MinerU 输出文件说明

概览

`mineru` 命令执行后，除了输出主要的 markdown 文件外，还会生成多个辅助文件用于调试、质检和进一步处理。这些文件包括：

- **可视化调试文件**：帮助用户直观了解文档解析过程和结果
- **结构化数据文件**：包含详细的解析数据，可用于二次开发

下面将详细介绍每个文件的作用和格式。

可视化调试文件

布局分析文件 (layout.pdf)

文件命名格式：`{原文件名}_layout.pdf`

功能说明：

- 可视化展示每一页的布局分析结果
- 每个检测框右上角的数字表示阅读顺序
- 使用不同背景色块区分不同类型的内容块

使用场景：

- 检查布局分析是否正确
- 确认阅读顺序是否合理
- 调试布局相关问题

aggregate cost values within the pixel neighborhoods defined by these windows. In 2005, Yoon and Kweon [4] proposed an adaptive matching cost aggregation scheme, which assigns a weight value to every pixel located in the support window of a given pixel of interest. The weight value is based on the spatial and color similarity between the pixel of interest and a pixel in its support window, and the aggregated cost is computed as a weighted average of the pixel-wise costs within the considered support window. The edge-preserving nature and matching accuracy of adaptive support weights have made them one of the most popular choices for cost aggregation in recently proposed stereo matching algorithms [3], [5]–[8].

Recently, Rheman *et al.* [9], [10] have revisited the cost aggregation step of stereo algorithms, and demonstrated that cost aggregation can be performed by filtering of subsequent layers of the initially computed matching cost volume. In particular, the edge-aware image filters, such as the bilateral filter of Tomasi and Manducci [11] or the guided filter of He [12], have been rendered useful for the problem of matching cost aggregation, enabling stereo algorithms to correctly recover disparities along object boundaries. In fact, Yoon and Kweon's adaptive support-weight cost aggregation scheme is equivalent to the application of the so-called joint bilateral filter to the layers of the matching cost volume.

It has been demonstrated that the performance of stereo algorithms designed to match a single pair of images can be adapted to take advantage of the temporal dependencies available in stereo video sequences. Early proposed solutions to temporal stereo matching attempted to average matching costs across subsequent frames of a video sequence [13]

temporal information, making it possible to process a temporal collection of cost volumes. The filtering operation was shown to preserve spatio-temporal edges present in the cost volumes, resulting in increased temporal consistency of disparity maps, greater robustness to image noise, and more accurate behavior around object boundaries.

