

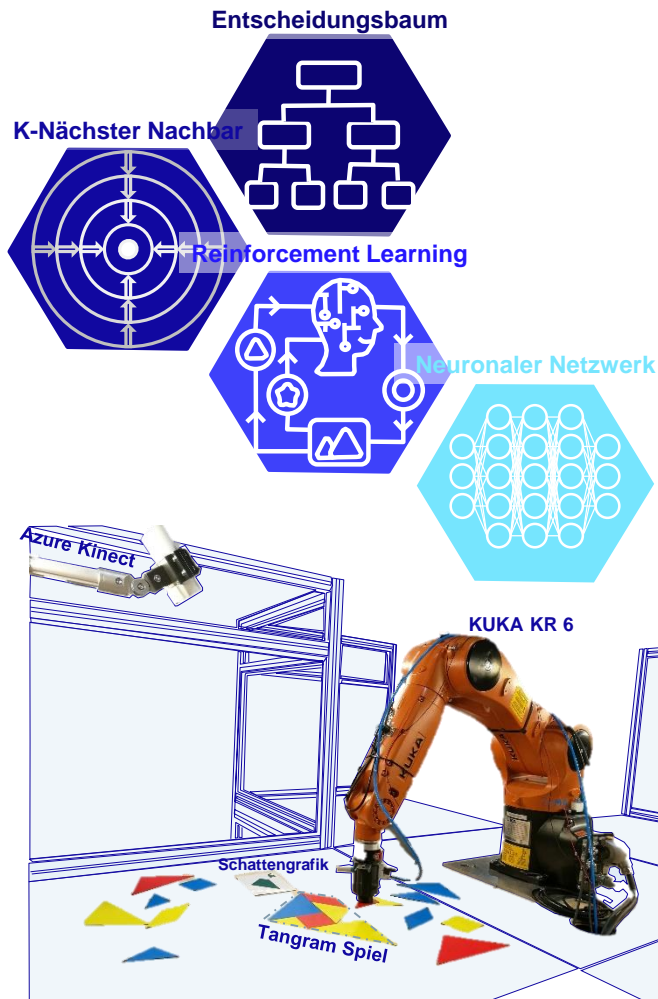
Intelligent Robotic System for Solving Dissection Puzzle Combining K-Nearest Neighbors, Decision Tree and Deep Q Network

Robotix-Academy Roadshow 2021

Xiaomei Xu, M.Sc.

ZeMA

29.04.2021, Saarbrücken



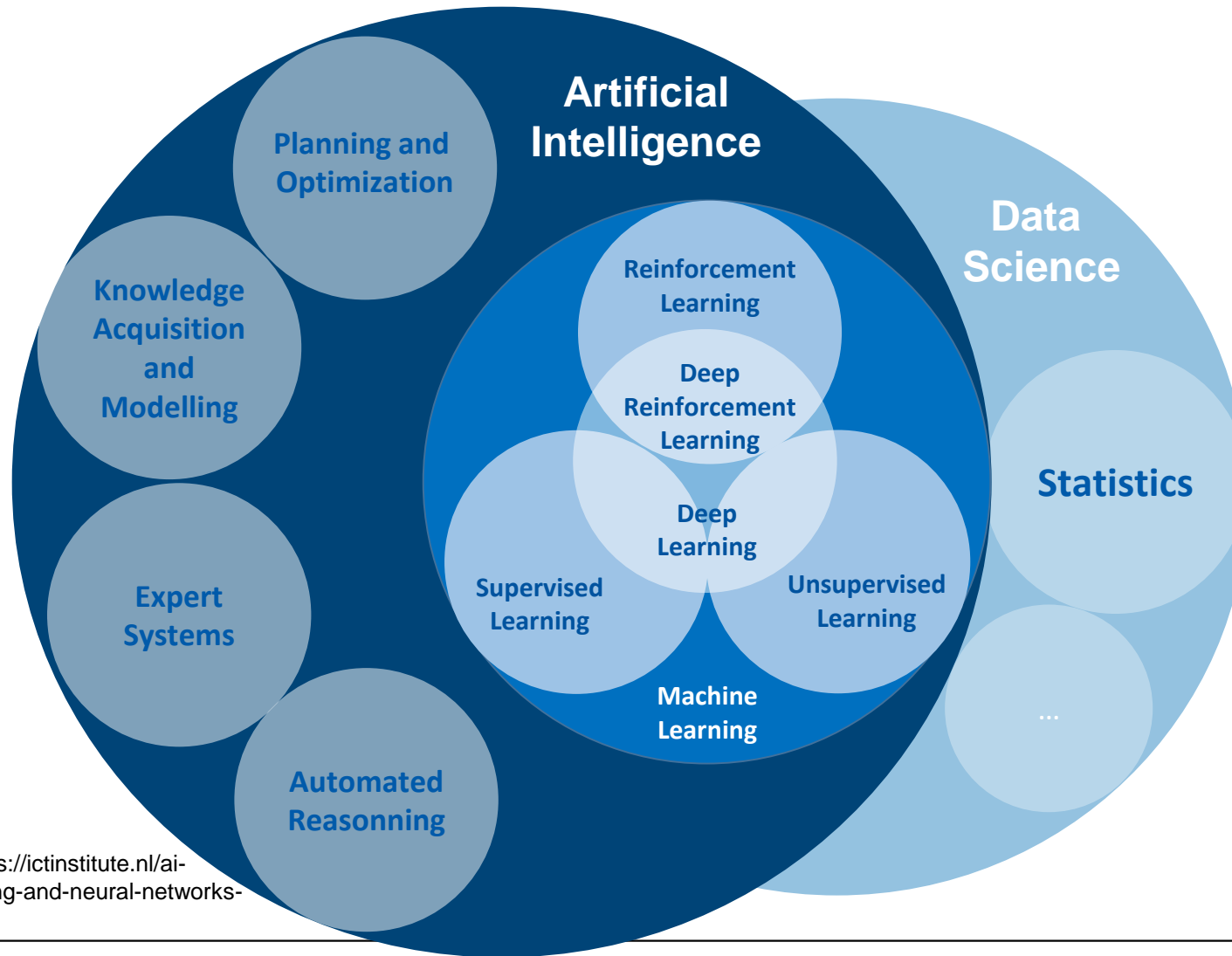
Agenda

- 1 Artificial Intelligence**
- 2 Machine Learning**
- 3 Reinforcement Learning**
- 4 Tangram Rule**
- 5 Structure of intelligent Robotic System in Tangram**
- 6 Camera System**
- 7 Intelligent Robotic System**
- 8 Path planning**
- 9 Implementation**

Agenda

- 1 Artificial Intelligence**
- 2 Machine Learning**
- 3 Reinforcement Learning**
- 4 Tangram Rule**
- 5 Structure of intelligent Robotic System in Tangram**
- 6 Camera System**
- 7 Intelligent Robotic System**
- 8 Path planning**
- 9 Implementation**

Artificial intelligence

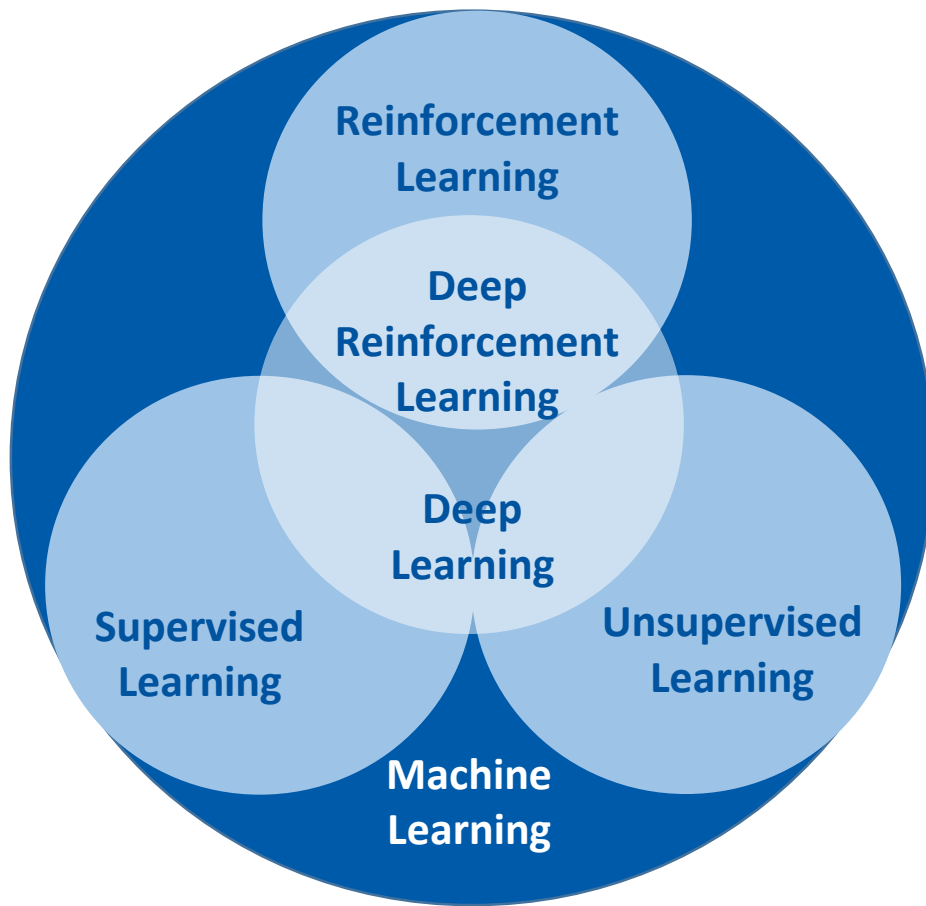


Reference: <https://ictinstitute.nl/ai-machine-learning-and-neural-networks-explained/>

Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System
- 7 Intelligent Robotic System
- 8 Path planning
- 9 Implementation

