Ben-Gurion University of Negev
Faculty of Engineering Sciences
Department of Mechanical Engineering

SIMULTANEOUS GRASPING OF MULTIPLE OBJECTS

A thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science

by Yosef Ovcharik
Advisor: Prof. Amir Shapiro

July 2018

Abstract
kfkfasjkdklsfaja sdfl;afjl;asdf

Prepared by: Yosef Ovcharik_______________

Advisor: Prof. Amir Shapiro____

Acknowledgments

I am very grateful to my arms for being by my side, to my legs for carrying me, and to my fingers - I can always count on them.

This page intentionally left blank.

Contents

List of Tables

List of Figures

Task		**Status**
Preliminaries	*Background*	✓
	Related work	✓
	Definitions	✓
	Assumptions	✓
	Problem Statement	✓
Solution	*Description*	☐
	Limitations	☐
	Algorithm	☐
	Complexity	☐
Simulations	*Software*	☐
	Setup	☐
	Results	☐
Experiments	*Setup*	☐
	Results	☐
Discussion		☐
Conclusions		☐
Introduction		☐
Abstract		☐
Title		✓
Acknowledgments		☐
References		✓

Table 1.: Advancement.

This page intentionally left blank.

1. Introduction

1.1. Motivation [1 page]

Robotics become a vast field of research and applications,

- Technologies
- Robotic manipulators
- Grasping
- Automation
- Home Assistance
- Defense applications
- Remote robotics (space etc.)

1.2. Research objectives [.5-1 page]

While vast amount of research is done on grasping in different forms (manipulators, fixturing, first\second order, machine learning approach, visual assessment, restraint analysis etc.) little work was done on grasping/manipulation of several objects an once.

The main objective of the research is to present a solution for immobilization of several objects simultaneously, while using minimal amount of actuators (fingers). That can be achieved by analyzing the spatial relationship of objects and the contacts between them.

The research can be divided to sub-objectives:

Classifying the types of contacts between bodies.

Assessment of additional constraints needed to immobilize the set of objects

Deriving an optimal way to arrange the objects in order to minimize the amount of external constraints needed for the immobilization.

1.3. Research contribution and innovation [.5-1 page]

The research yielded solid and reproducible results of multiple object immobilization that can be integrated in various robotics systems.

The methods described are novel solutions for the problems yet to arise.

The research can be further extended to manipulation of multiple objects, interaction with soft objects / fingers and, of course, all of this in 3D space.

Some of the products of the research can be used for more intelligent manipulation in cases where variety of items exist and decisions as to selection of groups to be moved simultaneously have to be made. Example of such use: home assistance robotic system that performs a task of sorting and storage of groceries.

Should mention ICRA here, and the book that could be published.

1.4. Literature Review

Important contribution to the future development of robotics in general and grasping in particular belongs to Reuleaux [29], Landau and Lifshitz [14] for bringing the basics of mechanics and mechanism kinematics, which serve as foundation for every analytical approach of robotics study. Lozano-Perez [17] for the introduction of Configuration Space use which contributed in vast majority of fields. Use of the Configuration Space representation has been a major interest for the simultaneous multi-body manipulation, yet because of high computation cost and lack of explicit analytical formulations was not fully incorporated in this research.

Ball [3] introduces an essential ideas of screw theory which allows use of wrench and wrench space concepts, tools that take major part in the contemporary robotics. Murray et al. [23] presents concise and yet thorough survey on appliance of the mathematical tools for the description of manipulations. With general tools for mechanisms and manipulators in particular, interaction of such manipulators with the environment can be formulated using Coulomb's friction, categorizing contacts for frictional, frictionless, hard or soft contacts Murray et al. [23]. Rimon and Burdick [31] proposed bounds on number of frictionless fingers for 2D polygonal object immobilization.

Given broad mathematical basics, grasping becomes of particular interest with Mishra et al. [20] showing existence of positive grasps, and allowing following grasp synthesis development: Bunis et al. [10], Borst et al. [8]. Derivations would not be possible without defining form and force closure by Asada and Kitagawa [2]Rimon and Burdick [30]. Nguyen [24] proposes a viable methods for constructing force closure grasps.

More methods on grasp construction include synthesizing grasps evolved from predefined configurations Pollard [27], mimicking human behavior in grasping tasks: using predefined shapes and modes for closure grasps are planned with numerical computations of trajectories intersections Wren and Fisher [37].

Contact modeling Xydas and Kao [40]

Rimon and Burdick [32] propose a novel idea of mobility index based on second order curvature of bodies in contact, intended for multi-finger grasping.

Grasp evaluation takes a long way with Mishra and Silver [21] discussing the stability of grasps, Ferrari and Canny [12] introducing quality measures based on wrenches, Roa and Suárez [33] surveying several methods of grasp quality measure, whileLin et al. [16] extend the measures for complaint grasps.

Grasp generation can be addressed with different approach, with randomized grasp generationBorst et al. [7] where grasp candidates are generated, tested for force closure and evaluated according to desired quality measures.

