Fullseye：基于类型化算子与物理仿真的图像检测和三维测量平台

○古瀬 和文†
†：独立研究人员
kazufumi@furuse.work

摘要： 本文提出开源平台 Fullseye。该平台将图像检测和三维测量处理构造成声明输入输出类型的算子链，使用物理与成像仿真生成的真值进行评价，并记录处理步骤以供复用。通过改变照明与光学条件生成模拟图像，可在搭建设备前确认处理能否成功及其失败条件。平台包含约 3,000 个算子、算子检索功能和 218 个概念验证程序。本文报告代表性示例的定量评价，并区分其验证阶段。
＜关键词＞ 图像检测，三维测量，物理仿真

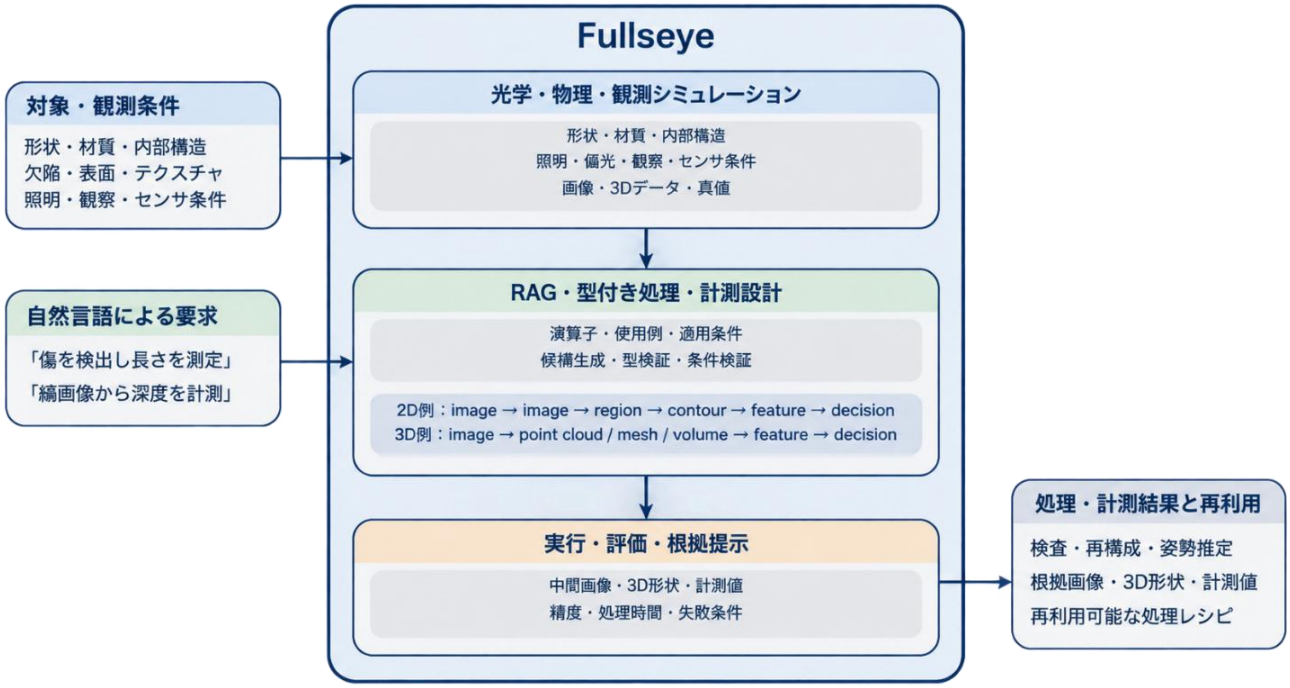

图 1 Fullseye 图像检测与测量管道的结构

1. 引言

在工业视觉中，外观检测、三维测量、光学与干涉测量、X 射线 CT 及机器人感知等课题，往往分别设计成像条件、处理步骤和评价方法。因此，在某一现场获得的处理与评价经验不易迁移到照明或对象略有差异的其他课题。此外，确认处理优劣需要正确答案（真值），但真实图像中往往无法得知缺陷的准确深度、位置和尺寸。Fullseye 以虚拟方式构建光学与成像系统并生成真值，再以此作为比较基准来解决这一问题。

本文提出的 Fullseye[1]（图 1）将应对上述两个困难——知识难以迁移和真值难以获取——的步骤连接在同一框架中。以“检测光泽零件表面的划痕并测量其长度”为例。用户首先给出零件形状与材质、划痕尺寸以及照明和相机条件，生成已知划痕真实位置与长度的模拟图像。其次，构建平滑→二值化→区域选择→轮廓提取→长度测量等处理链（第 4 章代码清单 1）；无法连接的组合会在执行前被拒绝。最后，依据模拟图像的真值定量确认检出率和长度误差，并记录在哪些照明与划痕条件下失败。同一流程也可用于点云、网格和体数据（三维数据）、干涉条纹与偏振图像（光学测量）以及 X 射线投影图像