Machine Learning Overview



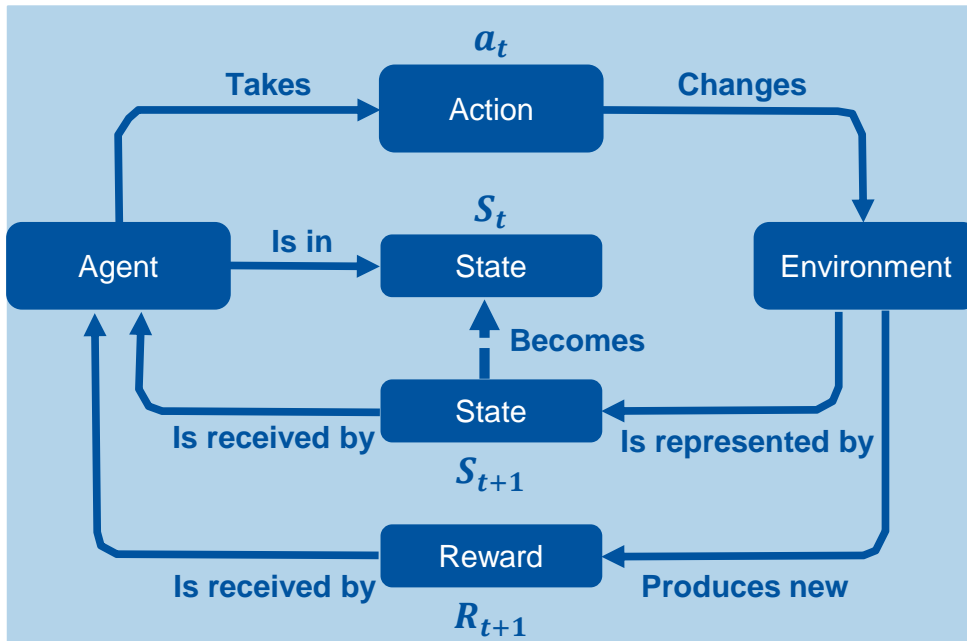
- **ML** is about learning from data and making predictions and/or decision. Usually it is categorized as:
 - **Supervised learning** algorithms are trained using **labeled** examples, such as an input where the desired output is known.
 - **Unsupervised learning** is used against data that has **no** historical **labels**.
 - **Reinforcement learning** algorithm discovers through trial and error which actions yield the greatest rewards.
- **Deep learning** mimics the workings of the human brain in processing data. It is able to learn without human supervision, drawing from data that is both unstructured and unlabeled.

Reference: Y. Li, Deep Reinforcement Learning
P. Ongsulee, Artificial Intelligence, Machine Learning and Deep Learning

Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning**
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System
- 7 Intelligent Robotic System
- 8 Path planning
- 9 Implementation

The Reinforcement Learning Framework



- The **agent** is the action-taking or decision-making learning algorithm in any RL problem.
- Ever **action** taken is the result of analyzing the current state of the environment and attempting to make the best decision based on that information
- The **environment** of a RL task is any dynamic process that produces data that is relevant to achieving our objective.
- To make things algorithm-friendly, we need to take this environment data and bundle it into discrete packets that we call the **state** (of the environment).
- The **reward** is a (local) signal of how well the learning algorithm is performing at achieving the global objective. It can be a positive signal (i.e. doing well, keep it up) or a negative signal (i.e. don't do that).

Policy-Based RL and Value-Based RL

Model Free RL: It can only take one step at a time, waiting for feedback from the real world, and then taking the next step based on that feedback.

Q learning, Sarsa, Policy Gradient

Value-Based RL: The output is the value of all actions, and we choose the action based on the highest value, so that the value-based approach is more definitive in the decision making part than the probability-based approach. **Q-learning, Sarsa**

Monte-Carlo update: Once the game has started, we wait for it to end and then take stock of all the turning points in the round before updating our code of behaviour. **Monte-Carlo Update**

On- Policy: It must be present in person and must be learning in person while playing.

Sarsa, Sarsa lambda

Model-based RL: It is able to anticipate all the scenarios that will happen next by imagining them. It then selects the best of these imagined situations. The next step in the strategy is based on this scenario.

Policy-Based RL: It analyses the environment it is in through the senses, directly outputs various probabilities for the next action, and then acts on the probabilities. **Policy Gradient**

Temporal-Difference update: Single step updates are updated at every step of the game, without waiting for the end of the game, so we can learn as we play. **Q-learning, Sarsa**

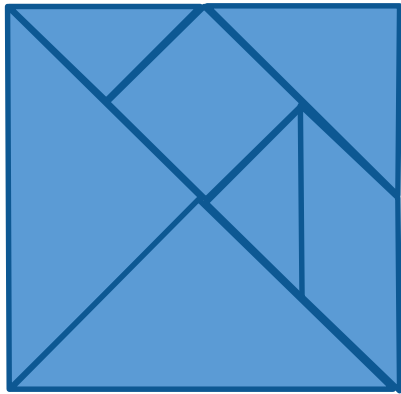
Off-Policy: It choose to play on ist own or someone else to play. The learning bases on the past experience. **Q Learning, Deep Q Network**

Agenda

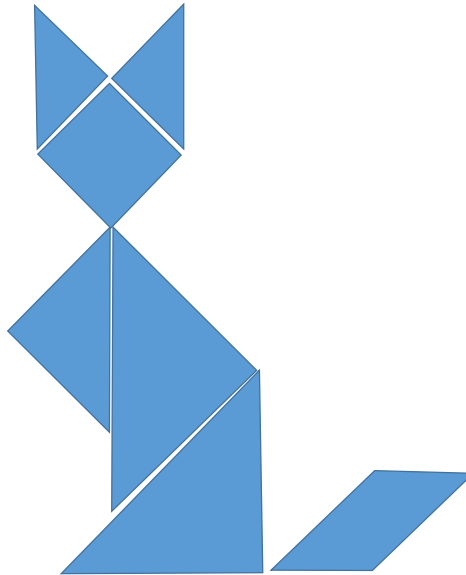
- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule**
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System
- 7 Intelligent Robotic System
- 8 Path planning
- 9 Implementation

Tangram Rule

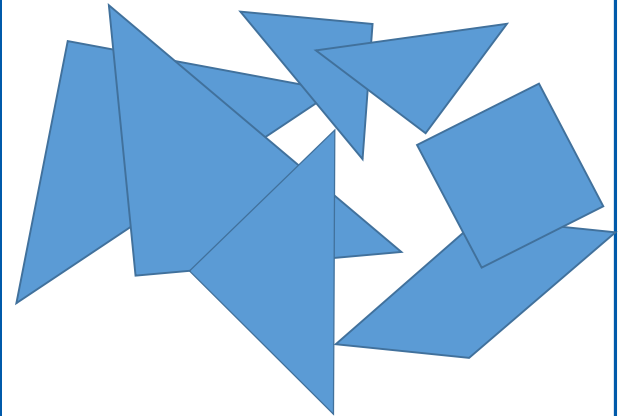
Use all 7 forms



The forms must touch



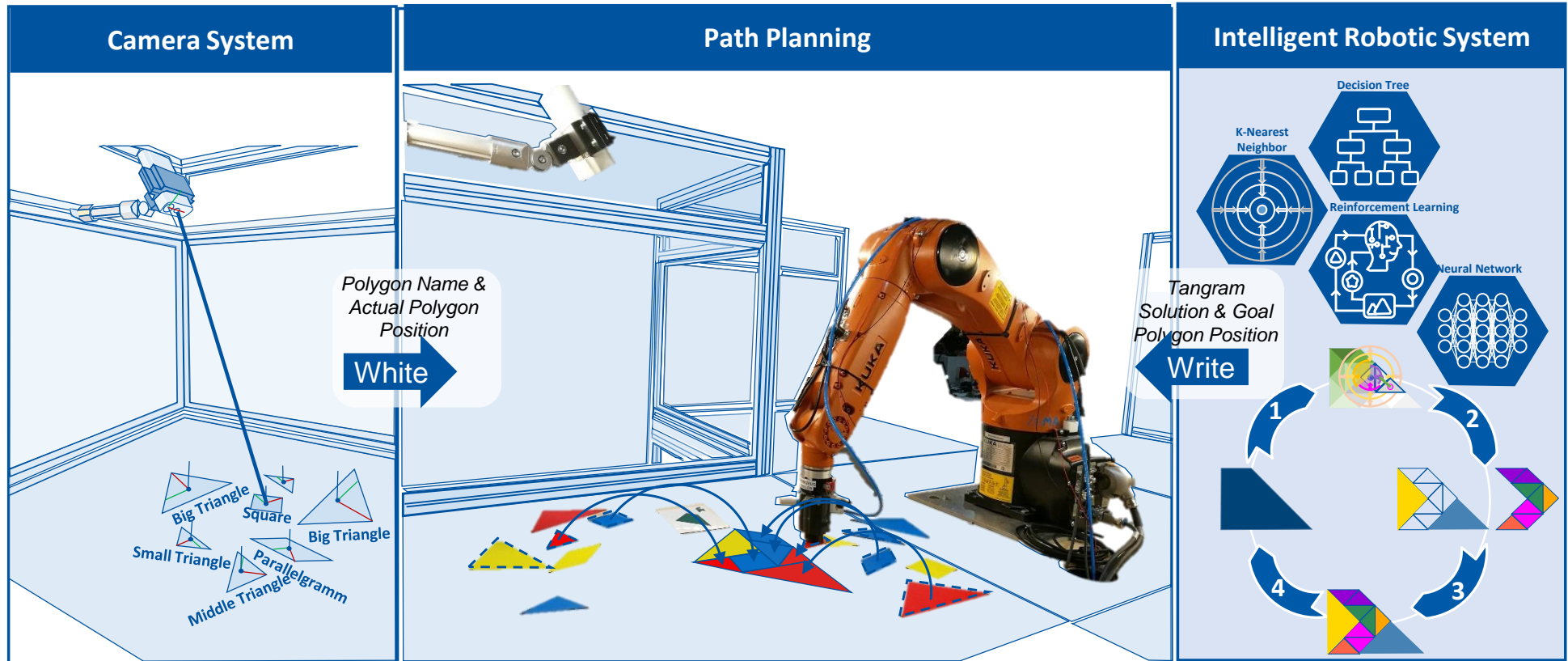
The shapes must not lie on top of each



Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram**
- 6 Camera System
- 7 Intelligent Robotic System
- 8 Path planning
- 9 Implementation

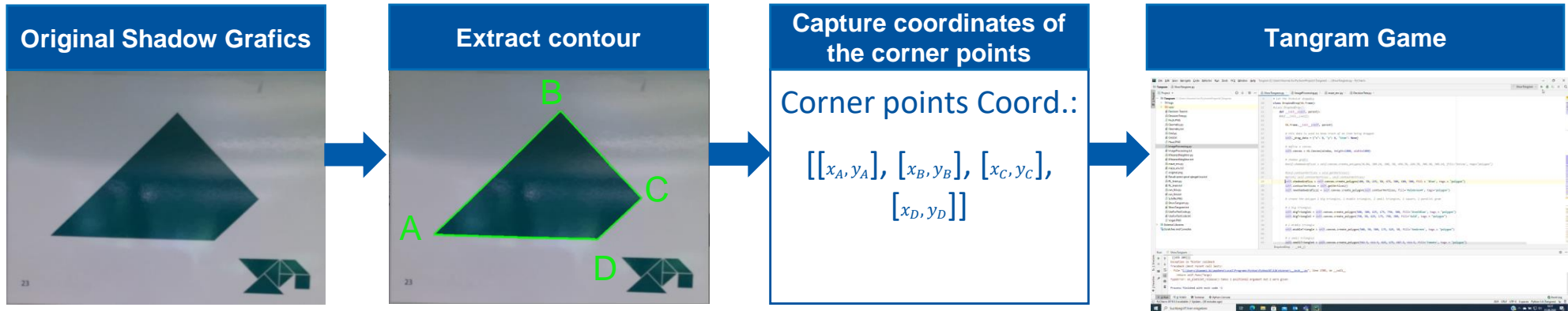
Structure of Intelligent Robotic System in Tangram



