良好的鞋具进行所有活动
- 定期进行步态分析以监控进展

三、鞋具选择指导

日常鞋款选择：

- **鞋型**：选择宽楦（2E或4E宽度）鞋款
- **鞋垫**：使用定制化支撑鞋垫

- **鞋底**: 中等硬度，具有良好的弯曲灵活性
- **后跟**: 稳固的后跟杯设计

运动鞋推荐特性：

1. **跑步鞋**：

- 稳定性跑鞋类型
- 宽楦设计（如New Balance 2E/4E系列）
- 良好的中足支撑
- 适度的后跟到前掌落差（6-8mm）

2. **训练鞋**：

- 平底设计提供稳定性
- 宽前掌空间
- 良好的侧向支撑

3. **避免的鞋型**：

- 窄楦设计
- 极简主义鞋款
- 高跟鞋或完全平底鞋

尺寸建议：

- **男码**: US 8-8.5（比常规大半码）
- **女码**: US 9.5（宽版设计）
- 务必在下午或运动后试鞋，此时足部略有肿胀

四、后续建议

1. **定期评估**: 每6个月重新评估足部状况
2. **渐进适应**: 新鞋具应逐步增加穿着时间
3. **专业咨询**: 建议每年前往足科医生处进行检查

这些建议基于您的特定足部测量数据，旨在为您提供最佳的舒适度和支撑性。如果您有任何不适或特殊需求，建议咨询专业的足科医生进行进一步评估。