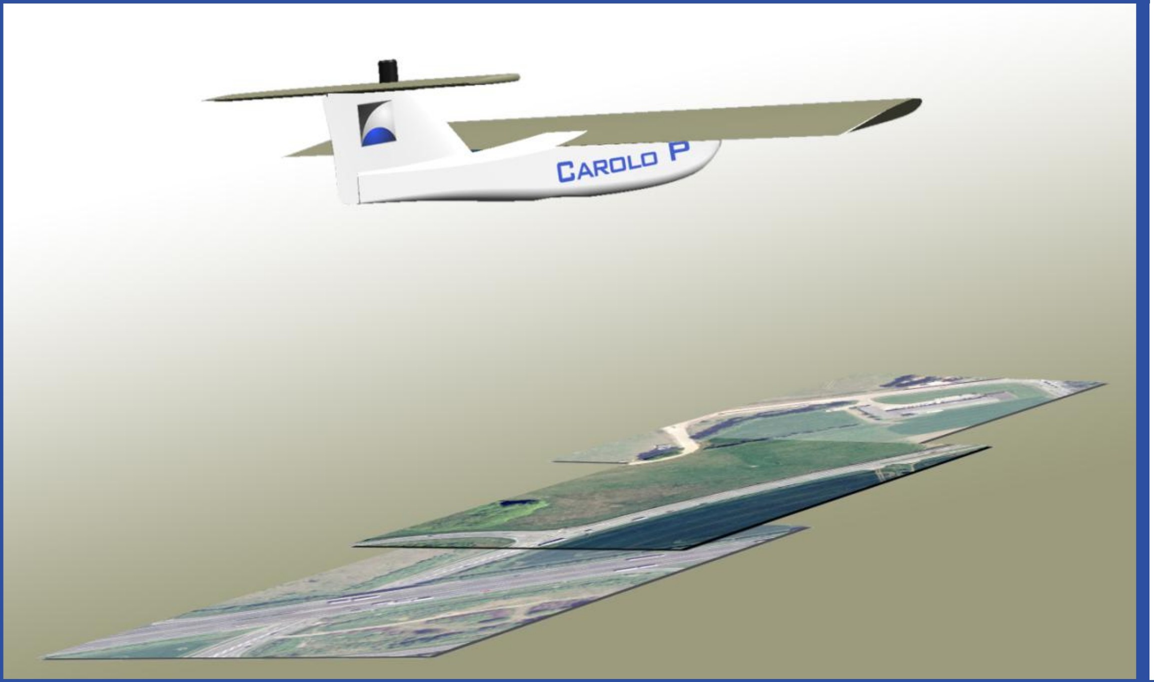
Technische Universität Braunschweig
Institut für
Luft- und Raumfahrtsysteme

Strategies for the Implementation of a Sense and Avoid System for Unmanned Aerial Vehicles



Jang-Bum Park
Institute of Aerospace Systems

Overview

- UAV & Laser Range Finder
- Simulation Model
- Strategies
- Conclusions & Outlook

UAV & Laser Range Finder

CAROLO P200

Span:	2.00 m
Velocity:	55 km/h
Takeoff Weight:	4,000 g
Payload Weight:	1,000 g
Endurance:	> 60 min.

Figure 1. UAV P 200

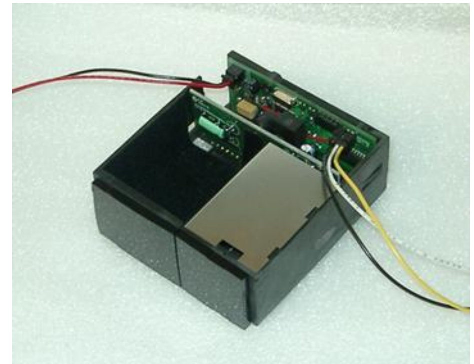


Figure 2. Opti-Logic RS400 [a]

Power	7~9 Volts DC
Protocol	RS232
Laser Wavelength	905 nm +/- 10 nm (IR)
Laser Divergence	5 x 5 cm at 100 m
Accuracy	+/- 1 m
Measuring Range	max. 360 m
Measuring Frequency	10 ~ 200 Hz

Table 1. Specification of the laser range finder RS 400

[a] http://www.opti-logic.com/industrial_rangefinders.htm

Overview

- UAV & Laser Range Finder
- Simulation Model
- Strategies
- Conclusions & Outlook