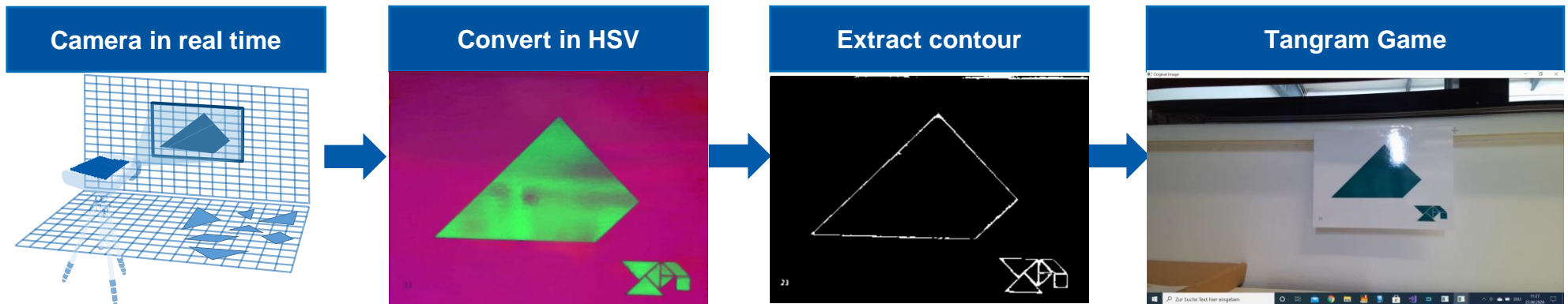
- Camera System : Contour Detection, Shape Detection, 3D Position Estimation
- Path Planning: Pick and Place
- Intelligent Robotic System: K-Nearest Neighbors, Decision Tree and Deep Q Network

Camera System – Contour Detection

Original Image -> Convert Grey Image -> Threshold Operation -> Draw Contour

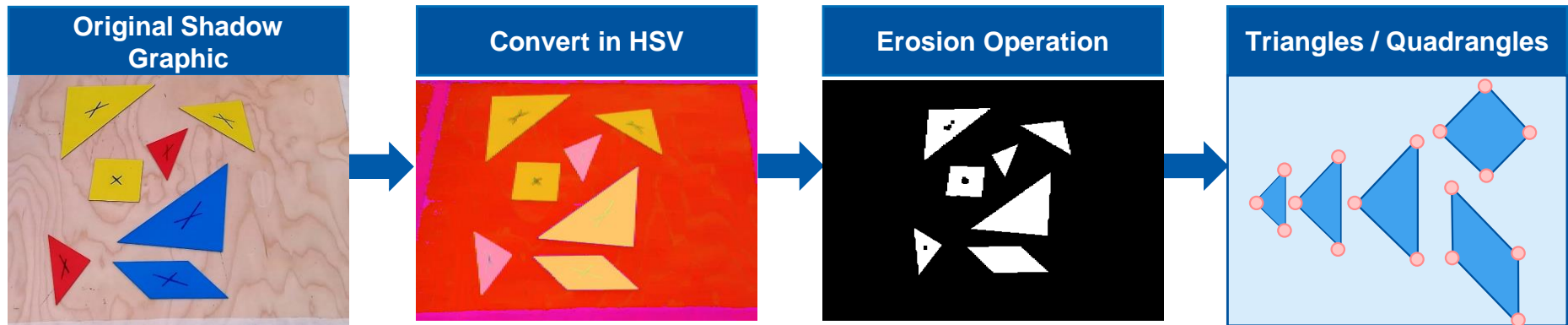


Start camera through API -> Convert HSV image -> Threshold operation -> Capture contour

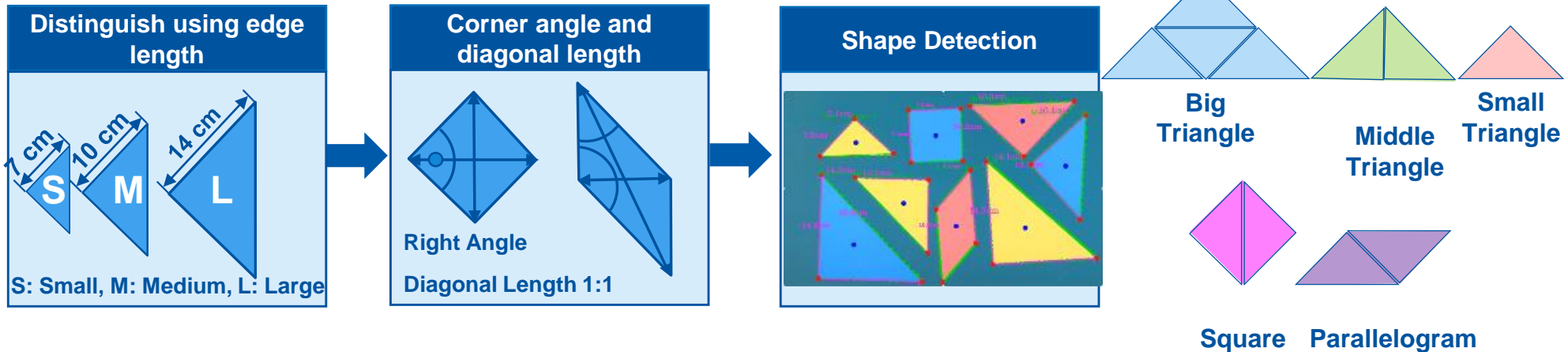


Camera System – Shape Detection

Original Image -> Convert HSV Image -> Erosion Operation -> Triangles / Quadrangles



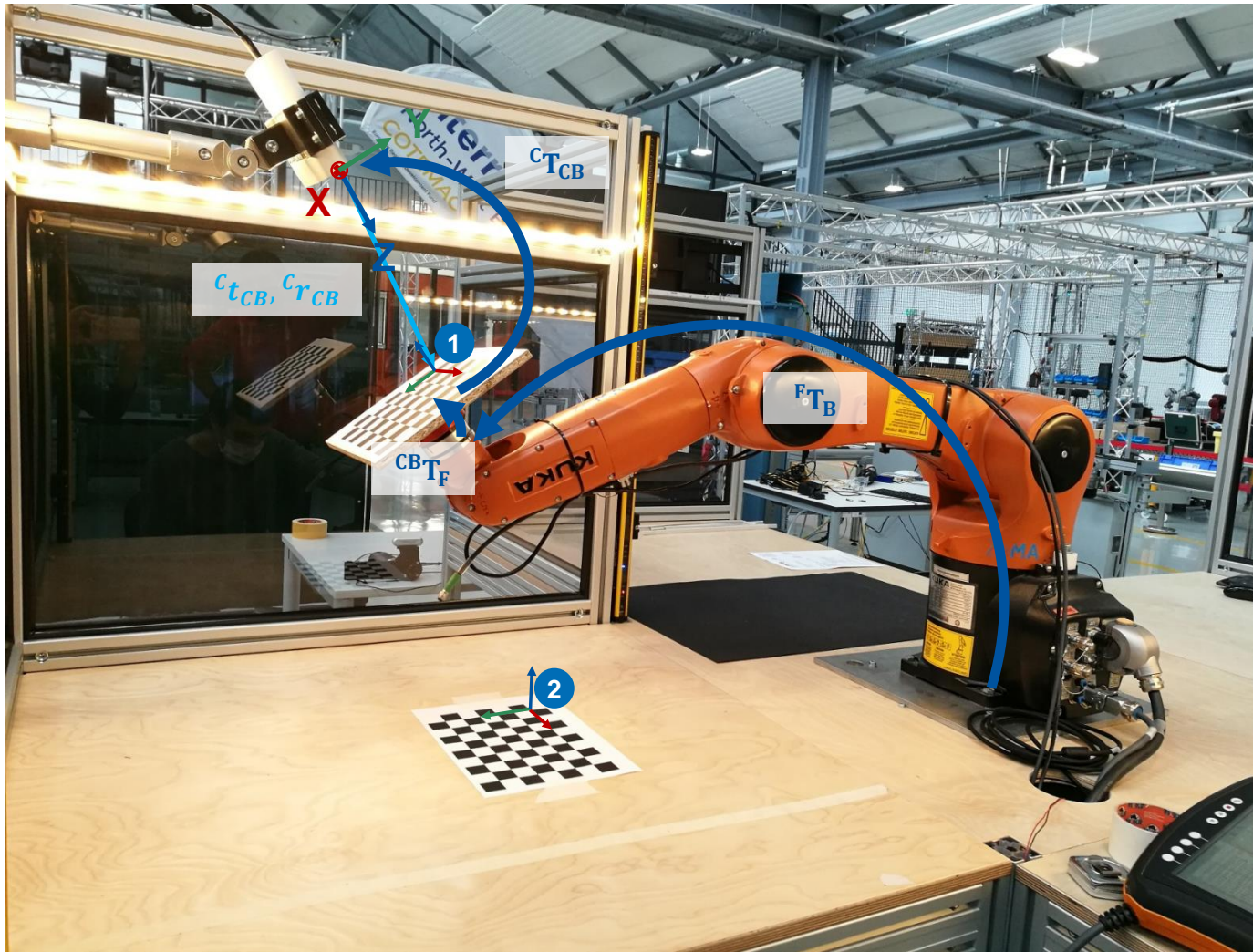
Distinguish Triangle with Small/ Medium/ Large Size -> Distinguish Square and Parallelogram



Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System**
- 7 Intelligent Robotic System
- 8 Path planning
- 9 Implementation

