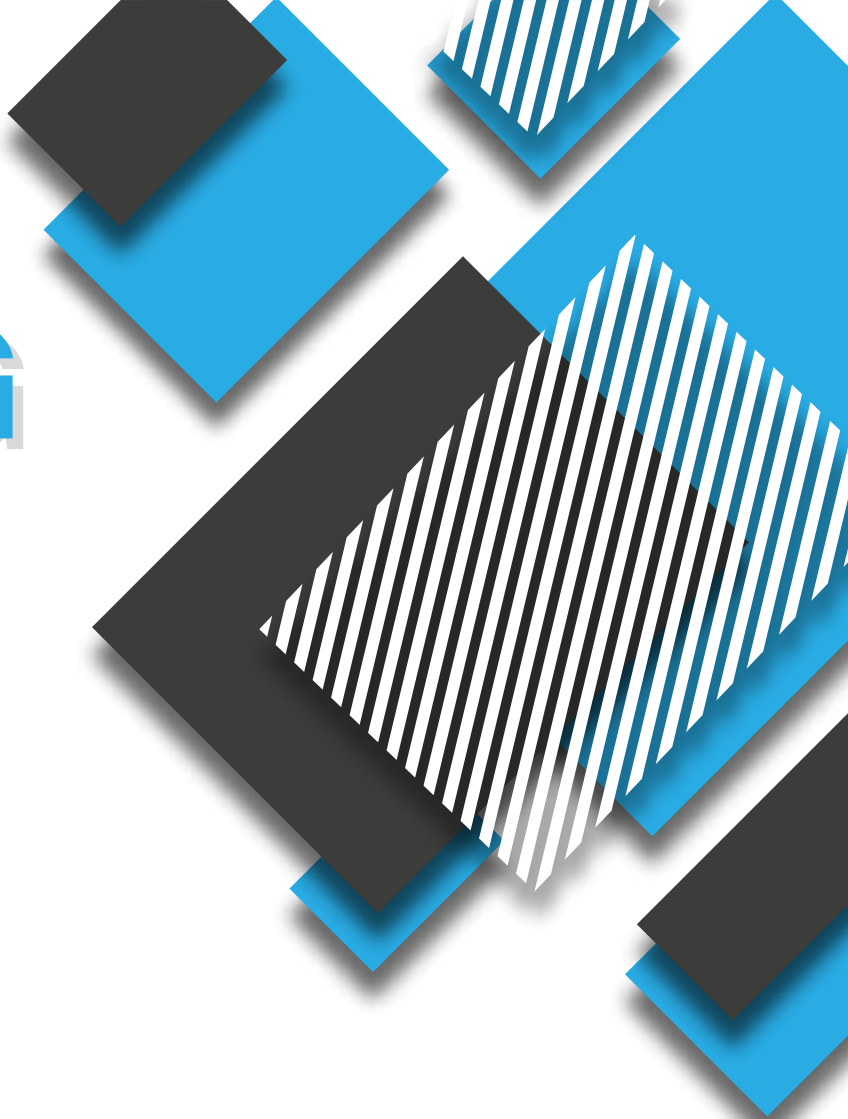
BIN PICKING

pocket reference

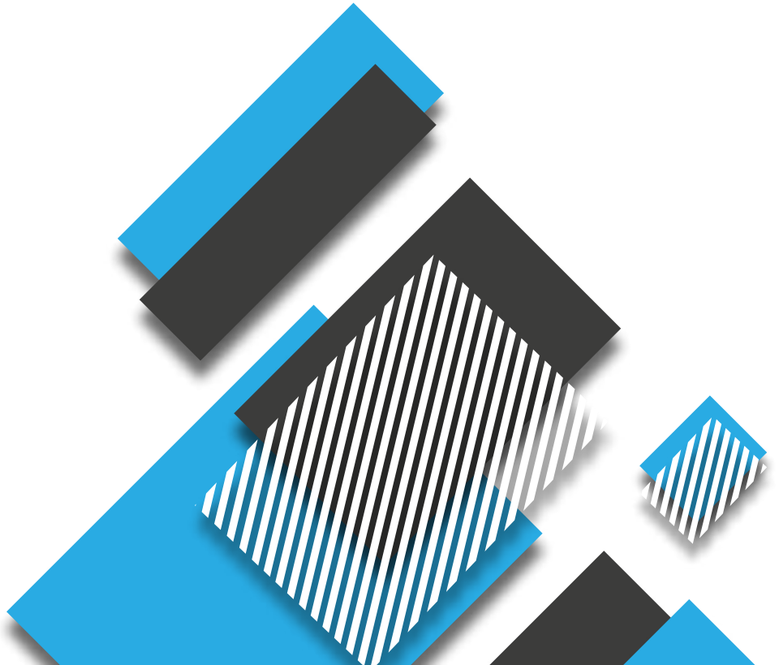


Roberto Polesel

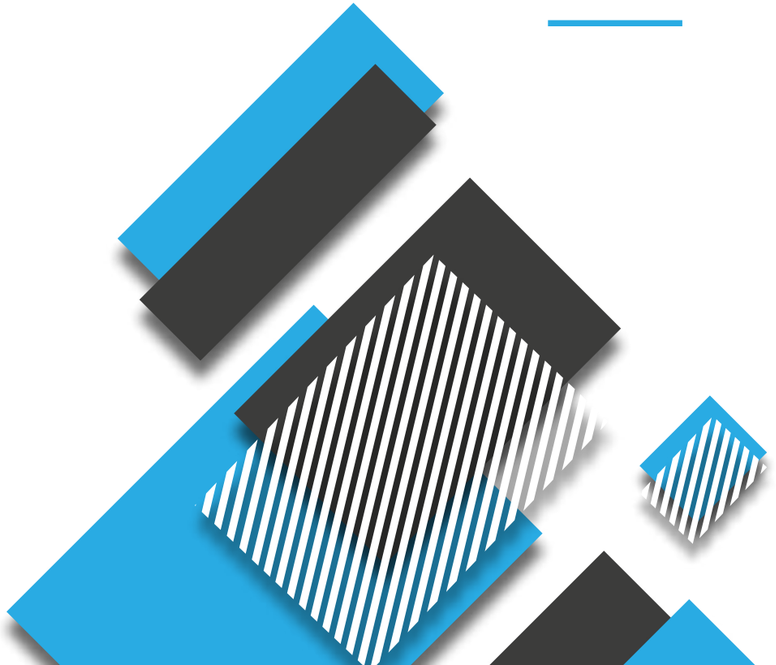
Euclid Labs
3D Vision and Robotics

BIN PICKING
POCKET REFERENCE

Roberto Polesel



CONTENTS

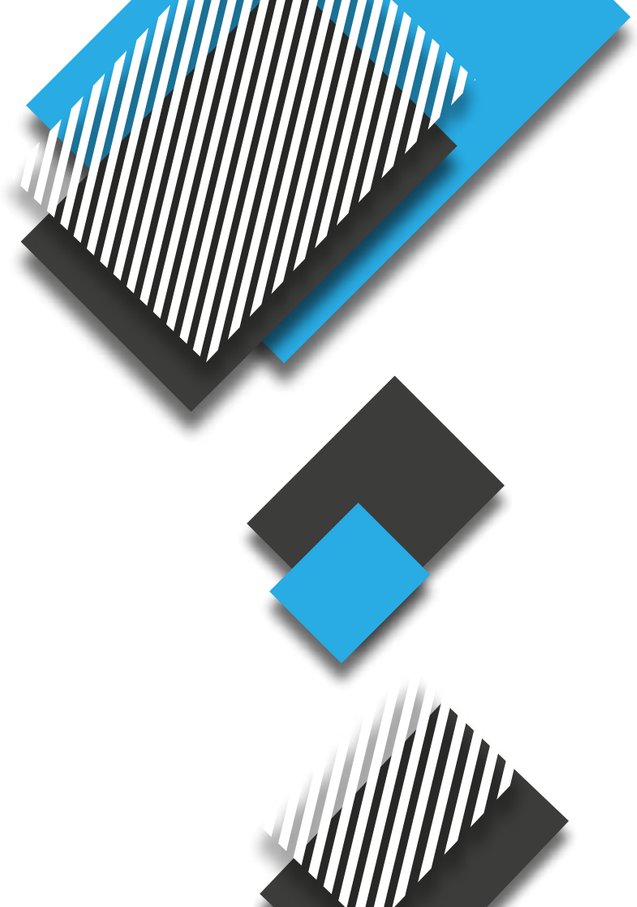


1. Introduction

This small book is a first attempt to design a kind of simple map of bin picking applications. It's not going to be a full compendium and there is no attempt to describe a general way to solve all situations. I just try do draw a simplified, schematic and out of scale map of this still little explored part of the automation world.
You may think such a thing to be useless but the most reproduced (and used) map of the world is exactly like that: the London Underground Map!
I don't expect to reach the enlightenment level of Harry Beck diagram but I will try to make a first step in that direction.

A quick Google search leads to some dozens of bin picking solutions, mostly claiming to have finally solved "the bin picking problem" and often just launching a new "groundbreaking" (before 2010) or "disruptive" (after 2010) product. Anyway, with a second search we can find hundreds of proofs that Earth is flat.
In the real (and pretty spheric) world only a few companies have succeeded in delivering a significant number (let's say more than fifty) of industrial installations and some of them are still not profitable.

After more than two hundred installations in ten years with Euclid Labs what I can still see is that bin picking growth is limited by poor knowledge of actual possibilities and that the most critical task for a successful result is the initial design of the project.
As it happens every time with rising technologies most of current users are approaching bin picking for the first time. What I write here is a little tool to help beginners figure out what questions to ask their supplier and how they could arrange their organization to take back the maximum advantages.
There is for sure some naive reasoning behind any high expectation from technological development and the core idea of this one can be summed up in this way: if you give a robot 3D vision and general grasping capabilities than you can replace any human labour at low cost and get rid of a lot logistic issues.
In real world internal logistic is a quite rigid characteristic of a modern factory since many years of development were spent to optimize it. Usually special fixtures and custom pallets are used for multiple products so adding a bin picking system to serve only a portion of them will

