Mobile Manipulator Project

3D Concrete Printing Robot

Project codename: Mobile Manipulator
Degree Program: MS-R
Department: Mechanical Engineering
Number of Project Units: 24
Project Course Number: 24794

Andrew ID: aramakr2
Email: aramakr2@andrew.cmu.edu
Mobile Phone: +1 412-954-8545

Aditya Ramakrishnan

Jan – Feb, 2022

.CERLAB.

Computational Engineering
& Robotics Lab.

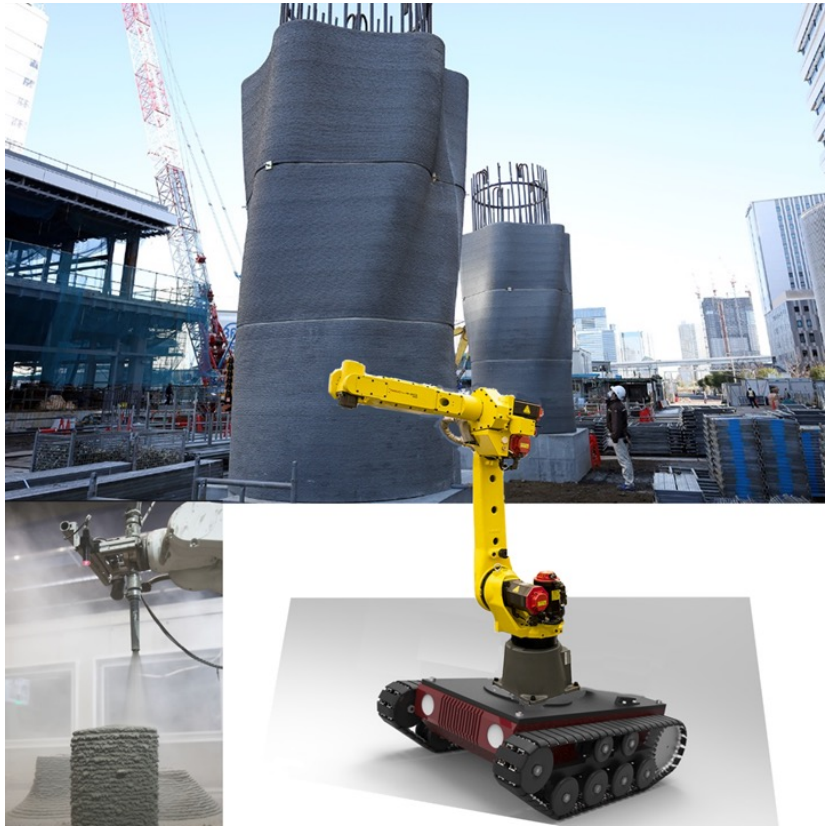
Carnegie Mellon University

Mobile Manipulator

3D Concrete Printing Localization

3D Concrete Printing Summary

- Project Overview:
  - Develop fully autonomous 3D spray printing mobile manipulator for construction of complex surfaces.
- Semester goals:
  - Select a wheeled base and Design support frame
  - Create simulation environments:
    - Robotic Gantry System
    - Wheeled Mobile Manipulator
  - Generate material laying simulation
  - Develop computer vision
    - localization and laser profile



2022.09.05

- Summary of your work:
  - Pros and cons of different localization methods.
  - Considerations for proprioceptive and exteroceptive sensors that can be implemented.
  - Consideration for a unified approach to both concrete printing systems.
  - Arrive at final localization approach for each system.

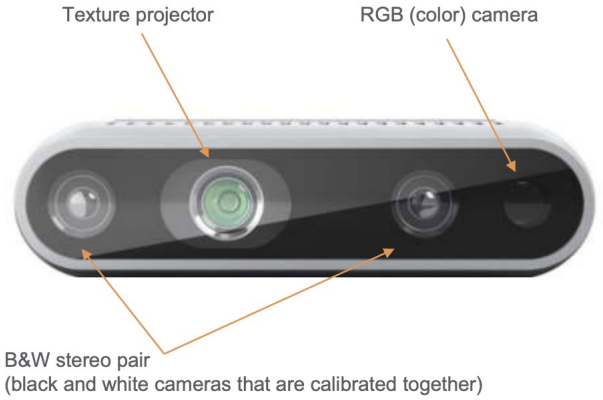
- Next steps:
  - Review semester milestones with Kyshalee
  - Reproduce demo with Jeremy manipulator controls for kogumi robot

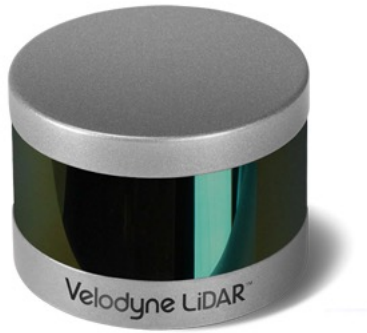
Hardware Selection and Test Requirements

Sensor fusion

Exteroperceptive Sensors:

- Intel D435i - aids AMR Obstacle Avoidance
  - Precalibrated + Depth accuracy reliable upto 6m
- Velodyne Puck – localization (accurate estimate of range & bearing) ~ +/- 3cm error for 100m range





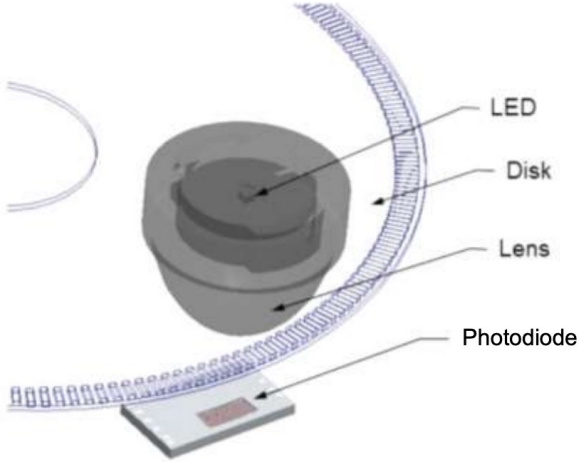
Propioceptive Sensors (Internal to Gantry/AMR):

- IMU
- Optical Encoder (accurate odometry for localization)

Test setup for Velodyne:

- Ground truth : sharp wall corner or edged surface
- Short range measurements for 10-15 meter distances
- Expected Veloview results well within range of error

IMU



Optical Encoder

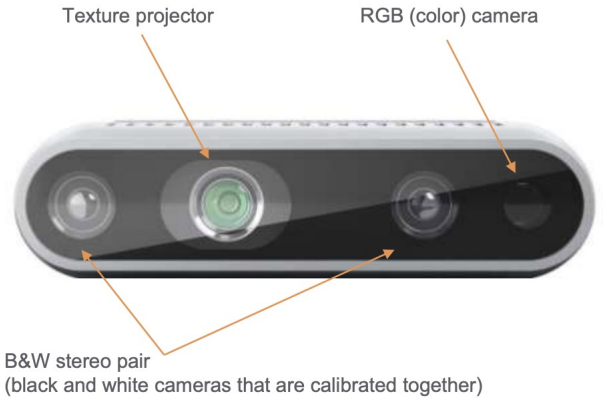
Hardware Selection and Test Requirements

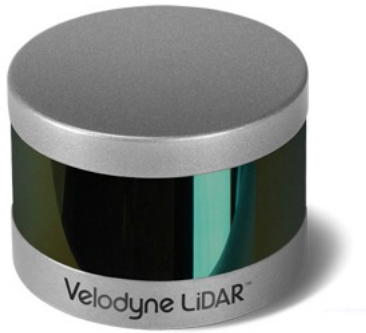
Sensor selected:

- Velodyne VLP-16 Puck
- Intel Realsense D435i Depth Sensor

Setup:

- 3 sensor setup ideal for accurate localization





AMR:

- Velodyne VLP-16 (robot-base mounted):
  + point-cloud data capture
  + 3D structure profiling
- Intel Realsense D435i Depth Sensor (base/ceiling):
  + simple robot base localization
  + obstacle detection/avoidance

Gantry:

- 2 Velodyne VLP-16 (gantry-mounted):
  + point-cloud data capture
  + 3D structure profiling
- Intel Realsense D435i Depth Sensor (gantry-mounted)
  + simple robot base localization

Brief Overview of Localization Methods considered

- Iterative Kalman Filter

  - Alleviates covariance approximation in EKF
  - IEKF iterates over the measurement update step over fixed number of iterations/ stopping criterion
  - Linearizes about better estimates and improves approximation
  - Increased computational cost (iterations)

- Unscented Kalman Filter (Implementing)

  - The difference between EKF and UKF is in the representation of the Gaussian random variables.
  - State distribution approximated with a Gaussian distribution
  - UKF is not based on the local linear approximation, however it still does use linearization, namely linear regression, but is an acceptable approach here.

- Particle Filter

  - Transforms a set of points via known nonlinear equations and combines the results to estimate the mean and covariance of the state
  - Number of points used in a particle filter generally needs to be much greater than the number of points in a UKF (computationally expensive)

- Batch Processing (Stretch Goal)

  - Formulates the problem using Kronecker product and finds the least square solution using n pairs of data
  - Ideal approach as it provides the most accurate results in literature.

2022.09.12

- Summary of your work:
  - Literature Review of Localization methods considered for gantry 3D concrete printing station.
  - Localization requirement for gantry simplified to 3 options:
    - Extended Kalman Filter (based on KF approach)
    - Iterative EKF
    - Point-cloud based article filter
- Next steps:
  - Review literature for AMR localization/mapping methods

Extended Kalman Filter

- EKF is well-suited for applications where it takes significant time to collect and process paired measurements.
- The implementation consumes a single pair of measurements to update the state at every time step and requires many iterations to converge.
- For highly nonlinear functions such as this application, where we estimate the pose, EKF can significantly underestimate the covariance.
- Although, EKF is probably the most widely used estimation algorithm for nonlinear systems, it is however, difficult to tune, and only reliable for systems that are almost linear on the time scale of the updates.

- Extended Kalman filter is easy in use and computably efficient.
- Its limitation is that it works well only with small nonlinearities (close to linearity) since it is based on local linear approximation, and it disregards probabilistic uncertainties that arise in linearization of equations of state. Thus, for highly nonlinear models the EKF is not the optimum estimator.
- Due to Jacobians, measurement model and dynamics model must be differentiable. Only additive and Gaussian process noises are allowed. These problems can be solved by the unscented Kalman filter.

Iterated EKF

- To alleviate the covariance approximation in EKF, an iterated extended kalman filter (IEKF) can be used.
- The main difference between EKF and IEKF is the measurement update step. IEKF iterates over the measurement update step for a fixed number of iterations or until a stopping criterion is met.
- This allows linearizing about better estimates and improves approximation each iteration.
- Innovation or the norm of the difference between the consequent state estimates can be used as a stopping criterion.
- This reduces the linearization error at the cost of increased computational requirements.

Note: *It's important to note that, the state covariance matrix is kept fixed during this iterative step and only the state estimate is iterated. If we were to also update the covariance matrix, it would mean we process two identical measurements. Once the iterations end, we use the same EKF update equations to update the state and covariance.*

Iterated Extended Kalman Filter:

while k $< n_{max}$:

while i $< i_{max}$ or $\|x_i - x_{i-1}\| < \varepsilon$:

$$K_i = P_k H_i^T (H_i P_k H_i^T + R)^{-1}$$
$$y_i = z_i - h_i - H_i^T (x_k - x_i)$$
$$x_i = x_k + K_i\, y_i$$

k # of poses
i: iteration

$$x_k = x_i$$
$$P_k = (I - K_i H_i) P_k$$

Particle Filter

- The particle filter transforms a set of points via known nonlinear equations and combines the results to estimate the mean and covariance of the state.
- Hence the nonlinearity of the system is more accurately modeled which is acceptable provided there are no processing constraints.
  (Note: in the particle filter the points are chosen randomly)
- The number of points used in a particle filter generally needs to be much greater than the number of points in a UKF.
- A key difference between the UKF and particle filters is that the estimation error in a UKF does not converge to zero in any sense, but the estimation error in a particle filter does converge to zero as the number of particles (and hence the computational effort) approaches infinity..

2022.09.19

- Summary of your work:
  - Literature Review of Localization methods considered for AMR (advanced SLAM approaches for increased complexity) :
    - Batch Processing and Factor Graph based methods
    - Unscented Kalman Filter
- Next Steps:
  - VLP 16 Calibration test for short range measurements (error < 1cm required)

Batch Processing / Factor graph

- Batch processing class formulates the problem using Kronecker product and finds the least square solution using n pairs of data. Kronecker product is a generalization of the outer product to matrices. It is useful when solving or optimizing a function where the unknown is a matrix.