Camera System – Camera Calibration



■ Camera Calibration

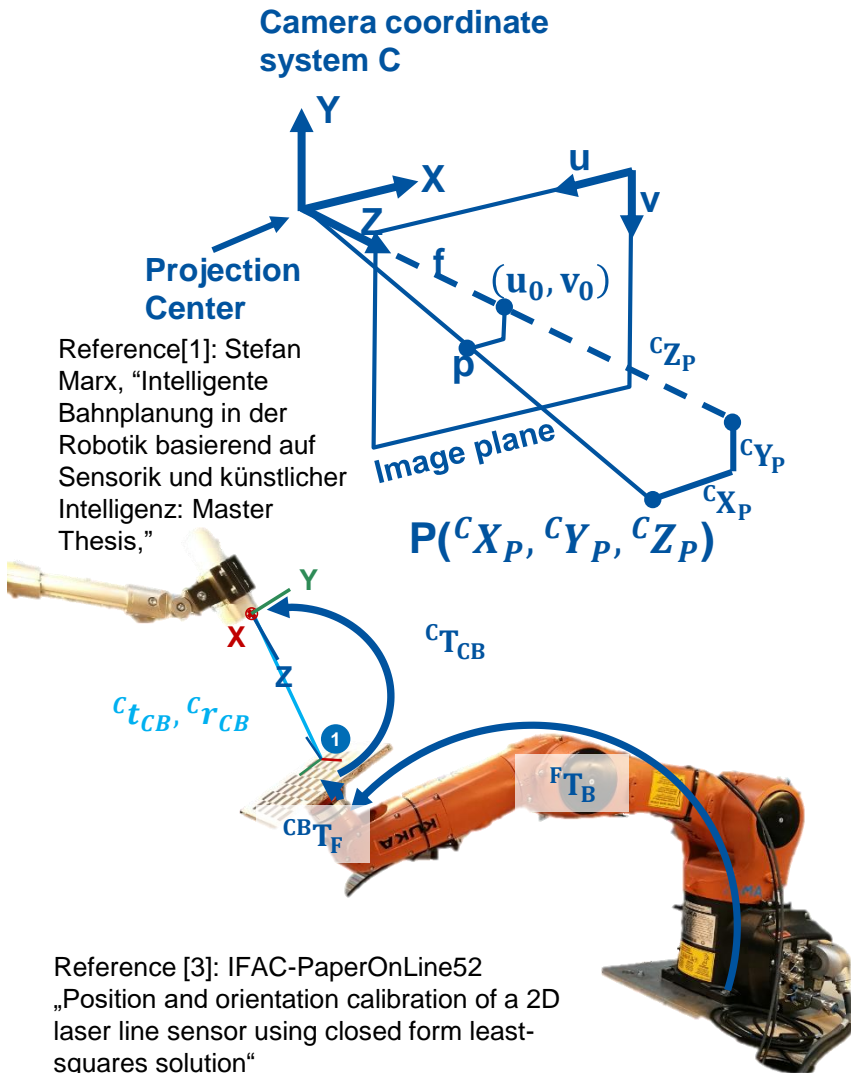
– Input:

- 9 x 7 dimension
- Edge length of a square in the pattern.
- 3840 x 2160 resolution
- 20 images of the chessboard pattern from different positions and orientations

– Output:

- Camera matrix: 3x3, intrinsic parameters
- Translational vector and rotational vector in Rodrigues form from camera coordinates to each chessboard image coordinate

Camera System – 3D Position Estimation



$P(cX_P, cY_P, cZ_P)$: relative to camera coord. system

(u_0, v_0) the image plane

f is the focal length of the camera

The Pixel coordinated can be transformed to camera coordinates as following:

$$\begin{pmatrix} cX_P/cZ_P \\ cY_P/cZ_P \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} u_P \\ v_P \\ 1 \end{pmatrix}$$

Reference[2]: Learning OpenCV

The transformation matrix cT_B from base coordinate system of the robot to camera is calculated in equation:

$$cT_B = cT_{CB} \cdot cT_F \cdot F T_B$$

$F T_B$ transformation matrix from base coord. of the robot to robot flange

cT_F is the transformation matrix from the robot flange to the chessboard [3]

cT_{CB} is the transformation matrix from chessboard origin coordinates at origin point (0,0,0) to camera coordinates

Camera System – 3D Position Estimation

The transformation matrix from chessboard coordinate system to the camera coordinate system is calculated from the translational vector ${}^C t_{CB}$ and rotational vector ${}^C r_{CB}$ OpenCV.

Since the polygons lie on the plane, ${}^{CB} Z_P$ is equal to zero:

$$\begin{pmatrix} {}^C X_P \\ {}^C Y_P \\ {}^C Z_P \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^{CB} X_P \\ {}^{CB} Y_P \\ {}^{CB} Z_P \\ 1 \end{pmatrix} \quad (1)$$

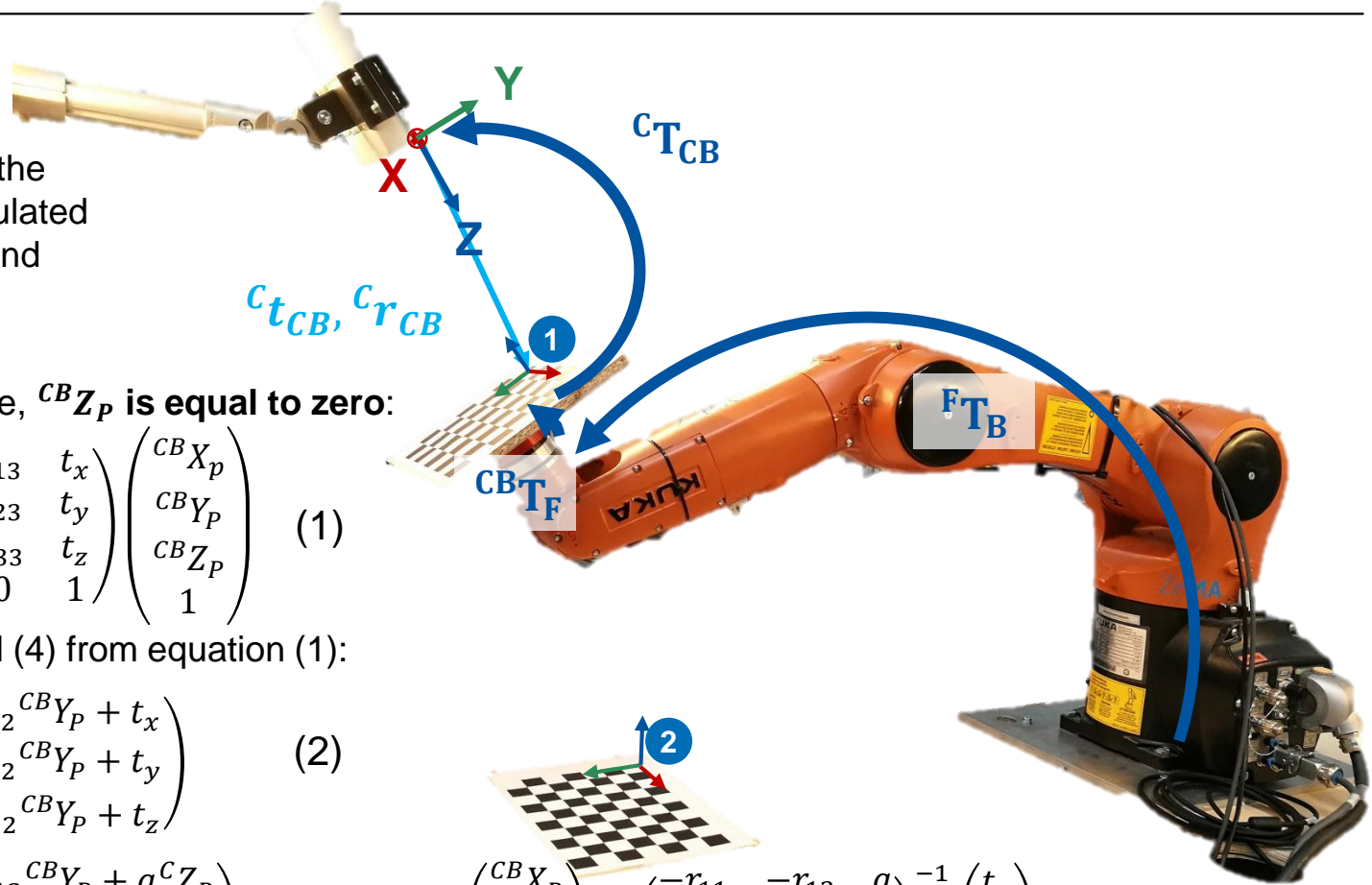
We deduce equations (2), (3), and (4) from equation (1):

$$\begin{pmatrix} {}^C X_P \\ {}^C Y_P \\ {}^C Z_P \end{pmatrix} = \begin{pmatrix} r_{11} {}^{CB} X_P + r_{12} {}^{CB} Y_P + t_x \\ r_{21} {}^{CB} X_P + r_{22} {}^{CB} Y_P + t_y \\ r_{31} {}^{CB} X_P + r_{32} {}^{CB} Y_P + t_z \end{pmatrix} \quad (2)$$

$$\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} -r_{11} {}^{CB} X_P - r_{12} {}^{CB} Y_P + a {}^C Z_P \\ -r_{21} {}^{CB} X_P - r_{22} {}^{CB} Y_P + b {}^C Z_P \\ -r_{31} {}^{CB} X_P - r_{32} {}^{CB} Y_P + c {}^C Z_P \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} {}^{CB} X_P \\ {}^{CB} Y_P \\ {}^C Z_P \end{pmatrix} = \begin{pmatrix} -r_{11} & -r_{12} & a \\ -r_{21} & -r_{22} & b \\ -r_{31} & -r_{32} & 1 \end{pmatrix}^{-1} \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (4)$$

${}^C Z_P$ is calculated in equation (9) and is then substituted in equation (4) to get the 3D position in the robot arm basis.



Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System
- 7 Intelligent Robotic System**
- 8 Path planning
- 9 Implementation

Intelligent robotic system – Analysis target photo puzzle

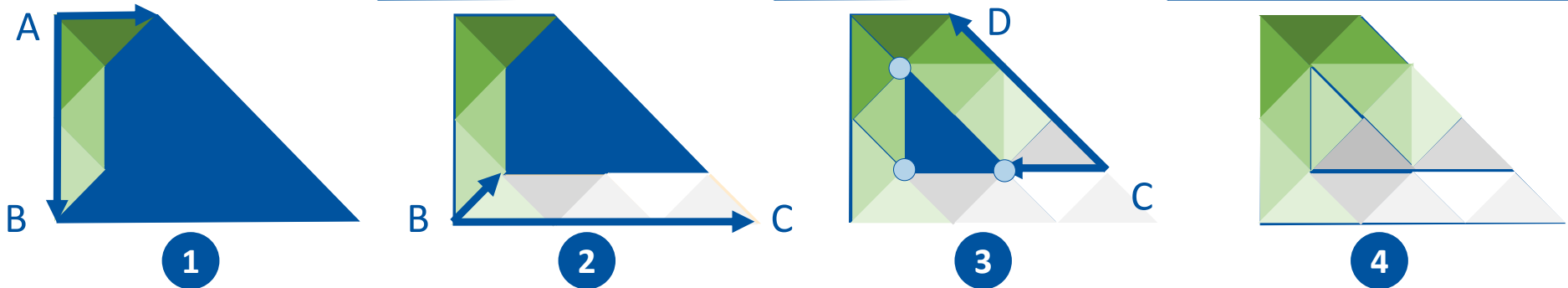
- Take the target puzzle and divide it into a grid of 16 small triangles

Extend in edge AB

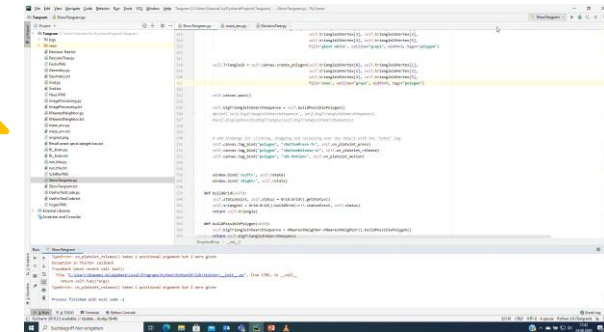
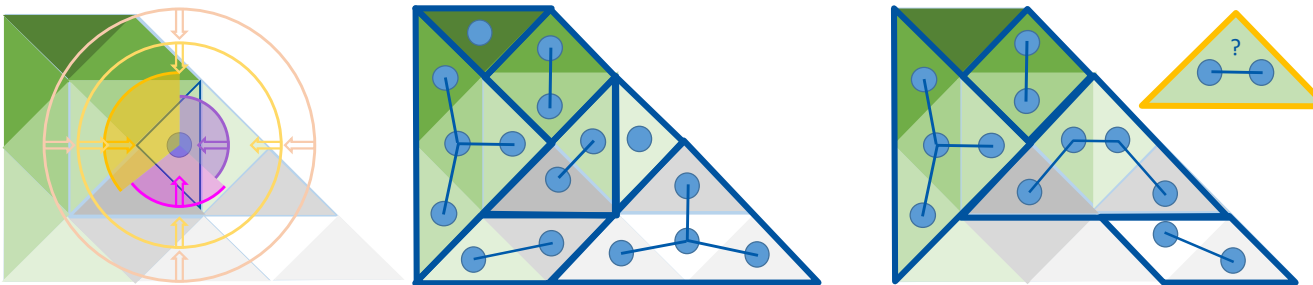
Extend in edge BC

Extend in edge CD

Build the grid

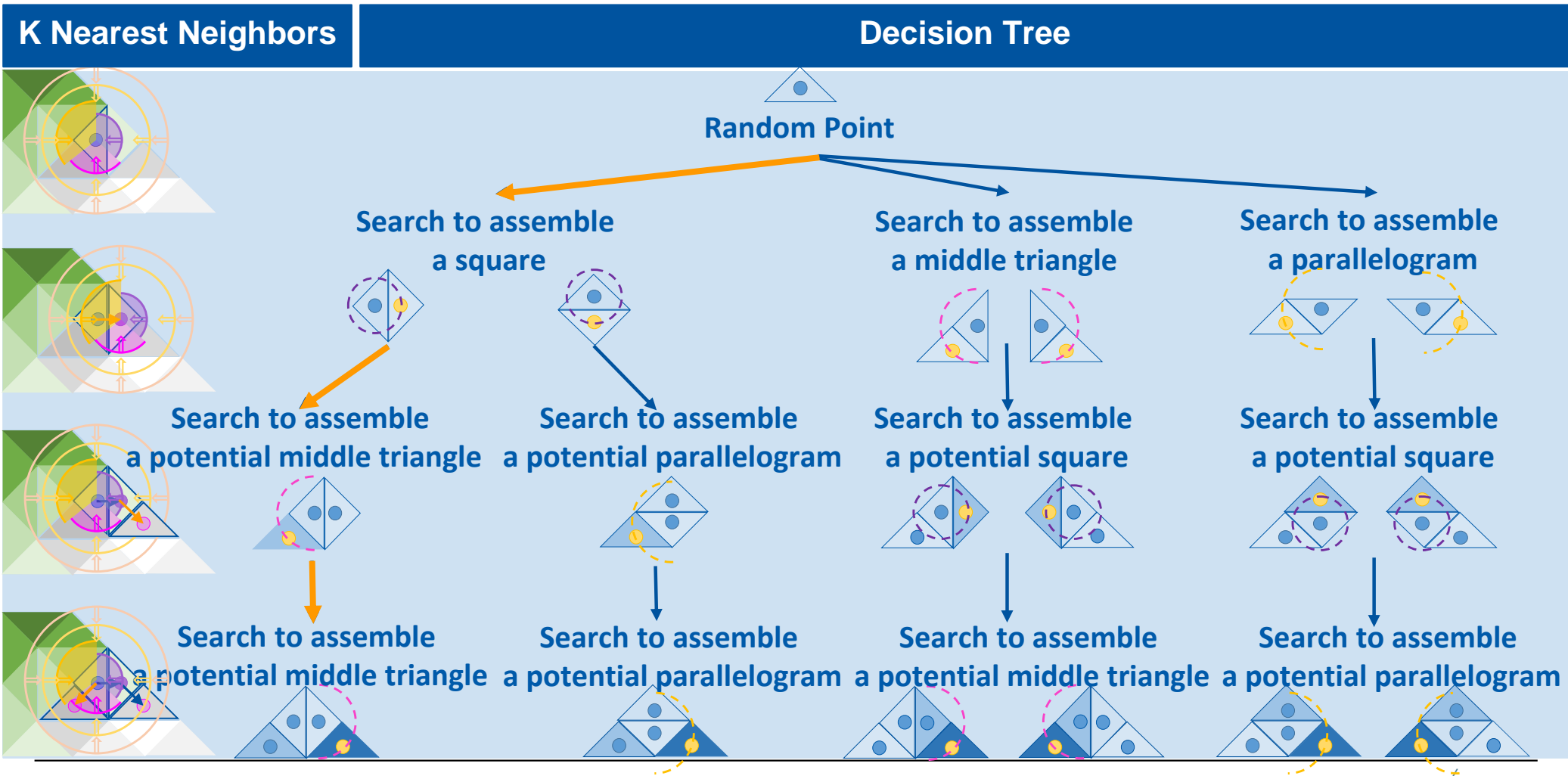


- The strategy for creating the shadow graphics with polygon is to first find the potential large triangle and then the potential medium polygon group, such as medium triangle, parallel graph, square and then the small triangle..



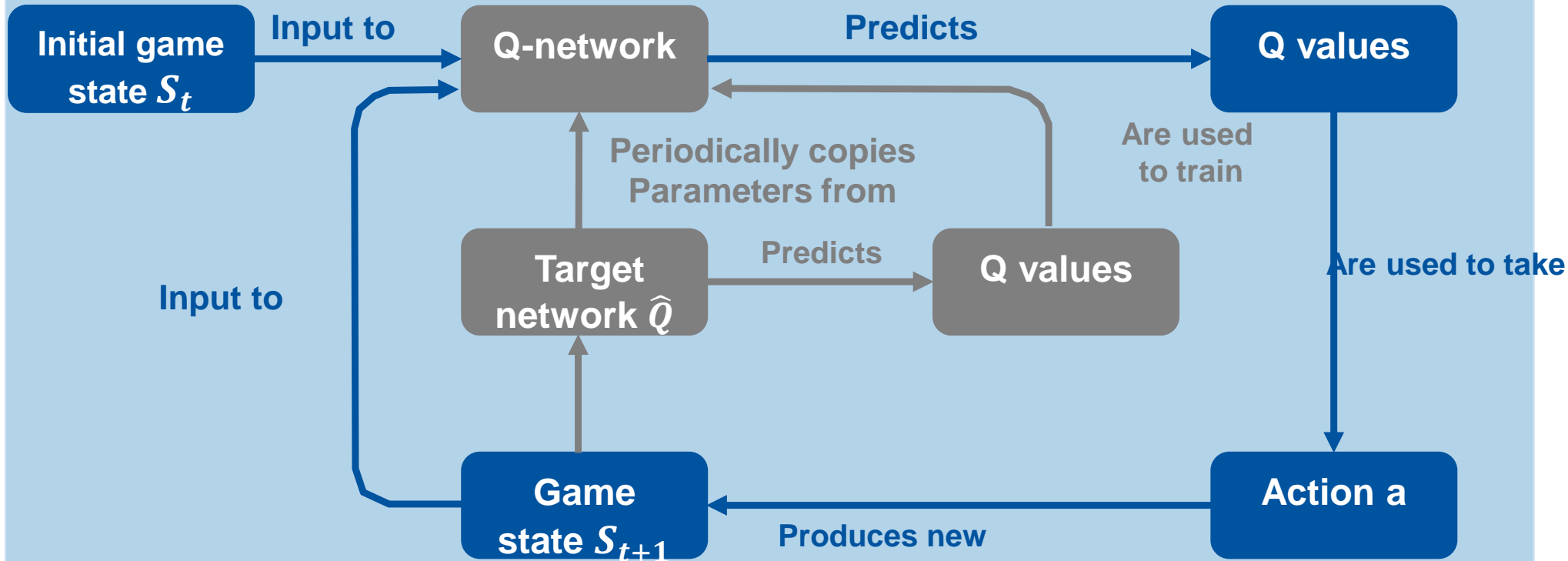
Intelligent robotic system – K Nearest Neighbor and decision tree