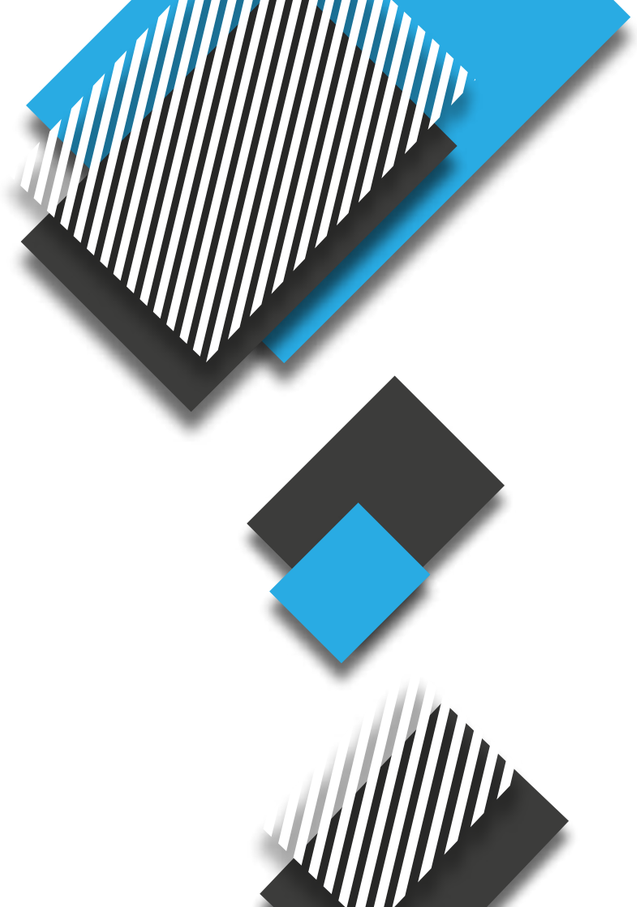


not reduce that side costs in the short term. It could also be that custom pallets are used to carry some of the production information (via RFID for example) or that some activities (usually parts inspections) are completed while preparing them and so changing handling systems requires to rethink all those processes.
Customer may be missing models of parts in intermediate steps of production and may consider this model creation(even if performed by same 3D vision system driving the robot) a big obstacle. Often automatic loading is the only missing step in a production line, but there are external limitation hard to walkaround, for example there could be no room enough for a robotic system that can fit demand.
All this will slow random bin picking introduction for a long time but I still think its market grow is never going to stop.

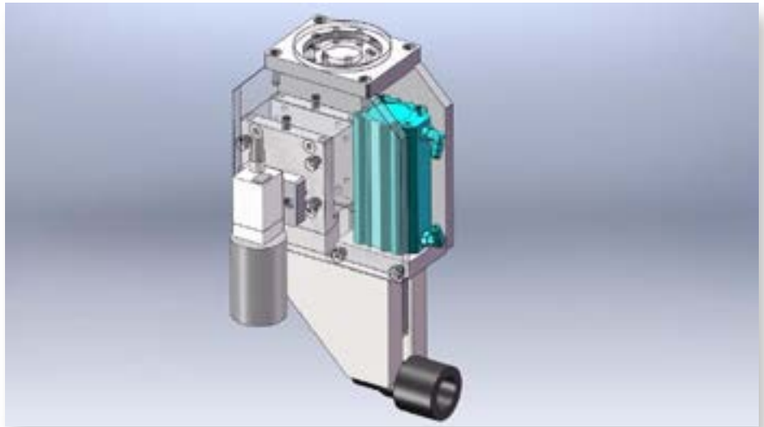
Some OEMs have already introduced bin picking as a standard way to load their machines. It happens mostly with new products because OEMs see bin picking as a strategic way to help customers receive their technology.
In some applications bin picking has positive side effects:

we have seen how a single small robot cell can successfully load two independent turning centres with a footprint that use room in a very efficient way.
As more working cells are going to be running around as more knowledge about real advantages of specific approaches is going to be common and customer demand is going to be clear.
I expect a lot of OEM will add bin picking as a typical plugin for their machines in next four years since end users are going to ask at least to be “ready for” it. I am not sure we will see a tipping point after which random bin picking is going to be mainstream and I think there is no way we can know it in advance but for sure it is already time for manufacturing companies to evaluate it as a valuable part of their internal logistic.

I have explicitly avoided any reference to commercial products, from Euclid Labs or any other vendor, even if in some cases it would have been helpful to explain differences in between similar problems. Loyalty in young technological companies is very close to faith in brand new religion and I am not looking to start esoteric discussions.



I hope that all the industrial ecosystem involved in this field (3D vision sensors, robots, grippers, software) will cooperate to enhance the awareness and further evolve the bin picking role in modern production lines.