(CT)。

预期使用场景如下。(a) 在搭建设备前，以改变照明和相机条件的模拟图像比较检测成败并筛选候选方案(第 4.2 节)。(b) 在缺陷实物较少的启动阶段，用尺寸和位置已知的缺陷评价处理方法与判定阈值(第 5 章)。(c) 以文本保存处理链、评价和失败条件，面对其他产线或照明老化等变化时用相同指标重新评价，并给出判断依据(代码清单 1)。(d) 让生成式 AI 组合处理候选方案，再根据类型与真值决定是否采用(第 5 章)。

2. 相关研究

BlenderProc2[2]和 Kubric[3]是生成带真值合成数据的平台，外观检测领域也有使用渲染的合成缺陷训练和评价缺陷分割的研究[4]。这些工作主要面向训练数据，而本文依据测量量的真值评价处理链。HALCON[5]采用将图像、区域和轮廓区分为类型的思想，Halide[6]则以声明方式描述处理管道。前者以执行处理为主，本文的重点是以同一类型系统处理 X 射线 CT、干涉和偏振等测量侧处理，并以开源形式供 LLM 调用。VisProg[7]、ViperGPT[8]、基于 API 文档检索的 Gorilla[9]以及面向工业检测的处理链生成[10]，均研究由 LLM 组合视觉处理；本文的区别在于依据类型一致性和真值决定候选方案的取舍。MVTec AD[11]和 MVTec 3D-AD[12]提供用于评价的真实图像数据。各组成要素均有先例，本文的贡献在于用一个平台连接测量量真值评价、基于类型的执行前检查与算子检索，以及记录失败条件的 PoC 回归测试。

3. 术语定义

本文术语定义如下。

【算子】对图像或数据实施的一项处理，例如高斯平滑、大津二值化和点云配准。算子实现为 Python 函数，具有名称、参数和输入输出类型。

【类型(数据类型)】算子接收和返回的数据种类。二维数据区分图像(image)、彩色图像、区域(region：像素集合)、轮廓(contour：亚像素点列)和特征量(feature：尺寸、形状指标等数值)；三维数据区分点云、网格和体数据。合格与否表示为针对特征量的阈值。

【类型一致性】处理链每一阶段中，前一阶段的输出类型与后一阶段的输入类型一致。例如，二值化(image → region)之后可以接轮廓提取(region → contour)，但不能接收图像的平滑处理(image → image)(图 2)。Fullseye 在执行首个阶段前检查二维处理链并拒绝不一致的组合；三维、光学等算子则在每次调用时检查输入类型。

图 2　处理链各阶段的输出与类型(图 3 金属表面的划痕)

【可复用处理】以算子名称和参数序列保存为文本的处理链。原则上，在相同版本中，只要输入和参数(包括随机种子)相同，就返回相同结果(确定性)。由于采用文本形式，可将其用于其他数据或课题进行比较，也可作为检索对象。

【真值】用于评价的正确答案。本文指生成模拟图像时已知的值(划痕位置、深度、台阶高度等)，以及解析解或公开值。

【PoC(Proof of Concept，概念验证程序)】针对一项课题，可一次执行输入生成、处理、与真值或比较对象的比较、合格标准及失败条件记录的程序。Fullseye 含有 218 个 PoC(也包括图像检测与测量以外的题材)，均由回归测试执行。

4. Fullseye 的结构

图 1 给出 Fullseye 的处理结构。光学、物理和观测仿真以对象和观测条件为输入，生成图像、三维数据及真值；类型化算子组成的处理链对其进行处理和测量，并以可复用形式保留评价结果和依据(中间图像、测量值与失败条件)。

4.1. 类型化算子与规模

Fullseye 是一个 Python 平台，将二维图像处理、三维几何、点云、网格与体数据处理、光学、干涉、偏振、结构光、光传输、X 射线 CT、声学和计量学统一为类型化算子。平台以 Apache-2.0 许可证发布，获

取方式和安装方法见代码清单 1 开头(Python 3.10 及以上;基本功能仅用 CPU 即可运行)。0.6.0 版(2026 年 10 月)共有 3,083 个算子。

为说明功能范围，本文以工业机器视觉的代表性算子体系 HALCON[5]作为比较基准。HALCON 算子参考手册将 2,313 个算子分为 30 章(Filters、Regions、Matching 等)。Fullseye 以机器可读形式记录功能最接近的 HALCON 算子名称。按名称对应，2,313 个算子中有 981 个(42.4%)存在对应项。这只是表示功能接近的对应表，并不保证参数或输出等价。30 章中有 23 章至少包含一个对应算子。未对应项多为绘图(Graphics)、元组运算、系统、文件输入输出、字符识别和深度学习等，并非图像处理算法本身;但模板匹配(Matching)仍有 58 个、三维匹配仍有 44 个未对应项，有待后续补充。Fullseye 并不声称与 HALCON 兼容或可替代 HALCON。