- Use K nearest neighbour and the decision tree to find the large triangle



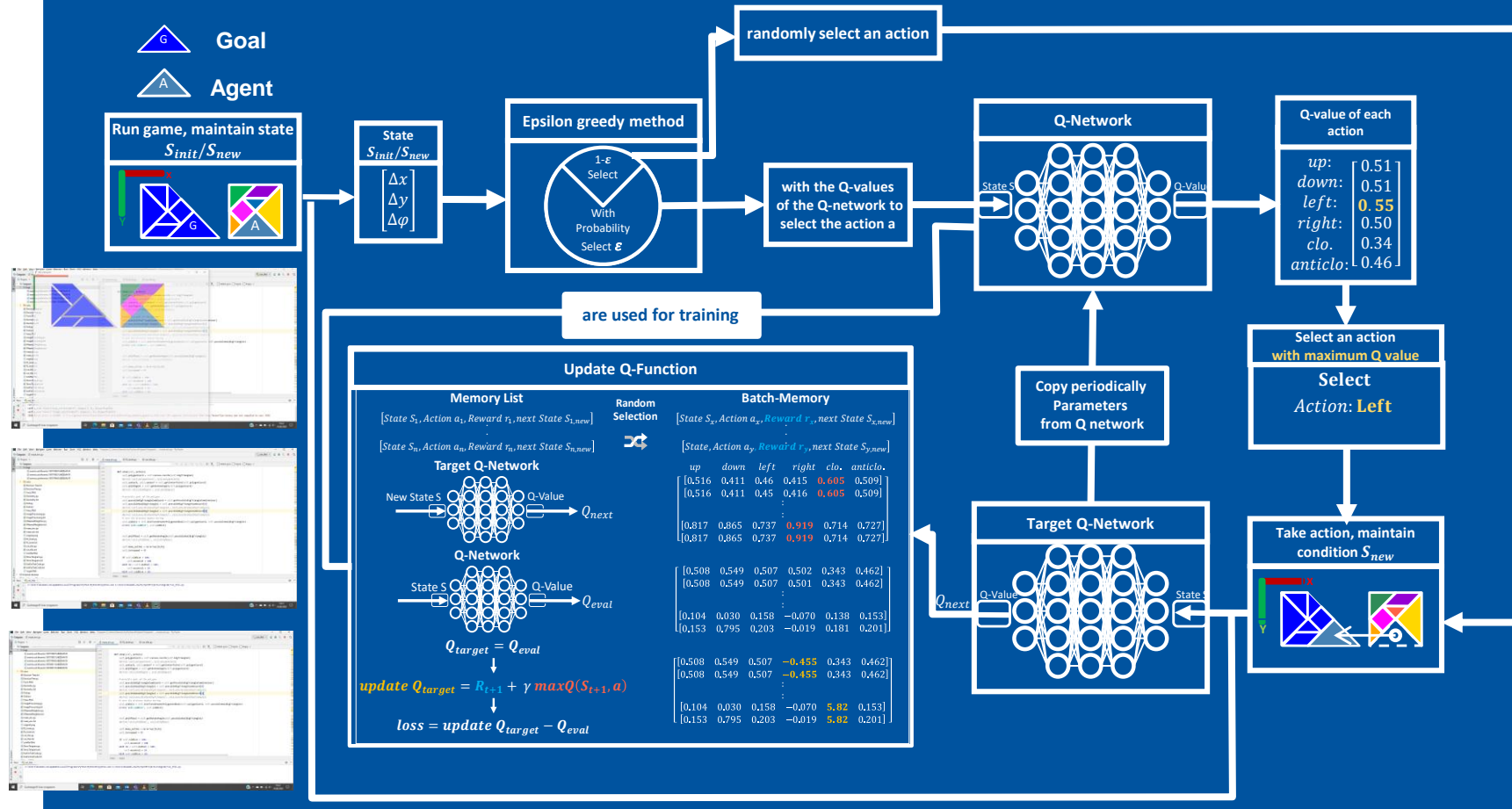
Q Learning with a target neural network

Q-Learning with target neural network



Intelligent robotic system – DeepQNetwork -> Q Learning + Neural Network

DeepQNetwork -> Q Lernen + Neuronaler Netzwerk



Update Q Function

Memory List

[State S_1 , Action a_1 , Reward r_1 , next State $S_{1,new}$]

[State S_n , Action a_n , Reward r_n , next State $S_{n,new}$]

Random
Selection

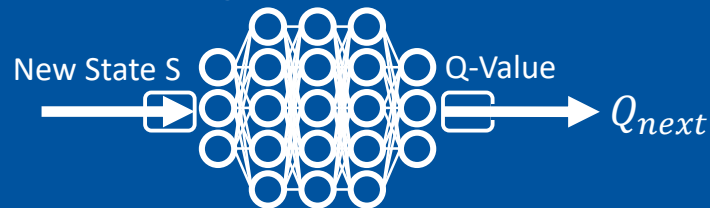


Batch-Memory

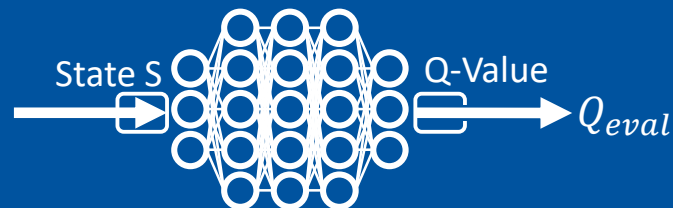
[State S_x , Action a_x , **Reward r_x** , next State $S_{x,new}$]

[State, Action a_y , **Reward r_y** , next State $S_{y,new}$]

Target Q-Network



Q-Network



$$Q_{target} = Q_{eval}$$



$$\text{update } Q_{target} = R_{t+1} + \gamma \max Q(S_{t+1}, a)$$



$$\text{loss} = \text{update } Q_{target} - Q_{eval}$$

<i>up</i>	<i>down</i>	<i>left</i>	<i>right</i>	<i>clo.</i>	<i>anticlo.</i>
[0.516	0.411	0.46	0.415	0.605	0.509]
[0.516	0.411	0.45	0.416	0.605	0.509]
			:		
			:		
[0.817	0.865	0.737	0.919	0.714	0.727]
[0.817	0.865	0.737	0.919	0.714	0.727]

[0.508	0.549	0.507	0.502	0.343	0.462]
[0.508	0.549	0.507	0.501	0.343	0.462]
[0.104	0.030	0.158	-0.070	0.138	0.153]
[0.153	0.795	0.203	-0.019	0.181	0.201]

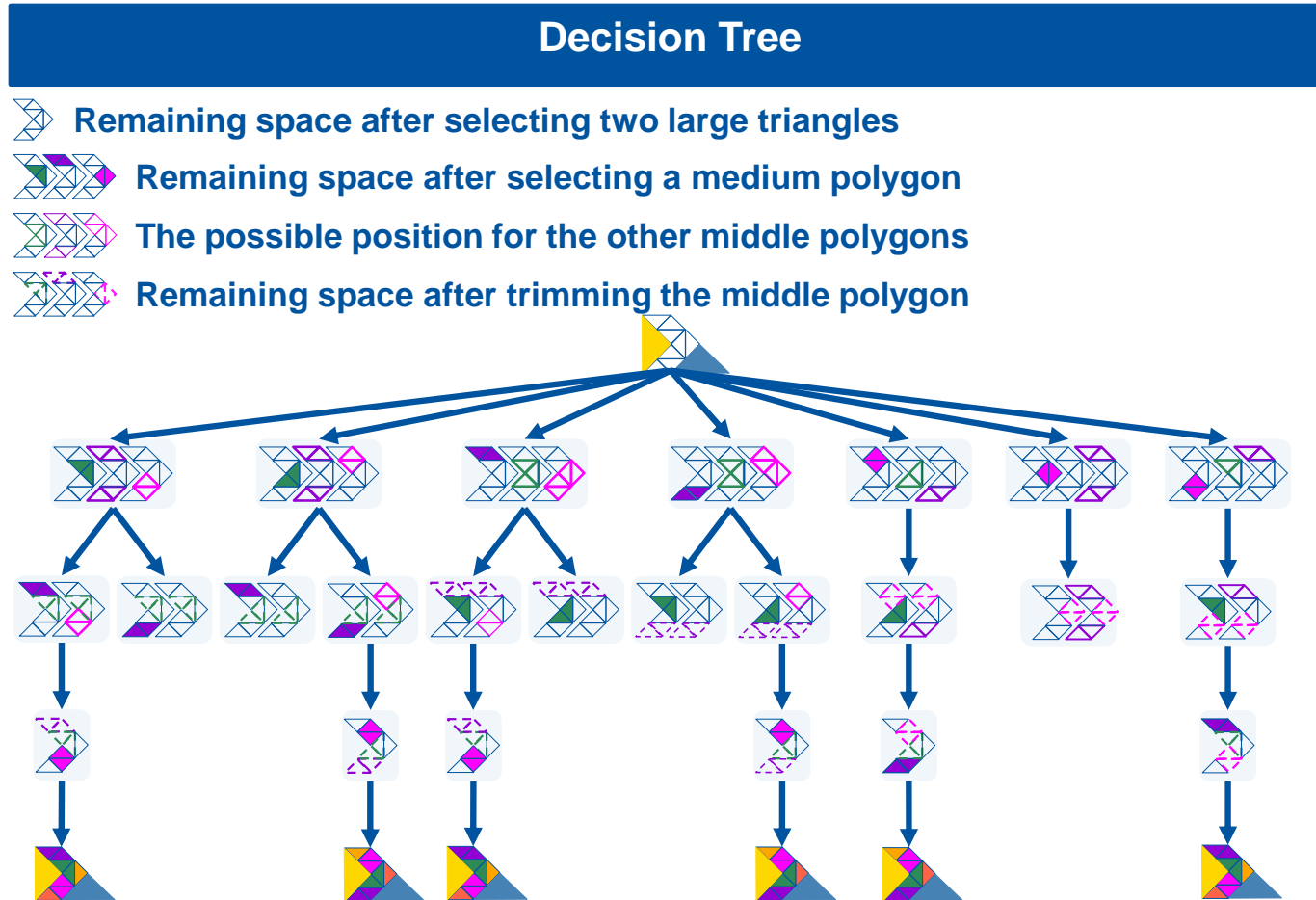
$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha [R_{t+1} + \gamma \max Q(S_{t+1}, a) - Q(S_t, A_t)]$$

Q_{target}

[0.508	0.549	0.507	-0.455	0.343	0.462]
[0.508	0.549	0.507	-0.455	0.343	0.462]
[0.104	0.030	0.158	-0.070	5.82	0.153]
[0.153	0.795	0.203	-0.019	5.82	0.201]

Intelligent robotic system – Decision Tree

- Use second decision tree to find the medium polygons combination group



Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System
- 7 Intelligent Robotic System
- 8 Path planning**
- 9 Implementation

Path Planning

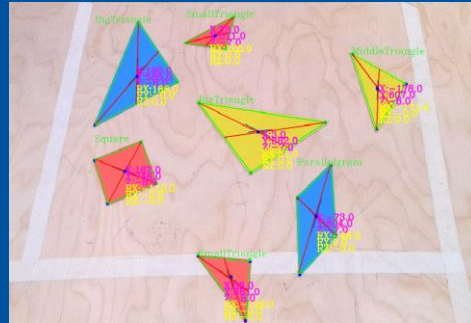
Intelligent Robotic System:

