Reactive Trajectory Planning for Robotic Operations in an Unstructured Environment

Simulated by Tecnomatix Process Simulate

Giulio Carpi Lapi

Université de Strasbourg

August 18, 2025

UFR de **mathématique** et d'**informatique**
Université de Strasbourg

SIEMENS

Context: Industry 4.0 & Smart Manufacturing

- **Industry 4.0**: Integration of AI, IoT, and data analytics into factories.
- **Digital Twins & Simulation**: Essential for virtual testing, process optimization, and reducing costs.
- **Automation**: Key to competitiveness, precision, and safety.

The Environment: Siemens & Process Simulate

- Internship hosted at **Siemens Digital Industries Software** in Toulouse.
- Primary software platform: **Tecnomatix Process Simulate**, a leading tool for robotic simulation and workcell design.
- Development followed the **Scrum** agile methodology.
- Work conducted within the **Kineo team (ARK - Advanced Robotic Kinematics)**.

Kineo Presentation

- Founded in **2001** as a spin-off from **LAAS-CNRS** in Toulouse.
- Acquired by **Siemens PLM Software** in **2012**.
- Specializes in technologies for **automatic motion planning** and **collision detection**.
- Develops **Software Development Kits (SDKs)** integrated into major PLM applications.

Kineo Technology Overview

- **KineoWorks**: Computes collision-free trajectories.
- **Kineo Collision Detector (KCD)**: Performs fast interference checks.
- **Kineo Flexible Cables**: Simulates the behavior of deformable cables.
- **Kineo Nesting**: Optimizes the arrangement of parts within a defined space.
- **KineoWorks Interact**: GUI toolkit for building 3D applications.

Project Objectives

- **Primary Goal**: Develop a **Proof-of-Concept** for a reactive planner in Process Simulate.
  - Enabling the system to **reactively avoid obstacles** based on **real-time virtual camera data**.
  - Allowing for **dynamic path adjustments** in response to **environmental changes**.
- **Secondary Goal**: Evaluate the performance and limitations of this approach.

The Perception Pipeline: An Overview

1. **Capture & Generate**: Convert depth data from virtual cameras into a 3D point cloud.
2. **Filter Stage 1**: Remove the robot's own geometry from the point cloud.
3. **Filter Stage 2**: Remove the known static environment to isolate dynamic objects.
4. **Detect Collisions**: Check for interference between the robot and the final (dynamic) point cloud.

A Multi-Layered Architecture

- **Presentation (C#/WPF)**: User Interface.
- **Bridge (C++/CLI)**: Connects managed C# to native C++.
- **Core Engine (C++)**: High-performance simulation and algorithms.

Test Scenario: Bin-Picking Operation

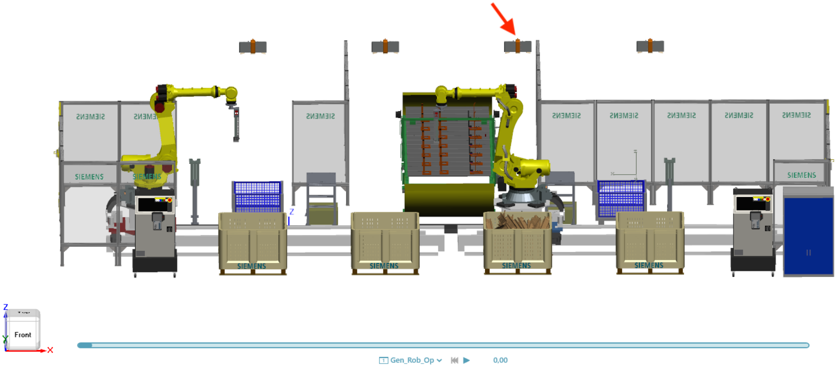

Figure: Initial state before the operation begins.