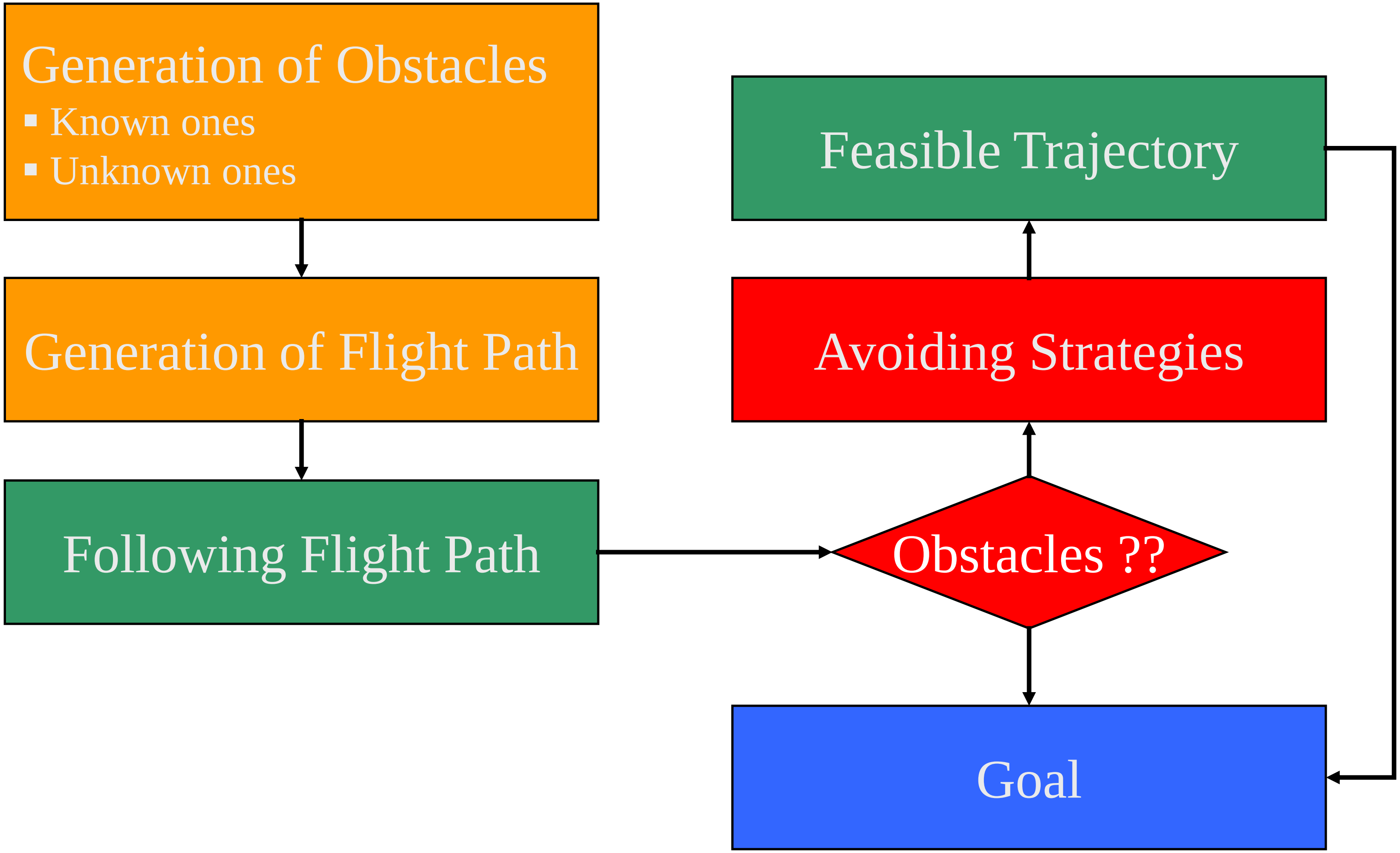


Figure 3. **Structure of the Simulation Model**

- UAV & Laser Range Finder
- Simulation Model
- Strategies
- Conclusions & Outlook

Strategies

- Adapted Flight Path Generation
- Distance Transformation
- A* Algorithm

Adapted Flight Path Generation

➢ **Path Generation by means of Bézier curves**

$$P(v) = (1-v)^3 b_0 + 3(1-v)^2 v b_1 + 3(1-v) v^2 b_2 + v^3 b_3 \ , \ v \in [0,1]$$

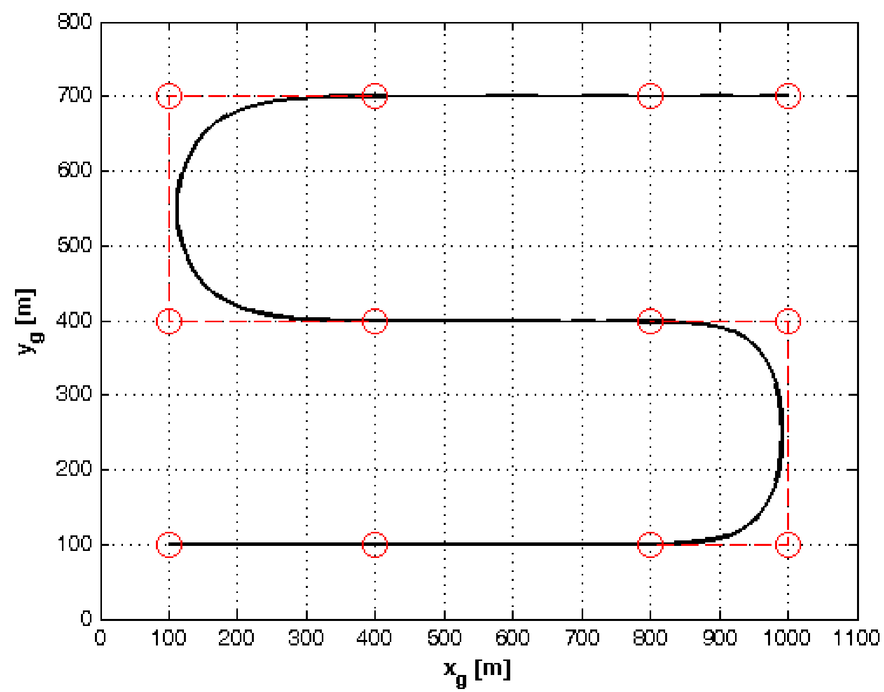


Figure 4. **Exemplary Generation of a Flight Path**

- **Black solid line: Flight Path**
- **Red circles: Predefined Waypoints**

Adapted Flight Path Generation

- Manoeuvre for avoiding obstacles
  - Depending on the detected distance D toward the obstacle