Integrating methods in real life applications usually require use of vision systems as showsCorke [11] and often extended to use of machine learning for vision based grasping:Quillen et al. [28], Bezak et al. [5], Le et al. [15].

The list would not be full without mentioningSiciliano and Khatib [35] providing an overview and composition of concepts and techniques in grasp synthesis, evaluation and optimization.

Most methods for grasp construction are searching for force closure grasps, some of them for optimal grasps. Usually the complexity of the computations depends on the number of edges / faces of the object to be grasped.

Multiple objects graspingPajarinen and Kyrki [26], Zeng et al. [41]

* ***Put more info on CONFIGURATION SPACE approach to grasping***

What about Stable pushing of stacked parts

1.5. Preliminary Background

Basically this part should explain to the reader all the basics that are known prior this work. Everything that is out of the scope of the B.Sc in mechanical engineering should be addressed and explained.

Starting with introduction to grasping, contact models and grasp matrix, follow with for closure and form closure, grasp planning (fingers number, conditions and grasps). Restraint analysis (form closure). Need to describe: workspace, closure, contacts, twist and wrench duality.

Notation in robotics makes a wide use in vectors and matrices, along with coordinate frames and transformations. A lot of concepts are drawn from linear algebra.

Manipulator forces, relations between joint forces/torques and end-effector forces, singularities, dynamics of manipulators etc. is out of the scope of this work.

* ***Convex cones and dual cones***

1.5.1. Motion and Forces

Traditionally, motion of rigid bodies can be described by utilizing a concept knows as twist: the description of both linear and angular velocities of a rigid body.Murray et al. [23] shows that any rigid body motion can be described using a twist, namely that every motion of a rigid body can be described by a rotation of a body about an axis and translation of that body in direction parallel to that axis – known as*screw* motion. The mapping between twist and screws can be done using matrix exponential.

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix} \in \mathbb{R}^6 \qquad v \in \mathbb{R}^3, \omega \in \mathbb{R}^3 \tag{1.1}$$

A generalized force, acting on the body is described as a*wrench*- force/moment pair, which consists from a linear component and an angular component acting on a point.

$$F = \begin{bmatrix} f \\ \tau \end{bmatrix} \in \mathbb{R}^6 \qquad f \in \mathbb{R}^3, \tau \in \mathbb{R}^3 \tag{1.2}$$

Combination of wrench and twist define power: the dot product of twist and wrench yield instantaneous power. Reciprocal wrenches and twist are combinations that have instantaneous zero power:$F \cdot V = 0$. Both concepts can be described as screws, and this way threat them in same way. The notion allows simple analysis of kinematics of mechanisms in general, and, more importantly, grasping in particular.

In grasping, the wrenches applied to the object act as a set of constraints, and twists are possible motions of the object. If acting wrenches allow no reciprocal twists, i.e. there is no twist that for every wrench the dot product will be 0, then the object is immobilized.

1.5.2. Grasping and Immobilization

Contacts

Basic concept in grasping is a contact between a finger and an object. Contact describes the mapping of forces applied by fingers at some point on the object boundary to resultant wrenches in object coordinate frame. A set of properties should be defined for a contact:

- Contact point location
- Contact type
- Forces and torques applied by the contact

Contact point location is usually defined in object-attached coordinate frame. It is convenient to define the origin of this coordinate frame at object's center of mass, and to set contact coordinate frame where*z-axis* is normal to the object surface, pointing inside of the object, this way positive$z$ value means pressure applied by a finger. Term pressure here means that finger can exert only pressing force and not pulling, as opposed to suction cups.Figure 1.1 illustrates coordinate frames:O - object coordinate frame,C - contact coordinate frame together with vectorr_c which defines the location of contact point in object coordinate frame. Contact type would describe the interaction between the surface of the object and the surface of the finger. Common cases are: normal force only, in cases where friction coefficient is low; normal and tangent forces, in cases where friction is significant; normal and tangent forces along with normal torque (torsional

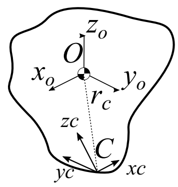


Figure 1.1.: Coordinate frames.

friction). Typical examples for given contact types are: pressing a pen tip against a glass - normal force only; pressing a pencil tip against rubber - can provide normal and tangent forces; pressing a hand against a paper sheet laid on a table: friction coefficient between a sheet and a table is considerably lower than this between the hand and the sheet, this way we could move the sheet aside and rotate it about axis normal to the table surface.

Frictionless point contact is a simplest type of contact which can apply forces only in the direction of the surface inward normal.