二维处理链可构造成 image → image → region → contour → feature 等类型序列(图 2)。照明校正、平滑和边缘增强属于 image → image，阈值处理与区域分割属于 image → region，轮廓提取属于 region → contour，形状与尺寸测量属于 contour → feature。

代码清单 1　处理链的执行及类型不一致处理链的执行前拒绝(含执行结果)

```
# https://github.com/furuse-kazufumi/fullseye
# pip install fullseye
import fullseye as fs
chain = ["gaussian", "otsu", "invert_region",
         "select_largest", "gen_contour_region_xld",
         "length_xld"]
L = fs.run_pipeline(img, chain)   # 周长 609.3 px
fs.run_pipeline(img,
    ["gaussian", "otsu", "gaussian"])
# ValueError: run_pipeline: type mismatch at
#  stage 3 (gaussian): it takes 'image' but
#  stage 2 (otsu) outputs 'region' -- the chain
#  was refused before any stage ran
```

代码清单 1 把从平滑到长度测量的 6 阶段处理链用于一幅 640×480 像素的模拟图像，其中放置了宽 4 px、长 300 px 的暗色划痕。划痕周长为 609.3 px(真值 608 px，换算为长度是 300.7 px)，处理时间约为每幅 15 ms(仅使用普通台式 PC 的 CPU，20 次中位数)。在二值化之后放置接收图像的平滑处理时，系统会在执行首个阶段前指出类型不一致的阶段并拒绝处理链。

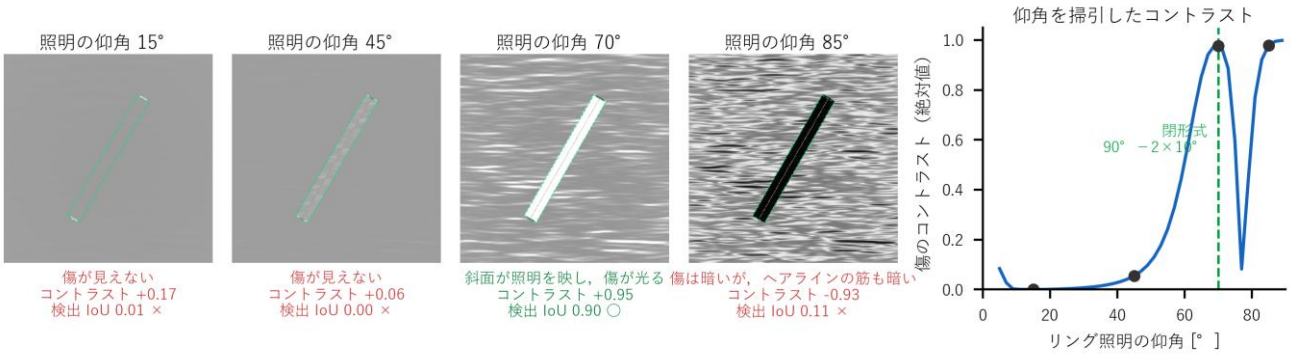

図 3　环形照明仰角对金属表面划痕外观的影响

4.2.　物理与成像仿真及真值

系统以对象形状、材质、表面反射特性、缺陷形状、照明与观察条件以及传感器条件为输入，通过光学与成像仿真同时生成观测图像和真值，包括深度、法线、缺陷区域、反射分量、物体位姿和内部结构等。因此，可以参照仅凭真实图像难以获得的正确答案，对处理进行定量评价。系统地改变条件，还可调查性能和失败条件。

光学算子覆盖从光源到像素的各阶段(0.6.0 版约 200 个)。照明部分按照划痕明暗差和背景均匀度，对低角度和高角度环形、穹顶、同轴及透射照明候选方案排序。材质部分依据金属复折射率、菲涅耳公式和微表面 BRDF 计算反射。镜头部分根据各表面参数进行光线追迹，求出畸变与色差，可用阻尼最小二乘法优化参数，并处理衍射极限 MTF 与离焦。传感器特性可按 EMVA 1288 流程求取，也可模拟线扫描成像。此外还支持波动光学传播、琼斯与穆勒矩阵偏振、白光干涉、光场(包括复眼型)及光子计数(dToF)。借助这些功能，可在搭建设备前通过模

拟图像比较照明、镜头和传感器候选方案。

图 3 是在拉丝金属表面绘制斜面为 10°的 V 形划痕，并仅改变环形照明仰角的示例。各像素亮度依据照明、表面方向和反射特性计算，划痕真实位置由表面形状确定。划痕在闭式解 90°−2×10°=70°附近明亮发光；85°时划痕呈暗色，但拉丝纹理也变暗，平滑与二值化处理链误检纹理而失败。每种照明条件下的检测成败及失败原因均可依据真值确认。右端曲线是照明算子（lighting_sweep）求得的斜面与平面明暗差绝对值，其定义与各图像数值不同。