Least Squares Solution in Batch Processing

- Outer Product for vectors:

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^{\top} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \\ u_4 v_1 & u_4 v_2 & u_4 v_3 \end{bmatrix}$$

- Kronecker Product for Matrices ($\otimes$):

$$A \otimes B = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{11}b_{13} & a_{12}b_{11} & a_{12}b_{12} & a_{12}b_{13} \\ a_{11}b_{21} & a_{11}b_{22} & a_{11}b_{23} & a_{12}b_{21} & a_{12}b_{22} & a_{12}b_{23} \\ a_{21}b_{11} & a_{21}b_{12} & a_{21}b_{13} & a_{22}b_{11} & a_{22}b_{12} & a_{22}b_{13} \\ a_{21}b_{21} & a_{21}b_{22} & a_{21}b_{23} & a_{22}b_{21} & a_{22}b_{22} & a_{22}b_{23} \\ a_{31}b_{11} & a_{31}b_{12} & a_{31}b_{13} & a_{32}b_{11} & a_{32}b_{12} & a_{32}b_{13} \\ a_{31}b_{21} & a_{31}b_{22} & a_{31}b_{23} & a_{32}b_{21} & a_{32}b_{22} & a_{32}b_{23} \end{bmatrix}$$

- Good for solving equations where unknown is a matrix:

$$A_{m\times m}X_{m\times n} + X_{m\times n}B_{n\times n}{}^{T} = C_{m\times n} \rightarrow \ (I_{n\times n} \otimes A_{m\times m} + B_{n\times n} \otimes I_{m\times m})\, vec(X_{m\times n}) = vec(C_{m\times n})$$

Has the form: $A\ x\ =\ b$

Batch Processing / Factor graph

Closed Form Solution in Batch Processing

- Using $vec(\underbrace{A_{m\times m}X_{m\times n}}_{\text{matrix}}B_{n\times n})_{\text{vector}} = \underbrace{(B_{n\times n}{}^{T}\otimes A_{m\times m})}_{\text{matrix}}\underbrace{vec(X_{m\times n})}_{\text{vector}}$

$$R_A R_X = R_Y R_B \rightarrow R_A R_X R_B{}^{T} = R_Y \quad \rightarrow (R_B \otimes R_A)\, vec(R_X) - vec(R_Y) = 0$$

$$\rightarrow (-I \quad R_B \otimes R_A)\begin{bmatrix} vec(R_Y) \\ vec(R_X) \end{bmatrix} = 0$$

- For n poses:

$$\begin{bmatrix} nI & \overbrace{-\sum_{i=1}^{n} R_{B_i} \otimes R_{A_i}}^{K} \\ -\sum_{i=1}^{n} R_{B_i}{}^{T} \otimes R_{A_i}{}^{T} & nI \end{bmatrix}\begin{bmatrix} vec(R_Y) \\ vec(R_X) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Longrightarrow \begin{matrix} n\, vec(R_Y) - K\, vec(R_X) = 0 \\ -K^{T}\, vec(R_Y) + n\, vec(R_X) = 0 \end{matrix}$$

Has the form: $A\ x = b$

Unscented EKF

- The difference between EKF and UKF is in the representation of the Gaussian random variables.
- EKF approximates the state distribution with a Gaussian distribution and propagates the Gaussian variables by linearizing the process model and/or measurement model.
- UKF uses a deterministic sampling approach by representing the Gaussian distribution with a set of samples around the mean, aka sigma points, and directly passes the sigma points through the nonlinear process and/or measurement model.

- Thus, the major advantage of the unscented Kalman filter is that it is not based on the local approximation.
- For the estimation of the state with non-Gaussian noises particle filters are used which are based on the sequential Monte Carlo method (Particle Filters)

State vector $x = [w_x, t_x]$

Process model: $x_k = x_{k-1}$

Sigma points:

n_x: number of states
λ: design parameter

$$X_k = \left[x_k, x_k + \sqrt{(\lambda + n_x)P_k}, x_k - \sqrt{(\lambda + n_x)P_k}\right]_{n_x \times (2n_x+1)}$$

Predicted mean: $\bar{x}_k = \sum_{i=1}^{n_x} w_i X_k^i$

$$w_i = \frac{\lambda}{\lambda + n_x}, \quad i = 0$$

$$w_i = \frac{\lambda}{2(\lambda + n_x)}, \quad i = 1 \cdots 2n_x + 1$$

Predicted covariance:

$$\bar{P}_k = \sum_{i=1}^{2n_x+1} w_i (X_k^i - \bar{x}_k)(X_k^i - \bar{x}_k)^T$$

Measurement model: $h_k(x_k) = AX - XB$

Pseudo sensor measurement: $z_k = 0$

Predicted mean measurement:

$$\bar{h}_k = \sum_{i=1}^{n_x} w_i h_k(X_k^i)$$

Predicted measurement covariance:

$$S_k = \sum_{i=1}^{2n_x+1} w_i (h_k(X_k^i) - \bar{h}_k)(h_k(X_k^i) - \bar{h}_k)^T + R$$

Unscented EKF

Note: *The difficulty in implementing any Kalman-type filters for nonlinear state transitions stems from the numerical stability issues required for precision, however the UKF does not escape this difficulty in that it uses linearization as well, namely linear regression.*

The stability issues for the UKF generally stem from the numerical approximation to the square root of the covariance matrix, whereas the stability issues for the EKF stem from possible issues in the Taylor Series approximation along the trajectory.

Unscented Kalman Filter:

while k $< k_{max}$:

$$X_k = \left[x_k \ , x_k + \sqrt{(\lambda + n_x)P_k} \ , x_k - \sqrt{(\lambda + n_x)P_k}\right]$$

$$\bar{x}_k = \sum_{i=1}^{n_x} w_i \, X_k^i$$

$$\bar{P}_k = \sum_{i=1}^{2n_x+1} w_i \left(X_k^i - \bar{x}_k\right)\left(X_k^i - \bar{x}_k\right)^T$$

$$\bar{h}_k = \sum_{i=1}^{n_x} w_i h_k(X_k^i)$$

$$S_k = \sum_{i=1}^{2n_x+1} w_i \left(h_k(X_k^i) - \bar{h}_k\right)\left(h_k(X_k^i) - \bar{h}_k\right)^T + R$$

$$T_k = \sum_{i=1}^{2n_x+1} w_i \left(X_k^i - \bar{x}_k\right)\left(h_k(X_k^i) - \bar{h}_k\right)^T$$

$$K_k = T_k {S_k}^{-1}$$

$$P_k = \bar{P}_k - K_k S_k {K_k}^T$$

$$X_k = \bar{x}_k - K_k\left(z_k - \bar{h}_k\right)$$

2022.10.03

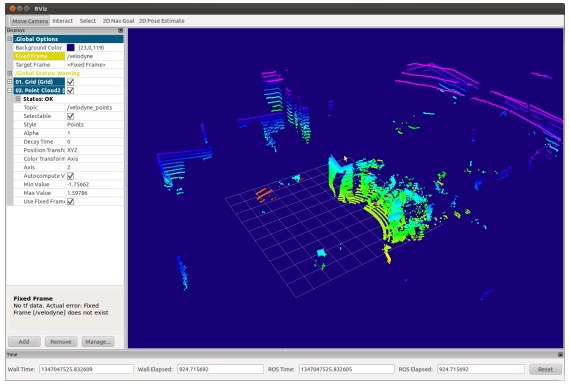
- Summary of your work:
  - Test rig for VLP 16 setup for true north yaw-error corrections.
  - Calibration procedure defined with Kyshalee and required components for rig purchased and printed as seen in images.
- Next steps:
  - MIL 19 test setup and error logging.

Concrete Spray Printing - Localization

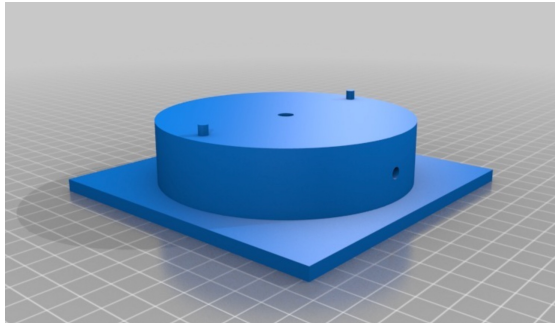
Test Specifications

The primary objective is to orient the VLP-16 sensor towards a distinct feature to identify discrepancies arising from innate yaw angle errors. How we accomplish this is detailed below:

- **Ground truth environment:**
  - Feature edge for measurement recording in Veloview -
    - Wall corners or edged surfaces contain minimal features, and a distinct edge-edge contour : 2 flat, straight walls intersecting at a vertical corner serves as an ideal feature to align the point cloud
  - Precisely demarcated measurement points of 5m, 10m and 15m to record test results



- **Precise positioning of LIDAR:**
  - Align the expected 0° yaw angle (90.0° perpendicular to the line between mounting alignment pins of the VLP-16) to a feature located at 5m, 10m, and 15m respectively
  - 3D printed base plate for LIDAR fixture, with precisely toleranced alignment pins to clock the puck



Concrete Spray Printing - Localization

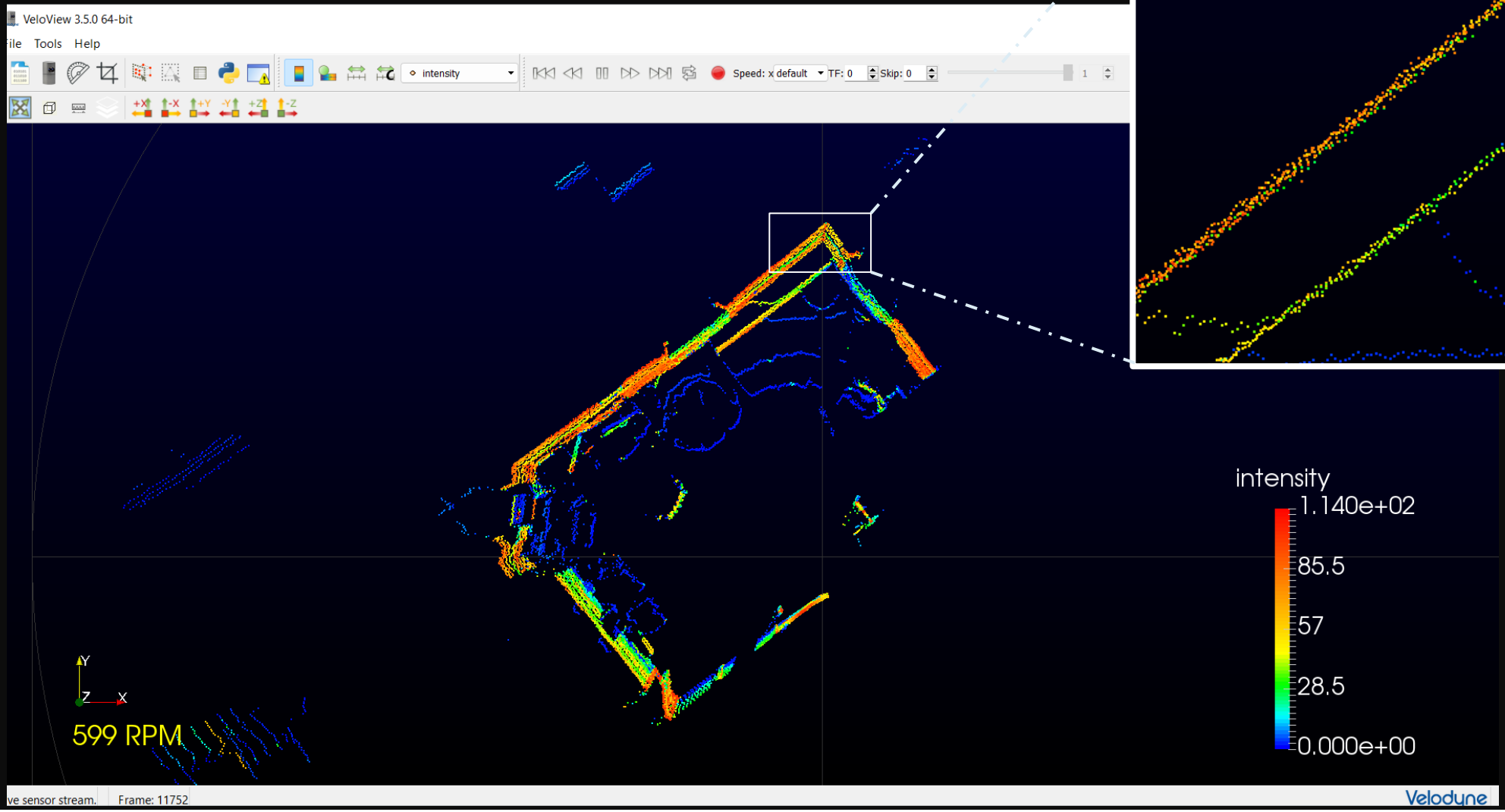
- **Precise positioning of LIDAR contd.:**
  - To ensure recording of absolute yaw error we take readings from VLP-16 True North:
    - Fix a co-axial bore-sighter laser on 3D printed base-pate fixture at 0° yaw.
    - Note: Co-axial lasers often don't fire perfectly in-axis, so alignment correction is required about a single axis (laser source is coaxial to the shaft)
    - We roll the bore-sighter laser linearly and mark the orientation at which only pitch error exists, and zero yaw error is observed (see gif).
  - 3D printed base plate for LIDAR fixture, with precisely toleranced alignment pins to clock the puck
    - Introduce an additional bore within the structure to mount the bore-sighter laser (Ensure that the location is oriented 90° to the alignment pins)
- **Calibration : Angular offset => Linear displacement**
  - Verify by correcting angular parameter on Veloview to get internal calibration offset
  - Increase + or - value until the corner lines up on the x=0 gridline
  - Detected yaw error of (-0.95° azimuth correction)
  - Procedure explained on next slide