Test Scenario: Bin-Picking Operation

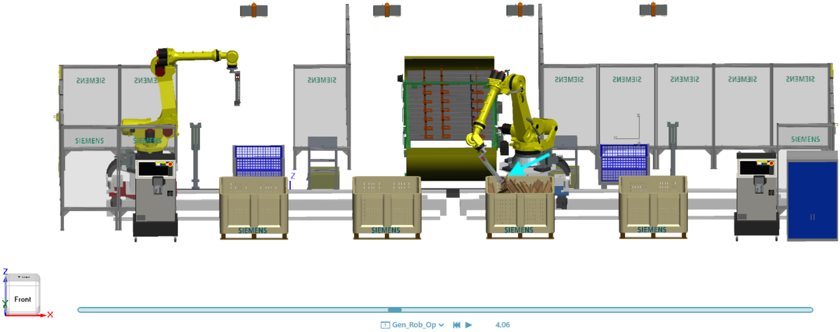

Figure: The robot arm descends into the bin to pick the target object.

Test Scenario: Bin-Picking Operation

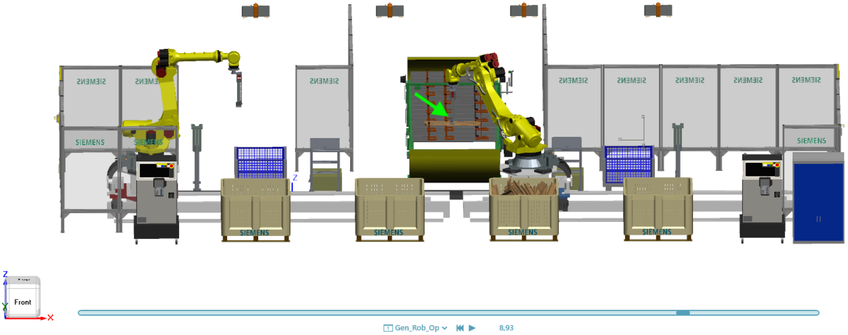

Figure: The robot places the object on the conveyor belt.

Virtual Camera RGB output

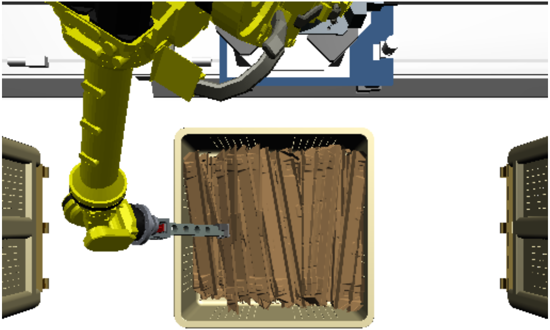

Figure: RGB virtual camera output at initial state.

Virtual Camera RGB-D output

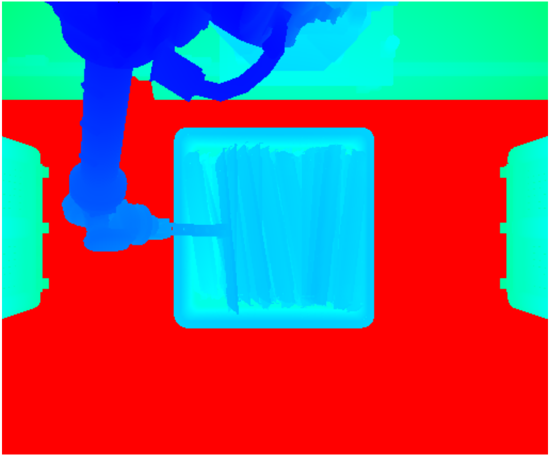

Figure: RGB-D virtual camera output at initial state.

WPF Dialog Box (C#/WPF)

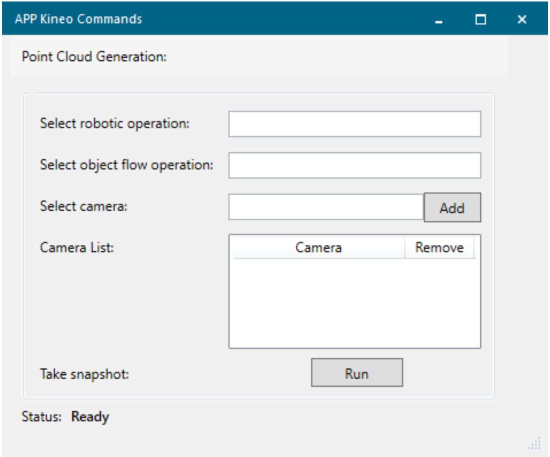

Figure: WPF dialog box for user interaction.