4.3. 知识语料库与算子检索（RAG）

每个算子都附有说明文档，记录说明、输入输出类型、用例、适用条件和已知限制；这些文档兼作算子帮助页面与检索知识语料库。通过全文检索（BM25[13]），可从“去噪”“相移 深度”等词语找到相关算子。基于大语言模型（LLM）的编码辅助可直接调用该检索，也可通过 MCP[14]服务器在对话中调用；MCP 是 AI 调用外部工具的协议，该服务器提供算子检索、说明查阅和处理链执行（RAG：检索增强生成）。与基于 API 文档检索的生成[9]相同，LLM 把需求拆分为检索词，读取检出的算子说明和输入输出类型，再构建处理候选方案。

4.4. GUI 与验证机制

图 4 给出 Fullseye Studio 操作界面。图中把图 2 截至区域提取的处理链用于图 3 中仰角 70°的图像，右侧三维显示将高度放大 8 倍。通过样例处理链选择、按类型排列的算子列表（如[image → image]、[region → region]）和代码面板，可交互式构建并执行保持类型一致的处理链，并获得 Python 代码。

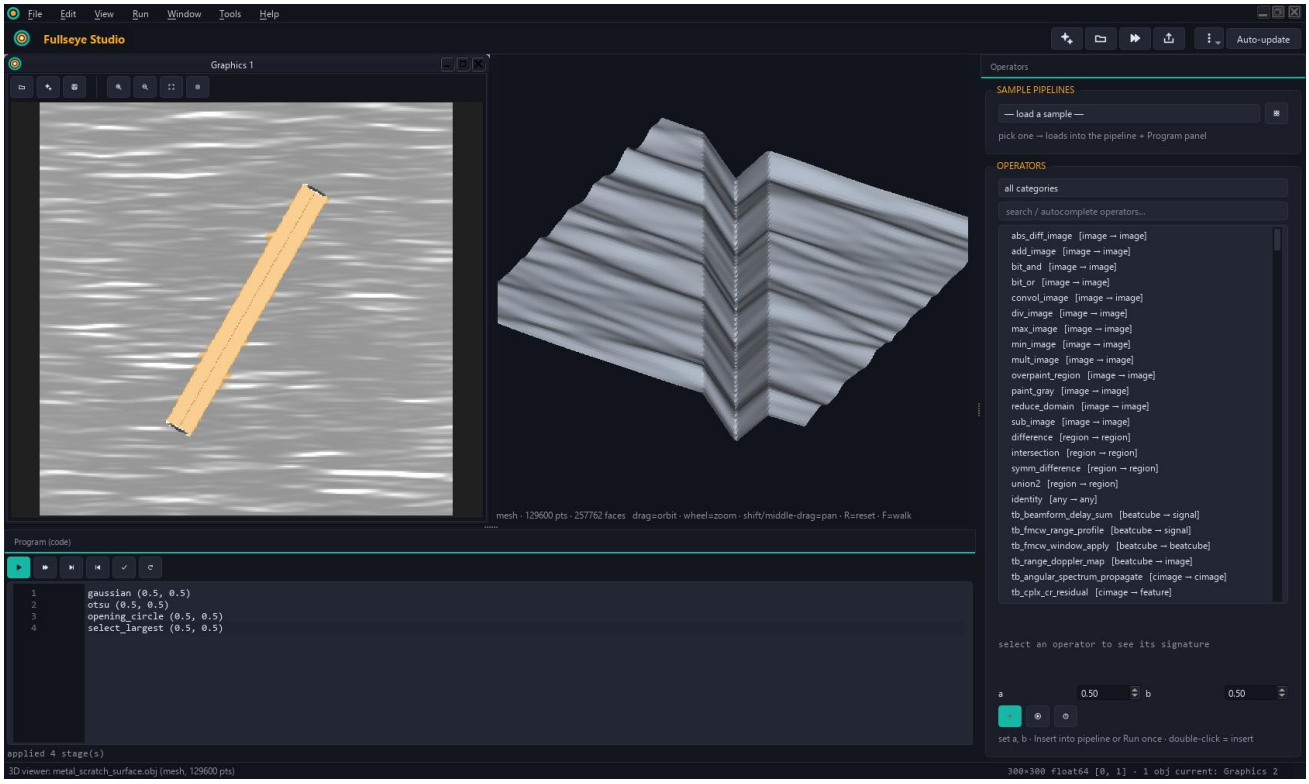

图 4　Fullseye Studio 操作界面

整个库通过约 1.8 万项回归测试，持续验证二维算子声明的输入输出类型与实际输出一致，以及文档数值与执行结果一致。

5. 代表性 PoC 验证

Fullseye 实现了面向二维图像检测、三维重建、点云、网格与体数据处理、光学、干涉与偏振测量、X 射线 CT、热与声学分析以及机器人与空间感知的 PoC（第 3 章）。这些程序用于确认对象与观测条件、类型化算子、评价量、真值或比较对象以及结果可