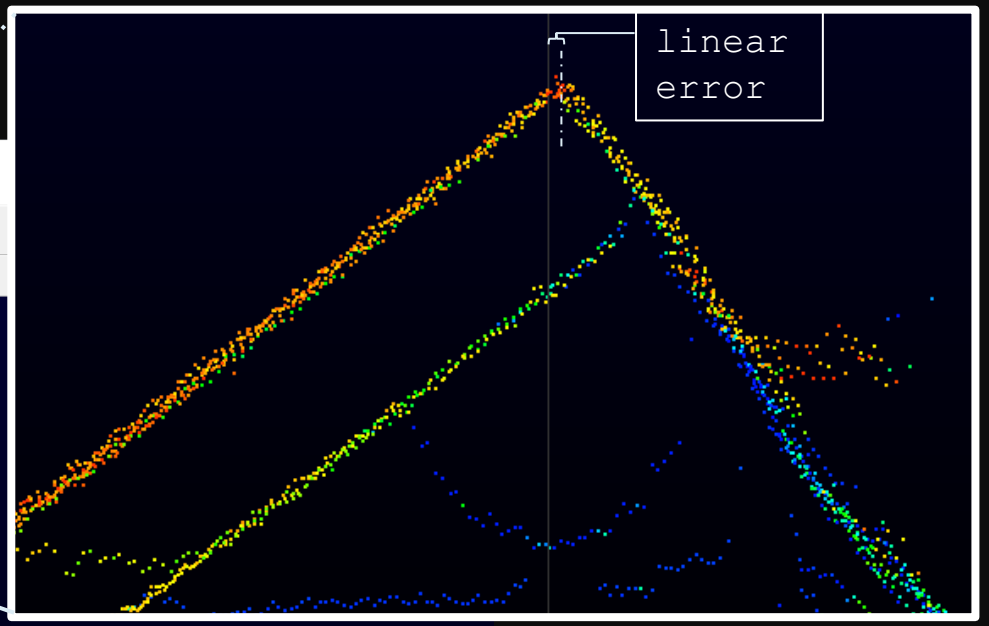
Concrete Spray Printing - Localization

- **Calibration Procedure:**
  1. Fire up Veloview and get to the point where you've got a live point-cloud from your LIDAR.
  2. Configure your view so you're looking top-down (-Z, I believe) with an orthographic rather than isometric projection (the box icon top left).
  3. Position the bore-sighter in the hole as above, with the "up" mark up, and aim the whole assembly at a far-off, clean corner in your test room. (The farther away the better, since a small angular misalignment will show up as a larger linear displacement the farther away you are).
     Approach the corner to verify the laser dot is dead-on the crease, as accurately as you can get it. It might help to have an assistant if your corner is far away or if the lighting is tough.
  4. In Veloview, the corner you're pointed at should be due-north in the window, on x=0. If you zoom in on that point, you may find that your two walls don't converge on that line; that there is an observable angular offset.
  5. Note down the angular displacement. If your corner is far enough away, you can use the linear measurement tools to calculate the linear displacement of your corner from the x=0 line,
  6. Using trigonometric computation, determine the angle and as a redundancy check, verify by changing the yaw.
  7. Click the "Sensor Stream" button (the icon looks like a VLP-32) and check "Advanced Configuration." Here, you can adjust the yaw value of the laser's position to determine how far off your internal calibration is.
  8. Increase the positive or negative value until your corner perfectly lines up on the x=0 gridline.

Concrete Spray Printing - Localization

- **Veloview Data:**





Concrete Spray Printing - Localization

- **Results**

Distance (m)	Azimuth Correction (deg)	Distance Error (cm)
2.5 m	…	…
5.0 m	…	…
7.5 m	…	…
10.0 m	0.95°	1.65 cm
12.5 m	…	…
15.0 m	…	…

- **Conclusion**
  - All the errors above were measured at a rotational speed of 600RPM. In the interest of being thorough, perhaps testing the sensor at 1200rpm (the VLP-16's max rotational speed) would be beneficial.

2022.10.10

- Summary of your work:
  - Calibration Test conducted in MIL 19
  - Recorded measured errors in VLP-16 puck for varying distances of 5-15 meters at 600rpm.
  - Lab presentation

- Next steps:
  - Review sensor fusion methodology (proposed ML-based auto-calibration for LiDAR and Realsense sensors)
  - Work with UGV and Activity Tracker team for sensor fusion

Concrete Spray Printing - Localization

Internal Calibration (Correcting Yaw Error)

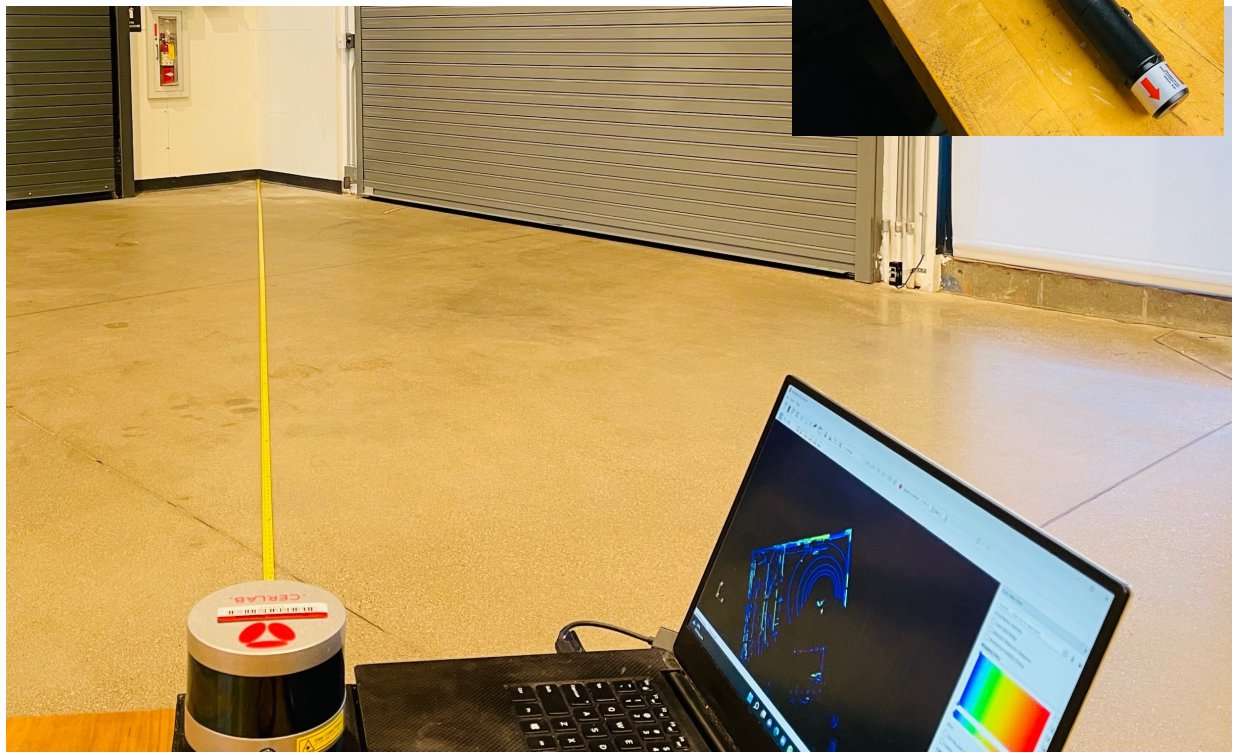
The primary objective is to orient the VLP-16 sensor towards a distinct feature to identify discrepancies arising from innate yaw angle errors.

Test Setup :

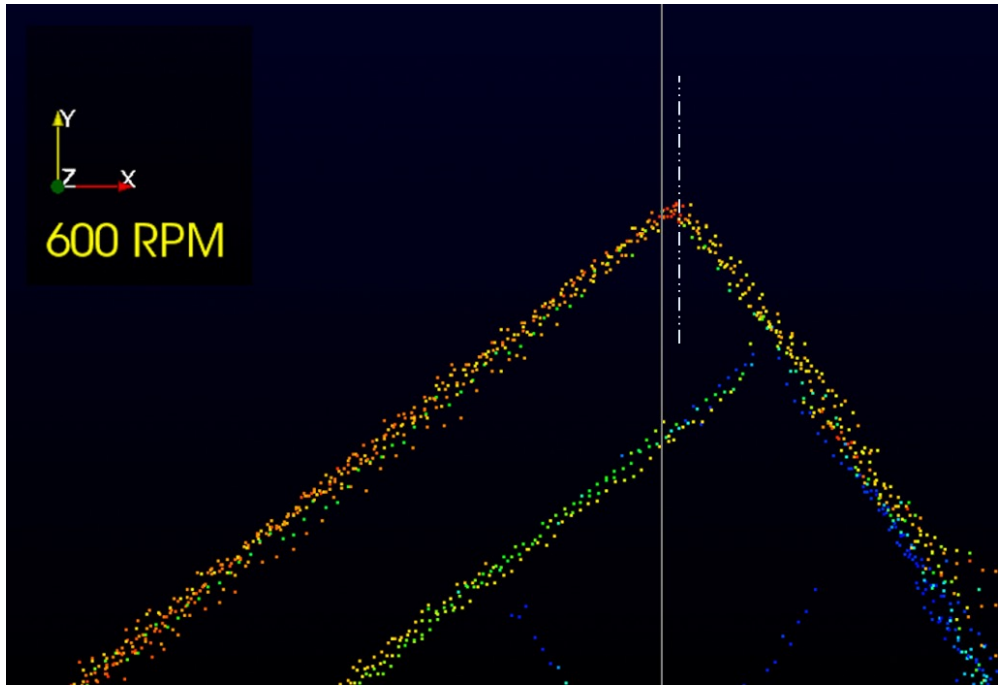
- Feature for recording measurements in Veloview -

  Edged surfaces with minimal features, and a distinct edge-edge contour : 2 flat, straight walls intersecting at a vertical corner serves as an ideal feature to align the point cloud

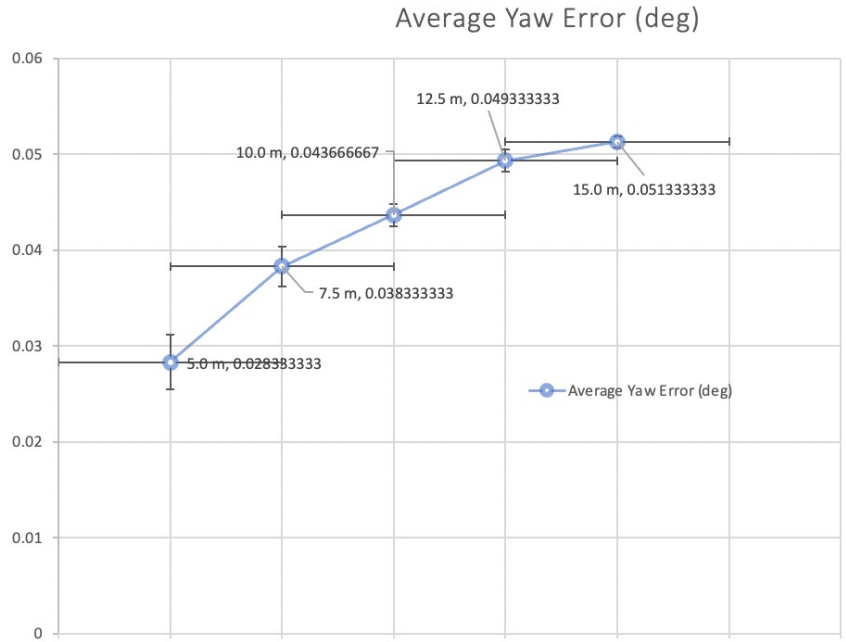
- Precisely demarcated measurement points of 5m, 10m and 15m to record test results



Concrete Spray Printing - Localization



Distance (m)	Azimuth Correction (deg)			Average Distance Error (cm)
5.0 m	0.030°	0.030°	0.025°	<0.5 cm
7.5 m	0.039°	0.04°	0.036°	0.50 cm
10.0 m	0.045°	0.043°	0.043°	0.76 cm
12.5 m	0.050°	0.050°	0.048°	1.07 cm
15.0 m	0.052°	0.051°	0.051°	1.34 cm



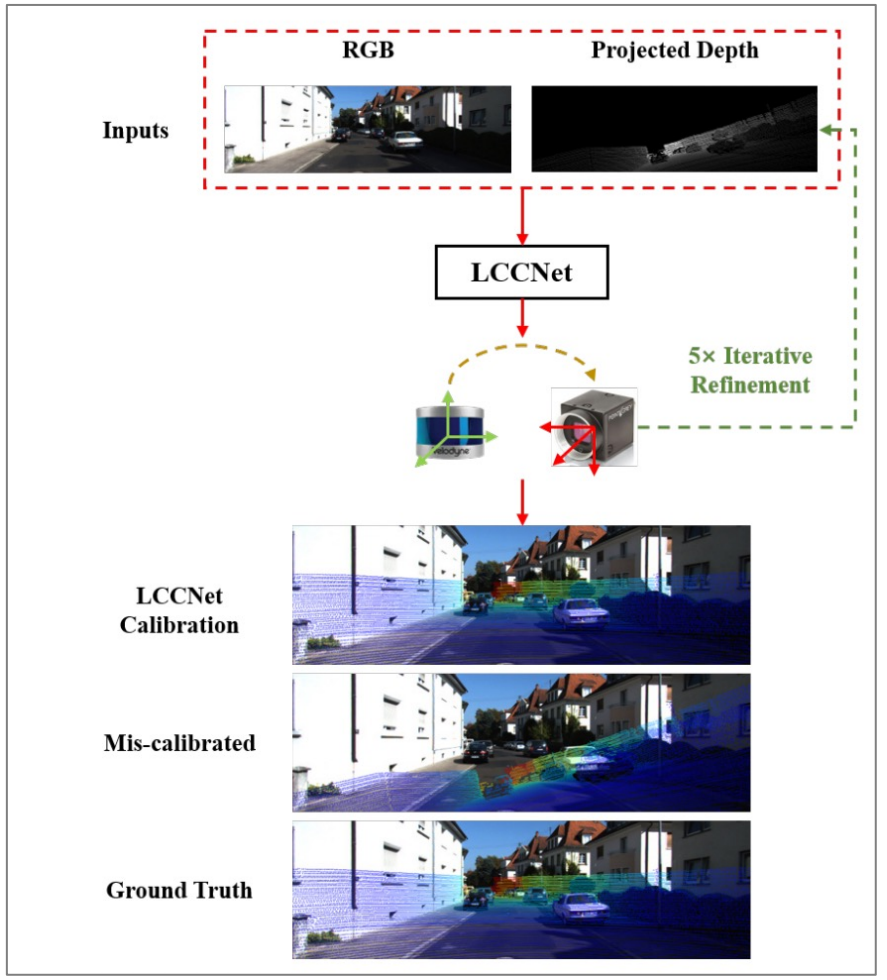
Note: All errors measured at rotational speed of 600RPM.