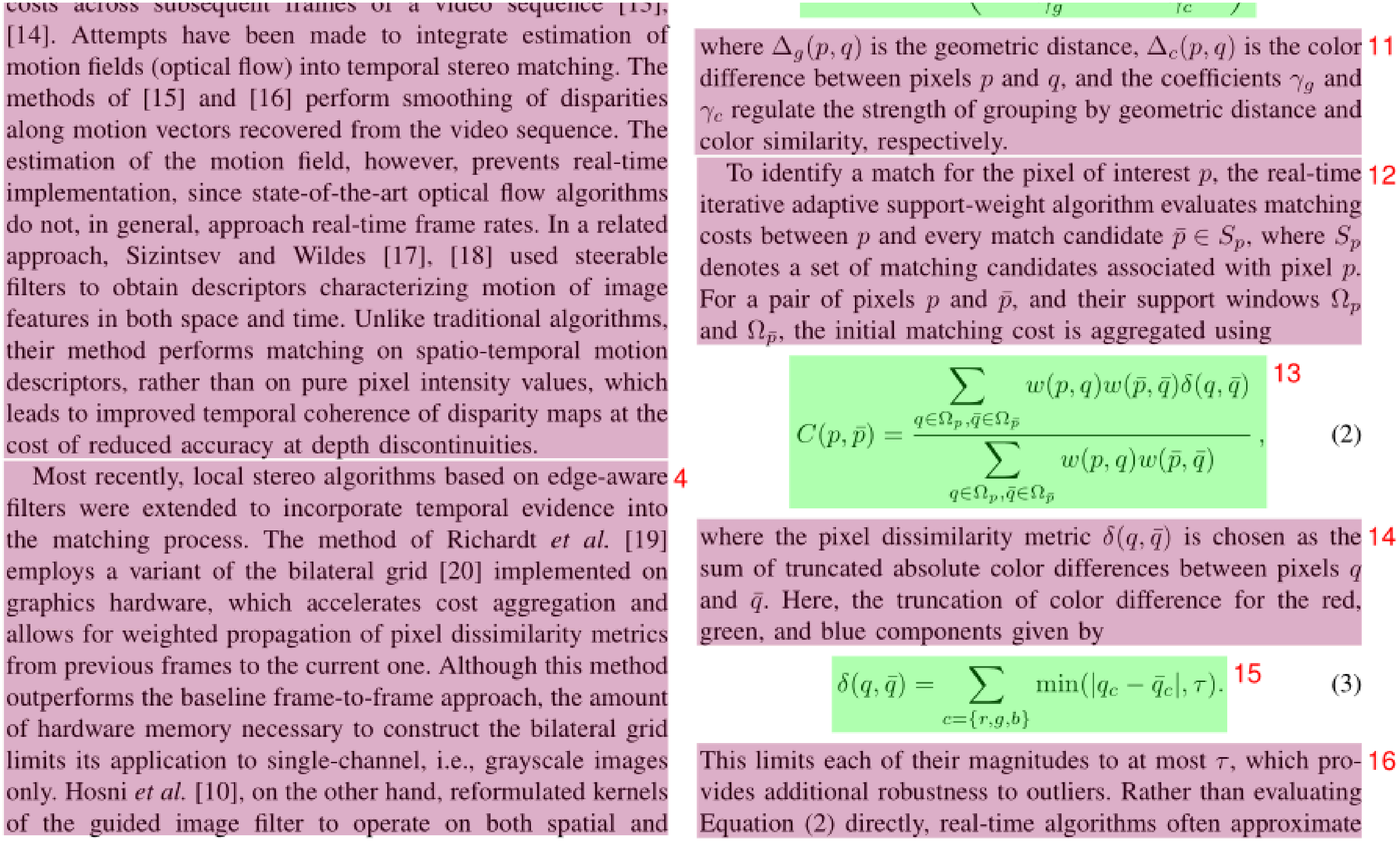
III. Method

The proposed temporal stereo matching algorithm is an extension of the real-time iterative adaptive support-weight algorithm described in [3]. In addition to real-time two-pass aggregation of the cost values in the spatial domain, the proposed algorithm enhances stereo matching on video sequences by aggregating costs along the time dimension. The operation of the algorithm has been divided into four stages: 1) two-pass spatial cost aggregation, 2) temporal cost aggregation, 3) disparity selection and confidence assessment, and 4) iterative disparity refinement. In the following, each of these stages is described in detail.

A. Two-Pass Spatial Cost Aggregation

Humans group shapes by observing the geometric distance and color similarity of points in space. To mimic this visual grouping, the adaptive support-weight stereo matching algorithm [4] considers a support window Ω_p centered at the pixel of interest p, and assigns a support weight to each pixel $q \in \Omega_p$. The support weight relating pixels p and q is given by

$$w(p,q) = \exp\left(-\frac{\Delta_g(p,q)}{\gamma_g} - \frac{\Delta_c(p,q)}{\gamma_c}\right), \qquad (1)$$



costs across subsequent frames of a video sequence [13], [14]. Attempts have been made to integrate estimation of motion fields (optical flow) into temporal stereo matching. The methods of [15] and [16] perform smoothing of disparities along motion vectors recovered from the video sequence. The estimation of the motion field, however, prevents real-time implementation, since state-of-the-art optical flow algorithms do not, in general, approach real-time frame rates. In a related approach, Sizintsev and Wildes [17], [18] used steerable filters to obtain descriptors characterizing motion of image features in both space and time. Unlike traditional algorithms, their method performs matching on spatio-temporal motion descriptors, rather than on pure pixel intensity values, which leads to improved temporal coherence of disparity maps at the cost of reduced accuracy at depth discontinuities.

Most recently, local stereo algorithms based on edge-aware filters were extended to incorporate temporal evidence into the matching process. The method of Richardt *et al.* [19] employs a variant of the bilateral grid [20] implemented on graphics hardware, which accelerates cost aggregation and allows for weighted propagation of pixel dissimilarity metrics from previous frames to the current one. Although this method outperforms the baseline frame-to-frame approach, the amount of hardware memory necessary to construct the bilateral grid limits its application to single-channel, i.e., grayscale images only. Hosni *et al.* [10], on the other hand, reformulated kernels of the guided image filter to operate on both spatial and

where $\Delta_g(p,q)$ is the geometric distance, $\Delta_c(p,q)$ is the color difference between pixels p and q, and the coefficients γ_g and γ_c regulate the strength of grouping by geometric distance and color similarity, respectively.

To identify a match for the pixel of interest p, the real-time iterative adaptive support-weight algorithm evaluates matching costs between p and every match candidate $\bar{p} \in S_p$, where S_p denotes a set of matching candidates associated with pixel p. For a pair of pixels p and $\bar{p}$, and their support windows Ω_p and $\Omega_{\bar{p}}$, the initial matching cost is aggregated using

$$C(p,\bar{p}) = \frac{\sum\limits_{q\in\Omega_p,\bar{q}\in\Omega_{\bar{p}}} w(p,q)w(\bar{p},\bar{q})\delta(q,\bar{q})}{\sum\limits_{q\in\Omega_p,\bar{q}\in\Omega_{\bar{p}}} w(p,q)w(\bar{p},\bar{q})}, \tag{2}$$

where the pixel dissimilarity metric $\delta(q,\bar{q})$ is chosen as the sum of truncated absolute color differences between pixels q and $\bar{q}$. Here, the truncation of color difference for the red, green, and blue components given by

$$\delta(q,\bar{q}) = \sum_{c=\{r,g,b\}} \min(|q_c - \bar{q}_c|, \tau). \tag{3}$$

This limits each of their magnitudes to at most τ, which provides additional robustness to outliers. Rather than evaluating Equation (2) directly, real-time algorithms often approximate

文本片段文件 (spans.pdf)

Note

仅适用于 pipeline 后端

文件命名格式：`{原文件名}_spans.pdf`

功能说明：

- 根据 span 类型使用不同颜色线框标注页面内容
- 用于质量检查和问题排查

使用场景：

- 快速排查文本丢失问题
- 检查行内公式识别情况
- 验证文本分割准确性

dependent on the service headway and the reliability of the departure time of the service to which passengers are incident.

After briefly introducing the random incidence model, which is often assumed to hold at short headways, the balance of this section reviews six studies of passenger incidence behavior that are motivated by understanding the relationships between service headway, service reliability, passenger incidence behavior, and passenger waiting time in a more nuanced fashion than is embedded in the random incidence assumption (2). Three of these studies depend on manually collected data, two studies use data from AFC systems, and one study analyzes the issue purely theoretically. These studies reveal much about passenger incidence behavior, but all are found to be limited in their general applicability by the methods with which they collect information about passengers and the services those passengers intend to use.