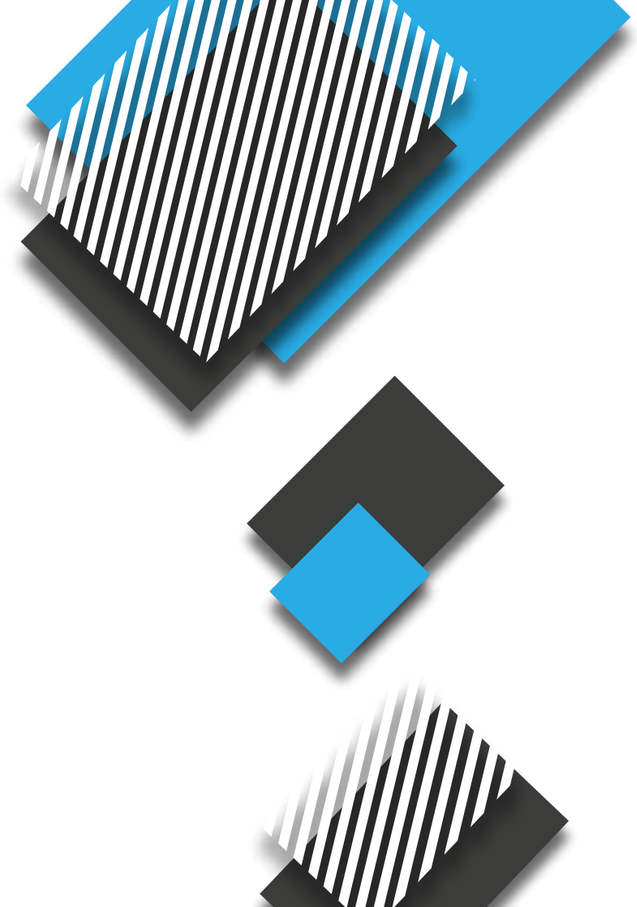
"I can't understand why it's so expensive: all you have to do is to pick a part from here and place it there".
The CEO of a company manufacturing industrial shelves told me this while evaluating a bin picking application on January 20 2009.
He didn't buy and went bankrupt in 2012, failing to understand that all a company has to do is to create value for customers and get money for that".

2. Definition of bin picking problem

Strictly speaking the bin picking problem involves the pick of a part from a box where many items are mixed. In an industrial application anyway we have to extend normally the analysis to the subsequent placing of the object somewhere else. Placing can have a really critical when analysing the project since it may add constraint in gripper design or create a deadly difference in cycle time.

It is common to refer to “Robot Vision” by Berthold and Horn (MIT, 1986) as the beginning of any discussion about bin picking in industrial applications.
Since then, often announced as an incumbent revolution, robot bin picking for a long time has been limited by the lack of 3D sensors with the necessary resolution and speed, computational power and experience in solving the pose estimation and path planning problem.

A bin picking system is build by providing a robot workcell with a. a sensor to build a 3D image b. a software that finds parts in a 3D image, calculates a safe pick position if it exists and plans a path there avoiding collision c. a gripper that can reach parts in a sufficient number of poses to ensure the box can be emptied d. a robot arm to perform the path.



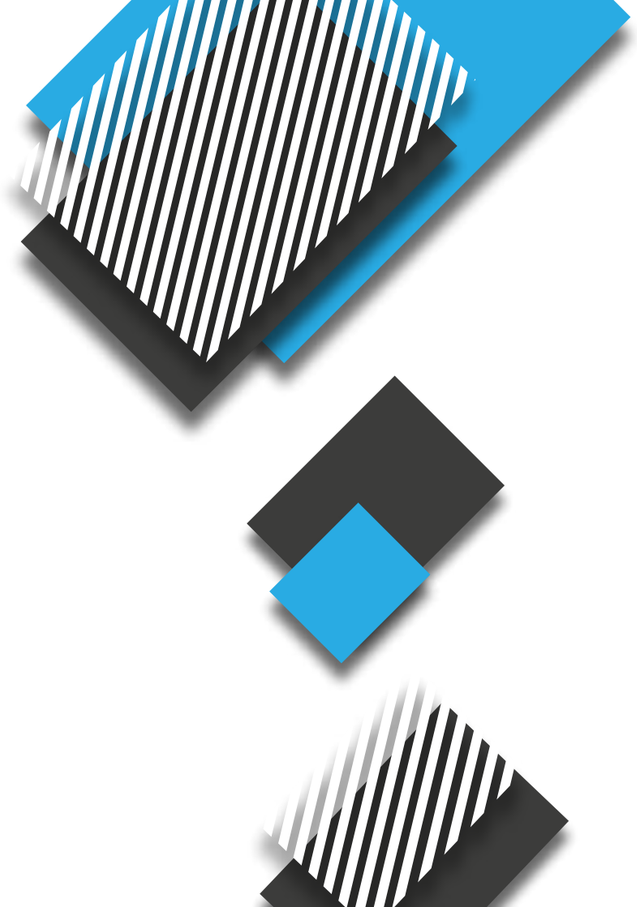
It is evident that while software is, at least in principle, universal and a sensor can address a really wide range of parts robot end effector has usually
to be designed each time. Robot grasping with contact compliance is still not evolved enough to allow the use of hand-like end effectors at real application speed. For sure this need for customization can reduce the flexibility advantage of a bin picking system compared to other loaders but if this is a main reason of slow market growth we should see a wider range of applications in cases where gripper device is easier to design.
For example many classes of machines are loaded with quite symmetrical objects (as billets in an oven for hot stamping or parts for turning centres) and it is possible to create a general gripping system for them.
A second clear example is a system that has to load a limited number of objects (sometimes only one) for its complete lifetime.
All semistructured bin picking have also easier gripping requirements since only a very limited number of poses

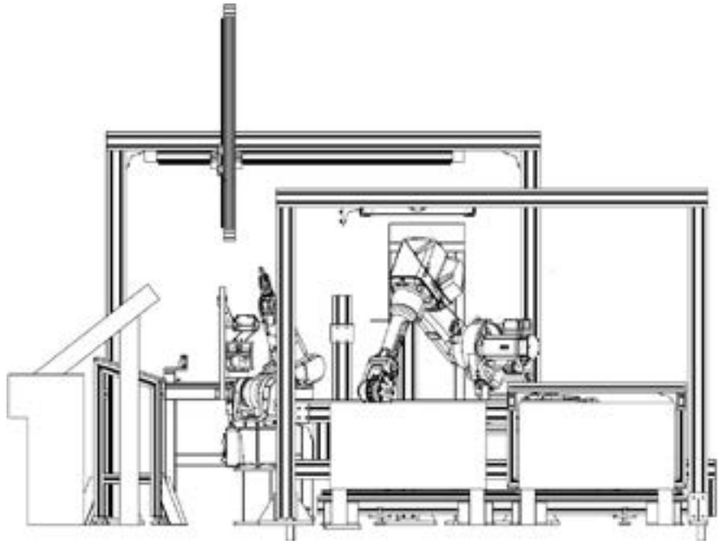
are going to be handled.
So there is for sure a market much larger than the one currently addressed even giving a maximum weight to the end effector design challenge.
We will take a look in next chapters to each component: **vision sensor**, **gripper**, **robot**, **software**.
When competing on complex task with humans machines have a hard time keeping same efficiency, in bin picking mayor limitations are:

1. **effectiveness in emptying the box**
2. **cycle time**

It is generally impossible to ensure that all parts are going to be picked unless a high level of complexity is added to the end effector and time cycle could be limited by the need to change gripping position.
It is true also that for a large part of the electronic and plastic industry often it is hard to fullfil these requirements but in the metal industry current typical average cycle time of 6 seconds is more than satisfying.
We will go through cycle time and its consequences in a specific chapter.

A little space will then be dedicated to the thorny topic of error handling. At the end I will define a few index to classify bin picking applications in different classes.





"A first quality classification of bins for robot picking may be: candies tray, troubles barrel, Schrödinger box, Pandora's jar".

https://www.euclidlabs.it/

3. Bin

Bin important properties are dimensions, shape and structure.