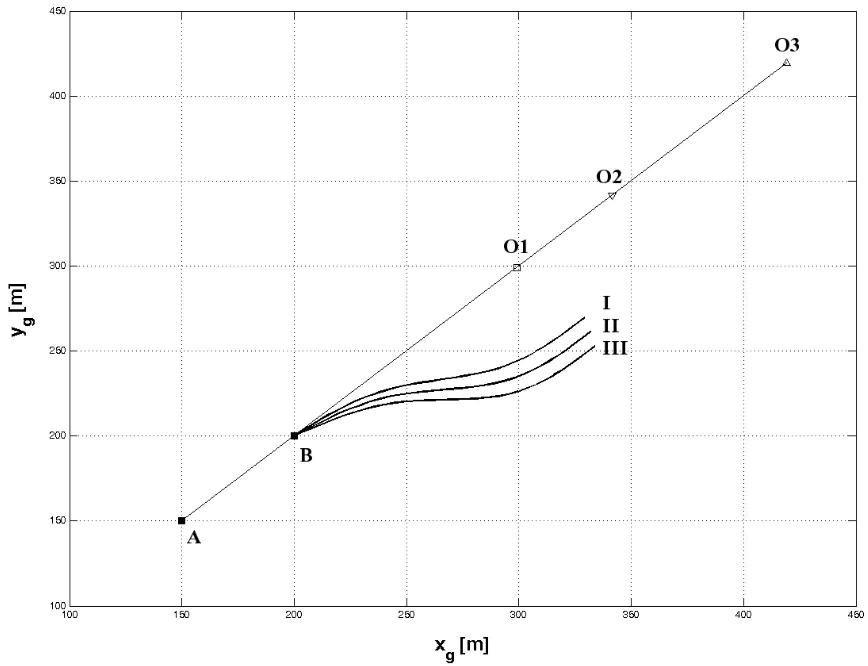


Figure 5. Manoeuvre depending on detected distances

Category	D [m]	dθ [°]
I	300 ≤ D	5
II	150 ≤ D < 300	10
III	D < 150	15

Table 2. Category for various manoeuvres

- Manoeuvre for avoiding obstacles
  - Depending on the curvature κ of the curve

$$\kappa = \frac{\dot{x}\,\ddot{y} - \dot{y}\,\ddot{x}}{\left(\dot{x}^2 + \dot{y}^2\right)^{3/2}}$$

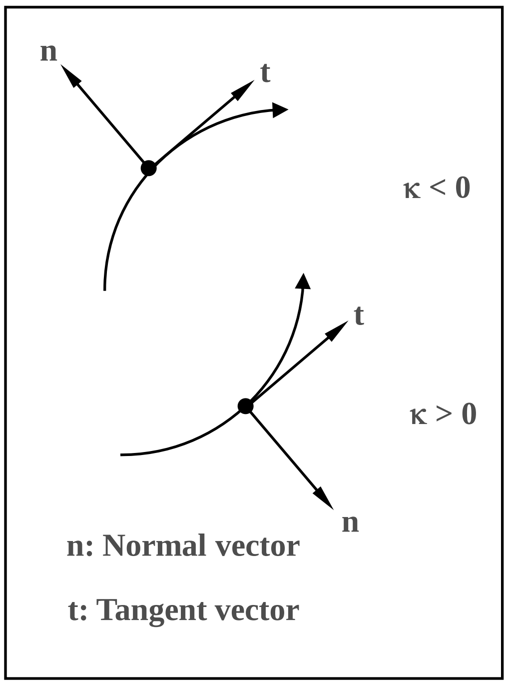
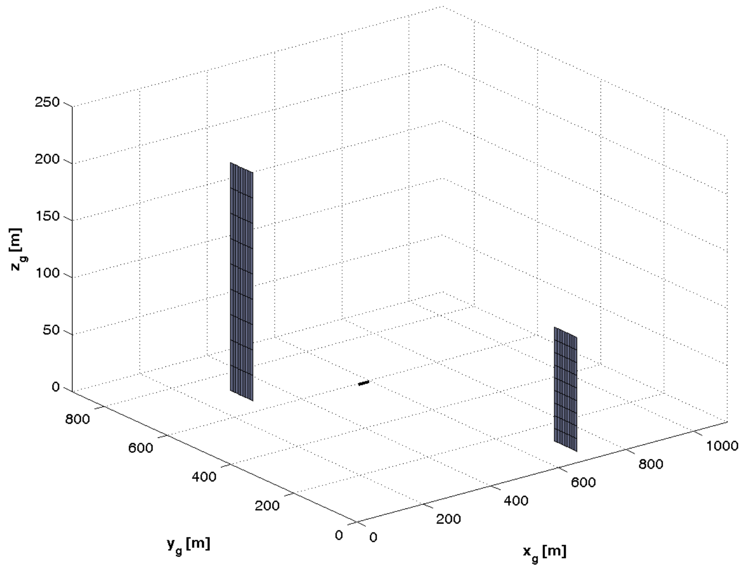