Performance-Driven Camera Configuration

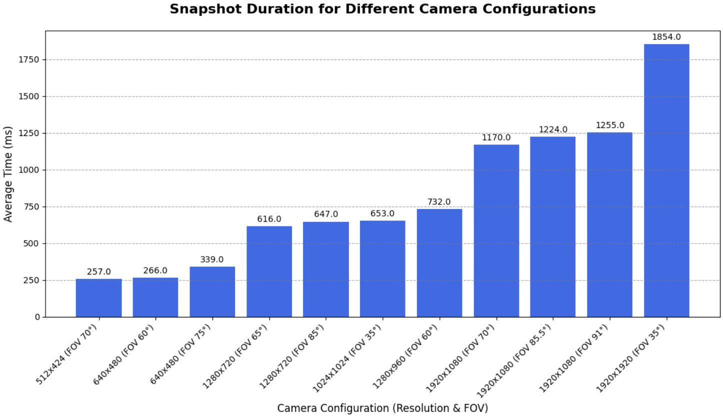

Figure: Snapshot duration vs. camera configuration.

- Benchmarked snapshot duration for various camera resolutions and FOVs.
- **Selected Config**:
  - Resolution: **512x424**
  - FOV: **70°**
- **Result**: Balances detail with performance, averaging **250ms** per snapshot.

Depth Data Processing: Depth buffer

- **Depth buffer**: 2D array stored as 1D float list in row-major order.
- Each value represents the **distance** from the **camera** to an **object** at **pixel (u, v)**.
- Access **pixel (u, v)** at index: $v \times \text{width} + u$.

Depth Data Processing: Camera parameters

- **Focal Length (f)**: Derived from field of view (FOV) and image dimensions.

$$f = \frac{\sqrt{\text{width}^2 + \text{height}^2}}{2\tan\left(\frac{\text{FOV}}{2}\right)}$$

- **Principal Points (c_x, c_y)**: Image center coordinates.

$$c_x = \frac{\text{width} - 1}{2}, \quad c_y = \frac{\text{height} - 1}{2}$$

Depth Data Processing: Camera Reference Frame

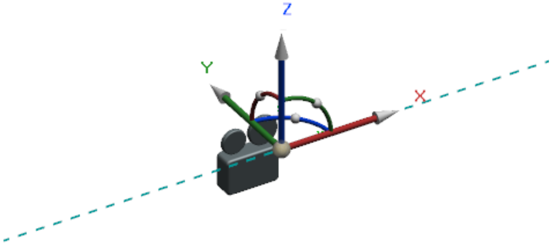

Figure: Camera reference frame in Process Simulate.

Depth Data Processing: Pixel to 3D point

For each pixel (u, v):

1. Retrieve **depth**: $Z = \mathrm{depthBuffer}[v \times \mathrm{width} + u]$
2. Compute **normalization factor**:

$$t = \sqrt{\left(\frac{u - c_x}{f}\right)^2 + \left(\frac{v - c_y}{f}\right)^2 + 1}$$

3. Calculate **3D coordinates** in camera space:

$$X = \frac{Z}{t}, \quad Y = \frac{(c_x - u) \times Z}{f \times t}, \quad Z = \frac{(v - c_y) \times Z}{f \times t}$$

Point Cloud Generation: Camera to world

Transform to world-space P_{world} using the camera's **transformation matrix**.

$$P_{world} = T_{camera \to world} \times P_{camera}$$

Generated Point Cloud

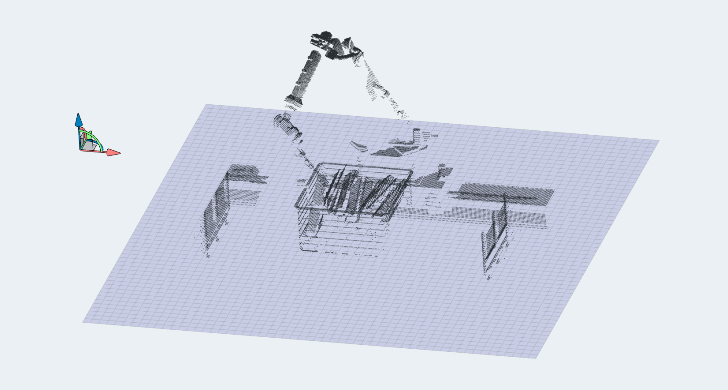

Figure: Generated oint cloud (includes robot).