Dimensions are obviously critical in defining sensor field of view but their proportions may be really important in defining the gripper shape. This argument will be explored in the chapter related to gripper.

Shape of bin is often forced by customer existing internal or external logistic.
The usual shape is a complete box with all four side coming up at 90 degrees from base.

In some cases there are no side walls. From the bin picking application point of view there are many advantages using boxes with open angles: parts are easier to reach, pick path is faster and safer, wall shadow is smaller or even

missing when scanning with camera and laser.
If bin shape can be selected it should be done by carefully matching it to the best option for sensor and part extraction.
Bin structure is the most characteristic property and it can be roughly be defined as structured, semi-structured or unstructured.
A structured bin is basically full of palletized parts that are in a position known with a small error usually only in a couple of directions.

The robot can pick each part of a structured bin with the same gripper configuration since part has only few (usually one) poses. It has to be clear that the property has to be constant along the process so that removing properly parts has no effect on others' pose.
Often cardboards are separating layers of structured pallet. When this happens an efficiency of 100% in emptiness is implicitly needed since trying to remove a cardboard with some parts on it can lead to big damages.
Parts in semi-structured bins are distributed with more freedom but with the essential detail that all necessary robot configurations are predictable from the beginning.

Semi structured bins are sometimes the result of a structured bin collapse, because of an intrinsic instability or in consequence of external forces such as transport.
Unstructured bins are the consequence of a random placement process. This is the classic, I would say iconic, case of random bin picking.
The real number of poses to take into account while designing a gripper is hard to calculate and so the poses' statistical distribution, which has often to be determined to calculate the average cycle time.
One parameter to estimate is the total number of parts that are available for picking at each scan (supposing an universal gripper).

At least in principle all robot trajectories are going to be unique, it is so impossible to test the robot behaviour in all cases.
If box volume is large compared to robot range this implicates that a random bin picking software has to support strategies to avoid singularity and walk around all possible robot limitations.

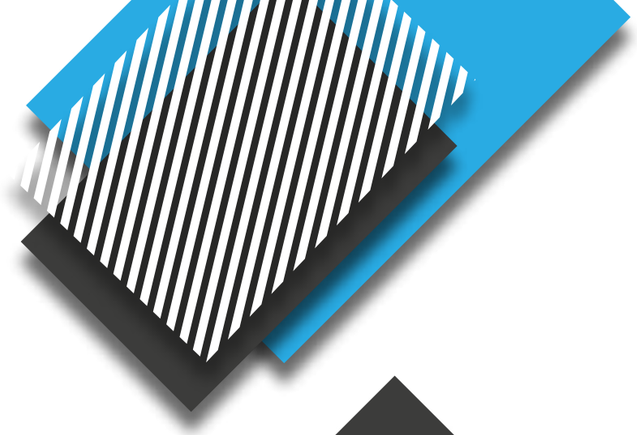
https://www.euclidlabs.it/

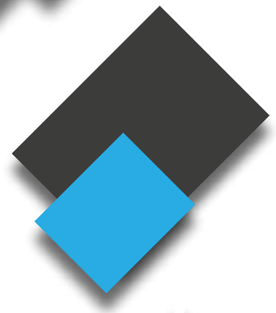
"I though bin picking was about picking bins". The honest comment of an American friend I was introduced to as "bin picking expert" after watching one of our application video".

4. Part

Part shape and weight has a direct impact on gripper design. Bin picking doesn't allow to define the area around parts, so as a general rule gripper has to pick in a reliable way using minimum volume around the target. So parts that are naturally creating an empty volume around themself, or inside themself, may be easier to pick than parts that are easily stacking even if at a first look this is unexpected.

This is one of the reasons of wide use of magnets in random bin picking. Often a part shape has in itself some natural gripping points (holes for example).



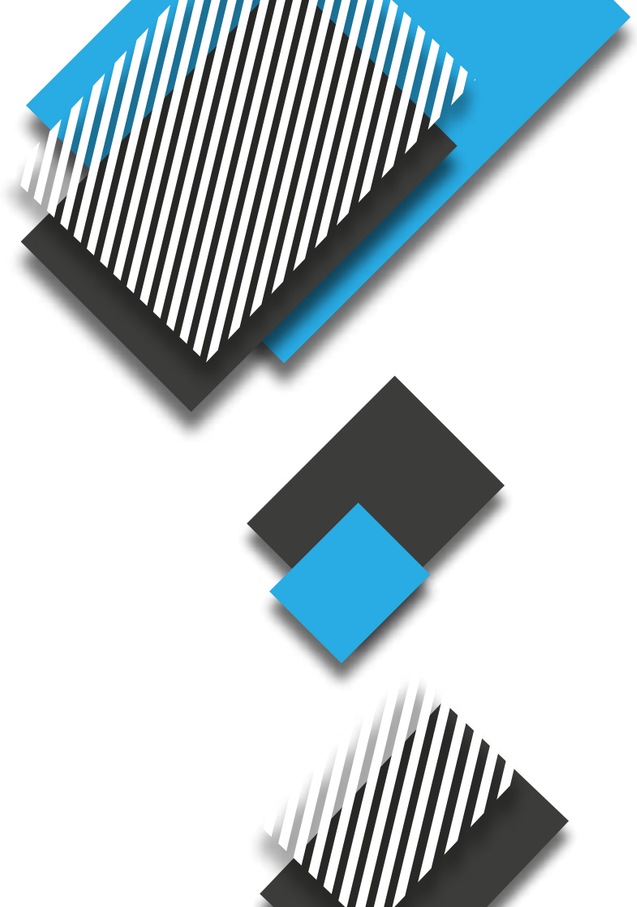


Part shape may also cause hooking of parts among themself. This is a geometrical problem that can not be fixed by the bin picking process itself, usually it is necessary to adopt a two step project: first pick and place on an intermediate station were parts occasionally linked are separated by robot or a mechanical device.
Part surface may be critical for the 3D sensor. Three main problems may rise: a black surface that does not allow a 3D reconstruction, a shiny surface that creates to much noise, some transparent side that is invisible to the camera.
The part may also have a significant detail (usually not visible in a point cloud) that has to be properly aligned on the place station; in this case a second vision system has to be added. Part dimensions are defining the resolution needed from the 3D sensors. In some cases there are not suitable sensors to pick small parts from a large bin by scanning the complete volume, the required number of points to obtain such a result is just too much. A solution with multiple devices may be adopted (a low resolution system for the complete volume and a high resolution sensor on robot hand for the single part).

"Videmus nunc per speculum in aenigmate tunc autem facie ad faciem (1 Corinthians 13:12). Paulus description of our limits in knowledge of God always sounded me better fitting our knowledge of the world we live in. Somehow he is saying that at now not all functions are computable with a Turing machine".

5. 3D sensor

The general output of a 3D camera is a point cloud.
Current 3D sensors are usually divided in four groups : stereo cameras,structured lights, laser triangulation, time of flight.
A stereo cameras system builds the point cloud by matching features in two or more images acquired from different points of view. I heard often this reasoning: "Human is the best implementation of intelligence created by nature. Human has stereo vision. So the best way to implement 3D vision is stereo system". If this was a good syllogism we should have airplanes flapping their wings like eagles.
Structured light is the process of projecting a known pattern (often grids or horizontal bars) on to a scene to record the way that these deform when striking surfaces and calculate the depth and surface information of the objects illuminated.
In laser-based triangulation systems, a narrow band of light projected on a 3D surface produces a line of illumination that will appear distorted from an observation perspective other than that of the projector. By collecting images of distorted laser line while moving the complete system or one of the components (with a linear movement or a rotation) a complete point cloud can be mounted.
Time of flight cameras are really measuring the distance of objects in a specific direction by sending some kind of wave signal and measuring the time needed to bounce back. Each approach has several different possible implementations with unique features, mostly related to behaviour on dark or reflective objects.
It is pretty straightforward that it's easier to see dark object when the light source is opposite to the camera, so that light reflected by the object is maximized. This explains why structured light vision with a large baseline are in general better at this task.
When parts are reflective best condition is the opposite. Each configuration leads also to different resolution and different need of camera field of view.
Following schema shows a first approach classification of 3D sensor features depending on functioning principle for commercial products.
It is a rough schema that I think helps to figure out what type of sensor main part of applications are going to use, some specific implementation of a principle may be classified



differently but it's a less common approach usually for cost's reason.
I have also to specify that I am referring to application with quite complex surfaces, limiting the evaluation to specific fields could lead to quite different results.