Image Processing:

Shape Detection

3D Position Estimation

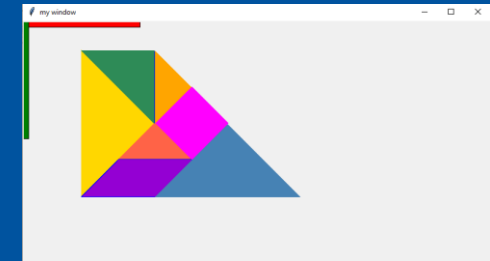
7 Polygon Actual
Position



AI Program:

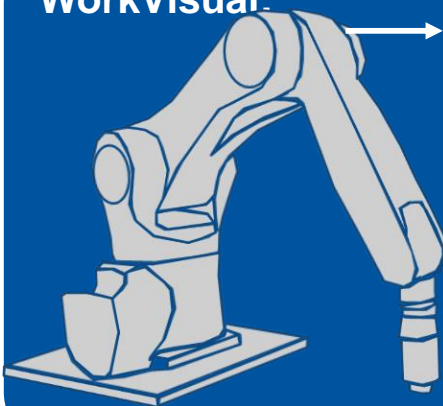
Solution

7 Polygon Goal
Position



OpenShowVar

WorkVisual:



$X, Y(\text{Rotate})$

$Z + 100\text{mm}$ $Z + 100\text{mm}$

Pick

Place

Pick

$X'Y'Z' + 100\text{mm}$ $Z' + 100\text{mm}$ $X''Y''Z'' + 100\text{mm}$ $Z'' + 100\text{mm}$

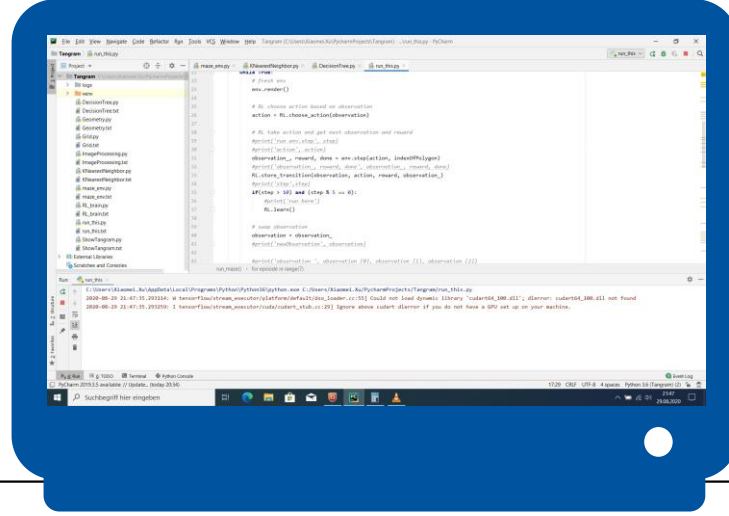
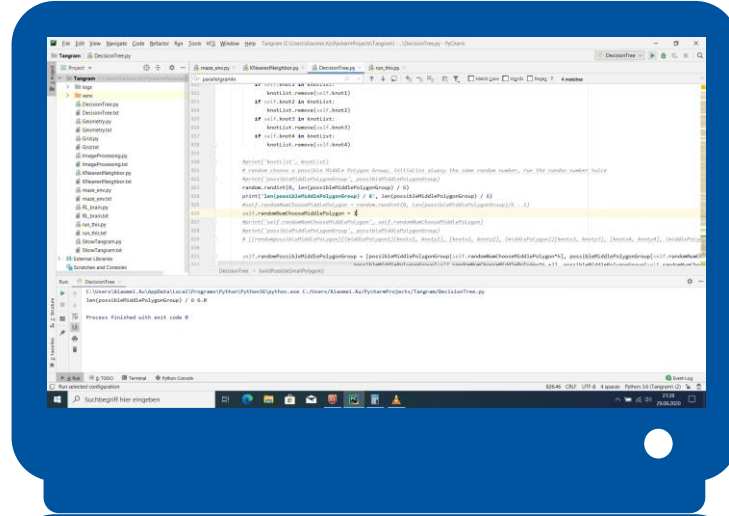
$Z' + 50\text{mm}(\text{not rotate})$

$Z'' + 50\text{mm}(\text{not rotate})$

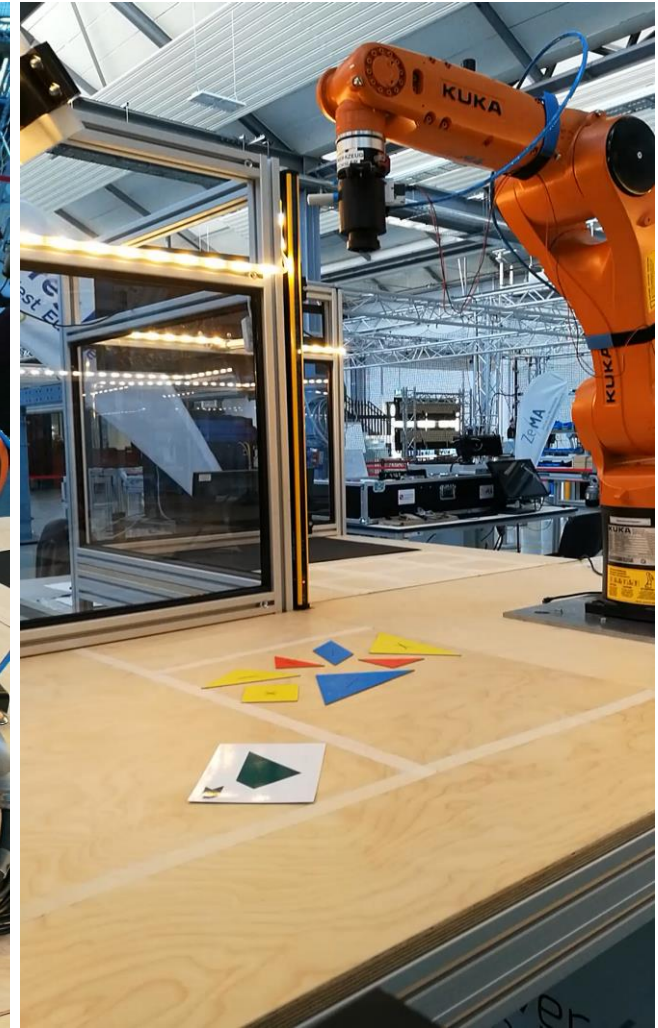
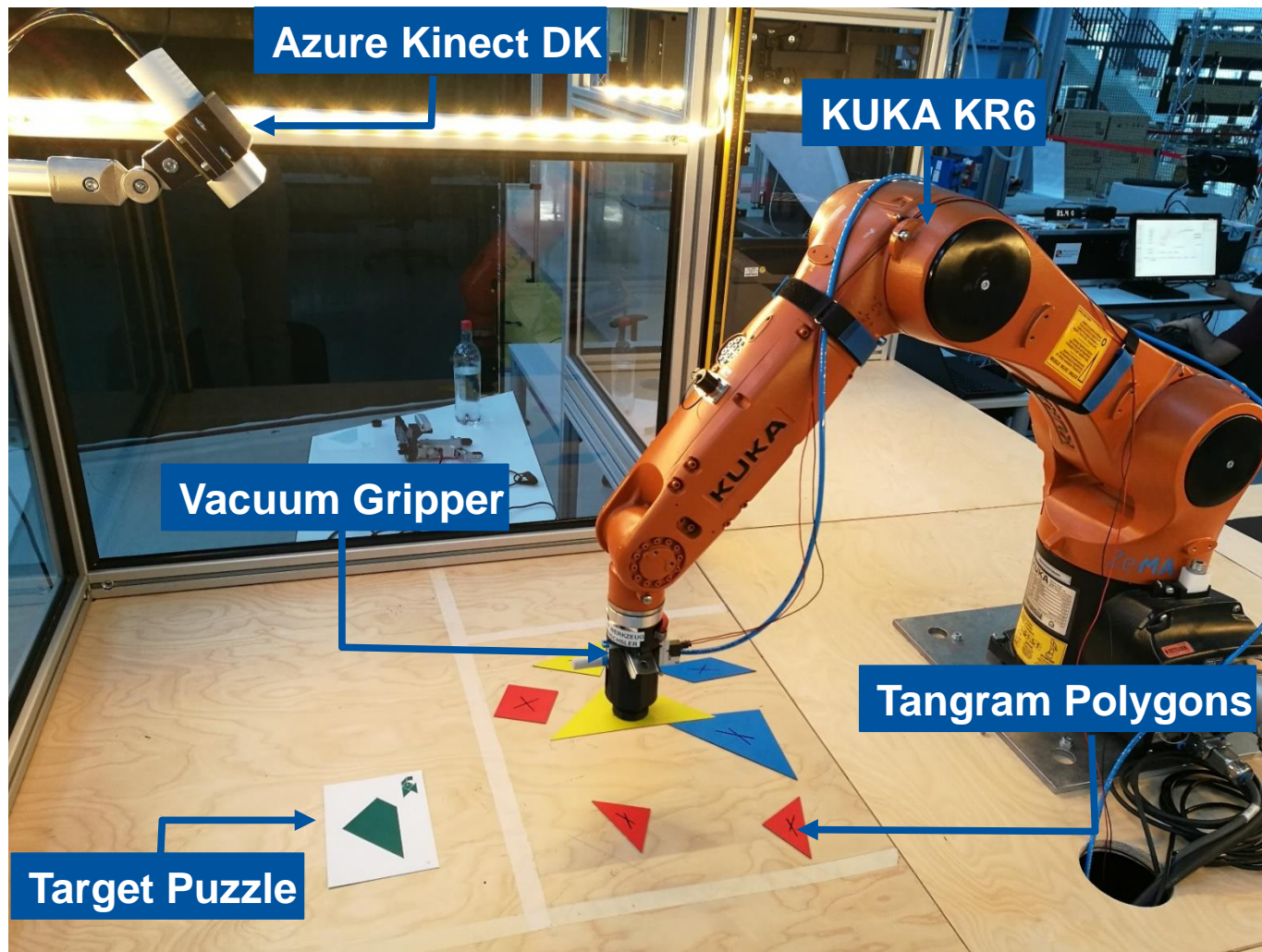
Agenda

- 1 Artificial Intelligence
- 2 Machine Learning
- 3 Reinforcement Learning
- 4 Tangram Rule
- 5 Structure of intelligent Robotic System in Tangram
- 6 Camera System
- 7 Intelligent Robotic System
- 8 Path planning
- 9 Implementation