Perhaps testing the sensor at 1200rpm (max rotational speed) would be beneficial

Concrete Spray Printing - Localization

- **Sensor Fusion:**

  - Multiple sensors (VLP + realsense camera) for AMR
  - Camera-LiDAR extrinsic auto-calibration using Learning
  - Convolutional neural networks trained to map sparse 3D point-cloud data and dense 2D image features to compute 6DoF transformation matrix
  - Feature extraction -> feature matching -> feature aggregation
  - Test on live data or available public datasets



From: *Xudong, Lv & Wang, Boya & Dou, Ziwen & Ye, Dong & Wang, Shuo. (2021). LCCNet: LiDAR and Camera Self-Calibration using Cost Volume Network. 2888-2895. 10.1109/CVPRW53098.2021.00324*

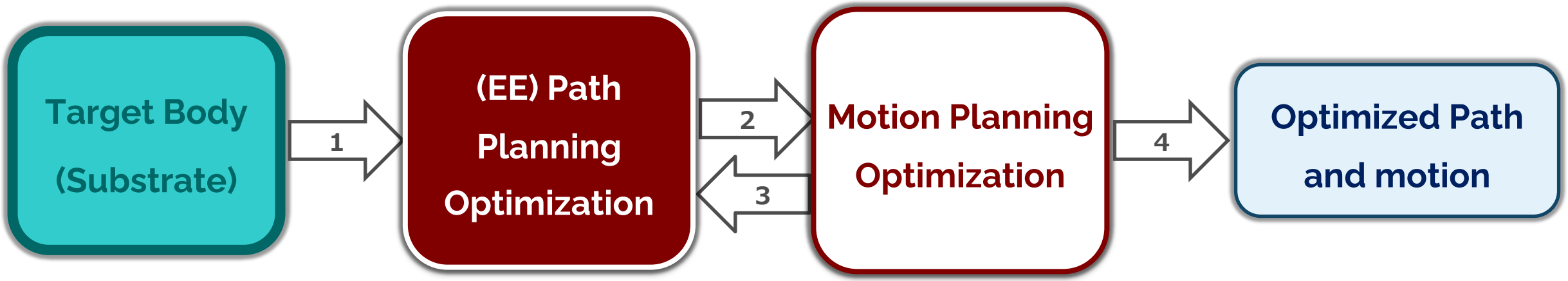
2022.10.24

- Summary of your work:
  - Reviewed Path/Motion Planning task for 3D Concrete Printing project.
  - Review Motion Planning and Path Planning Objective functions and Minimization criteria
- Next steps:
  - Literature Review input for path planner for Bubble Mesh and Surfel based Path Planning methodologies.
  - Review segmentation of 3D model and implementation consideration for 3D bubble mesh structure
  - Work on AMCL localization repository adaptation for Velodyne VLP-16 simulation

Shotcrete Path and Motion Planning Optimization Framework

Planning + Optimization Process

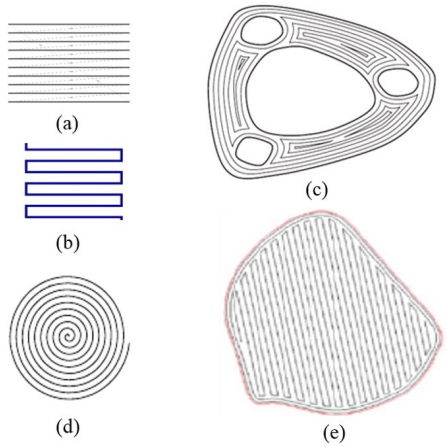
1) Acquire target body position + orientation
2) End-Effector trajectory planning and optimization
3) Motion optimization based on EE path planning loop closure



Shotcrete End-Effector Path Planning

End-effector path planning criteria:

- Optimization Criteria
  - Minimize Distance travelled by the end-effector
  - Minimize orientation changes
  - Maintain constant End-effector velocity
- Constraints:
  - Area Coverage
  - Layer thickness
  - EE perpendicular to surface
- Knows(out of scope):
  - Targeted travel velocity of end-effector
  - End-effector distance from substrate
  - Desired Thickness



The key challenge, is to understand the target configuration and printing processes to build optimized traversability maps of the end-effector for shotcrete printing!

Shotcrete End-Effector Path Planning

Objective Function to optimize end-effector path planning:

$$Min \quad f(\boldsymbol{P},\boldsymbol{q}) = \underbrace{\sum_{i=1}^{n} |\boldsymbol{P}_i|}_{\text{Distance Travelled}} + \underbrace{\sum_{i=1}^{n} |\boldsymbol{q}_i|}_{\text{End-effector orientation}} + \underbrace{\sum_{i=1}^{n} \left|\frac{\boldsymbol{P}_i}{t_i^2}\right|}_{\text{Acceleration (constant velocity/layer thickness)}}$$

$$\boldsymbol{S.T.:} \quad \boldsymbol{x} * \boldsymbol{y} = S_A \quad \text{..... (Surface Area Coverage)}$$

$$h - a \leq L \leq h + a \quad \text{..... (Layer thickness } \pm a\text{)}$$

$$(\boldsymbol{q_i} \cdot \boldsymbol{q}_{s_i}) = 0 \quad \text{..... End-effector perpendicularity to surface}$$

Where:

- n = the number of discretized points travelled
- $\boldsymbol{q}$ = the orientation of the end-effector at each position $\mathbf{P} = (x, y, z)$ in quaternions.
- L = the layer thickness, function of end-effector velocity. $L \propto \frac{1}{\left|\frac{(x,y,z)_i}{t_i}\right|}$

Shotcrete Motion Planning

Motion planning defines the robotic system motion for the End-Effector(EE) to adhere to the path planning result.

- Optimization Criteria
  - Minimize manipulator joint velocities
    - Avoid singularities (since these occur in high velocity scenarios)
    - Increases smoothness of manipulator motion
  - Minimize Base movements (applicable to Gantry and wheeled base)

- Constraints:
  - End-Effector position and orientation
  - Desired End-Effector velocity

Shotcrete Motion Planning

Objective Function to optimize end-effector motion planning:

$$Min \quad f(\boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{n} \left| \frac{\boldsymbol{\theta}_i}{t_i} \right|}_{\text{Joint velocities}} + \underbrace{\sum_{i=1}^{n} |(\theta_{B1}, \theta_{B2})_i|}_{\text{Base motions (Left, Right Wheel) or (X,Y Gantry axes)}}$$

$$\boldsymbol{S.T.:} \quad F(\boldsymbol{\theta}) = \boldsymbol{P} \ldots\ldots\ldots\ldots\ldots\ldots \quad \text{Forward Kinematics adhere to EE Path \& Orientation}$$

$$\frac{F(\boldsymbol{\theta})}{\mathrm{t}} = \mathrm{V} \quad \text{End-Effector velocity is constant}$$

Where:

- n = the number of discretized points travelled
- $\boldsymbol{\theta}$ = vector of manipulator and mobile base joints

2022.10.31

- Summary of your work:
  - Discussed Literature with Kyshalee for Planning in Additive Manufacturing and Concrete Spray Printing
  - Explored 3D segmentation methods for target body representation and planning.
- Next steps:
  - Review Bubble Mesh advantages and drawbacks, include feedback from Prof. Shimada and Tomotake from lab presentation
  - Find alternate 3D target representation/segmentation

Shotcrete End-Effector Path Planning

Survey on Artificial Intelligence for Additive Manufacturing (2017)

TABLE 1. SUMMARY OF AI METHODS FOR 3D PRINTING

3D Printing Process	Status	Application	Method	Problems &Future Work
Printability Checking (Design & Preparation)	Offline	Original printability checker	PC scheme (FE, PM, VE)	1. Multi-indicator test and optimal effect proportion using GA. 2. Lower complexity threshold.
		Automatic checking	ML (SVM)	
Prefabrication (Planning)	Offline	Slicing acceleration	Slicing Algorithm (TRI, TS, LE) GPU (PPS and FPS parallelism)	1. Evaluation of time consumption closed to a practical situation. 2. Consolidation of print segments and threshold control. 3. Parallel computing of printability checking, slicing and path planning.
		Path Optimiser	TSP based optimisation Christofides algorithm	
Service Platform & evaluation (Design & Printing &Service& Control)	Online	Cloud Service Platform	Demand matching algorithm Resource allocation algorithm	1. Security enhancement. 2. Real-time control. 3. Design for printing.
		Evaluation Model	ML (Multi-criteria fuzzy decision based on Hamming Distance algorithm; GA)	
Security (Control)	Online	Attack Detection	kNN, anomaly detection, random forest	Defect detection for mass customisation.

- Similar to **Travelling Salesman Problem** (TSP), Fok et al. proposed a relaxation scheme of 3DP path optimizer to compute the nozzle traversing time.
- The path optimisation problem is formulated as TSP by comparing each print segment to a city and finding the fastest or shortest tour to visit the whole country. In each layer, TSP implements optimisation at inter-partition level (print area) and intra-partition level (blank area) based on the boundaries of these areas.

Shotcrete End-Effector Path Planning

Survey on Artificial Intelligence for Additive Manufacturing (cotd.)

Between each interpartition, TSP can compute the visit order and start the tour to visit each area. Then the **Christofides** algorithm is used to find the shortest time of this visit. The transition segment in intra-partition level is the connection of two nearest end points located in two adjacent print areas

Christofides Algorithm:

- Let G = (V,w) be an instance of the travelling salesman problem. (V of vertices, w assigns a nonnegative real weight to every edge of G.
- Create a minimum spanning tree T of G. Let O be the set of vertices with odd degree in T.
- Find a minimum-weight perfect matching M in the induced subgraph given by the vertices from O.
- Combine the edges of M and T to form a connected multigraph H in which each vertex has even degree.
- Form a Eulerian circuit in H.
- Make the circuit found in previous step into a Hamiltonian circuit by skipping repeated vertices (shortcutting).
- The steps 5 and 6 do not necessarily yield only one result. As such the heuristic can give several different paths.

Shotcrete End-Effector Path Planning - Literature review

Evaluation of the Infill Algorithm for Trajectory Planning of Pointed Ends for Droplet-Generating 3D Printers (2014)

- This paper presents an evaluation of an algorithm that optimizes the infills of the geometries, a droplet generating 3D printer that extrudes plastic droplets on a moving platform

- The algorithm suggests the optimal positioning of the droplets such that the faults in the sharp corners are minimized

- Depending on the angle, additional droplets ought to be inserted in the angular bisector so long until the overall error decreases. In addition, a certain number of droplets positioned in the very sharp angle that overlap for more than 50% ought to be removed.

- The parts printed according to the repetitive contours infill with additional insertions and deletions according to the look-up table resulted in minimal oscillations

Dynamic programming approach to adaptive slicing for optimization under a global volumetric error constraint (2018)

- Proposed an adaptive slicing algorithm that balances accuracy and print time. The proposed, near-optimal, dynamic programming (DP) based algorithm for adaptive slicing minimizes the number of layers subject to a global volumetric error constraint.

- Goal is to minimize cusp height.

- Essentially the slicing adapts to the model to minimize error with original model.

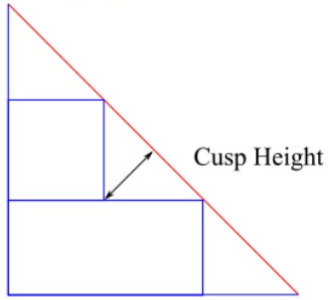


Figure 1: An illustration of cusp height error

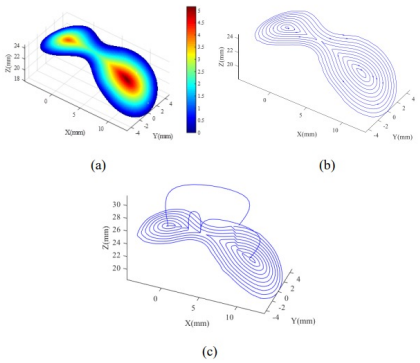
Process Planning in Multi-Axis Machining and Printing of Complicated Parts (2020)

- The model is decomposed into curved layers by interpolating a number of iso-geodesic distance surfaces (IGDSs).