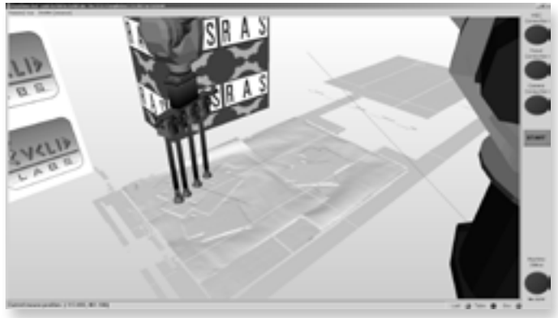
	Stereo vision	Structured light	Laser triangulation	ToF Cameras
RANGE	medium	medium	low/medium	high
RESOLUTION XY	medium	high	high	low
DEPTH ACCURACY	medium	high	very high	low
3D FRAME RATE	high	medium	low	high
DARK OBJECT	low	high	medium	low
SHINY OBJECT	low	medium	high	medium to low
GLOBAL COST	medium to low	high	high including the moviment cost	medium to low

https://www.euclidlabs.it/

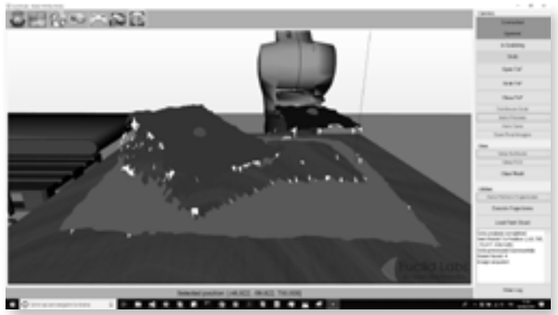
Optical image

Laser triangolation point cloud

Time of flight point cloud



A laser triangolation point cloud that allows a perfect match of metalsheet parts



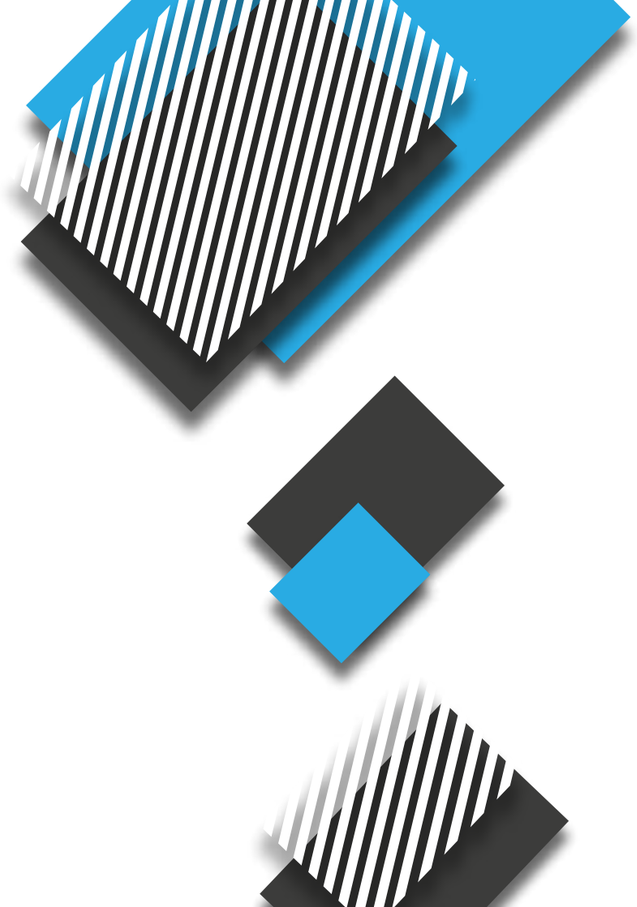
A time of flight point cloud to pick boxes

https://www.euclidlabs.it/

A point cloud generated by a structured light sensor

A 3D match on a ToF point cloud

Should the sensor be on top of the box or on the gripper? Advantages of having the sensor on robot end effector : accuracy (smaller calibration error, better image fit), costs when many pallets are used, freedom to build a point cloud from multiple acquisitions. In some cases the part to pick is so big that only moving a sensor using the robot a significant point cloud can be used. Disadvantages of such solution are : minimum cycle time increased to stop the robot for scanning, higher volume of robot end effector, which means more limits in picking inside the box, risk of calculatin trajectories without considering possible collisions with the part of box outside the field view.
In my experience if something gets broken it's on robot tool (and usually because of a manual operation) so I really try to avoid placing there the most expensive piece of hardware.



"The art of reading grippers to understand a project outcome should be much more popular than the art of reading palms".

6. Gripper

Vacuum systems, magnets and mechanical gripper are all suitable for bin picking application and in some cases the could be used simultaneously.
A bin picking gripper has to face some unique challenges:

1. **volume of interference has to be minimum to reduced the probability of collisions with surrounding parts**

2. **end effector has to reach all box corners with the largest freedom of orientation**

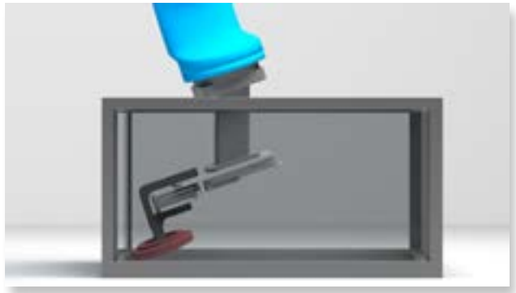
3. **there should be a way of picking for all possible poses**

A further constraint comes from the deposit configuration: gripper has to match it or time cycle has to be sufficient to allow a tool change.
Failing the gripper design will lead to deadlock conditions where parts are localized but there are no available picking positions. If this conditions happens only on the low corners it may be an accettable limitation of emptiness efficiency.

The gripper in picture could be perfect to pick from conveyors but has clear limitation when picking from boxes. Even if dealing with short box walls and a small angle the gripper quickly ends in a collision.

A first improvement could be done adding an offset to gripper fingers.

With higher side walls or a just smaller change of angle this gripper design is not going to pick parts.
The basic idea may be somithing similar to the following one.



It is easy to figure out some limits even when end effector has a proper offset and it is constructed as thin as possible.

It should be evident that also if part is simple a good emptiness result requires a device capable of picking effectively from multiple points. In not trivial applications a good gripper

design is in almost always a task more complicated than software startup.
At least in principle robot could change tool in between picking different parts, but this usually adds too much costs and increase too much cycle time to be an effective possibility.
A second alternative could be to shake somehow the bin to flip parts that are impossible to pick, miming a technique common in 2D pick and place applications. This is obviously limited to light objects.

Magnets, when dealing with suitable objects, are by nature easy to use on different sides of parts. As a side effect anyway part position once gripped is not certain (the same may happen when using gripper fingers on some not safe clamping point, used only when no other option to empty the bin is left).