视化能够在共同框架中处理。

其中许多 PoC 以交互方式创建。作者给出课题和真值生成方法后，基于 LLM 的编码辅助查阅算子文档与输入输出类型，组合经典图像处理及光学、成像仿真算子，编写处理链与评价。作者将结果与真值核对后决定取舍；若失败，则指出原因并要求重新组合。类型不一致的组合在执行前被拒绝，结果又由真值判定，因此类型错误和偏离真值的结果会在采用前被发现。借助这一流程，个人也能积累跨领域的 218 个 PoC。

图 5 给出代表性 PoC 的输出示例。各面板的课题、输入和评价指标不同，因此本图仅表示实现范围，不能比较 PoC 间的优劣。【H】的形状采用丝川小行星真实形状模型[15]，细部起伏和岩石为合成。表 1 汇总各观测与测量类别的代表性评价对象。

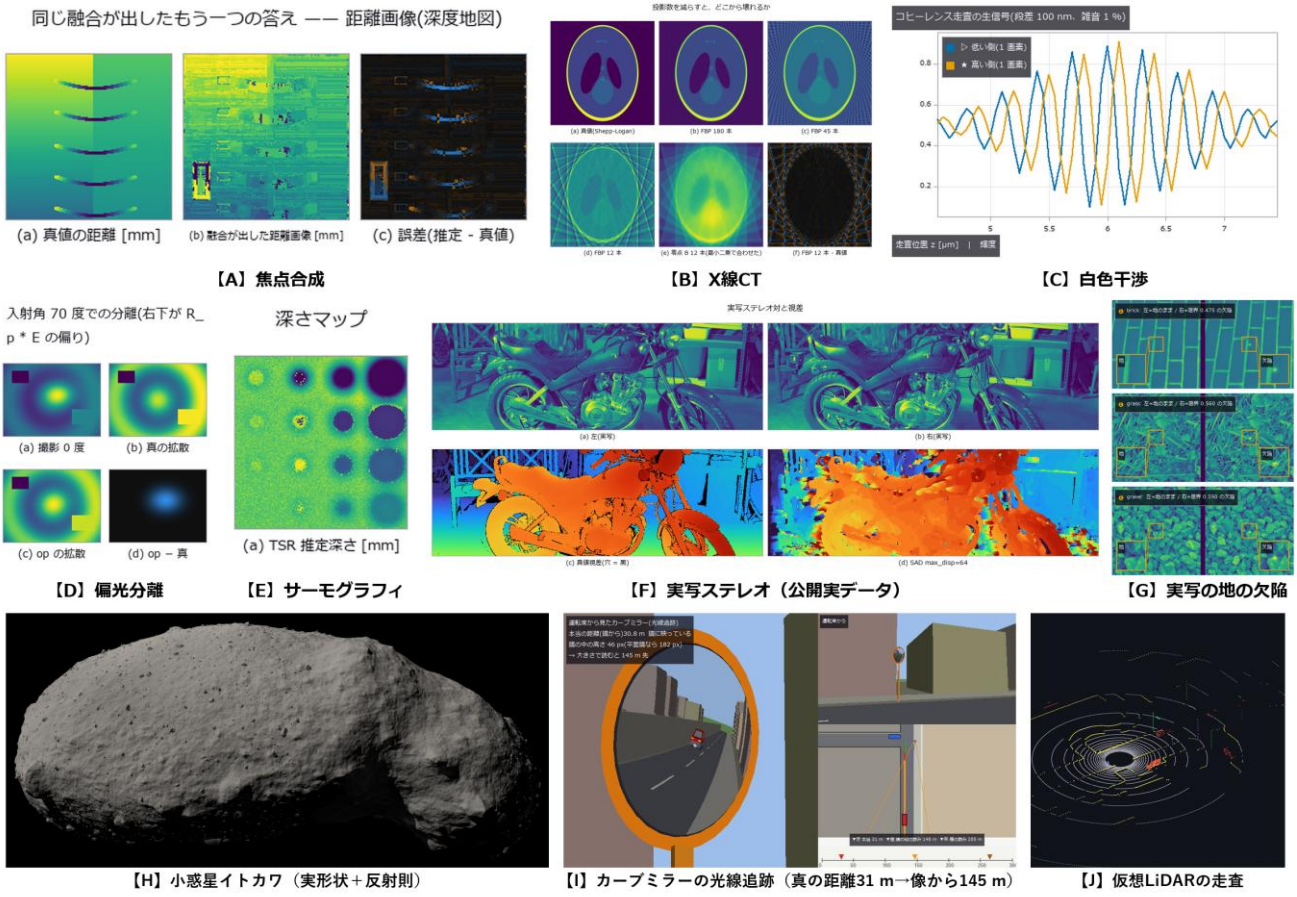

图 5　代表性 PoC 输出示例（【F】为公开真实数据，【G】为半合成，其余为合成）

表 1　PoC 类别与代表性评价对象

类别	代表性评价对象
图像检测与外观测量	缺陷候选、尺寸、形状、纹理
三维测量与几何处理	深度、点云、网格、位置与位姿
X 射线 CT 与体数据处理	重建图像、体积、空隙、内部形状
光学、干涉与偏振	相位、反射、偏振特性、光学图像
热、声学与时间序列	温度场、频率特性、异常指标
机器人与空间感知	物体位姿、抓取候选、安全距离