Robot's Geometries Filtering

1. Get **current robot link poses**.
2. Compute **Oriented Bounding Boxes (OBBs)** for each link.
3. Discard any point from the cloud that falls inside one of these OBBs.

Prevent the robot from detecting itself as an obstacle.

Filtered Point Cloud

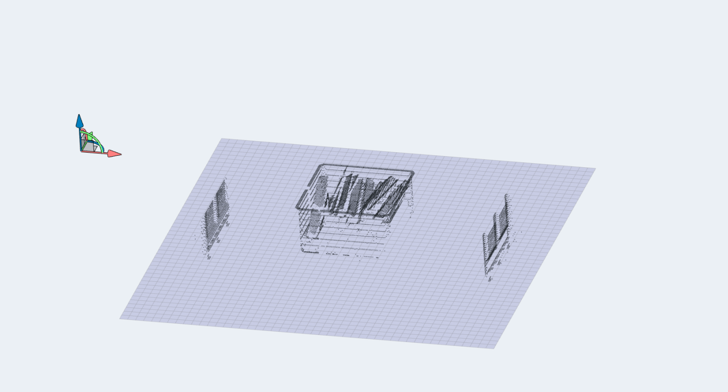

Figure: Filtered point cloud (After robot points are removed).

Object Flow Operation

A standard industrial task used to validate the system.

- **Robot Operation**: Pick a part from a bin and place it on a conveyor.
- **Dynamic Obstacle**: An "Object Flow" operation moves a box to intersect the robot's path, simulating an AGV or drone.

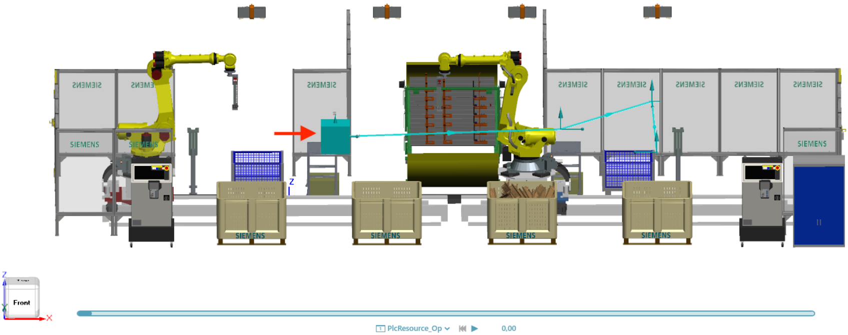

Figure: The dynamic obstacle (box) and its planned trajectory (blue).

Virtual Camera RGB output

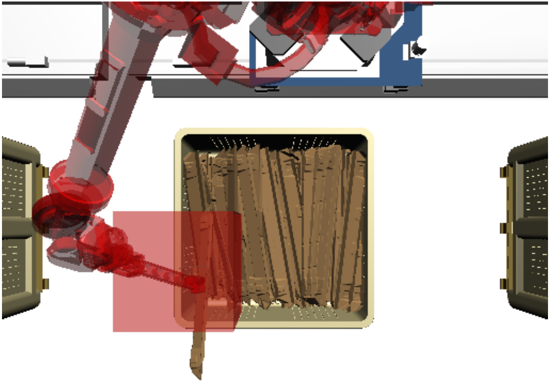

Figure: RGB virtual camera output at the moment of detected collision.