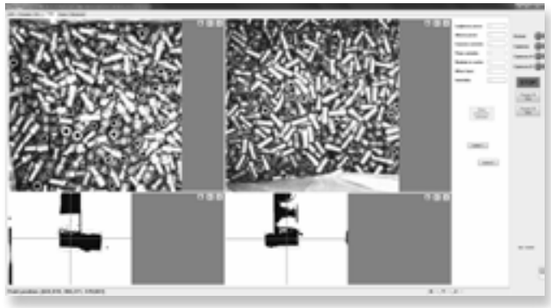
In this situation possible solutions are :

1. taking a second picture of the gripped part

2. moving to a regrip stand or centering unit before delivering to final station

3. adding some qualifying movement to the gripper or to the delivery station

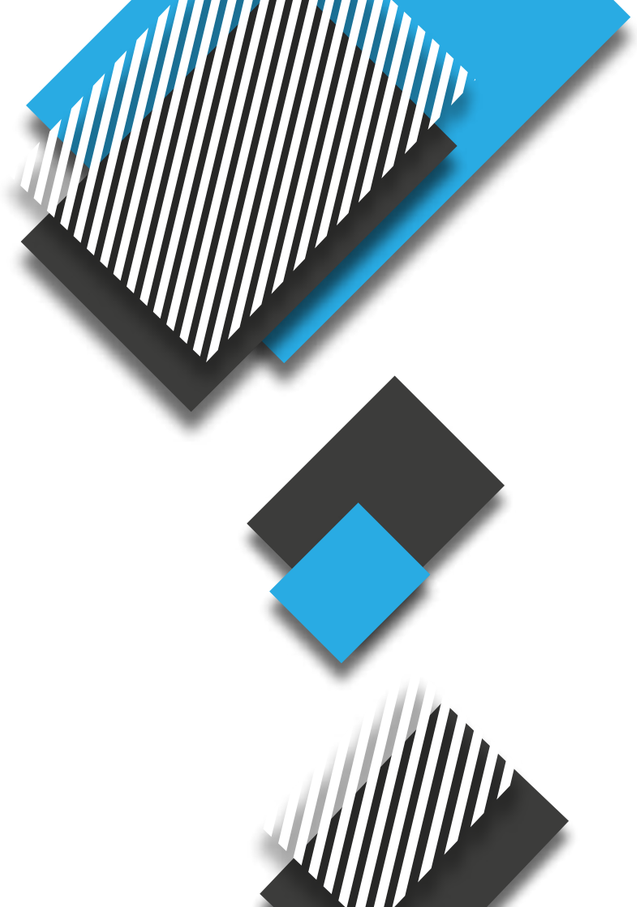
Both options have a negative impact on cycle time.



Previous picture shows an example of a second vision step used to understand the real positioning of an object on the magnet.

A second unique and usually underestimated challenge in bin picking gripper design comes from the fact that part support surface is always different and if objects are small they will be somehow floating when dealing with box top level while they

will be on a solid floor at the bottom.
As a result it may be necessary to push the part differenty along the bin to make a gripper device work in the proper way. This may be done by software or by building a compliant end effector.
Limitations of current grasping tools prevent the use of bin picking technology in all those cases where speed of process depends on taking a series of components in one shot and then releasing them one by one. With very light objects this is quite common.



"The God of Robotics shall send in singularity all robots with evil software".

7. Robot

Since I am generally referring to random bin picking robots involved are six or seven axis industrial arms, where the external axis could be part of the arm kinematic (as in ABB Yumi or KUKA Iiwa) or a linear unit on the base of it. Obviously any baroque configuration with six degrees of freedom or more would work, but for any practical pourpose we can ignore them.

Robot controller is required to receive a trajectory by an external device (a smartcamera or a computer) and perform it. All points of the trajectory are complete of arm configuration. For a basic application this would be enough. Each robot will then have its own unique performance in terms of speed and accuracy.
possibility to run a parallel task, which will allow for example to stack multiple trajectories in memory, and a good torque control system to skip collisions.
A bin picking system has to walk around all movement limitations of a robot arm, first of all singularities.
The most common is related to the alignment of axis four and six in a Pumalike arm when performing a linear movement.
If box volume is small the case may be removed with a proper gripper design or using a robot large enough to stay far from that configuration, but in general a definitive solution comes only from implementing in the software a good robot motion simulation and some strategy to walk around singularities by adding not linear movements.

"I confess that after reading all main classical and heretic books about software development I have concluded that there are only two cruel rules:

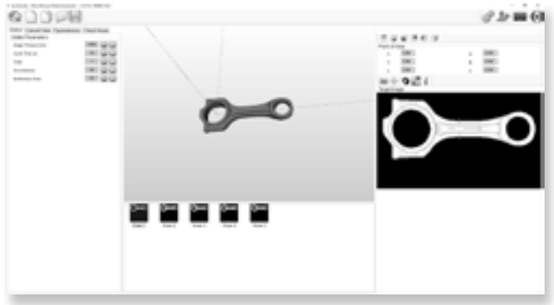
1. in the first stage of a project take the road that gives you more freedom to change later

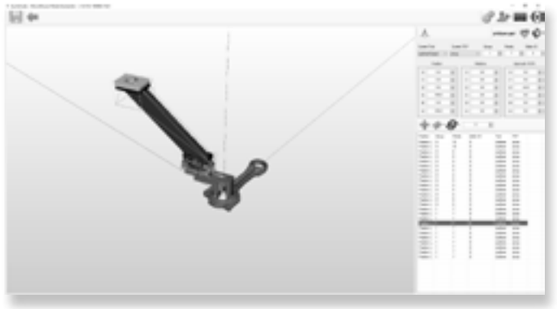
2. in the last stage of development take a cautious decision based on costs and benefits.

Software manager job is to decide when to switch from stage 1 to stage 2 and when to start a new cycle".

8. Software

Bin picking software has to purposes : executing bin picking (inline) and programming a new part picking (offline).
The new part programming software may not be necessary, a program could contain enough knowledge to recognize what to pick without introducing new information and, at least theoretically, could have enough rules to define which gripper to use and how to use it. In real applications anyway this is mostly related to very specific environments where a single versatile gripper picks each item at same way.
In manufacturing we can assume that starting point of each production is a 3D drawing and that a lot of design of the robot cell has been already done on top of that. So the best thing a bin picking software can provide is a simple user interface to load a 3D cad model, define vision parameters if needed, test them on live or saved images, define which set of end effectors to use and how graphically.





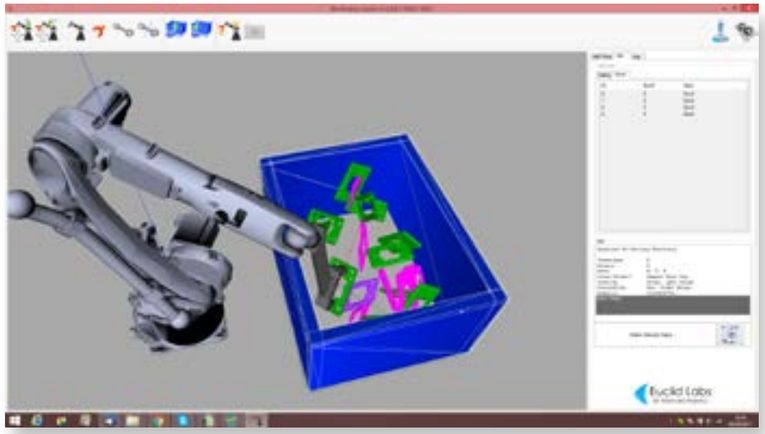
The inline software has to perform several tasks:

1. **configure the workcell (sensor, robot model, obstacles...)**
2. **calibrate sensor and robots to match positions (once)**
3. **load a point cloud for each bin used at same time**
4. **match 3D cad model into loaded point cloud**
5. **check that a trajectory to pick the matched model exists and send it to robot.**

A second piece of code runs on robot controller to execute the calculated trajectory.