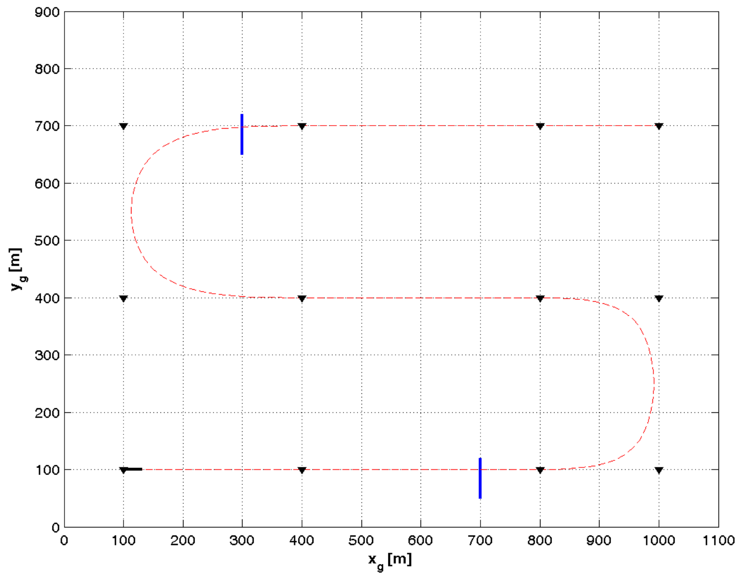
Figure 6. Sign of curvature κ

Case	κ_{Detect}	$\kappa_{Re\text{-}entry}$	Avoidance Curve
1	0	0	
2	0	+	
3	0	–	
4	+	0	
5	+	+	
6	+	–	
7	–	0	
8	–	+	
9	–	–	

Table 3. Different cases of curvatures

Adapted Flight Path Generation





Distance Transformation

- In order to generate an “Occupancy Map”
- Widely used in image processing (medicine, robotics)
- Can be constructed by means of “Metrics” and “Distance Mask”

- Chessboard metric:

$$D_{chess}\left((x_1, y_1), (x_2, y_2)\right) = \max\left(|x_1 - x_2|, |y_1 - y_2|\right)$$

- City-Block metric:

$$D_{city}\left((x_1, y_1), (x_2, y_2)\right) = |x_1 - x_2| + |y_1 - y_2|$$

- Euclidean metric:

$$D_{euclid}\left((x_1, y_1), (x_2, y_2)\right) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