每个 PoC 不仅记录最终判定或测量值，还记录输入数据、中间图像或中间形状、所用算子链、主要参数、评价指标、真值或比较对象及已知限制。由此可确认处理结果的依据、重新执行步骤，并将算子与评价方法复用于其他课题。

各 PoC 的成熟度并不一致，因此区分带真值合成数据验证、公开真实图像或实测数据验证，以及连接真实传感器与设备的验证。合成数据用于系统改变形状、材质、照明、观察方向、噪声和缺陷条件，调查性能与失败条件；真实图像和实测数据用于验证仅靠合成难以捕捉的反射、背景、对象差异、校准误差、缺失数据和成像噪声。未验证范围明确标为研究原型，与已验证功能区分。0.6.0 版成熟度台账由代码库内容自动生成：35 项能力中 32 项处于合成数据验证阶段，3 项处于公开真实数据验证阶段，尚无实机验证。

代表性定量评价见表 2。表 2 中的数值均针对合成数据的解析解或生成时嵌入的真值。运行标签 v0.6.0 的各 PoC（examples/poc_*.py）即可复现。偏振一行是依据与合成相同的菲涅耳公式校验实现。

表 2　代表性 PoC 的定量评价（带真值或比较对象，均为合成数据）

PoC（课题）	真值／比较对象	评价指标	结果
焦点堆叠（聚焦法）	已知深度和全聚焦图像	全聚焦 PSNR、深度误差	全聚焦 33.69 dB（中央帧 28.52 dB，最佳单帧 29.44 dB）；整体深度 RMS 误差 0.92 mm（常数估计 1.17 mm），有纹理 0.47 mm，无纹理 2.20 mm（劣于区域最佳常数 0.36 mm）
X 射线 CT 保真度（Shepp-Logan[16]）	解析总质量	质量守恒	投影总质量相对误差 1.3×10^{-5}；滤波反投影（FBP）[17]重建为−0.0099%
白光干涉台阶测量	台阶真值 50～500 nm	偏差与离散度	噪声 1%（40 次、144 像素平均）时，偏差不超过 2.4 nm，标准差 10～14 nm（台阶 50 nm 时相对约 20%）
偏振镜面反射分离	菲涅耳闭式解	分离误差	与闭式解 R_p·E（布儒斯特角为 0）在舍入误差（不超过 4.7×10^{-17}）范围内一致（实现校验）
热成像无损检测	已知缺陷深度	深度误差	直径 8 mm 以上时为−6～+2%；将拟合时间窗从 25 秒缩短到 4 秒后，8 个小缺陷的最差误差由+612%降至−14%（时间窗用同一数据选择）
复眼型光场测量	合成真值（已知视差与深度）	SNR 增益（与√N 比较）、深度误差	N=5 时 2.25（√5=2.24），N=49 时 5.33（√49=7.00）；深度 RMSE 为 0.010 px/视点（远离不连续处的可靠像素）
由高速相机视频求恢复系数与摩擦系数（乒乓球）	公开实测值[18]、合成真值	恢复系数的速度依赖、摩擦系数、弹跳类型	斜率−0.00572 /(km/h)（与实测−0.0058 相差 1.4%），摩擦系数 0.250（真值 0.25），滑动/滚动判定 15/15 一致

热成像 PoC 最初得到的结果是：缺陷直径越小，深度高估越严重，看似是物理检测极限。对照实验表明，原因不是物理现象，而是拟合时间窗的选取。把时间窗从 25 秒缩短到 4 秒后，原为+612%的缺陷误差降至−9%，8 个小缺陷的最差值也降至−14%。但时间窗是观察同一 8 个缺陷后选择的，按深度选择

时间窗的泛化能力尚未验证。记录失败条件并区分原因也被视为评价的一部分。

在把复眼光学系统建模为光场传感器的 PoC[19, 20]中，将同一物点的 N 个视点叠加所得 SNR 增益与独立噪声理想值√N 比较。这相当于用多眼相机降低暗目标噪声。N=49 时增益仅为 5.33，原因是整数像素平移使超出图像的边缘像素产生误差；去除边缘 8 个像素后为 7.19（√49=7.00）。视差非整数时仍存在插值误差导致的饱和。