Some criteria to compare softwares:

1. **how many possible picking position could be defined and how this affect calculation time**
2. **what level of collision check is implemented (only gripper or full robot and gripper check, only final position or complete trajectory)**
3. **how fast is the result sequence (if stored results are available)**
4. **how robot behaviour is taken into account (configurations, singularities, approximations)**
5. **how automatic the calibration of the system after some dramatic event is**



"My father has been for years a boxing coach. Once while training newbies on how to avoid a jab he was asked ‹‹Coach, what should I do if I get hit?››. I saw my father thinking for a second, not making up an answer but recalling to memory hundreds of fights, before saying with the certain tone of experience ‹‹Smile››".

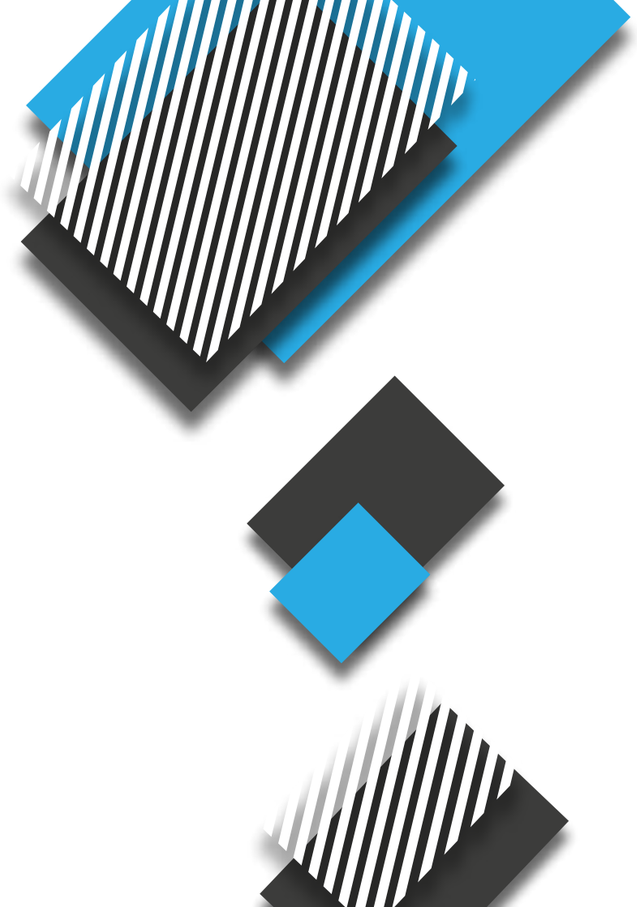
https://www.euclidlabs.it/

9. Error Handling

Collisions are an inevitable eventuality in a bin picking process for the obvious reason that 3D image and movement calculated on that image are happing at a time distance. This means that a movement of an object can not be excluded even assuming a perfect vision system.
The shorter the time, the less likely a collision is.

In many cases several picking positions are calculated from the same point cloud and used in sequence. This use of stored information is often a good way to improve cycle time but increases both the time in between image acquisition and execution and the number of external events influencing the bin status, since each pick somehow modify the complete system equilibrium.

If robot software or some gripper sensors are preventing damages to parts and robot arm itself than the quantity of information that can be stored from a single point cloud analysis can be optimized.
This is a statistical optimization and has to be evaluated on a large test set. It has also to be coupled with a buffer for extracted parts to lead to a real advantage.



"To explain customers how hard it could be to short cycle time of one second I remember them that to move the Olympic record of 100m from 10.6s ti 9.6s it took one hundred years (Donald Lippincott, Stockholm 1912 – Usain Bolt, London 2012)".

10. Cycle time

The easiest cycle would be:

1. **3D Scan Ts**

2. **Point cloud analysis and trajectory calculation Tc**

3. **pick trajectory execution Tr**

4. **gripper pick Tp**

5. **place trajectory execution (when robot exit the bin a new scan starts) Te**

Assuming that operation after the extraction are faster than previous part of the process cycle time would be Ts + Tc + Tr + Tp + Te.
All this parameters are depending on equipment (sensor, computer, software , robot, gripper) and part (scanning time is likely higher on black objects for example).

Tr and Te (time to reach the target and extraction time are usually different since robot speed when part is not attached can be much higher) are dependent on the level of bin filling.
In the case of no buffer this time has to be calculated in worst case, which is usually on a bottom corner of bin. If a buffer of length N is available than this time is the worst average on a sequence of N parts.

Once the robot brand and model are selected this is also the bottom line of cycle time (reachable with a gripper that works in zero time and all scan and calculation in hidden time).

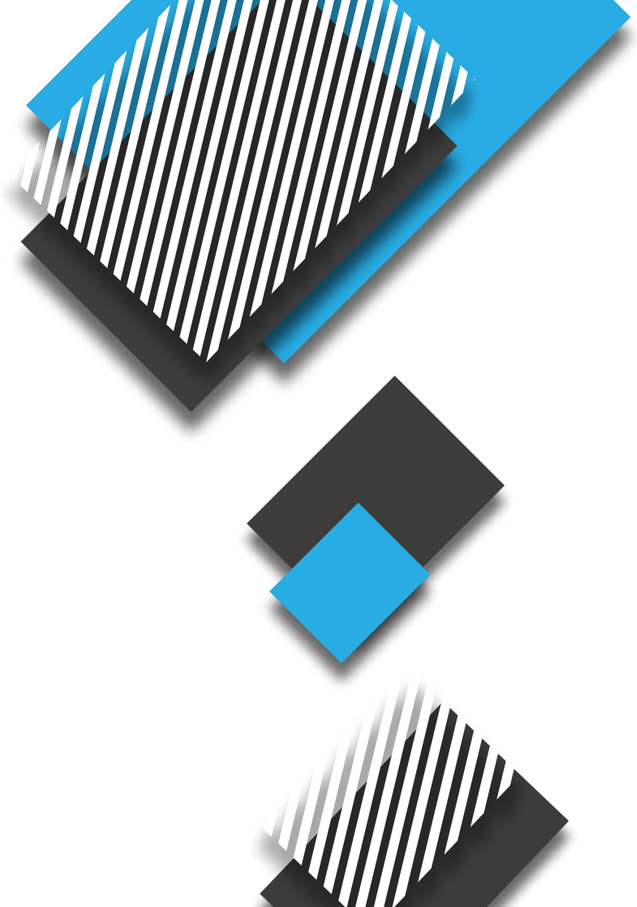
On a common bin as an europallett of 800mm height and with a 10 to 20 kg payload robot Tr+Te could be something between 3 and 6 seconds.