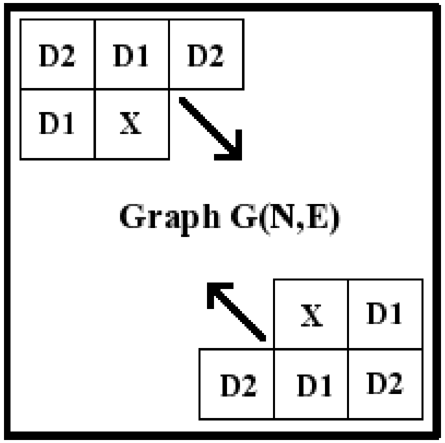
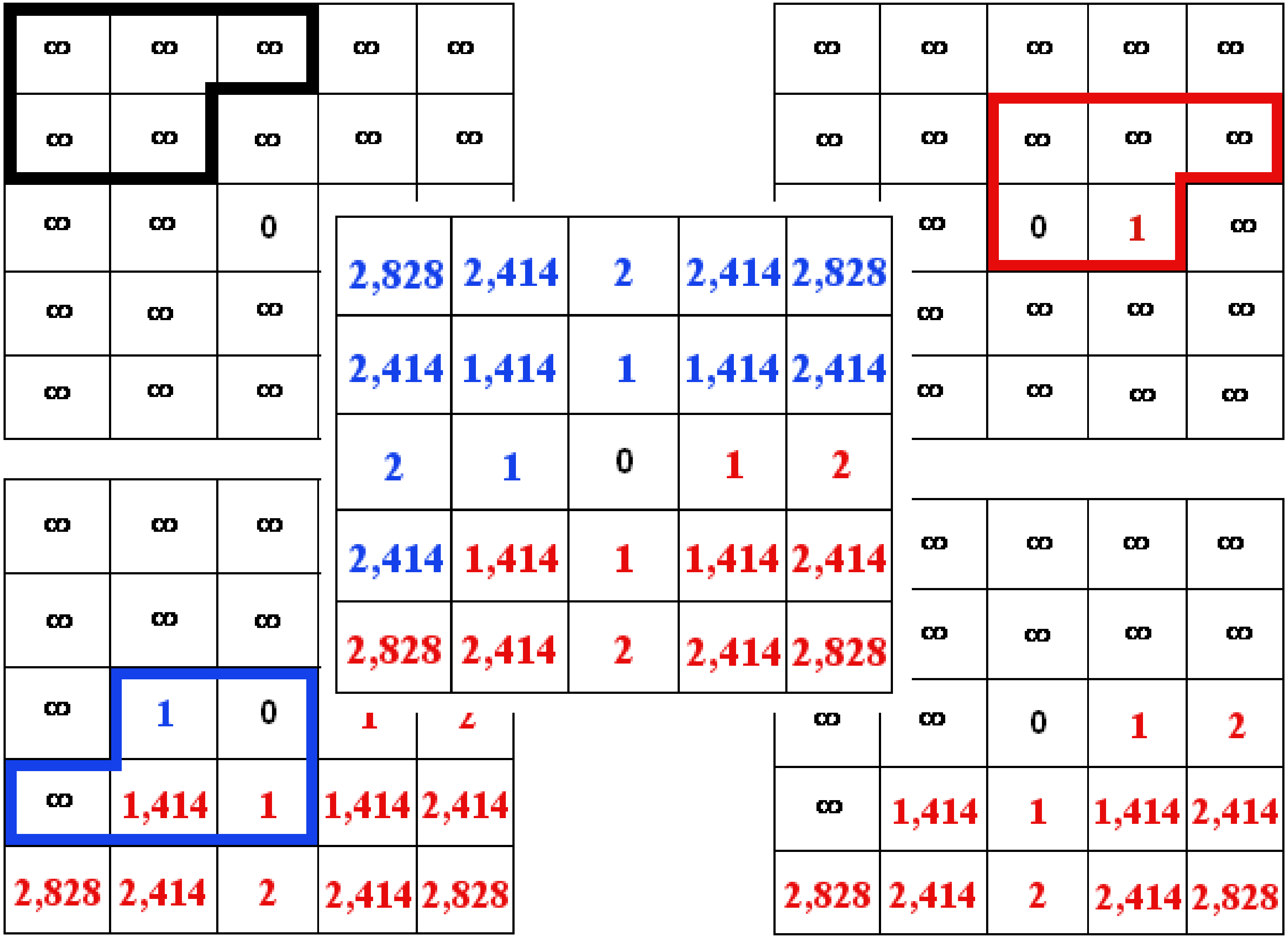
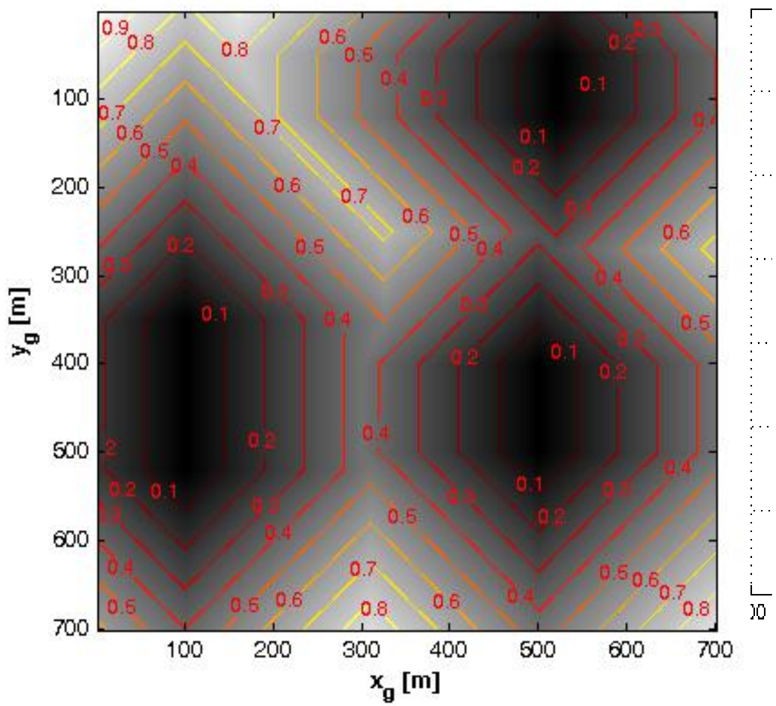


Figure 7. Scheme of Distance Transformation

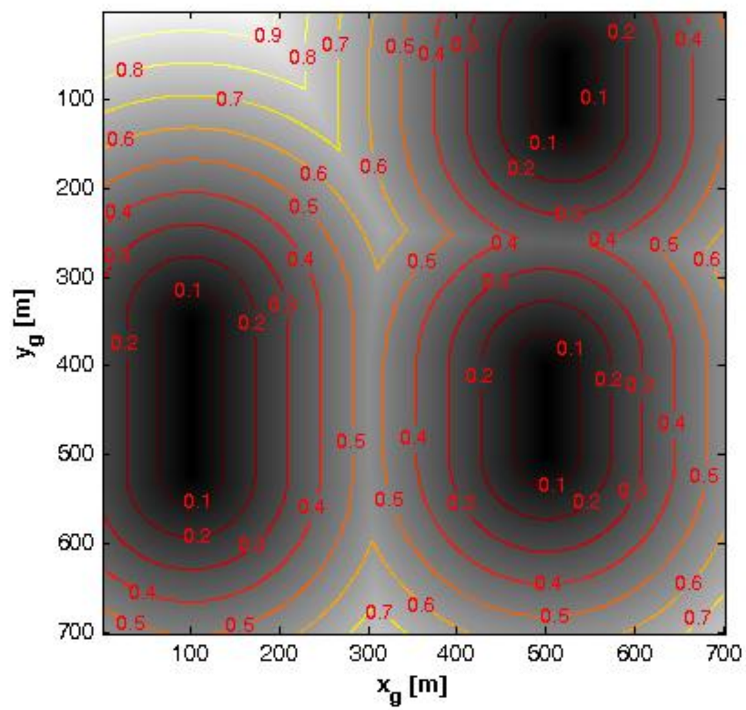
Distance Transformation



- Applying different metrics



Distance Transformation based on "City-Block" metric



Distance Transformation based on "Euclidean" metric

→ **The darker the region in the occupancy map, the higher the danger of collision with obstacles**