从高速相机（1000 fps）拍摄的球体弹跳中，仅凭图像读取恢复系数与摩擦系数，并与公开实测值核对（表 2）。这一方法也可用于评价零件投放和输送中的弹跳与滑动。

Fullseye 算子不仅用于独立 PoC，还被其他研究项目调用，作为物理仿真中机器人的“眼睛”（表 3）。控制（腿部或推进）由各项目的物理模型承担，Fullseye 只负责从点云与距离图像感知地形和目标。在不需要视觉的条件下（无台阶、目标无偏移），两者结果一致，确认 Fullseye 不会干扰控制。四足机器人则不存在应由视觉选择的更佳步态，加入视觉后几乎没有改善。这类负面结果也作为适用范围边界予以记录。平台还提供入口（g1_walk_policy），用于在物理仿真中执行 GPU 训练的人形机器人策略（图 6）。另行训练的奔跑策略经此入口在 MuJoCo 中执行，12 秒前进 50 m 且未跌倒。表 3 中的改善来自各项目利用 Fullseye 估计的台阶高度或目标位置所采取的策略；相关项目代码不包含在本文公开内容中。

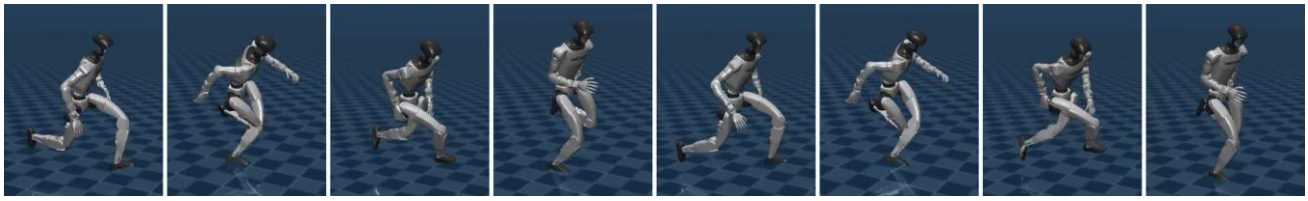

图 6　人形机器人 G1 奔跑

表 3　使用 Fullseye 进行机器人感知的示例（物理仿真，与无视觉比较）

课题	Fullseye 的作用（算子）	无视觉	有 Fullseye 视觉
双足行走（平面双足模型，台阶 2～14 cm 的 7 种条件，最长 5 秒的行走距离）	地形点云→高程图→台阶高度估计（elevation_map, step_edges）。点云由台阶形状生成（无噪声）。每种条件 1 次（确定性）	有台阶时平均 3.35 m	平均 4.10 m（+0.75 m，无恶化条件）
火箭着陆（着陆垫偏移 0.5～2.5 m）	向下距离点云（噪声 1 cm）→高度阈值→重心（centroid），估计着陆垫位置（误差不超过 4 mm）。5 种偏移条件×各 30 次	着陆垫内着陆率 21%，着陆误差 1.58 m	着陆率 67%，着陆误差 0.74 m

6. 讨论

Fullseye 的要点并非把多领域处理集中到一个代码库，而是关联对象与观测条件、数据表示、算子、评价指标和可视化，并以可复用方式管理。类型一致性同时保障处理链构建与评价流程的安全，并作为按目的检索和复用算子的索引。

物理仿真生成的带真值合成数据可以评价稀有缺陷和极端观测条件，但不能完全再现实机的照明不均、像差、传感器噪声与个体差异。不能仅凭合成评价声称真实环境性能。平台已经开始使用公开真实数据验证。对于带真值视差的真实立体图像（图 5【F】，Middlebury 2014[21]的 1 对，缩小至 1/4），默认设置中误差超过 2 像素的比例为 95.06%，比仅输出恒定视差更差；使用半全局匹配（SGM）[22]后仍为 15.81%。在 3 种真实纹理（图 5【G】，scikit-image[23] 附带，CC0）中植入已知缺陷的半合成评价中，检测极限比残差噪声量相同的合成背景高 2.03～3.47 倍。极限取决于背景图案而非噪声量，这是仅用合成背景无法发现的差异。需要逐步加入公开实测数据和实机验证，以明确平台的实用性与适用范围。

LLM 组合处理候选方案可提高效率，但不能只依据语言模型输出决定是否采用。采用前检查类型、参数范围、确定性、评价数据上的性能和已知失败条件，可兼顾 AI 辅助效率与工业测量所需的可复现性、可追溯性和可解释性。平台还让未参与开发的通用 AI 代理（GenSpark）使用公开版本检查易用性，并复现其指出的问题后修复。本文未对算子检索和 LLM 组合处理候选方案的准确率进行定量评价。

7. 总结

本文介绍了 Fullseye 这一图像检测与三维测量平台，它集成类型化算子、物理与成像仿真的真值评价以及算子检索（RAG）。本文依据真值或比较对象定量评价代表性 PoC，区分合成数据与公开真实数据的验证阶段，并明确实机验证是今后的课题。

今后将扩大公开真实图像、实测数据和真实传感器的验证范围，推进其在半导体封装、精密光学零件、无损检测和 Physical AI 中的应用，并使用 MVTec AD[11]等公开基准评价。代表性 PoC 的图与视频可在图 7 所示页面（https://furuse.work/view2026/）查看。此外，还将扩充 HALCON 未对应项较多的模板匹配和三维匹配，并研究将复眼型光场成像置于前级、把基于昆虫视觉神经系统（连接组）的处理[24]接在后级的扩展。该方向也可能用于只以低延迟输出亮度变化的事件相机[25]。

图 7　代表性 PoC 图像与视频的索引页面

致谢　本文所示 PoC 的部分实现、文献调查和文字修改使用了生成式 AI（Claude）。作者根据执行结果和原始文献核实了所列数值与内容。

参考文献

[1] 古瀬和文：Fullseye, https://github.com/furuse-kazufumi/fullseye, PyPI fullseye 0.6.0（tag v0.6.0）, 2026.