Tp, the time to grip a part, could be from 0.1 to 1 seconds

Ts depends on what type of camera is used. With a time of flight system Ts is less then 0.1 second but if a laser triangulation is used the vision system has to move over the pallet to collect all data and this is going to last easily more than 3 seconds.
Tc depends a lot on what is the global point cloud size, how much noise there is and so on. To make some example I will assume a Tc of 3 seconds for first part in a point cloud and 1 sec for all the next ones. I am simplifying here one

important thing: Tc depends a lot on how many parts are detected without finding a way to reach them. So it is coming out from a property of software that almost no customer ask about: what is the software success rate in elaborating first parts that are really going to be picked? Modern hardware that allows a wide parallel processing is reducing this variability.
In an example with a laser scanner and a mechanical gripper mounted on a quite fast robot we may find modeling

Ts	4s
Tc (first part)	3s
Tc2 (nexts)	1s
Tr+Te	4s
Tp	1s



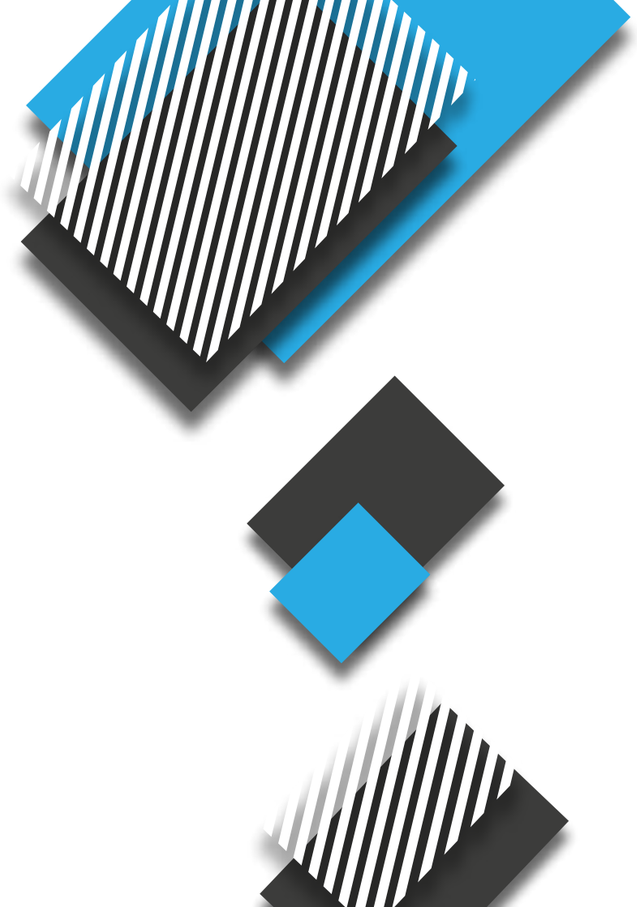
The global cycle time scanning before each pick is going to be 12 seconds.
There are two immediate ways to improve : use stored data (since Tc2 is less than Tc and there are going to be less scans) or use of two bins (under the same scanning system).
Let's simulate what happens in both cases. First of all it has to be underlined that if stored information is used a buffer is mandatory, since we are improving an average cycle time not the worst case.

By calculating 2 results per point cloud the result is:
cycle time (average on 2) = (Ts + Tc + Tc2)/2 + Tr+Te + Tp = 9 seconds

By calculating 4 results per point cloud the result is:
cycle time (average on 2) = (Ts + Tc + Tc2)/4 + Tr+Te + Tp = 7 seconds

To keep the example clear I am assuming that efficiency in detecting parts and picking remain constant while increasing the number of stored information used, which is for sure not true. Anyway it is usually close to that at

least when N is small. One more thing to mention is that, by writing a little more complicated robot code, it may be possible to start the scan while still using old data and synchronized with tasks outside the box, basically deleting Ts from the formula (or at least having part of it in hidden time).
Let's now look what may happen if the system scans (for example moving the sensor) two boxes. The effect is that Ts+Tc on one box will be parallel to Tr+Te+Tp on the other. This leads to a cycle time of 7 seconds in our example without storing data.
This example has the only purpose to show how without changing the basic technology very different achievements are possible by acting on logic and layout.
Since usually bin picking is used to load some kind of machines it is interesting to make a few considerations about the relationship between its cycle time (Tbp) and the one of the tended machine (Tm).
In actual applications Tbp is just a little lower than Tm or than Tm/2, which means it can be used to directly load one or two machines.



If Tbp is a lot less than Tm customers are today more likely to select a loading system from fixed positions since the total labour deleted by bin picking adoption would be little (as an example, if Tm is 12 minutes, a machine can work autonomously for 8 hours just loading 40 parts in fixed positions, which may take just a few minutes to a worker).
If Tbp is a lot more than Tm than several systems should be used to supply a single machine.
I expect that, while defining an internal efficient logistic system, the opportunity of using bin picking to prepare structured trays for other automated systems will be quickly more important thanks also to the progress of mobile robotics.

"Sometimes you do have to compare apples and oranges".

11. Some indexes for random bin picking classification

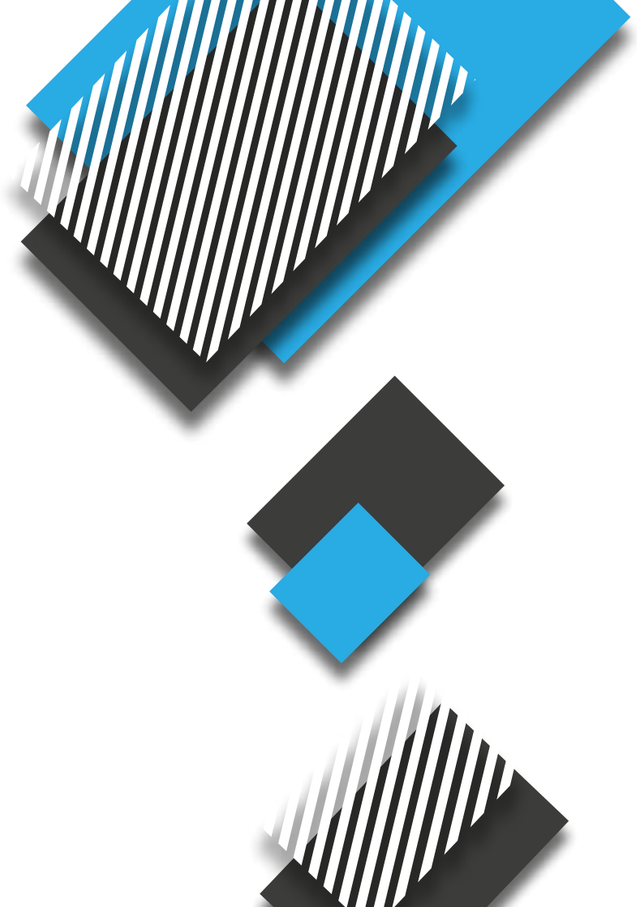
I have found very hard to compare bin picking applications just looking at part shape, dimensions and surface. This is enough to evaluate the vision challenge but not the real problem of maximizing workcell performances.
It is also hard often to predict how much installing a new version will improve performance of an old workcell (the same customer that tells us we will be required to do some small one day job with an uncertainty of three months is expecting a prediction of cycle time improvement with an accuracy of 0.1 seconds).

I am here just suggesting there may be some good indexes to classify bin picking applications, as an example four data that looks useful are:

1. **Te**= cycle time efficiency index, expected time cycle / base time cycle (the pure robot movement time in worst case)

2. **Vr**= volume ratio, part volume / bin volume

3. **Dr1 and Dr2** = dimension ratio, bin minimum dimension/ minimum part dimension and bin maximum dimension/ maximum part dimension



4. **Pr**= number of possible poses / number of picking positions (not counting symmetrical poses and positions)

Acknowledgements

While I have all the responsibility of typos, mistakes and incomplete data of this presentation I have to share credits for all the value you may find in it with all my colleagues of Euclid Labs.

You may submit any request or comment to

roberto.polesel@euclidlabs.it

Every two years, Euclid Labs is proud to organize an event in connection with Automatica's trade show in Munich. Everyone interested in advanced robotics, automation and computer vision can join, share and enjoy three talks about a crucial topic in an informal environment. This little book was written and realized for “The Future of Robot Programming” event on Tuesday, 19th June 2018. As CEO of Euclid Labs I take the chance to thank my Team that work hard to make the event successfull and all the participants that make this possible.