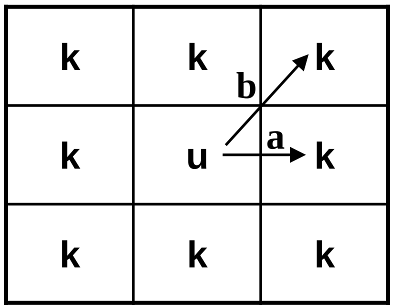
A* Algorithm

- In order to find the best (shortest) path to the goal
- Applied in e.g. automobile navigation
- Extending the path with a “cost function”

F(k) = G(k) + H(k)

H(k): Cost of moving from node *k* to the goal

G(k): Cost of moving from node *u* to its neighbour nodes *k*



$\mathbf{a} = \mathbf{10} = \mathbf{G(k_i)}$

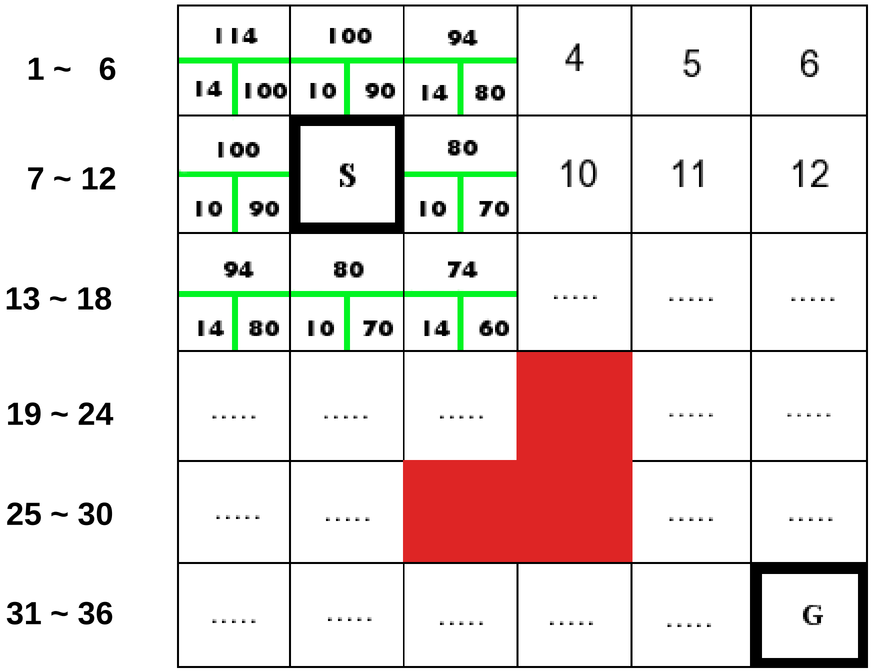
$\mathbf{b} = \mathbf{14} = \mathbf{G(k_i)}$

$\mathbf{a : b} = 10 : 14 \approx 1 : \sqrt{2}$

Figure 8. Distance Mask

Step 1

Enlist the starting node S with its neighbours and predecessors in the "open list"



Q	F(k)	G(k)	H(k)	P
1	114	14	100	S
2	100	10	90	S
3	94	14	80	S
7	100	10	90	S
S	-	-	-	-
9	80	10	70	S
13	94	14	80	S
14	80	10	70	S
15	74	14	60	S

Figure 9. A* Algorithm

Step 2

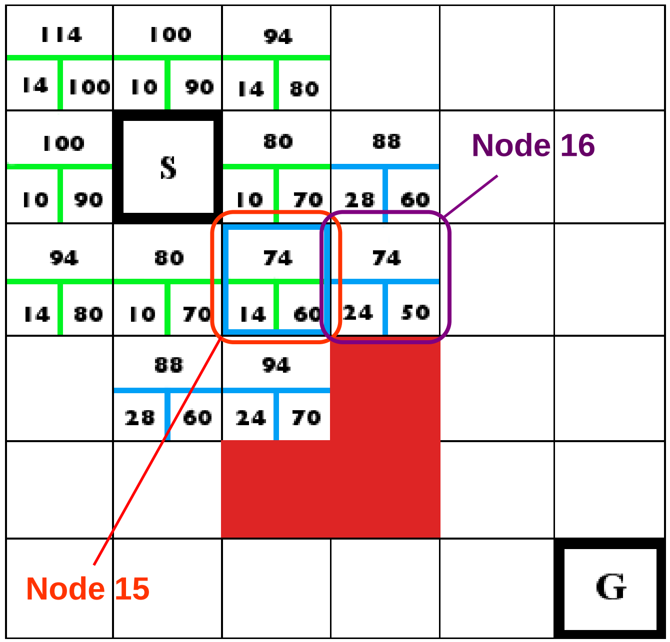
Searching for the node having the lowest F-value (e.g. node 15) and enlist in the “closed list”

Step 3

Calculating the F-values of the neighbours of the actual node (e.g. node 15)