Virtual Camera RGB-D output

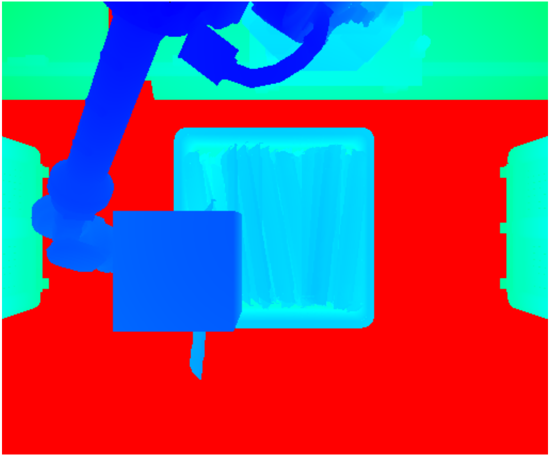

Figure: RGB-D virtual camera output at the moment of detected collision.

Static Environment Filter

Goal: Isolate *only* dynamic, unexpected obstacles.

- **Method**:
  1. At t=0, capture a **static baseline** point cloud of the environment.
  2. At each subsequent time step, compare the current point cloud to the baseline.
  3. Discard any point that is close to a point in the baseline.
- **Result**: A point cloud containing only new or moved objects.

Collision Detection: Identifying Threats

The system correctly identifies the dynamic obstacle and halts the robot.

- **Input**: The final, filtered point cloud containing only the moving box.
- **Mechanism**: The **Kineo Collision Detector (KCD)** checks for interference between the robot's geometry and the dynamic point cloud.
- **Action**: When a collision is predicted, the custom simulator stops the robot's motion, preventing impact.

Conclusion

- **Successfully integrated** a reactive perception system into Process Simulate.
- **Developed a robust pipeline**:
  - Custom simulation loop for dynamic scenarios.
  - Event-driven data capture from virtual RGB-D cameras.
  - Two-stage filtering algorithm to isolate dynamic obstacles.
- **Validated** the "sense and react" capability in a realistic scenario.
- **Limitation**: Performance is bound by the simulation's virtual camera snapshot speed (250ms), not by the algorithms themselves.

Future Work

Performance Optimization

- Improve filtering algorithms using spatial data structures (k-d trees, voxel grids) to handle denser point clouds faster.

Path Replanning: The "Act" Phase

- The critical next step for full autonomy.
- **Workflow**:
  1. On collision detection, *pause* the robot.
  2. Pass the dynamic obstacle's point cloud to **KineoWorks**.
  3. Compute a new, collision-free path to the original goal.
  4. Resume motion along the new path.

The End

Thank you for your attention!

IBM.
What is Industry 4.0?
https://www.ibm.com/think/topics/industry-4-0.

McKinsey & Company.
What are Industry 4.0, the Fourth Industrial Revolution, and 4IR?
https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-are-industry-4-0-the-fourth-industrial-revolution-an
2022.

SAP SE.
What is Industry 4.0?
https://www.sap.com/products/scm/industry-4-0/what-is-industry-4-0.html.

Patrick Ruane, Patrick Walsh, and John Cosgrove.
Using simulation optimization to improve the performance of an automated manufacturing line.
Procedia Computer Science, 217:630–639, 2023.

Accessed: 2025-07-31.

FlexSim Software Products, Inc.
Manufacturing Simulation.
https://www.flexsim.com/manufacturing-simulation.

Visual Components.
Manufacturing simulation: how it works and why you should do it?
https://www.visualcomponents.com/what-is-manufacturing-simulation.

Sean Camarella, Michael P. Conway, Kevin Goering, and Mark Huntington.
Digital twins: The next frontier of factory optimization.
https://www.mckinsey.com/capabilities/operations/our-insights/digital-twins-the-next-frontier-of-factory-optimization, 2024.

Mohsen Attaran and Bilge Gokhan Celik.
Digital twin: Benefits, use cases, challenges, and opportunities.

Decision Analytics Journal, 6:100165, 2023.
Accessed: 2025-07-31.

Matterport LLC.
Industry 4.0: Guide to Smart Manufacturing with Digital Twin Technology.
https://matterport.com/it/learn/digital-twin/manufacturing?srsltid=AfmBOorqBU2r1gVnj6xvwrJRYfuSFcR6i2GCi9XHi1rqAV4LheywZnD_.