Random Passenger Incidence Behavior

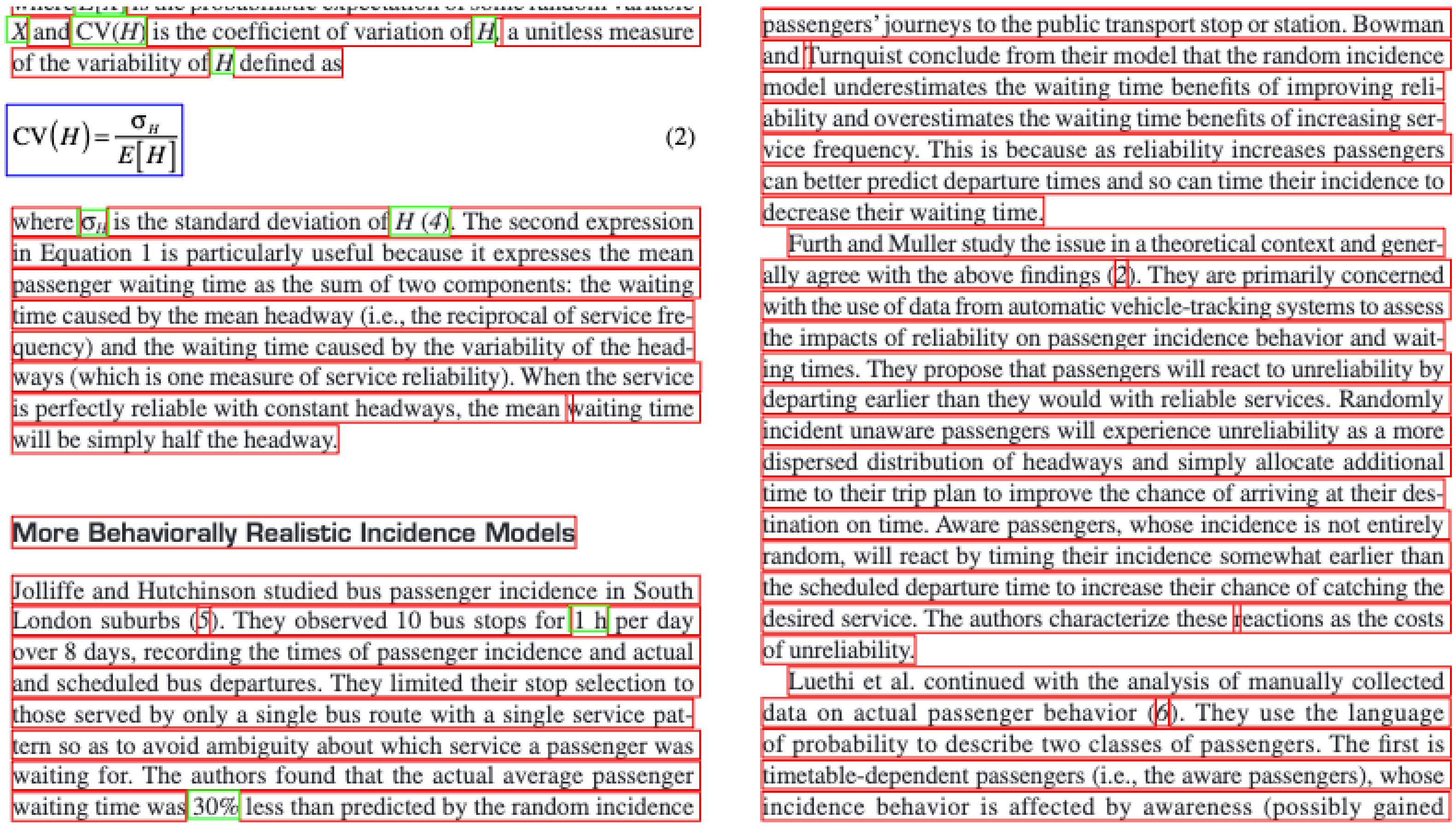
One characterization of passenger incidence behavior is that of random incidence (3). The key assumption underlying the random incidence model is that the process of passenger arrivals to the public transport service is independent from the vehicle departure process of the service. This implies that passengers become incident to the service at a random time, and thus the instantaneous rate of passenger arrivals to the service is uniform over a given period of time. Let W and H be random variables representing passenger waiting times and service headways, respectively. Under the random incidence assumption and the assumption that vehicle capacity is not a binding constraint, a classic result of transportation science is that

$$E(W)=\frac{E\left[H^{2}\right]}{2E\left[H\right]}=\frac{E\left[H\right]}{2}\left(1+\mathrm{CV}\left(H\right)^{2}\right) \qquad (1)$$

where $E[X]$ is the probabilistic expectation of some random variable

model. They also found that the empirical distributions of passenger incidence times (by time of day) had peaks just before the respective average bus departure times. They hypothesized the existence of three classes of passengers: with proportion q, passengers whose time of incidence is causally coincident with that of a bus departure (e.g., because they saw the approaching bus from their home or a shop window); with proportion $p(1-q)$, passengers who time their arrivals to minimize expected waiting time; and with proportion $(1-p)(1-q)$, passengers who are randomly incident. The authors found that p was positively correlated with the potential reduction in waiting time (compared with arriving randomly) that resulted from knowledge of the timetable and of service reliability. They also found p to be higher in the peak commuting periods rather than in the off-peak periods, indicating more awareness of the timetable or historical reliability, or both, by commuters.