- A layer can only be printed if its lower IGDS(s) have already been printed. (collision avoidance)

- A layer can only be printed if the potential collision surfaces of the unprinted layer do not include this layer

- Printing path planning: by using Crane's heat method on the triangular mesh of the layer, the iso-geodesic distance contours can be computed

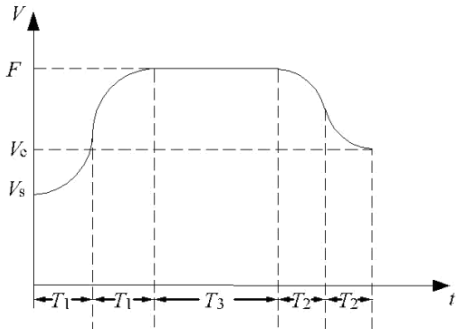


& they are then connected as the filling path

Shotcrete End-Effector Path Planning – Literature review

Application of the Five-phase S-curve Velocity Model on FDM (Fused Deposition Modeling) Three-dimensional Printer (2020)

- Introducing a modified acceleration strategy to prevent sudden acceleration jump is believed to improve quality of models. Conventionally described by trapezoidal model to control motor speed.
- The 5-phase S-curve model divides the acceleration and decelerations curves into 5 intervals, generating a simpler piece-wise function for motion planning.
- "Look Ahead" algorithm is applied to make planning between steps efficient.



$$a(t)=\begin{cases} Jt, & 0\le t<T_m \\ 2JT_1-Jt, & T_1\le t<2T_1 \\ 0, & 2T_1\le t<2T_1+T_3 \\ 2JT_1-J(t-T_3), & 2T_1+T_3\le t<2T_1+T_3+T_2 \\ -2JT_1-2JT_2-JT_3+Jt, & 2T_1+T_3+T_2\le t<2T_1+T_3+2T_2 \end{cases} \quad (1)$$

$$V(t)=\begin{cases} V_s+\frac{1}{2}Jt^2, & 0\le t<T_1 \\ V_s-JT_1^2+2JT_1t-\frac{1}{2}Jt^2, & T_1\le t<2T_1 \\ V_s+JT_1^2, & 2T_1\le t<2T_1+T_3 \\ V_s+JT_1^2-\frac{1}{2}J(t-(2T_1+T_3))^2, & 2T_1+T_3\le t<2T_1+T_3+T_2 \\ V_s+J(T_1^2-T_2^2)+\frac{1}{2}J(t-(2T_1+T_3+2T_2))^2, & 2T_1+T_3+T_2\le t\le 2T_1+T_3+2T_2 \end{cases} \quad (2)$$

3DCP: Path Planning Literature Review Summary

Current Literature focuses on how to best pre-process the shape and planning best path for printing.

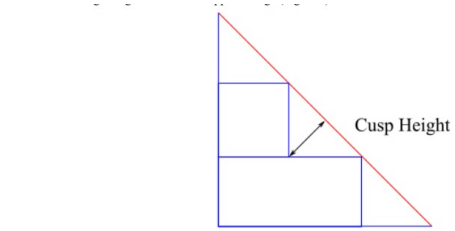


Figure 1: An illustration of cusp height error

- Preprocessing:
  - Slicing adaptive to model shape by minimizing cusp height
  - Optimizing build direction orientation for minimal amount of voids
  - Curved layers are defined by interpolating a desired number of iso-geodesic distance
- Planning algorithms:
  - Travelling Salesman Problem + Christofides Algorithm (not optimal, minimizes at least <50%)
  - Common 2D tool paths are projected into 3D view for equidistant toolpath.
  - Optimize printhead manipulability, orientation, and distance along a central printing column.
- Motion Planning:
  - Change from Trapezoidal to 5-phase S-curve acceleration model in motion planning

2022.11.7

- Summary of your work:
  - Literature reviewed for Bubble Mesh, Surfel, IsoGeodesic segmentation methods.
  - Reviewed Bubble Mesh advantages and drawbacks, include feedback from Prof. Shimada and Tomotake from lab presentation

- Next steps:
  - Review Learning based methods of Path Planning:
    - Particle Swarm Optimization, Ant Colony algorithm, Genetic Algorithms, etc.
  - Thesis preparation and presentation – Topic: Exploring Heuristic Learning Based Path Planning in Shotcrete Printing

Shotcrete End-Effector Path Planning

Target Body substrate input format:

Point cloud:

- A point cloud of the target body would detail many (x,y,z) points on the surface,
- Point-cloud should be processed for continuous smooth surface (de-noise)

Tessellation (STL of CAD):

- Fine meshing would allow for curvature to be represented accurately
- Vertices would be used as input for path travelling

Square Mesh Partition of Surface:

- Could be adaptable to the shotcrete deposit size
- Smaller computational cost

Surfel:

- Zero-dimensional *n*-tuple with shape and shade attributes that locally approximate an object surface.

3DCP Path planning input

Surfel:

- Surfels are ‘surface elements’ or ‘surface voxels’ in the volume rendering literature.
- Others describe it as a Objects can be represented by a dense set of points (surfels) holding lighting information.
- Common applications of Surfels are:
  - medical scanner data representations
  - rendering surfaces of volumetric data
  - real-time rendering of particle systems
  - Surfels are also used to improve the efficiency of GI (Global Illumination) as Tomasz Stachowiak from EA SEED describes.

Surfel is another data structure to represent surfaces, which can be used as input for path planning solution.
https://teamwisp.github.io/research/surfels.html

Bubble mesh method review:

Brief summary of bubble mesh approach:

A. Initial bubble placement:

- Bubbles are placed on segments, recursively subdivided based on bubble diameter.
- $\frac{d_1}{2}+\frac{d_2}{2}$, where $d\big(C(s)\big), C(s)$ is a curve segment

B. Interbubble Forces:

- Repulsive/Attractive forces are applied based on stability distance.
- $l_0 = \frac{d(x_i,y_i,z_i)}{2} + \frac{d(x_j,y_j,z_j)}{2}$

C. Physically based relaxation:

- Correct and improve bubble placement with bubble motion governed in parametric space of the curve by:
- $f_i(t) = m_i \frac{d^2 \boldsymbol{x_i}(t)}{dt^2} + c_i \frac{d\boldsymbol{x_i}}{dt}$,
- Where: $m\frac{d^2\,x(t)}{dt^2} + c\,\frac{dx}{dt} + kx(t) = 0, \quad k_0 = f'(l_0), \quad \zeta = \frac{c}{2\sqrt{mk}} \approx 0.7$

Bubble mesh method review:

Brief summary of bubble mesh approach (Cont.):

D. Adaptive Population Control:

- Deletes or add bubbles as needed, defined by the overlapping ratio.
- $\alpha_i = \frac{2}{d(\boldsymbol{x}_i)} \sum_{j=0}^{n} (d(\boldsymbol{x}_i) + \frac{d(\boldsymbol{x}_j)}{2} - \overline{\boldsymbol{x}_i \boldsymbol{x}_j})$

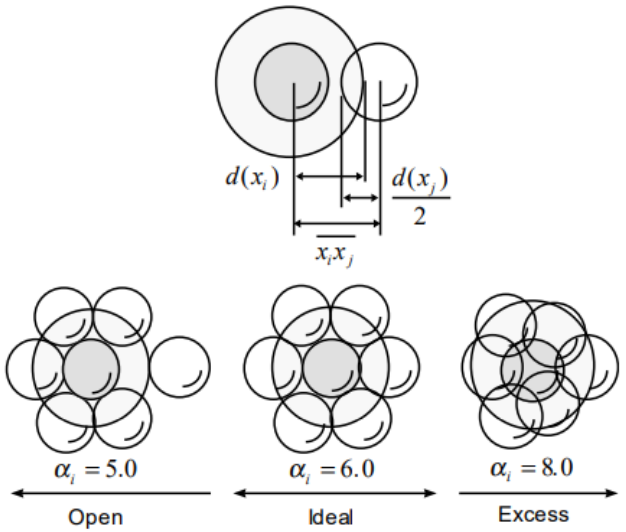
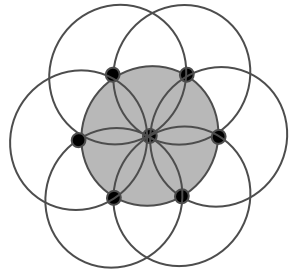


Figure 6: Overlapping ratio

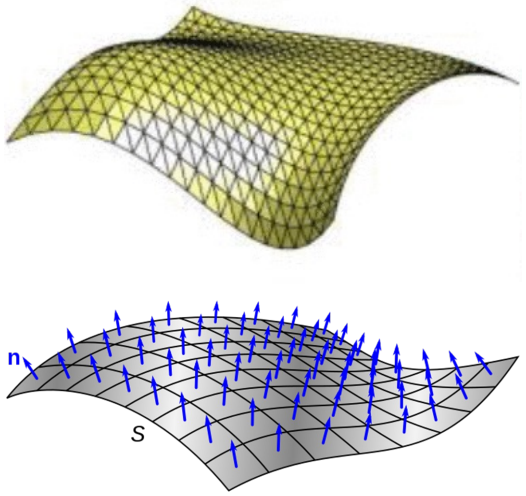


Ideal overlap for even spray layering

3DCP: Path Planning - Tesselation

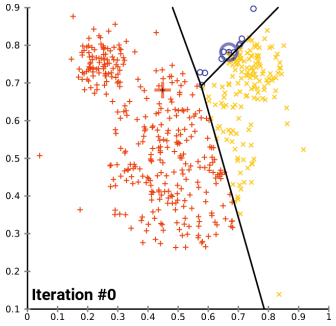
Mesh is to be post-processed to partition based on shotcrete spray cone size and partitioned into discrete locations to be sprayed over.

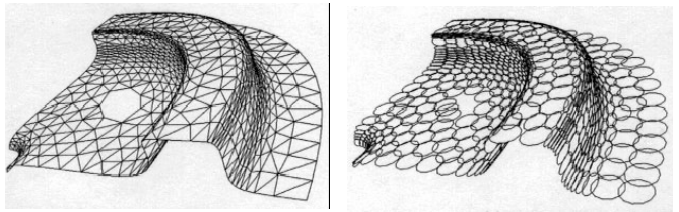
Input:
CAD -> STL Mesh

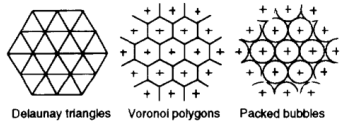


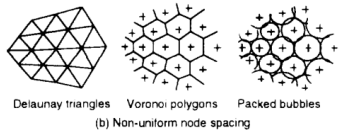
Post-processing of Mesh:

K-means Or Bubble Mesh

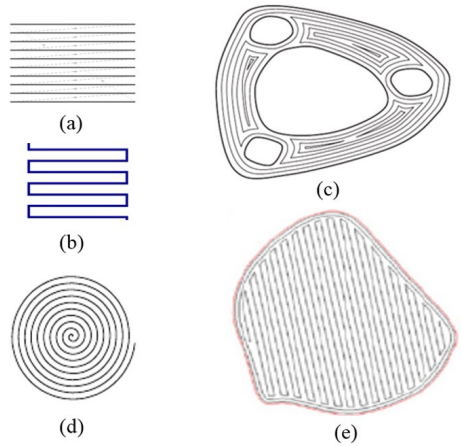








Path Planning



All vertices in STL need to be covered & size of partition based on concrete spray cone (adjustable to any desired cone size).

Overlap between partitions
- Necessary for coverage and smooth path
- 50% based on spray painting

Each partition will have a normal
- $p_n = average(t_n)$
- To be used as a criteria for path planning

2022.11.14

- Summary of your work:
  - Explored heuristic learning-based alternate path planning algorithms for 3DCP project and thesis.
  - Path Planning Literature Review :
    - Optimized combination of spray painting trajectory on 3D entities (2019)
- Next steps:
  - Path Planning Literature Review :
    - Toolpath generation for freeform surface models
    - Adaptive Isocurve-Based Rendering for Freeform Surfaces

3DCP: Path Planning

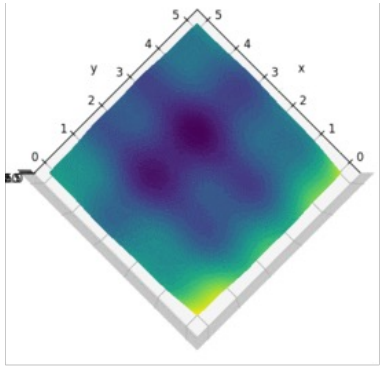
Three main steps will compose the path planning optimization where 4 optimization algorithms can be applied that are compatible with a discrete(discontinuous) domain

1. $P([\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]_i) \leftarrow Minimized(STL)$
2. $Alg \leftarrow Objective\ Function(\text{Dist.}, \text{EE orn}, \text{EE acc.})$
3. $return\ Optimized\ Path$

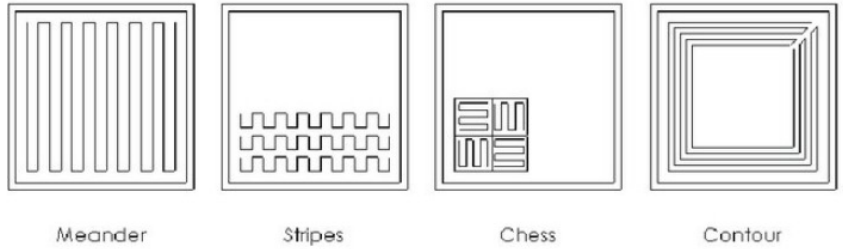
- Christofides
- Common coverage path
- Genetic Algorithm
- Particle Swarm optimization

Optimization algorithm needs to handle discontinuous function, such as the forward kinematics:

- Stochastic Optimization Algorithms:
  - Simulated Annealing
  - Genetic Algorithm
  - Swarm Algorithms
- Machine Learning:
  - Reinforcement Learning
  - Neural Network



Swarm Algorithms



Common coverage paths

3DCP: Path Planning - PSO

Particle Swarm Optimization:

- Research on PSO were mostly on how to determine the hyperparameters w, c1, and c2 or varying their values as the algorithm progressed. For example, there are proposals making the inertia weight linear decreasing. There are also proposals trying to make the cognitive coefficient c1 decreasing while the social coefficient c2 increasing to bring more exploration at the beginning and more exploitation at the end.