SAP SE.
What is product lifecycle management (PLM)?
https://www.sap.com/products/scm/plm-r-d-engineering/what-is-product-lifecycle-management.html.

Heather Krebsbach.
What is PLM (Product life cycle management)?
https://www.atlassian.com/agile/product-management/plm.

Oracle Corporation.
What is PLM (Product Lifecycle Management)?

https://www.oracle.com/scm/product-lifecycle-management/what-is-plm/.

Oliver Munro.
Automation in Manufacturing: Uses, Examples, & Trends.
https://www.unleashedsoftware.com/blog/automation-in-manufacturing, 2023.

Apan Dastider, Hao Fang, and Mingjie Lin.
Retro: Reactive trajectory optimization for real-time robot motion planning in dynamic environments.
In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
Accessed: 2025-07-31.

Alexander Dahlin and Yiannis Karayiannidis.
Trajectory scaling for reactive motion planning.
In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5242–5248, 2022.
Accessed: 2025-07-31.

Siemens Digital Industries Software.
Tecnomatix Process Simulate.
https://plm.sw.siemens.com/en-US/tecnomatix/process-simulate-software.
Accessed: 2025-07-31.

Siemens PLM Software.
Polarion.
https://://polarion.plm.automation.siemens.com/products/overview.
Accessed: 2025-07-31.

Siemens Digital Industries Software.
Tecnomatix Documentation, 2025.
Internal Siemens documentation, not publicly accessible.

Siemens Product Lifecycle Management Software Inc.
Kineo Components Reference Documentation, 2025.
Kineo Components documentation, not publicly accessible.

Scratchapixel.

The Lookat Function for Framing a Camera.
https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/lookat-function/framing-lookat-function.html, 2024.
Accessed: 2025-03-20.

Scratchapixel.
Computing the Pixel Coordinates of a 3D Point.
https://www.scratchapixel.com/lessons/3d-basic-rendering/computing-pixel-coordinates-of-3d-point/mathematics-computing-2d-coordinates-of-3d-points.html, 2024.
Accessed: 2025-03-20.

Kenji Hata, Silvio Savarese.
Camera Models.
https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf, 2024.
Course Notes, Stanford University, Accessed: 2025-03-25.

Mustafa Boyuk.
Obtaining Point Cloud from Depth Images with Intel RealSense D-435 Camera.
https://medium.com/@mustafaboyuk24/obtaining-point-cloud-from-depth-images-with-intel-realse
2023.
Medium, Accessed: 2025-03-31.

Kyle Simek.
Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix.
http://ksimek.github.io/2013/08/13/intrinsic/, 2013.
Blog post, Accessed: 2025-03-31.

David Lenaerts.
Reconstructing Positions from the Depth Buffer.
https://www.derschmale.com/2014/01/26/reconstructing-positions-from-the-depth-buffer/, 2014.
Blog post, Accessed: 2025-04-01.

Nicolas Burrus.

Kinect Calibration with RGBDemo.
https://nicolas.burrus.name/oldstuff/kinect_calibration/, 2011.
Accessed: 2025-04-01.

Tully Foote and Mike Purvis.
Standard Units of Measure and Coordinate Conventions.
https://www.ros.org/reps/rep-0103.html, 2010.
Accessed: 2025-08-08.

Gregorij Kurillo, Evan Hemingway, Mu-Lin Cheng, and Louis Cheng.
Evaluating the accuracy of the azure kinect and kinect v2.
Sensors, 22(7):2469, 2022.

Intel Corporation.
Intel® realsense™ depth camera d435 - product specifications.
https://www.intel.com/content/www/us/en/products/sku/128255/intel-realsense-depth-camera-d435/specifications.html, 2018.
Accessed: 2025-08-13.

Orbbec.
Femto bolt hardware specifications.
https://www.orbbec.com/documentation/femto-bolt-hardware-specifications/.
Accessed: 2025-08-13.

Siemens Digital Industries Software.
Kineo SDK Documentation, 2025.
Internal documentation, not publicly accessible. Accessed August 2025.

Kineo Wiki.
Process Simulate general architecture, 2025.
Internal documentation, not publicly accessible. Accessed August 2025.