Bowman and Turnquist built on the concept of aware and unaware passengers of proportions p and $(1-p)$, respectively. They proposed a utility-based model to estimate p and the distribution of incidence times, and thus the mean waiting time, of aware passengers over a given headway as a function of the headway and reliability of bus departure times (1). They observed seven bus stops in Chicago, Illinois, each served by a single (different) bus route, between 6:00 and 8:00 a.m. for 5 to 10 days each. The bus routes had headways of 5 to 20 min and a range of reliabilities. The authors found that actual average waiting time was substantially less than predicted by the random incidence model. They estimated that p was not statistically significantly different from 1.0, which they explain by the fact that all observations were taken during peak commuting times. Their model predicts that the longer the headway and the more reliable the departures, the more peaked the distribution of incidence times will be and the closer that peak will be to the next scheduled departure time. This prediction demonstrates what they refer to as a safety margin that passengers add to reduce the chance of missing their bus when the service is known to be somewhat unreliable. Such a safety margin can also result from unreliability in



where $E[X]$ is the probabilistic expectation of some random variable X and CV(H) is the coefficient of variation of H, a unitless measure of the variability of H defined as

$$\mathrm{CV}(H)=\frac{\sigma_H}{E[H]} \qquad (2)$$

where σ_H is the standard deviation of H (4). The second expression in Equation 1 is particularly useful because it expresses the mean passenger waiting time as the sum of two components: the waiting time caused by the mean headway (i.e., the reciprocal of service frequency) and the waiting time caused by the variability of the headways (which is one measure of service reliability). When the service is perfectly reliable with constant headways, the mean waiting time will be simply half the headway.

More Behaviorally Realistic Incidence Models

Jolliffe and Hutchinson studied bus passenger incidence in South London suburbs (5). They observed 10 bus stops for 1 h per day over 8 days, recording the times of passenger incidence and actual and scheduled bus departures. They limited their stop selection to those served by only a single bus route with a single service pattern so as to avoid ambiguity about which service a passenger was waiting for. The authors found that the actual average passenger waiting time was 30% less than predicted by the random incidence passengers' journeys to the public transport stop or station. Bowman and Turnquist conclude from their model that the random incidence model underestimates the waiting time benefits of improving reliability and overestimates the waiting time benefits of increasing service frequency. This is because as reliability increases passengers can better predict departure times and so can time their incidence to decrease their waiting time.

Furth and Muller study the issue in a theoretical context and generally agree with the above findings (2). They are primarily concerned with the use of data from automatic vehicle-tracking systems to assess the impacts of reliability on passenger incidence behavior and waiting times. They propose that passengers will react to unreliability by departing earlier than they would with reliable services. Randomly incident unaware passengers will experience unreliability as a more dispersed distribution of headways and simply allocate additional time to their trip plan to improve the chance of arriving at their destination on time. Aware passengers, whose incidence is not entirely random, will react by timing their incidence somewhat earlier than the scheduled departure time to increase their chance of catching the desired service. The authors characterize these reactions as the costs of unreliability.

Luethi et al. continued with the analysis of manually collected data on actual passenger behavior (6). They use the language of probability to describe two classes of passengers. The first is timetable-dependent passengers (i.e., the aware passengers), whose incidence behavior is affected by awareness (possibly gained

结构化数据文件

Important

2.5版本vlm后端的输出存在较大变化，与pipeline版本存在不兼容情况，如需基于结构化输出进行二次开发，请仔细阅读本文档内容。

pipeline 后端 输出结果

模型推理结果 (model.json)