[2] M. Denninger, et al.：BlenderProc2: A Procedural Pipeline for Photorealistic Rendering, J. Open Source Software, Vol.8, No.82, 4901, 2023.

[3] K. Greff, et al.：Kubric: A Scalable Dataset Generator, Proc. CVPR 2022, pp.3749-3761, 2022.

[4] J. Fulir, L. Bosnar, H. Hagen and P. Gospodnetić：Synthetic Data for Defect Segmentation on Complex Metal Surfaces, Proc. CVPR Workshops 2023, pp.4424-4434, 2023.

[5] MVTec Software GmbH：HALCON Operator Reference, https://www.mvtec.com/doc/halcon/, 访问日期 2026 年 9 月 11 日.

[6] J. Ragan-Kelley, et al.：Halide: A Language and Compiler for Optimizing Parallelism, Locality, and Recomputation in Image Processing Pipelines, Proc. PLDI 2013, pp.519-530, 2013.

[7] T. Gupta and A. Kembhavi：Visual Programming: Compositional Visual Reasoning Without Training, Proc. CVPR 2023, pp.14953-14962, 2023.

[8] D. Surís, S. Menon and C. Vondrick：ViperGPT: Visual Inference via Python Execution for Reasoning, Proc. ICCV 2023, pp.11888-11898, 2023.

[9] S. G. Patil, T. Zhang, X. Wang and J. E. Gonzalez：Gorilla: Large Language Model Connected with Massive APIs, Proc. NeurIPS 2024, pp.126544-126565, 2024.

[10] A. Martikkala, A. Daareyni, A. Queguineur and I. Flores Ituarte：LLM-assisted Workflow Generation for Low-cost Visual Inspection and Image Analysis, IET Conference Proceedings, Vol.2025, No.28, pp.54-61, 2025.

[11] P. Bergmann, M. Fauser, D. Sattlegger and C. Steger：MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection, Proc. CVPR 2019, pp.9584-9592, 2019.

[12] P. Bergmann, X. Jin, D. Sattlegger and C. Steger：The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization, Proc. VISAPP 2022, pp.202-213, 2022.

[13] S. Robertson and H. Zaragoza：The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends in Information Retrieval, Vol.3, No.4, pp.333-389, 2009.

[14] Model Context Protocol：Specification, https://modelcontextprotocol.io/specification, 访问日期 2026 年 10 月 12 日.

[15] R. Gaskell, et al.：Gaskell Itokawa Shape Model V1.1, NASA Planetary Data System, doi:10.26033/3b2j-yy57, 2007.

[16] L. A. Shepp and B. F. Logan：The Fourier Reconstruction of a Head Section, IEEE Trans. Nuclear Science, Vol.21, No.3, pp.21-43, 1974.

[17] A. C. Kak and M. Slaney：Principles of Computerized Tomographic Imaging, IEEE Press, 1988.

[18] Y. Inaba, et al.：Effect of Changing Table Tennis Ball Material from Celluloid to Plastic on the Post-Collision Ball Trajectory, J. Human Kinetics, Vol.55, pp.29-38, 2017.

[19] M. F. Land and D.-E. Nilsson：Animal Eyes, 2nd ed., Oxford University Press, 2012.

[20] R. Ng, et al.：Light Field Photography with a Hand-held Plenoptic Camera, Stanford Tech. Report CSTR 2005-02, 2005.

[21] D. Scharstein, et al.：High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth, Proc. GCPR 2014, LNCS 8753, pp.31-42, 2014.

[22] H. Hirschmüller：Stereo Processing by Semiglobal Matching and Mutual Information, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.30, No.2, pp.328-341, 2008.

[23] S. van der Walt, et al.：scikit-image: image processing in Python, PeerJ, Vol.2, e453, 2014.

[24] J. K. Lappalainen, et al.：Connectome-constrained networks predict neural activity across the fly visual system, Nature, Vol.634, pp.1132-1140, 2024.
[25] G. Gallego, et al.：Event-Based Vision: A Survey, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.44, No.1, pp.154-180, 2022.

古瀬和文：独立研究人员，正在开发跨图像处理、三维测量和物理仿真的开源图像检测平台 Fullseye。研究方向包括用带真值评价验证图像处理、可解释的处理自动组合，以及借鉴生物视觉的测量技术。https://furuse.work