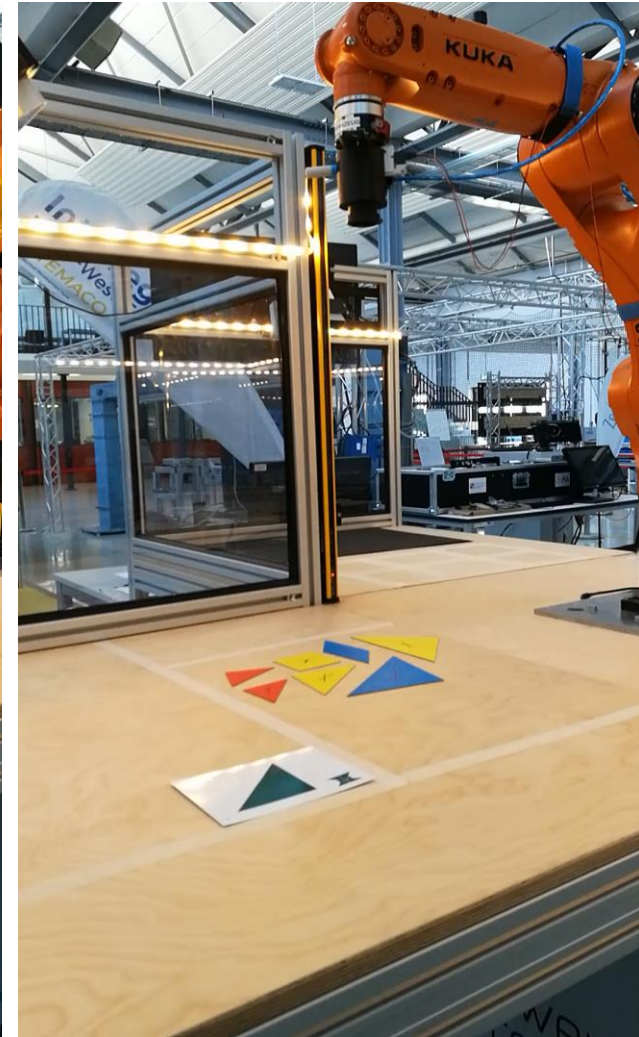
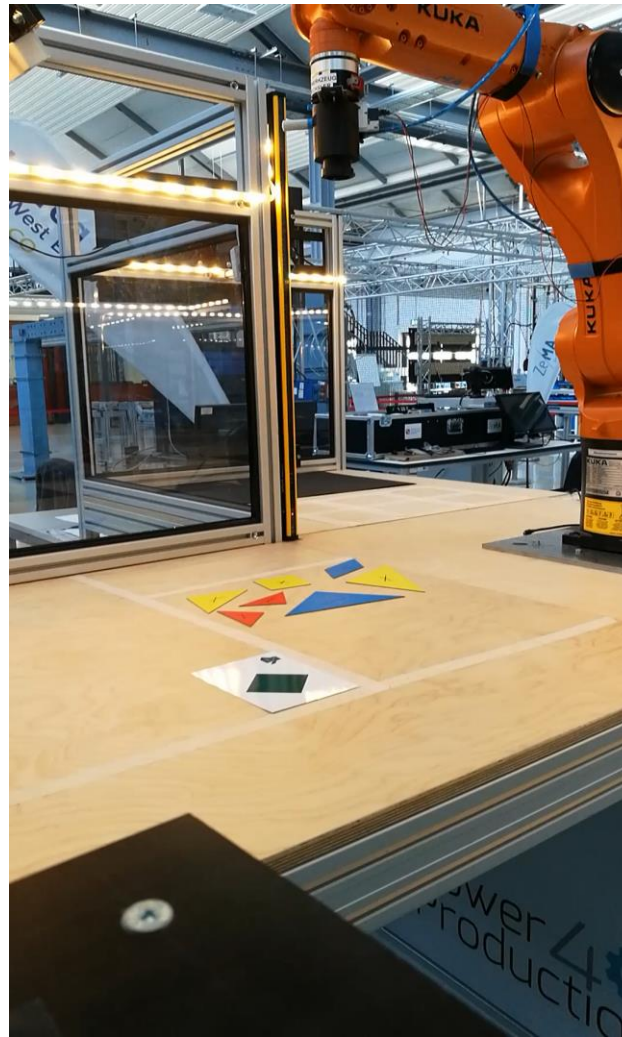
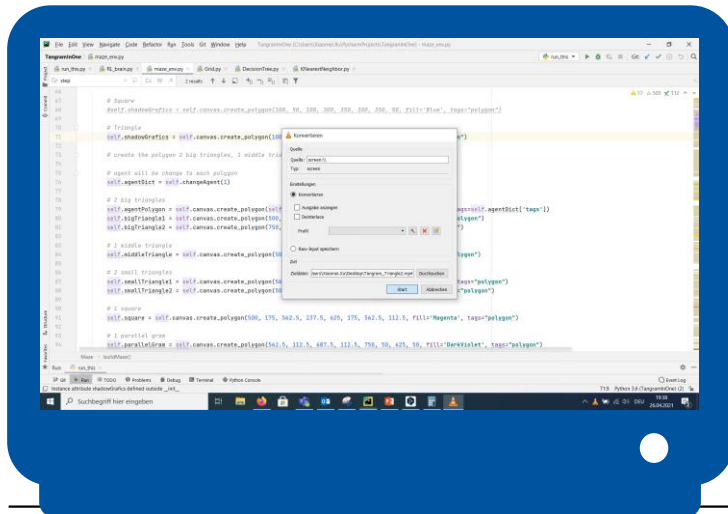
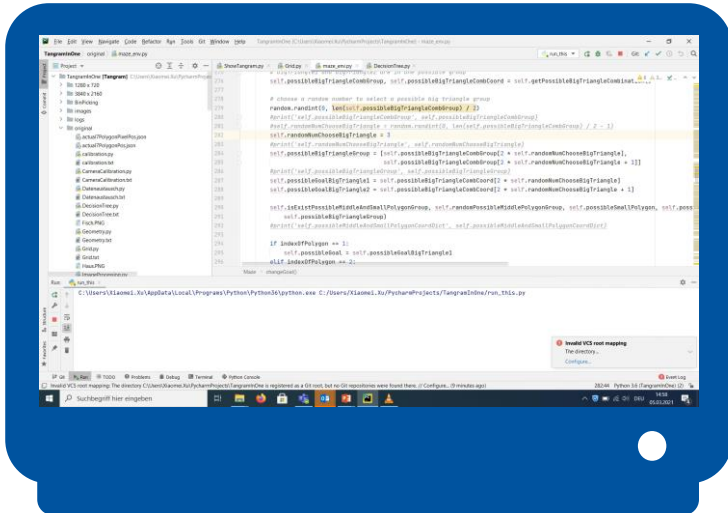
- AI programme finds all possible solutions in the Tangram game

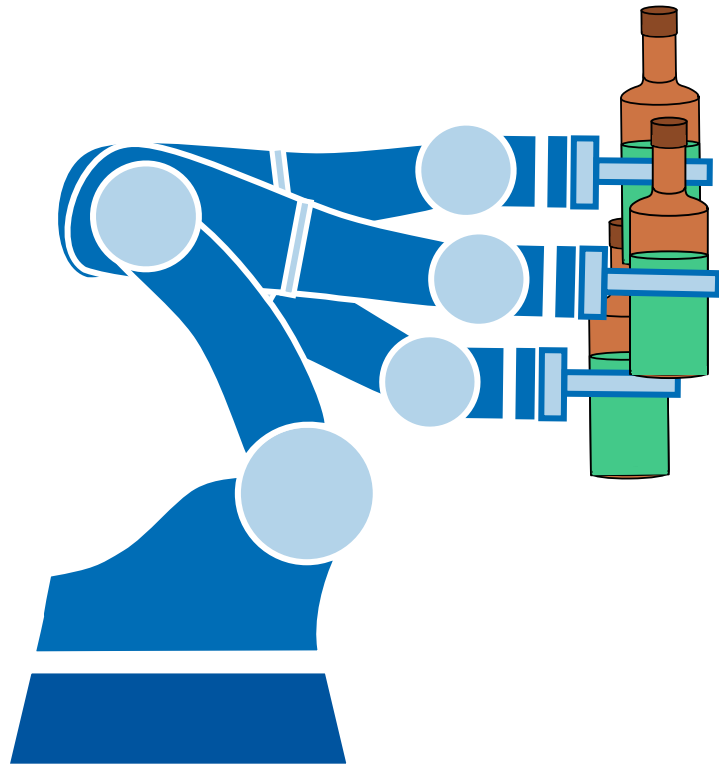


Experimental setup



Extension in the Other Shadow Graphic





Thank you for your attention!

Roadshow - Xiaomei Xu

E-Mail: xiaomei.xu@zema.de

Overview of the different methods of machine learning


Machine Learning		
Supervised Learning	Unsupervised Learning	Reinforcement Learning
Classification	Formation of affiliation	Interaction with environment
Regression	Categorization	Reward principles

- There are three classes learning supervised learning, unsupervised learning and reinforcement.
- The supervised learning is mostly used for classification and regression.
- With the methods unsupervised learning methods, existing patterns and groups within the data within the data and the individual data points can be assigned to these groups.
- Reinforcement learning is based on the principle of reward and punishment.
- In Figure shows an overview of the three principles. This action guide deals with reinforcement learning.

Reference: Einführungsstrategie für Reinforcement Learning in der industriellen Praxis

Application of machine learning in the various fields of automation with different levels of complexity

	Process Monitoring	Process Optimisation	Process Control
provides	Situational awareness and predictive information	Planning and decision support	Automated response to changes in the environment
offers	Increased quality, reduced downtime, reduced shortages	Increased efficiency, improved utilisation, greater yields, more effective design	Increased production and productivity, lower labor costs, less waste
needs	Data sources e.g. networked sensors	Process Monitoring + sophisticated analytical tools	Process optimization + Integration of the physical systems, e.g. robot
Methods	Visualization and descriptive statistics	Supervised und unsupervised Methods	Reinforcement Learning

Complexity 

- The area of **process monitoring** benefits directly from the increasing sensor equipment of production plants.
- This type of **process optimization** offers companies great potential in the form of increased efficiency and cost reduction.
- The complexity increases further when a process via machine learning methods, since the optimization and the execution on the physical system are very much intertwined. These steps in the **process control** are carried out alternately or simultaneously.

Reference: Einführungsstrategie für Reinforcement Learning in der industriellen Praxis