文件命名格式：`{原文件名}_model.json`

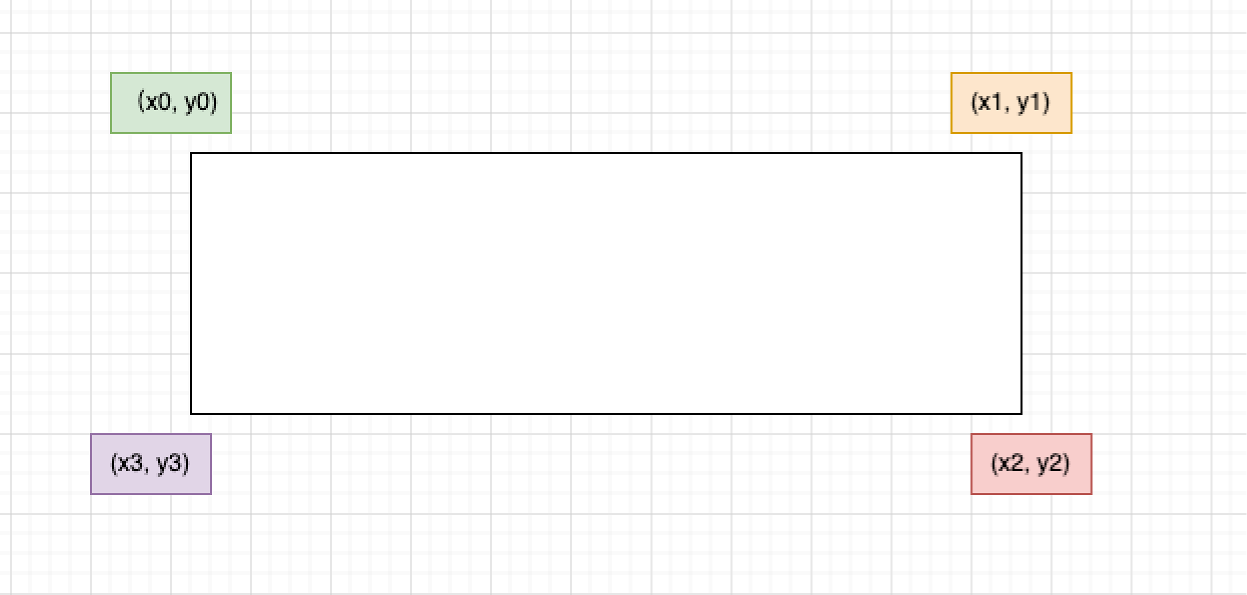
数据结构定义

```python
from pydantic import BaseModel, Field
from enum import IntEnum

class CategoryType(IntEnum):
    """内容类别枚举"""
    title = 0               # 标题
    plain_text = 1          # 文本
    abandon = 2             # 包括页眉页脚页码和页面注释
    figure = 3              # 图片
    figure_caption = 4      # 图片描述
    table = 5               # 表格
    table_caption = 6       # 表格描述
    table_footnote = 7      # 表格注释
    isolate_formula = 8     # 行间公式
    formula_caption = 9     # 行间公式的标号
    embedding = 13          # 行内公式
    isolated = 14           # 行间公式
    text = 15               # OCR 识别结果

class PageInfo(BaseModel):
    """页面信息"""
    page_no: int = Field(description="页码序号，第一页的序号是 0", ge=0)
    height: int = Field(description="页面高度", gt=0)
    width: int = Field(description="页面宽度", ge=0)

class ObjectInferenceResult(BaseModel):
    """对象识别结果"""
    category_id: CategoryType = Field(description="类别", ge=0)
    poly: list[float] = Field(description="四边形坐标，格式为 [x0,y0,x1,y1,x2,y2,x3,y3]")
    score: float = Field(description="推理结果的置信度")
    latex: str | None = Field(description="LaTeX 解析结果", default=None)
    html: str | None = Field(description="HTML 解析结果", default=None)

class PageInferenceResults(BaseModel):
    """页面推理结果"""
    layout_dets: list[ObjectInferenceResult] = Field(description="页面识别结果")
    page_info: PageInfo = Field(description="页面元信息")

# 完整的推理结果
inference_result: list[PageInferenceResults] = []
```

坐标系统说明

`poly` 坐标格式：`[x0, y0, x1, y1, x2, y2, x3, y3]`

- 分别表示左上、右上、右下、左下四点的坐标
- 坐标原点在页面左上角



示例数据

```
[
    {
        "layout_dets": [
            {
                "category_id": 2,
                "poly": [
                    99.1906967163086,
                    100.3119125366211,
                    730.3707885742188,
                    100.3119125366211,
                    730.3707885742188,
                    245.81326293945312,
                    99.1906967163086,
                    245.81326293945312
                ],
                "score": 0.9999997615814209
            }
        ],
        "page_info": {
            "page_no": 0,
            "height": 2339,
            "width": 1654
        }
    },
    {
        "layout_dets": [
            {
                "category_id": 5,
                "poly": [
                    99.13092803955078,
                    2210.680419921875,
                    497.3183898925781,
                    2210.680419921875,
                    497.3183898925781,
                    2264.78076171875,
                    99.13092803955078,
                    2264.78076171875
                ],
                "score": 0.9999997019767761
            }
        ],
        "page_info": {
            "page_no": 1,
            "height": 2339,
            "width": 1654
        }
    }
]
```

中间处理结果 (middle.json)

文件命名格式： `{原文件名}_middle.json`

顶层结构

字段名	类型	说明
`pdf_info`	`list[dict]`	每一页的解析结果数组
`_backend`	`string`	解析模式： `pipeline` 或 `vlm`
`_version_name`	`string`	MinerU 版本号

页面信息结构 (PDF_INFO)

字段名	说明
`preproc_blocks`	PDF 预处理后的未分段中间结果
`page_idx`	页码，从 0 开始
`page_size`	页面的宽度和高度 `[width, height]`
`images`	图片块信息列表
`tables`	表格块信息列表
`interline_equations`	行间公式块信息列表
`discarded_blocks`	需要丢弃的块信息
`para_blocks`	分段后的内容块结果

块结构层次

```
一级块 (table | image)
└── 二级块
    └── 行 (line)
        └── 片段 (span)
```

一级块字段

字段名	说明
`type`	块类型：`table` 或 `image`
`bbox`	块的矩形框坐标 `[x0, y0, x1, y1]`
`blocks`	包含的二级块列表

二级块字段

字段名	说明
`type`	块类型（详见下表）
`bbox`	块的矩形框坐标
`lines`	包含的行信息列表

二级块类型

类型	说明
`image_body`	图像本体
`image_caption`	图像描述文本

类型	说明
`image_footnote`	图像脚注
`table_body`	表格本体
`table_caption`	表格描述文本
`table_footnote`	表格脚注
`text`	文本块
`title`	标题块
`index`	目录块
`list`	列表块
`interline_equation`	行间公式块

行和片段结构

行 (line) 字段： - `bbox`： 行的矩形框坐标 - `spans`： 包含的片段列表

片段 (span) 字段： - `bbox`： 片段的矩形框坐标 - `type`： 片段类型（`image`、`table`、`text`、`inline_equation`、`interline_equation`） - `content` | `img_path`： 文本内容或图片路径