Step 4

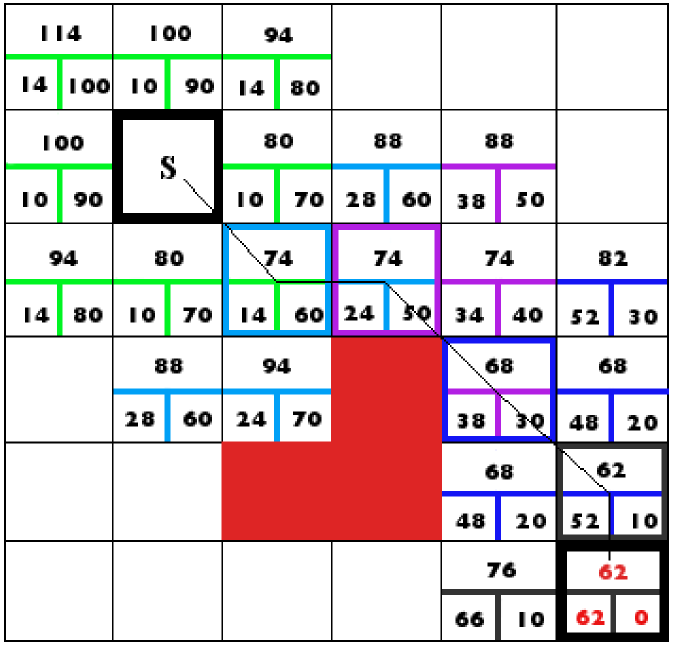
Repeating the steps for the new actual node (e.g. node 16)



Closed List

Q	F(k)	G(k)	H(k)	P
S	-	-	-	-
15	74	14	60	S
16	74	24	50	15

Figure 9. A* Algorithm (continued)



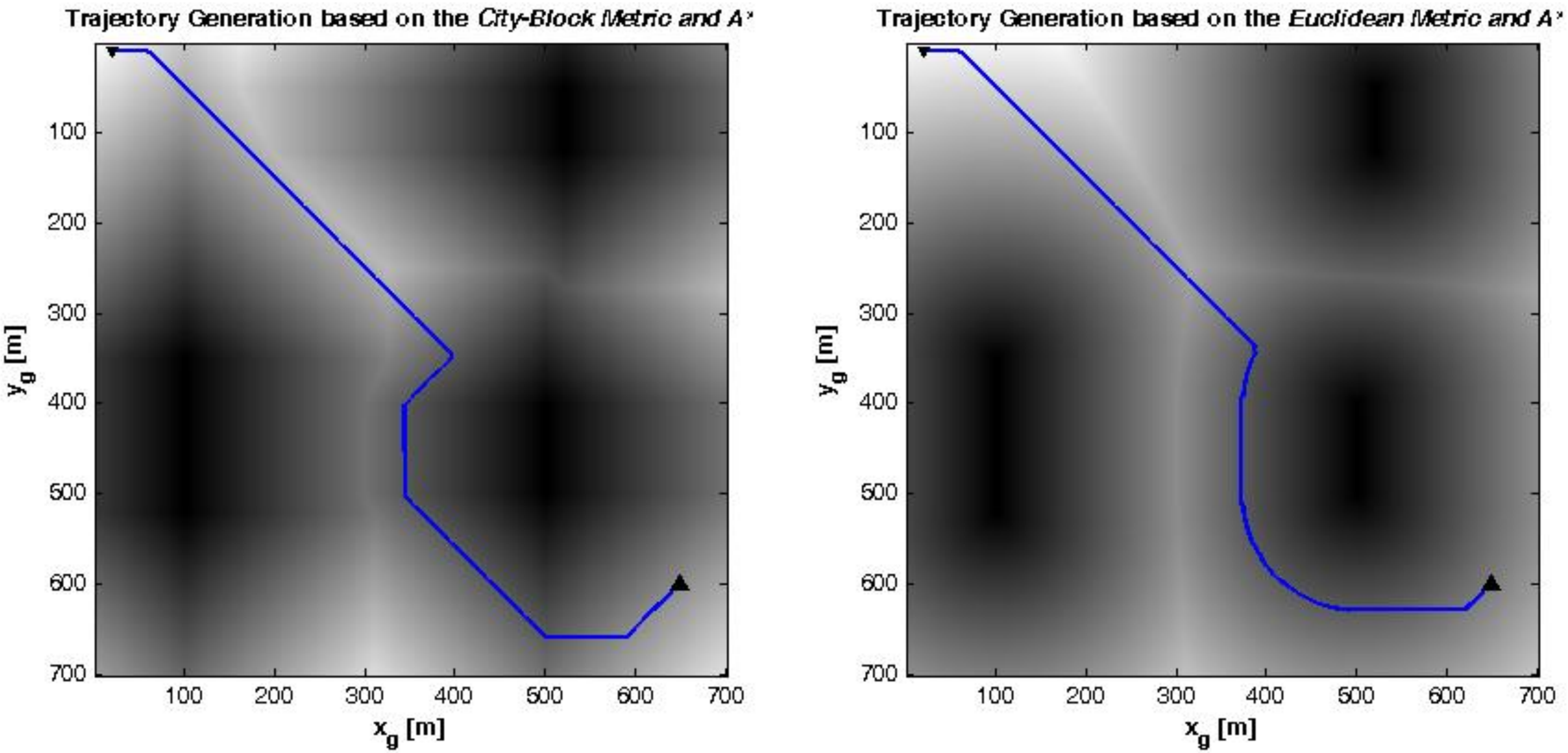
Closed List

Q	F(k)	G(k)	H(k)	P
S	-	-	-	-
15	74	14	60	S
16	74	24	50	15
23	68	38	30	16
30	62	52	10	23
36 = G	62	62	0	30