- This is a **heuristic solution** because we can never prove the real **global optimal** solution can be found and it is usually not. However, often the solution found by PSO is quite close to global optima.

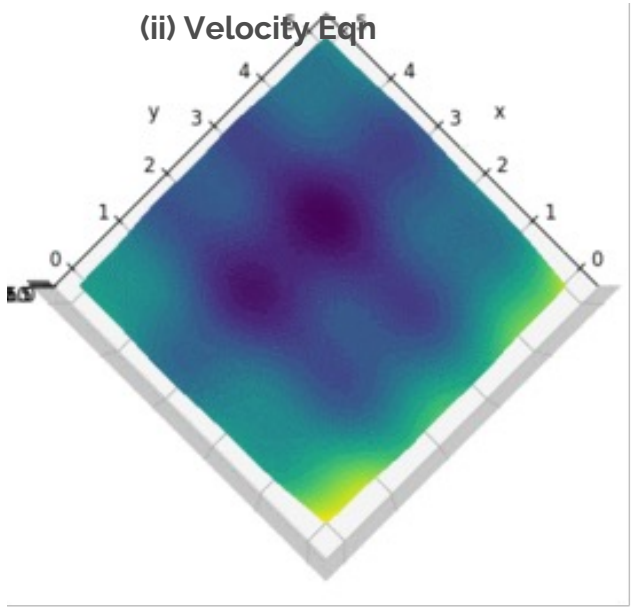
Assume we have P particles and we denote the position of particle i at iteration t as $X^i(t)$, which in the example of above, we have it as a coordinate $X^i(t) = (x^i(t), y^i(t))$. Besides the position, we also have a velocity for each particle, denoted as $V^i(t) = (v_x^i(t), v_y^i(t))$. At the next iteration, the position of each particle would be updated as

$$X^i(t+1) = X^i(t) + V^i(t+1)$$

(i) Position Eqn

$$V^i(t+1) = wV^i(t) + c_1 r_1 (pbest^i - X^i(t)) + c_2 r_2 (gbest - X^i(t))$$

(ii) Velocity Eqn



3DCP: Path Planning – Genetic Algorithms

Genetic Algorithms:

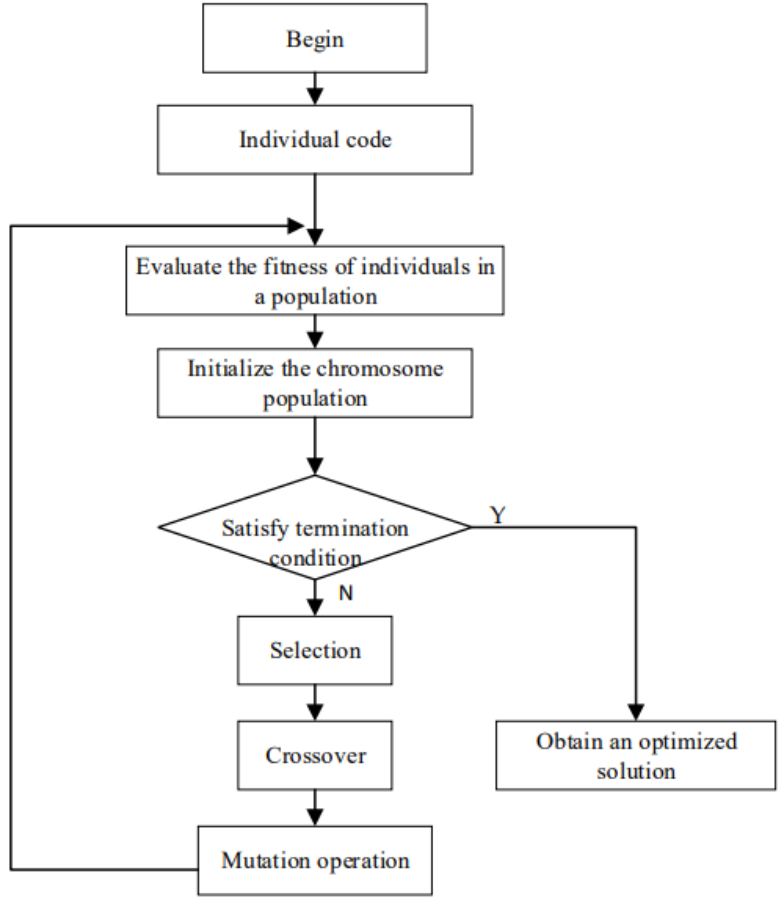
- Genetic algorithms are a method of searching for the best solution by emulating the natural evolution process. These algorithms are particularly effective in solving non-polynomial problems in combinatorial optimization. Here, genetic algorithms require specialized individual coding and genetic manipulation techniques, such as crossover and mutation.

Individual Code: The length of the individual code is |VH|. Each vertex in the Hamilton graph represents one edge of the original graph.The individual code contains the binary code, Psi, representing the direction of each edge in the original graph.

Fitness function: The fitness function is used to determine which individuals can enter the next round of evolution and which individuals need removing from the population.

Crossover: Crossover is the process of exchanging the partial codes between two individuals with a certain probability to generate new individuals. Here, order crossover (OX) is used on Pi while two-point crossover is used on Psi. OX ensures that the original order of each vertex is almost the same when the effective sequence of the individual itinerary is modified

Mutation operation: Pi is subjected to inversion mutation to generate a new individual. A basic variation is applied to Psi, where one or more loci are randomly selected for individual code and the gene values of these loci are inverted



3DCP: Path Planning –Ant Colony Optimization

Ant Colony Optimization:

Ant colony optimization (ACO) is a probabilistic algorithm that is inspired by the behavior of ants searching for food. It is used to find the optimal path and has strong resistance to interference and good compatibility. In the algorithm, individual information such as unvisited vertices (NVV), unvisited edges (NVE), visited edges (VE), and tour length (TL) are initialized. The algorithm uses memory and communication between ants to improve the search process and find the optimal solution.

After time, Δt, the pheromone on trajectory (i, j) i adjusted as follows:

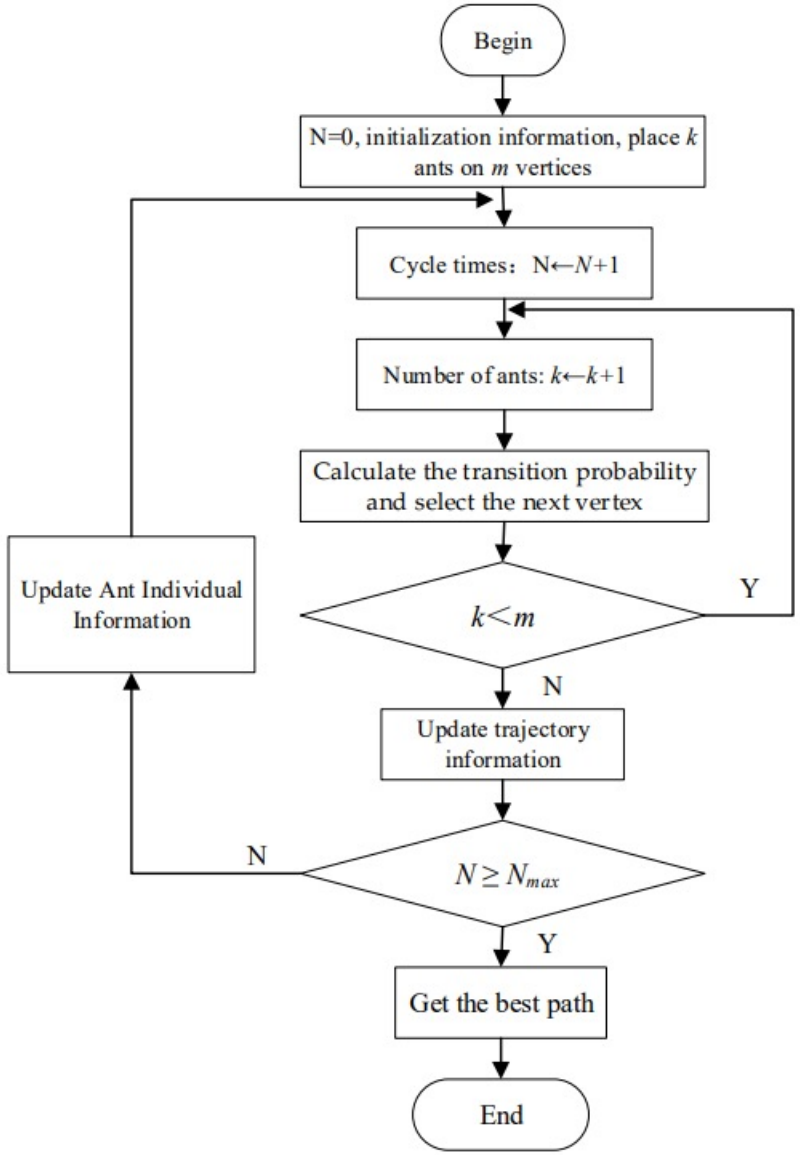
$$\tau_{ij}(t+\Delta t) = \rho\tau_{ij}(t) + \Delta\tau_{ij}$$

'ρ' represents the volatilization rate of pheromone, 'τij(t)' represents the accumulation amount of pheromone on the track (i, j) at time t, 'Δτij' represents the increment of the pheromone on the trajectory (i, j) after the time, Δt, which can be calculated as follows:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad \Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{TL[k]}, & ant\ k\ pass\ path\ (i,\ j) \\ 0, & else \end{cases}$$

Δτ k ij denotes the pheromone on the trajectory (i, j) during the searching process of the k-th ant, among which, Q is a constant

$$p_{ij}^{k} = \begin{cases} \frac{(\tau_{ij}(t))^{\alpha}(\eta_{ij}(t))^{\beta}}{\sum_{s\in alowed_k}(\tau_{is}(t))^{\alpha}(\eta_{is}(t))^{\beta}}, & j \in alowed_k \\ 0, & otherwise \end{cases}$$



2022.11.28

- Summary of your work:
  - Path Planning Literature Review :
    - Toolpath generation for freeform surface models
    - Adaptive Isocurve-Based Rendering for Freeform Surfaces
    - Efficient Tool-Path Planning for Machining Free-Form Surfaces.

- Next steps:
  - Path Planning Literature Review :
    - Real-time Optimal Trajectory Planning for Robotic Manipulators with Functional Redundancy
    - Approach summary for Trajectory Optimization of Electrostatic Spray Painting Robots on Curved Surface
    - Path Planning for Spray Painting Robot of Horns Surfaces in Ship Manufacturing
    - Planning Method of Offset Spray Path for Patch considering Boundary Factors

Title: Adaptive Isocurve-Based Rendering for Freeform Surfaces

In the ensuing discussion we need the concept of *valid coverage*.

Definition 1.1. A set of isocurves $\mathcal{C}$ of a given surface S is called a *valid coverage* with respect to some constant δ if for any point p on S there is a point q on one of the isocurves in $\mathcal{C}$, such that $\|p - q\|_2 < \delta$, where $\|\cdot\|_2$ denotes Euclidean distance.

Surface rendering algorithms using isocurves should comply with Definition 1.1 where δ is approximately half the image pixel size. All pixels representing S in the image are then guaranteed to be covered by at least one isocurve. In the ensuing discussion S is assumed to be represented in the viewing space. A surface in viewing space has its x and y coordinates aligned with the image plane coordinates i_x and i_y; that is, $i_x = x$ and $i_y = y$. However, the z coordinate of the surface is still accessible. The viewing space automatically accounts for distant and small surfaces that require less effort to render because the perspective transformation has already been applied. In many cases it is sufficient to compute the isodistance using only the x and y surface components, because coverage of the image plane is the concern. However, ignoring z may result in missed pixels when the surface is partially hidden. We discuss this issue further later.

Definition 1.2. A coverage for a given surface is considered *optimal* if it is valid and the accumulated pixel drawing cost function is minimal over all valid coverages.

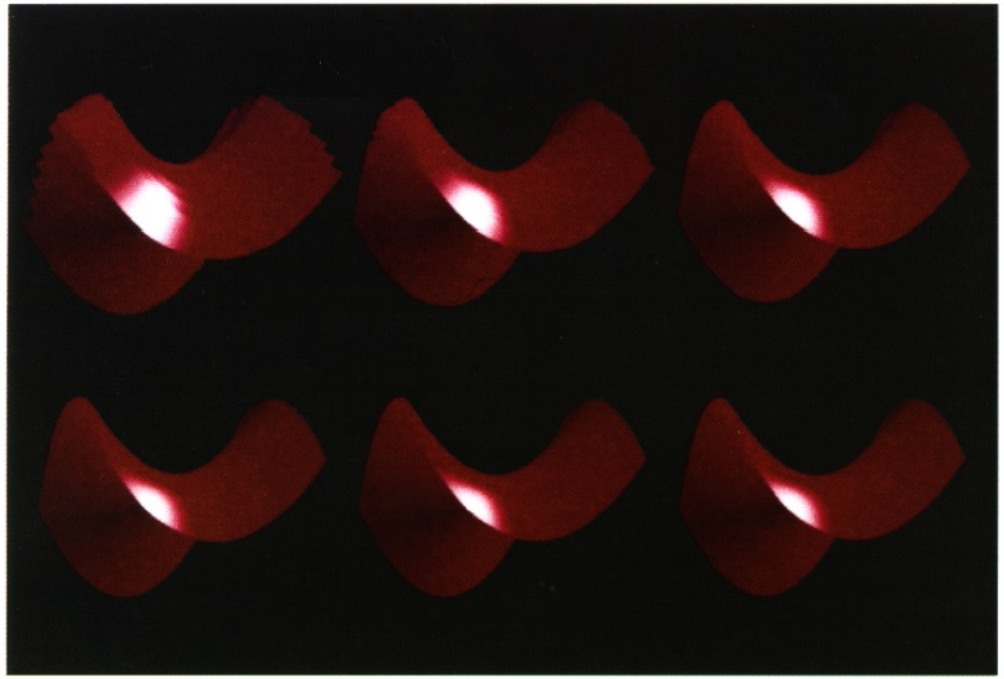


Fig. 11. Six steps in coarse to fine rendering using adaptive isocurves.