示例数据

```
{
    "pdf_info": [
        {
            "preproc_blocks": [
                {
                    "type": "text",
                    "bbox": [
                        52,
                        61.956024169921875,
                        294,
                        82.99800872802734
                    ],
                    "lines": [
                        {
                            "bbox": [
                                52,
                                61.956024169921875,
                                294,
                                72.0000228881836
                            ],
                            "spans": [
                                {
                                    "bbox": [
                                        54.0,
                                        61.956024169921875,
                                        296.2261657714844,
                                        72.0000228881836
                                    ],
                                    "content": "dependent on the service headway and the reliability of the departure ",
                                    "type": "text",
                                    "score": 1.0
                                }
                            ]
                        }
                    ]
                }
            ],
            "layout_bboxes": [
                {
                    "layout_bbox": [
                        52,
                        61,
                        294,
                        731
                    ],
                    "layout_label": "V",
                    "sub_layout": []
                }
            ],
            "page_idx": 0,
            "page_size": [
                612.0,
                792.0
            ],
            "_layout_tree": [],
            "images": [],
            "tables": [],
            "interline_equations": [],
            "discarded_blocks": [],
            "para_blocks": [
                {
                    "type": "text",
                    "bbox": [
                        52,
                        61.956024169921875,
                        294,
                        82.99800872802734
                    ],
                    "lines": [
                        {
                            "bbox": [
                                52,
                                61.956024169921875,
                                294,
                                72.0000228881836
                            ],
                            "spans": [
                                {
                                    "bbox": [
                                        54.0,
                                        61.956024169921875,
                                        296.2261657714844,
                                        72.0000228881836
                                    ],
                                    "content": "dependent on the service headway and the reliability of the departure ",
                                    "type": "text",
                                    "score": 1.0
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "_backend": "pipeline",
    "_version_name": "0.6.1"
}
```

内容列表 (content_list.json)

文件命名格式：`{原文件名}_content_list.json`

功能说明

这是一个简化版的 `middle.json`，按阅读顺序平铺存储所有可读内容块，去除了复杂的布局信息，便于后续处理。

内容类型

类型	说明
`image`	图片
`table`	表格
`text`	文本/标题
`equation`	行间公式

文本层级标识

通过 `text_level` 字段区分文本层级：

- 无 `text_level` 或 `text_level: 0`：正文文本
- `text_level: 1`：一级标题
- `text_level: 2`：二级标题
- 以此类推...

通用字段

- 所有内容块都包含 `page_idx` 字段，表示所在页码（从 0 开始）。
- 所有内容块都包含 `bbox` 字段，表示内容块的边界框坐标 `[x0, y0, x1, y1]` 映射在0-1000范围内的结果。

示例数据

```
[
        {
        "type": "text",
        "text": "The response of flow duration curves to afforestation ",
        "text_level": 1,
        "bbox": [
            62,
            480,
            946,
            904
        ],
        "page_idx": 0
    },
    {
        "type": "image",
        "img_path": "images/a8ecda1c69b27e4f79fce1589175a9d721cbdc1cf78b4cc06a015f3746f6b9d8.jpg",
        "image_caption": [
            "Fig. 1. Annual flow duration curves of daily flows from Pine Creek, Australia, 1989–2000. "
        ],
        "image_footnote": [],
        "bbox": [
            62,
            480,
            946,
            904
        ],
        "page_idx": 1
    },
    {
        "type": "equation",
        "img_path": "images/181ea56ef185060d04bf4e274685f3e072e922e7b839f093d482c29bf89b71e8.jpg",
        "text": "$$\nQ _ { \\% } = f ( P ) + g ( T )\n$$",
        "text_format": "latex",
        "bbox": [
            62,
            480,
            946,
            904
        ],
        "page_idx": 2
    },
    {
        "type": "table",
        "img_path": "images/e3cb413394a475e555807ffdad913435940ec637873d673ee1b039e3bc3496d0.jpg",
        "table_caption": [
            "Table 2 Significance of the rainfall and time terms "
        ],
        "table_footnote": [
            "indicates that the rainfall term was significant at the $5 \\%$ level, $T$ indicates that the time term was
significant at the $5 \\%$ level, \\* represents significance at the $10 \\%$ level, and na denotes too few data points for
meaningful analysis. "
        ],
        "table_body": "<html><body><table><tr><td rowspan=\"2\">Site</td><td colspan=\"10\">Percentile</td></tr><tr>
<td>10</td><td>20</td><td>30</td><td>40</td><td>50</td><td>60</td><td>70</td><td>80</td><td>90</td><td>100</td></tr><tr>
<td>Traralgon Ck</td><td>P</td><td>P,*</td><td>P</td><td>P</td><td>P,</td><td>P,</td><td>P,</td><td>P,</td><td>P</td>
<td>P</td></tr><tr><td>Redhill</td><td>P,T</td><td>P,T</td><td>, *</td><td>**</td><td>P.T</td><td>P,*</td><td>P*</td><td>P*
</td><td>*</td><td>, *</td></tr><tr><td>Pine Ck</td><td></td><td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>T</td>
<td>T</td><td>T</td><td>na</td><td>na</td></tr><tr><td>Stewarts Ck 5</td><td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td>
<td>P.T</td><td>P.T</td><td>P,T</td><td>na</td><td>na</td><td>na</td></tr><tr><td>Glendhu 2</td><td>P</td><td>P,T</td><td>P,*
</td><td>P,T</td><td>P.T</td><td>P,ns</td><td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td></tr><tr><td>Cathedral Peak 2</td>
<td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>*,T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>T</td></tr>
<tr><td>Cathedral Peak 3</td><td>P.T</td><td>P.T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>T</td><td>P,T</td><td>P,T</td>
<td>P,T</td><td>T</td></tr><tr><td>Lambrechtsbos A</td><td>P,T</td><td>P</td><td>P</td><td>P,T</td><td>*,T</td><td>*,T</td>
<td>*,T</td><td>*,T</td><td>*,T</td><td>T</td></tr><tr><td>Lambrechtsbos B</td><td>P,T</td><td>P,T</td><td>P,T</td>
<td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>P,T</td><td>T</td><td>T</td></tr><tr><td>Biesievlei</td><td>P,T</td>
<td>P.T</td><td>P,T</td><td>P,T</td><td>*,T</td><td>*,T</td><td>T</td><td>T</td><td>P,T</td><td>P,T</td></tr></table></body>
</html>",
        "bbox": [
            62,
            480,
            946,
            904
        ],
            "page_idx": 5
        }
    ]
```