Path : S → 15 → 16 → 23 → 30 → 36 = G

Figure 9. A* Algorithm (continued)

Distance Transformation & A* Algorithm

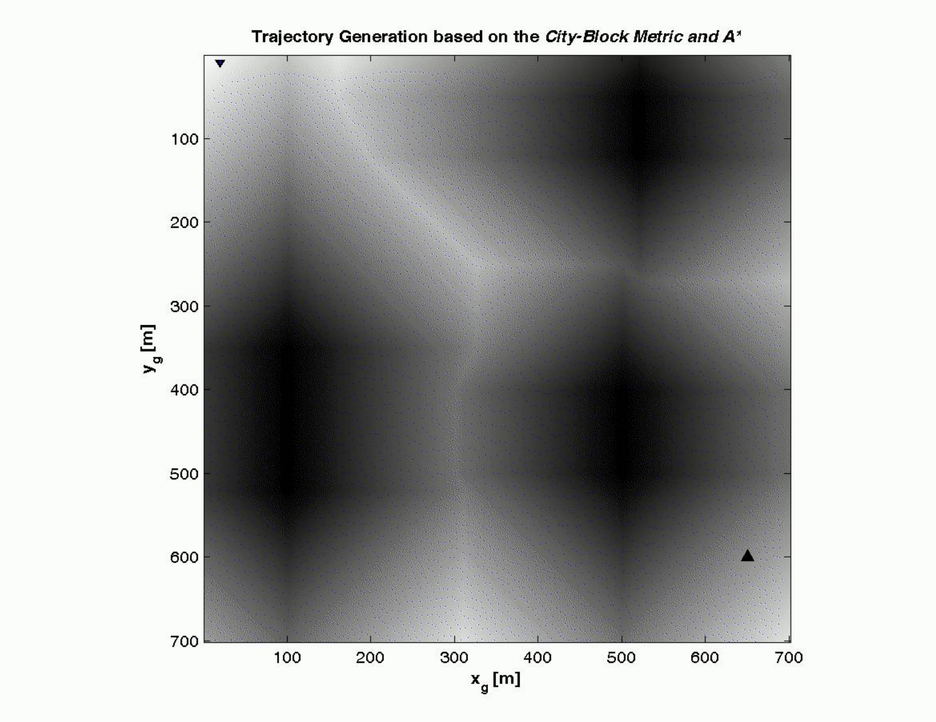


L = 1440 m

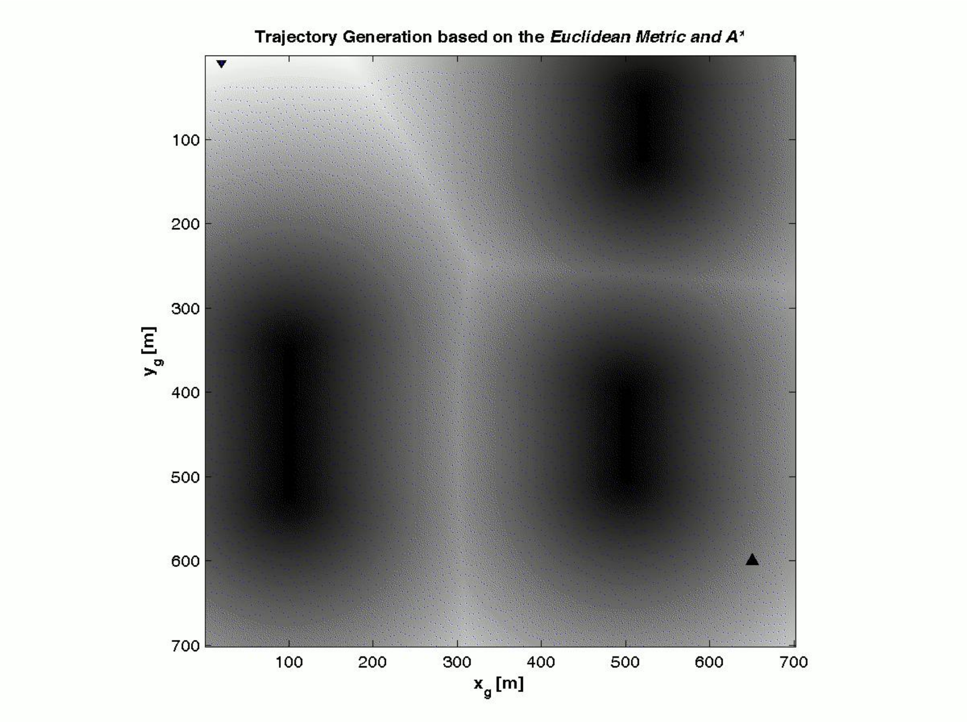
L = 1304 m

Figure 10. Distance Transformation and A* Algorithm for $\delta = 0.35$

Distance Transformation & A* Algorithm



L = 1232 m



L = 1214 m

Figure 11. **Distance Transformation and A* Algorithm for $\delta = 0.15$**

- UAV & Laser Range Finder
- Simulation Model
- Strategies
- Conclusions & Outlook

Conclusions & Outlook

- Adaptive Flight Path Generation by means of *Bézier curves* provides a possible solution for the implementation of a “Sense & Avoid” – system
- *Distance Transformation* and *A* Algorithm* are appropriate for constructing a feasible trajectory
- Investigation of the feasibility in implementing the above mentioned algorithms
- Validation of the simulation with flight tests