One might also consider rendering the surface adaptively using variable width curves. Starting with very few but widely drawn isocurves, one could immediately provide a coarse shape of the surface which could be refined hierarchically into a more accurate image using more isocurves. Figure 11 shows six steps of such a process.

* Varying curve width has been used but only to later refine, not to use multiple width in one rendering

3DCP: Path Planning Literature Review

Title: Adaptive Isocurve-Based Rendering for Freeform Surfaces

There are some subtleties that have not yet been considered. If the surface *V Min* boundary is the same as the *V Max* boundary, the algorithm will find their (zero) isodistance below the distance tolerance δ and quit immediately. A cylinder is one such example in which the *V Min* and *V Max* boundary seams are shared. One should guarantee such cases are detected before invoking Algorithm 1. One way to guarantee the prevention of such cases is to ensure the surface is silhouette free from the rendering direction (see Elber and Cohen [1990] and Heflin and Elber [1993] for silhouette detection). A surface is silhouette free if its normal is never perpendicular to the viewing direction. An alternative may be to use a heuristic that always enforces at least one subdivision of the surface, which solves the problem for surfaces such as cylinders. Another consideration is determination of the parametric direction for isocurves extraction; u isocurves or v isocurves. In our implementation we compute the maximum isodistance between the u and v surface boundaries, and prefer the direction with the smaller maximum. This heuristic promotes fewer, longer isocurves over numerous shorter ones in the hope that it will minimize the number of curves to be drawn.

Other image rendering aspects should be considered as well. The valid coverage is only one necessary condition. The surface normal for each pixel is also required for shading. An unnormalized representation of the surface normal $\hat{n}(u, v) = (\partial S/\partial u) \times (\partial S/\partial v)$, can be computed symbolically [Elber 1992] and represented as a vector surface whose coordinate functions are products and differences of surface partial derivatives.

Each isocurve output from Algorithm 1 is then piped into the curve renderer and is accompanied by the associated isocurve from the normal surface $\hat{n}$. The curve renderer uses the normal curve to compute the normal

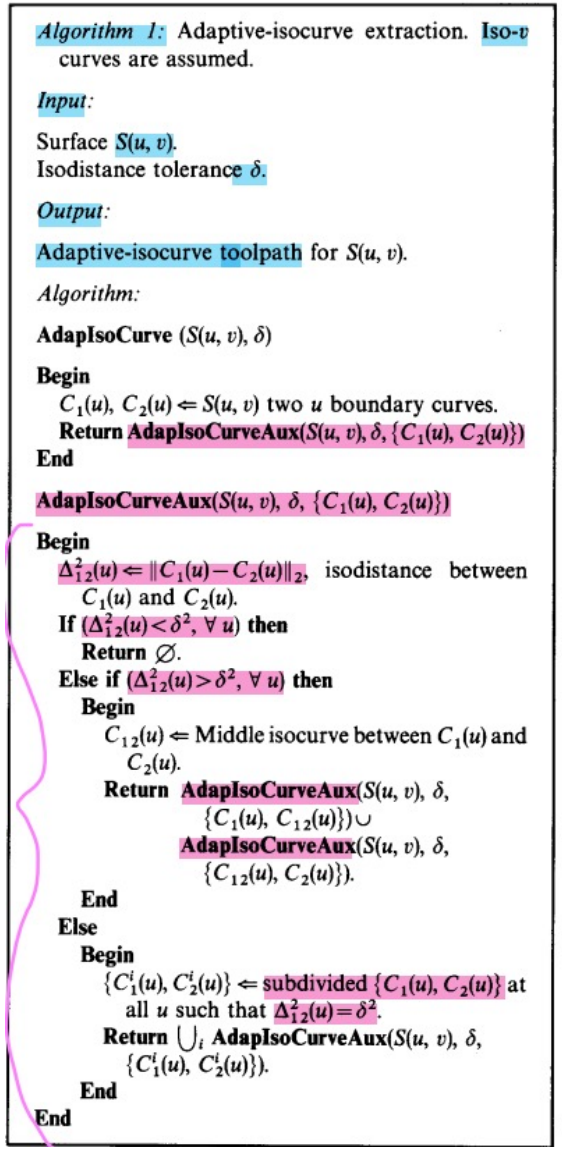
```
Input:
    S(u,v), input surface.
    δ, maximum distance between isocurves.

Output:
    S, the set of constant v isocurves of S(u,v) adjacent within δ, covering .

Algorithm:
    adapIsoCrvs( S, δ )
    begin
        C1(u), C2(u) ⇐ isocurves of S in u direction at VMin, VMax.
        return
            { C1(u) } ∪
            adapIsoCrvsAux( S, δ, u, VMin, VMax, C1(u), C2(u) ) ∪
            { C2(u) }.
        end
    end

    adapIsoCrvsAux( S, δ, VMin, VMax, C1(u), C2(u) )
    begin
        UMax, UMin ⇐ C1(u), C2(u) common u domain.
(1)     Δ²12(u) ⇐ squared iso-distance between C1(u) and C2(u).
        Z ⇐ zero set of (Δ²12(u) − δ²).
        if Z empty then
            R̂ ⇐ S subsurface between C1(u) and C2(u).
            R̂ ⇐ C1(u) * v + C2(u) * (1 − v), v ∈ (0,1).
(2)         if Δ²12((UMax + UMin)/2) < δ² and (R̂ ∼ R) valid then
(3)             return φ.
            else
                VMid ⇐ (VMin + VMax)/2.
                C12(u) ⇐ isocurve of S at VMid from UMin to UMax.
                return
(4)                 adapIsoCrvsAux( S, δ, VMin, VMid, C1(u), C12(u) ) ∪
                    { C12(u) } ∪
(5)                 adapIsoCrvsAux( S, δ, VMid, VMax, C12(u), C2(u) ).
            end
        else
(6)         Subdivide C1(u), C2(u) at all u^i ∈ Z into {C¹1(u), C¹2(u)} pairs.
(7)         return ⋃i adapIsoCrvsAux( S, δ, VMin, VMax, C¹1(u), C¹2(u) ).
        end
    end
```

Fig. 4. Algorithm 1.

3DCP: Path Planning Literature Review

Title: Toolpath generation for freeform surface models

- The model is generated into multiple isocurves.
- Isocurves are evaluated with respect to their relative distances.
- Isocurves are added or deleted depending on distance criteria.



Algorithm 1: Adaptive-isocurve extraction. Iso-v curves are assumed.

Input:

Surface $S(u, v)$.
Isodistance tolerance δ.

Output:

Adaptive-isocurve toolpath for $S(u, v)$.

Algorithm:

AdapIsoCurve $(S(u, v), \delta)$

Begin
  $C_1(u)$, $C_2(u) \Leftarrow S(u, v)$ two u boundary curves.
  Return AdapIsoCurveAux$(S(u, v), \delta, \{C_1(u), C_2(u)\})$
End

AdapIsoCurveAux$(S(u, v), \delta, \{C_1(u), C_2(u)\})$

Begin
  $\Delta^2_{12}(u) \Leftarrow \|C_1(u) - C_2(u)\|_2$, isodistance between $C_1(u)$ and $C_2(u)$.
  If $(\Delta^2_{12}(u) < \delta^2, \forall u)$ **then**
    Return $\varnothing$.
  Else if $(\Delta^2_{12}(u) > \delta^2, \forall u)$ **then**
    Begin
      $C_{12}(u) \Leftarrow$ Middle isocurve between $C_1(u)$ and $C_2(u)$.
      Return AdapIsoCurveAux$(S(u, v), \delta, \{C_1(u), C_{12}(u)\}) \cup$
        AdapIsoCurveAux$(S(u, v), \delta, \{C_{12}(u), C_2(u)\})$.
    End
  Else
    Begin
      $\{C^i_1(u), C^i_2(u)\} \Leftarrow$ subdivided $\{C_1(u), C_2(u)\}$ at all u such that $\Delta^2_{12}(u) = \delta^2$.
      Return $\bigcup_i$ **AdapIsoCurveAux**$(S(u, v), \delta, \{C^i_1(u), C^i_2(u)\})$.
    End
End

3DCP: Path Planning Literature Review

Title: Efficient Tool-Path Planning for Machining Free-Form Surfaces.

4.2 The Procedure of Finding the Efficient Tool-Path. A summary of the efficient tool-path determination procedure is given below.

(1) Choose either one of the surface parameters (say v) as the tool-path direction. Thus the first isoparametric curve at $u = 0$ is the first CC path.
(2) Offset the CC path to the CL path by Eq. (21).
(3) Curve-fit the CL path, which is the first tool path.
(4) Compute the CC path intervals in the direction of the tool-path normal by increasing at a small interval, (Δv) along the first CC path (Fig. 11). The equation for calculating the accurate CC path interval is given in Section 2.
(5) Convert the CC path interval into the u-domain by Eqs. (13) and (20), also store the u-values in an array corresponding to the v-values.
(6) The profile of these u and v values is the next efficient CC path.
(7) Offset this CC path to a CL path.
(8) Curve-fit the CL path, which is the next efficient tool path.
(9) The searching for the next path stops when the accumulated u's reach the end point, $u = 1$ (see Section 4.1).

Following above procedure, a set of tool-paths for the entire surface that does not violate the h constraint can be obtained.

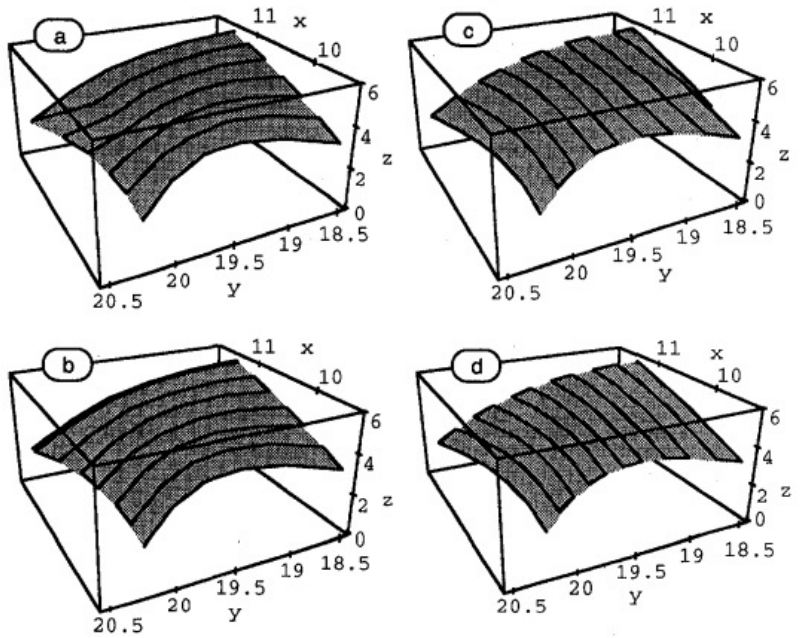


Fig. 16 (*a*) Efficient tool-paths in *v*-direction (*b*) isoparametric tool-path in *v*-direction (*c*) efficient tool-paths in *u*-direction (*d*) isoparametric tool-path in *u*-direction

- The example demonstrated that the cut-length efficiency of the proposed method improves 39 percent compared with the conventional isoparametric method.
- The efficiency depends highly on the local geometric properties, such as the curvature and the parameterization, of the part surfaces.

2022.12.05

- Summary of your work:
  - Path Planning Literature Review :
    - Real-time Optimal Trajectory Planning for Robotic Manipulators with Functional Redundancy
    - Approach summary for Trajectory Optimization of Electrostatic Spray Painting Robots on Curved Surface
    - Path Planning for Spray Painting Robot of Horns Surfaces in Ship Manufacturing
    - Planning Method of Offset Spray Path for Patch considering Boundary Factors
- Next steps:
  - Thesis : Algorithm implementation and testing for basic target bodies
  - Finish setup of VLP-16 localization package.

3DCP: Path Planning Literature Review

Title: Real-time Optimal Trajectory Planning for Robotic Manipulators with Functional Redundancy

- This paper casts constraints on the orientation error into logarithmic barrier functions.
- The search direction of each time step is found, and the problem is solved very efficiently by the gradient method.
- It is shown that the optimal orientation error can be found very fast and how this can be implemented in real-time path planning to improve performance.

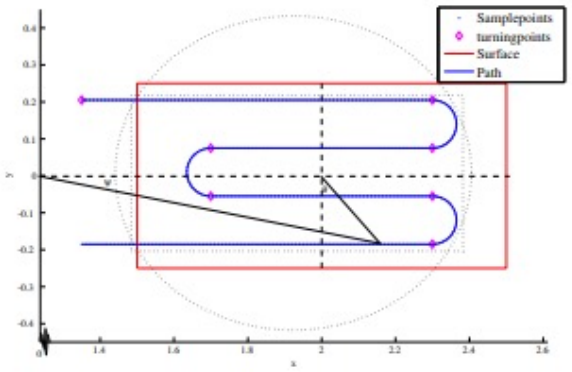


Fig. 1. The path of the tool centre point (TCP) in the xy-plane.