VLM 后端 输出结果

模型推理结果 (model.json)

文件命名格式：`{原文件名}_model.json`

文件格式说明

- 该文件为 VLM 模型的原始输出结果，包含两层嵌套list，外层表示页面，内层表示该页的内容块
- 每个内容块都是一个dict，包含 `type`、`bbox`、`angle`、`content` 字段

支持的内容类型

```
{
    "text": "文本",
    "title": "标题",
    "equation": "行间公式",
    "image": "图片",
    "image_caption": "图片描述",
    "image_footnote": "图片脚注",
    "table": "表格",
    "table_caption": "表格描述",
    "table_footnote": "表格脚注",
    "phonetic": "拼音",
    "code": "代码块",
    "code_caption": "代码描述",
    "ref_text": "参考文献",
    "algorithm": "算法块",
    "list": "列表",
    "header": "页眉",
    "footer": "页脚",
    "page_number": "页码",
    "aside_text": "装订线旁注",
    "page_footnote": "页面脚注"
}
```

坐标系统说明

`bbox` 坐标格式：`[x0, y0, x1, y1]`

- 分别表示左上、右下两点的坐标
- 坐标原点在页面左上角
- 坐标为相对于原始页面尺寸的百分比，范围在0-1之间

示例数据

```
[
    [
        {
            "type": "header",
            "bbox": [
                0.077,
                0.095,
                0.18,
                0.181
            ],
            "angle": 0,
            "score": null,
            "block_tags": null,
            "content": "ELSEVIER",
            "format": null,
            "content_tags": null
        },
        {
            "type": "title",
            "bbox": [
                0.157,
                0.228,
                0.833,
                0.253
            ],
            "angle": 0,
            "score": null,
            "block_tags": null,
            "content": "The response of flow duration curves to afforestation",
            "format": null,
            "content_tags": null
        }
    ]
]
```

中间处理结果 (middle.json)

文件命名格式： `{原文件名}_middle.json`

文件格式说明

vlm 后端的 middle.json 文件结构与 pipeline 后端类似，但存在以下差异：

- list变成二级block，增加 `sub_type` 字段区分list类型:
  - `text` （文本类型）
  - `ref_text` （引用类型）
- 增加code类型block，code类型包含两种"sub_type":
  - 分别是 `code` 和 `algorithm`
  - 至少有 `code_body` , 可选 `code_caption`

- `discarded_blocks`内元素type增加以下类型:
  - `header` （页眉）
  - `footer` （页脚）
  - `page_number` （页码）
  - `aside_text` （装订线文本）
  - `page_footnote` （脚注）
- 所有block增加`angle`字段，用来表示旋转角度，0，90，180，270

示例数据

- list block 示例

```
{
    "bbox": [
        174,
        155,
        818,
        333
    ],
    "type": "list",
    "angle": 0,
    "index": 11,
    "blocks": [
        {
            "bbox": [
                174,
                157,
                311,
                175
            ],
            "type": "text",
            "angle": 0,
            "lines": [
                {
                    "bbox": [
                        174,
                        157,
                        311,
                        175
                    ],
                    "spans": [
                        {
                            "bbox": [
                                174,
                                157,
                                311,
                                175
                            ],
                            "type": "text",
                            "content": "H.1 Introduction"
                        }
                    ]
                }
            ],
            "index": 3
        },
        {
            "bbox": [
                175,
                182,
                464,
                229
            ],
            "type": "text",
            "angle": 0,
            "lines": [
                {
                    "bbox": [
                        175,
                        182,
                        464,
                        229
                    ],
                    "spans": [
                        {
                            "bbox": [
                                175,
                                182,
                                464,
                                229
                            ],
                            "type": "text",
                            "content": "H.2 Example: Divide by Zero without Exception Handling"
                        }
                    ]
                }
            ],
            "index": 4
        }
    ],
    "sub_type": "text"
}
```