IV. Spray Painting

A. Conflicting Objective Functions

We now show an example where the direction of the z-axis is determined by two cone-shaped sets of orientations. The direction given by the two sets at each time step is in general conflicting and the solution is given by the minimum of a cost function of the sum of the two orientation errors.

Assume a manipulator that is to paint a surface in the xy-plane by following the path in Figure 1. There are two main criteria that will guarantee uniform paint coating, the orientation of the spray gun with respect to the surface and the velocity of the paint gun. The first restriction is ensured by the constraint

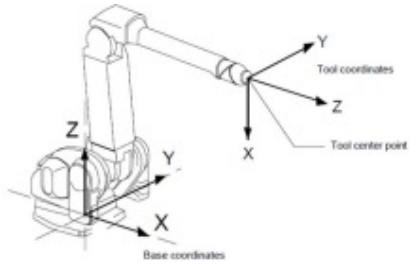


Fig. 2. General structure of a robotic manipulator.

Assume we want to paint the surface in the xy-plane with a constant distance z_{des} between the tool and the surface. Let c be the vector from the centre of the surface, at height z_{des}, denoted p_{cent}, to the current position p_{tcp} on the surface

$$c = p_{tcp} - p_{cent}. \tag{29}$$

This is the direction of the end effector for which the main axes don't need to move at all, i.e. pure rotation of the wrist. This is chosen as the desired direction of the paint gun when the orientation error is not considered.

Let the quaternion describing the desired direction of the z-axis be given by Q_d. Then the set of orientations for which the z-axis points in approximately the direction of the z-axis of Q_d is found by writing

$$Q_p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = Q_d^* * Q = \begin{bmatrix} * \\ -d_1q_0 + d_0q_1 - d_2q_3 + d_3q_2 \\ -d_2q_0 + d_0q_2 - d_3q_1 + d_1q_3 \\ * \end{bmatrix} \tag{30}$$

3DCP: Path Planning Literature Review

Title: Approach summary for Trajectory Optimization of Electrostatic Spray Painting Robots on Curved Surface (2017).

- The approach begins with determining the optimal width between two strokes
- Golden-section method is used to subdivided between 0 and spray radius R
- Then the width is evaluated for optimization.
- Then the surface is divided into triangular mesh
- Large patches are grouped based on normal vector similarity and adjacency

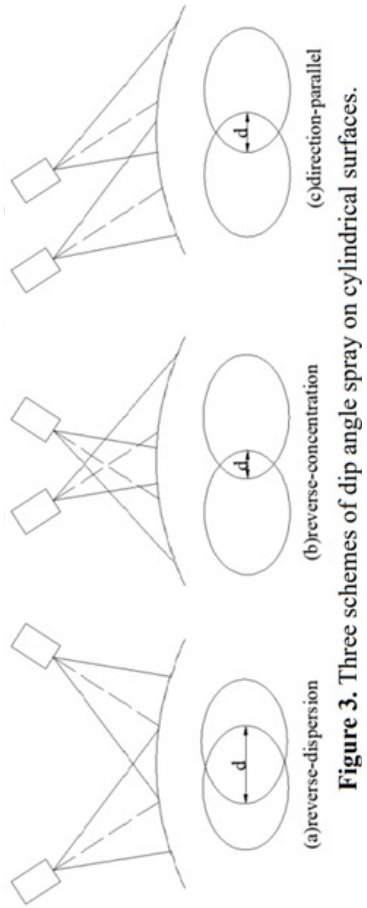
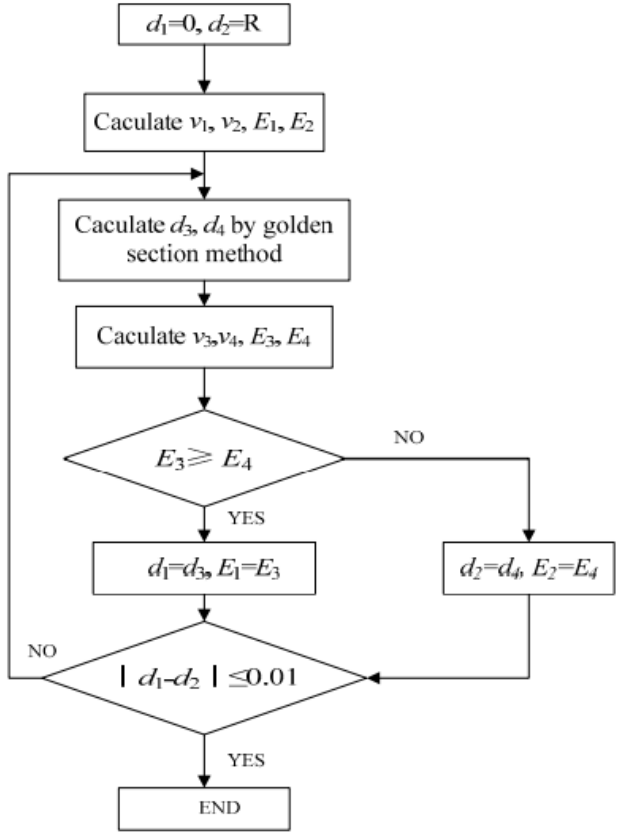


Figure 3. Three schemes of dip angle spray on cylindrical surfaces.

Flow chart for optimal width d



3DCP: Path Planning Literature Review

Title: Approach summary for Trajectory Optimization of Electrostatic Spray Painting Robots on Curved Surface (2017).

many 3D curved surfaces. After connecting the triangular facets into patches, each patch can be processed by the 3L algorithm, resulting in a smooth surface that retains the original properties [17–20].

3) The cuboid model is established on each patch, and the spatial path of the spray-painting robot on each patch is generated. Figure 5a shows the cuboid model established on a patch. The cuboid model is a cuboid that contains exactly the entire patch, which has two main properties: (i) its leading direction is opposite to the direction of the normal vector of the entire patch, and (ii) the area of the rectangle on each facet is as small as possible. In order to generate the spatial path of the spray-painting robot, firstly, a number of tangent planes whose distance are l (l is taken as $R/2$–R, R is the spray radius) are taken along the direction perpendicular to the right side of the cuboid model. Then, the intersection between the tangent plane and several segments of the curved surface can be obtained, and a series of points whose distance is d (the optimum value of the overlapping area's width formed by two spray-painting strokes) are uniformly made on the intersecting line. Finally, these points are connected along the right side of the cuboid mode to generate the spatial path of the spray-painting robot (as shown in Figure 5b).

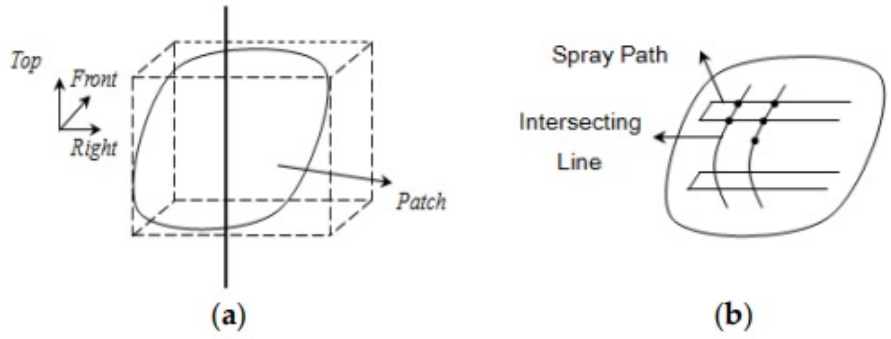


Figure 5. (**a**) The cuboid model; (**b**) The generated spatial path.

After finding the optimal value of the width d of the overlapping area of the two spray painting ›kes on the plane, the following steps are taken to design the spatial path of the spray-painting ›ot by the cuboid model method:

The surface is determined by the CAD (Computer Aided Design) model of the workpiece, and the surface is divided into triangular meshes. After triangulating the surface, we can express it as a mathematical expression:

$$M = \{T_i : i = 1, \ldots, M\} \quad (9)$$

where T_i is the i-th triangle in the triangular facet, and M is the total number of triangular facets in the triangular mesh.

2) Calculate the normal vector of each triangular facet, and generate a number of large patches according to the topology between adjacent triangular facets. Then, suppose that the normal vector of the surface and the normal vector of the projection plane of the surface both have the maximum angle β_{th} (only consider that the normal vectors of the two are on the same side of the surface). After determining β_{th}, each patch of the surface can be generated. The steps of connecting the respective triangular facets into patches are as follows:

(1) Specify any of the triangular facets as the initial triangular facet.
(2) Find all triangular facets that are less than the spray radius from the center point of the initial triangular facet.
(3) Calculate the angle between the normal vector of all triangular facets found in Step (2) and the normal vector of the original triangular facet. If the angle is smaller than β_{th}, connect the triangular facet with the initial triangular facet.
(4) Look for the triangular facets that have not been connected into patches as a new initial triangular facet, and repeat Steps (2) and (3) until all of the triangular facets are connected into patches.

3DCP: Path Planning Literature Review

Title: Path Planning for Spray Painting Robot of Horns Surfaces in Ship Manufacturing (2019)

- Even distribution of painting thickness was used as the optimization parameter for trajectory planning as seen in Equation 6.
- The optimal velocity, overlapping distance and dip angle were considered to minimize the thickness error.
- By examining three different dip angle spray strategies on cylindrical surfaces, it was concluded that reverse-dispersion yields better painting uniformity compared to reverse-concentration and parallel painting

3. Optimal Trajectory Planning for Cylindrical Surfaces

The trajectory of the dip coating on cylindrical surfaces can be divided into three situations: reverse-dispersion, reverse-concentration and direction-parallel, which is shown in figure 3.

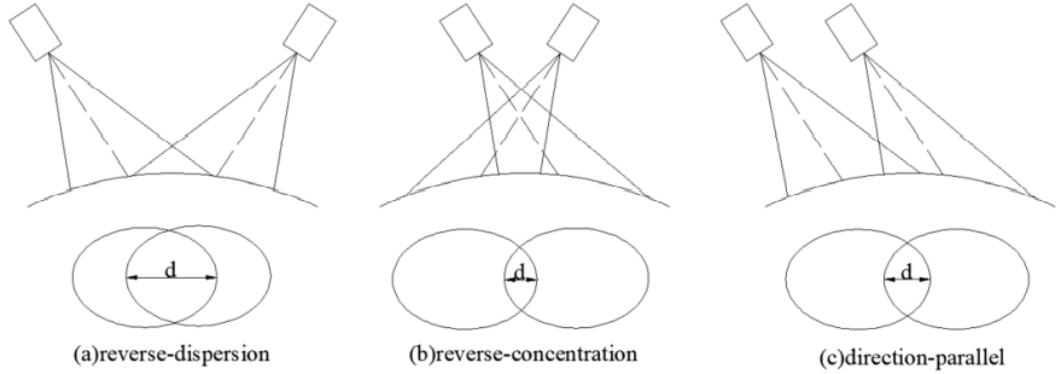


Figure 3. Three schemes of dip angle spray on cylindrical surfaces.

We employ figure 3(a) and 3(b) to calculate the painting thickness on a point s, figure 3(c) can also be calculated similarly. The painting thickness of a given point s can be expressed as follow:

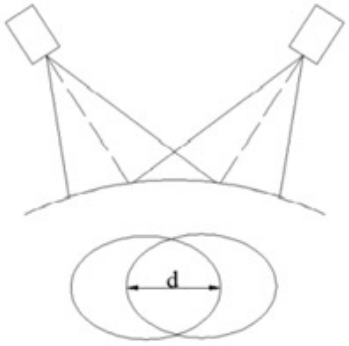
$$T_s(x)=\begin{cases} T_1(x) & 0<x<d-a(\pm\beta) \\ T_1(x)+T_2(x) & d-a(\pm\beta)<x<a(\pm\beta). \\ T_2(x) & a(\pm\beta)<x<d \end{cases} \quad (6)$$

Where a is the length of the long axes of dip angle spray model of the cylindrical surface; x is the position coordinate; T_1 and T_2 are dynamic single stroke coating deposition thickness which is calculated from static coating growth rate model:

3DCP: Path Planning Literature Review

Title: Planning Method of Offset Spray Path for Patch considering Boundary Factors (2018)

- The boundary curve is used to define the spray path.
- Equal division method if plane intercept line is used to generate the spray path in the length direction of the patch.
- A connection algorithm is used to generate the path.



(a)reverse-dispersion

In order to find the optimal velocity v, the overlap distance d and the dip angle of the spray gun β, the mean square error of the thickness deviation from the average thickness T_d must be minimized. Here the coating uniformity is taken as the objective for trajectory optimization:

$$\min E = (T_d - T_{\max})^2 + (T_d - T_{\min})^2. \quad (9)$$

The optimal parameters d, v, α can be obtained by the genetic algorithm.

To solve the problem of paint waste at the boundary of the patch when spraying, a planning method of offset spray path considering boundary of the patch is developed. By analyzing the causes of excessive paint waste at the boundary of the patch, a spray path planning method based on boundary curve of the patch is proposed, and the distance between the spray path and the boundary of the patch is optimized to reduce excessive paint waste. According to the allowable range of the spray height and the error range of the coating thickness, the variable range of the spacing distance is established; on this basis, the equal division method of plane intercept line is used to generate the discrete points of the spray path in the length direction of the patch, and a connection algorithm of the discrete points is used to generate the spray path of the patch. The simulation results show that the method can automatically generate a spray path based on the shape of the patch boundary; under the premise of meeting the uniformity requirements of the coating thickness, paint waste can be effectively reduced during spraying.

Thank you.