- code block 示例

```
{
    "type": "code",
    "bbox": [
        114,
        780,
        885,
        1231
    ],
    "blocks": [
        {
            "bbox": [
                114,
                780,
                885,
                1231
            ],
            "lines": [
                {
                    "bbox": [
                        114,
                        780,
                        885,
                        1231
                    ],
                    "spans": [
                        {
                            "bbox": [
                                114,
                                780,
                                885,
                                1231
                            ],
                            "type": "text",
                            "content": "1 // Fig. H.1: DivideByZeroNoExceptionHandling.java  \n2 // Integer division
without exception handling.  \n3 import java.util.Scanner;  \n4  \n5 public class DivideByZeroNoExceptionHandling  \n6 {
\n7 // demonstrates throwing an exception when a divide-by-zero occurs  \n8 public static int quotient( int numerator, int
denominator )  \n9 {  \n10 return numerator / denominator; // possible division by zero  \n11 } // end method quotient
\n12  \n13 public static void main(String[] args)  \n14 {  \n15 Scanner scanner = new Scanner(System.in); // scanner for
input  \n16  \n17 System.out.print(\"Please enter an integer numerator: \");  \n18 int numerator = scanner.nextInt();
\n19 System.out.print(\"Please enter an integer denominator: \");  \n20 int denominator = scanner.nextInt();  \n21"
                        }
                    ]
                }
            ],
            "index": 17,
            "angle": 0,
            "type": "code_body"
        },
        {
            "bbox": [
                867,
                160,
                1280,
                189
            ],
            "lines": [
                {
                    "bbox": [
                        867,
                        160,
                        1280,
                        189
                    ],
                    "spans": [
                        {
                            "bbox": [
                                867,
                                160,
                                1280,
                                189
                            ],
                            "type": "text",
                            "content": "Algorithm 1 Modules for MCTSteg"
                        }
                    ]
                }
            ],
            "index": 19,
            "angle": 0,
            "type": "code_caption"
        }
    ],
    "index": 17,
    "sub_type": "code"
}
```

内容列表 (content_list.json)

文件命名格式：`{原文件名}_content_list.json`

文件格式说明

vlm 后端的 content_list.json 文件结构与 pipeline 后端类似，伴随本次middle.json的变化，做了以下调整：

- 新增`code`类型，code类型包含两种"sub_type":
  - 分别是`code`和`algorithm`
  - 至少有`code_body`, 可选`code_caption`
- 新增`list`类型，list类型包含两种"sub_type":
  - `text`
  - `ref_text`
- 增加所有所有`discarded_blocks`的输出内容
  - `header`
  - `footer`
  - `page_number`
  - `aside_text`
  - `page_footnote`

示例数据

- code 类型 content

```
{
    "type": "code",
    "sub_type": "algorithm",
    "code_caption": [
        "Algorithm 1 Modules for MCTSteg"
    ],
    "code_body": "1: function GETCOORDINATE(d)  \n2:  $x \\gets d / l$ ,  $y \\gets d$  mod  $l$   \n3: return  $(x, y)$
\n4: end function  \n5: function BESTCHILD(v)  \n6:  $C \\gets$  child set of  $v$   \n7:  $v' \\gets \\arg \\max_{c \\in
C} \\mathrm{UCTScore}(c)$   \n8:  $v'.n \\gets v'.n + 1$   \n9: return  $v'$   \n10: end function  \n11: function BACK
PROPAGATE(v)  \n12: Calculate  $R$  using Equation 11  \n13: while  $v$  is not a root node do  \n14:  $v.r \\gets v.r +
R$ ,  $v \\gets v.p$   \n15: end while  \n16: end function  \n17: function RANDOMSEARCH(v)  \n18: while  $v$  is not a
leaf node do  \n19: Randomly select an untried action  $a \\in A(v)$   \n20: Create a new node  $v'$   \n21:  $(x, y)
\\gets \\mathrm{GETCOORDINATE}(v'.d)$   \n22:  $v'.p \\gets v$ ,  $v'.d \\gets v.d + 1$ ,  $v'.\\Gamma \\gets v.\\Gamma$
\n23:  $v'.\\gamma_{x,y} \\gets a$   \n24: if  $a = -1$  then  \n25:  $v.lc \\gets v'$   \n26: else if  $a = 0$  then
\n27:  $v.mc \\gets v'$   \n28: else  \n29:  $v.rc \\gets v'$   \n30: end if  \n31:  $v \\gets v'$   \n32: end while
\n33: return  $v$   \n34: end function  \n35: function SEARCH(v)  \n36: while  $v$  is fully expanded do  \n37:  $v
\\gets$  BESTCHILD(v)  \n38: end while  \n39: if  $v$  is not a leaf node then  \n40:  $v \\gets$  RANDOMSEARCH(v)  \n41:
end if  \n42: return  $v$   \n43: end function",
    "bbox": [
        510,
        87,
        881,
        740
    ],
    "page_idx": 0
}
```

- list 类型 content

```
{
    "type": "list",
    "sub_type": "text",
    "list_items": [
        "H.1 Introduction",
        "H.2 Example: Divide by Zero without Exception Handling",
        "H.3 Example: Divide by Zero with Exception Handling",
        "H.4 Summary"
    ],
    "bbox": [
        174,
        155,
        818,
        333
    ],
    "page_idx": 0
}
```

- discarded 类型 content

```
[{
    "type": "header",
    "text": "Journal of Hydrology 310 (2005) 253-265",
    "bbox": [
        363,
        164,
        623,
        177
    ],
    "page_idx": 0
},
{
    "type": "page_footnote",
    "text": "* Corresponding author. Address: Forest Science Centre, Department of Sustainability and Environment, P.O. Box 137, Heidelberg, Vic. 3084, Australia. Tel.: +61 3 9450 8719; fax: +61 3 9450 8644.",
    "bbox": [
        71,
        815,
        915,
        841
    ],
    "page_idx": 0
}]
```

总结

以上文件为 MinerU 的完整输出结果，用户可根据需要选择合适的文件进行后续处理：

- **模型输出**(使用原始输出):
  - model.json
- **调试和验证**(使用可视化文件):
  - layout.pdf
  - spans.pdf
- **内容提取**(使用简化文件):
  - *.md
  - content_list.json
- **二次开发**(使用结构化文件):
  - middle.json