Point contact with friction is a more realistic contact which can apply both normal and tangent forces on an object. The modeling of a friction between a finger and an object surface could be simplified by assuming that the finger can exert small tangent forces, linearly depended with a normal force. This would produce a*friction cone* as shown inFigure 1.2. Normal force and tangent force written asf_n andf_t respectively. Tangent force depends on normal force. and thus can be described mathematically by inequality:

$$f_t \leq f_n \cdot \mu \tag{1.3}$$

whereμ stands for coefficient of friction between a finger and a surface. For practical applications, friction cone is approximated by finite set of vectors lying on cone's lateral surface. In 2 dimensional space it takes a form of an isosceles triangle with a vertex between two equal edges located at the contact point, and the bisector lies along the normal direction. The half angle of the aforementioned vertex is defined by$\alpha = \arctan(\mu)$.

Soft finger is the most"capable" contact, able to apply both normal and tangent forces, and a torque about the contact normal as well. In$\mathbb{R}^2$ the concept is not used since rotations only possible around 1 axis.

** *INSERT CONTACT TYPES SUMMARIZING FIGURE HERE***

Grasp and Grasp Matrix

A grasp is a collection of contacts (types and locations) that can apply forces and torques on a body. Contact forces applied to the object will result in a resulting force and torque - wrench. A matrix transforming contact wrenches to object

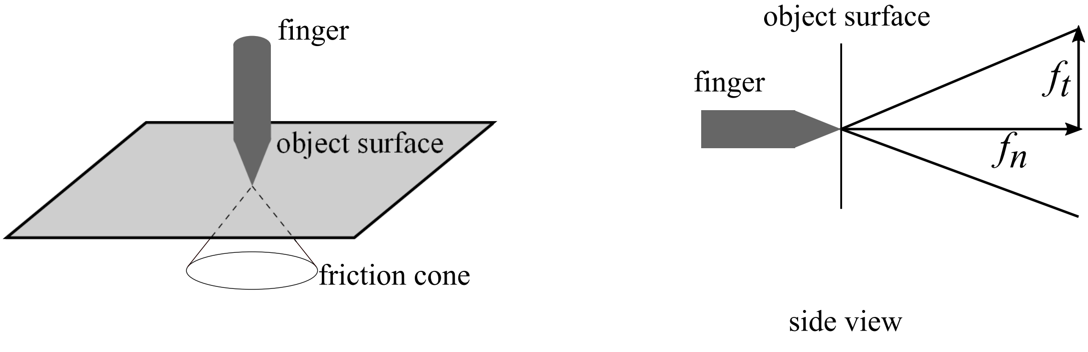


Figure 1.2.: Friction cone.

wrenches can be derived from contact locations and types, and forces applied. Construction of the grasp map can be automatized for accurately known objects and fingers. Derivation of the matrix is not shown here, reader can refer toMurray et al. [23] for more detailed explanation.

*** *MAYBE IT SHOULD BE SHOWN HERE.***
GM maps the contact forces at contact locations to wrenches induces in the body.

Example 1. Grasp map derived by[23] p.222 is presented below inFigure 1.3: a planar rectangular object held by two fingers which apply forces in the plane. Resulting wrench on the object from one finger can be described as following:

$$F_o = \begin{bmatrix} f_o \\ \tau_o \end{bmatrix} = \begin{bmatrix} R_c & 0 \\ \begin{bmatrix} -p_y & p_x \end{bmatrix} R_c & 1 \end{bmatrix} \cdot \begin{bmatrix} f_c \\ \tau_c \end{bmatrix} \tag{1.4}$$

whereR_c represents*** *WHAT?*** . Grasp map for this example is given by:

$$G = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ r & 0 & r & 0 \end{bmatrix}. \tag{1.5}$$

Although the derivation of the grasp matrix is not shown here, reader can intuitively see that locations of contacts and orientation of contact coordinate frames are reflected in this representation. Using grasp map, total wrench on the object could be defined as:

$$F_o = G \cdot f_c = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ r & 0 & r & 0 \end{bmatrix} \cdot \begin{bmatrix} f_x^1 \\ f_y^1 \\ f_x^2 \\ f_y^2 \end{bmatrix}, \tag{1.6}$$

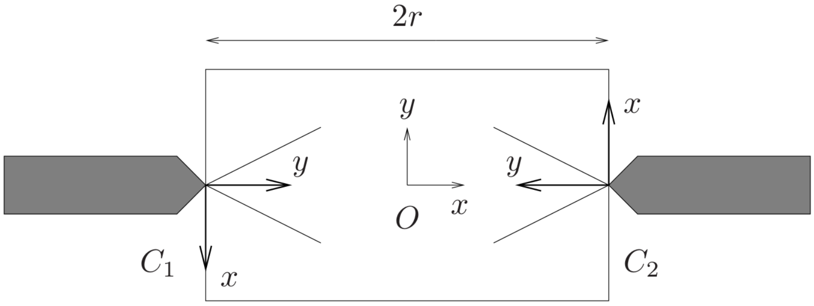


Figure 1.3.: Planar grasping.

whereF_o denotes a total wrench (force and torque combined) applied to the object,G is the grasp map as shown inEquation 1.5 andf_c is contact forces vector (normal and tangent forces applied by each and every finger). Expanding the matrix-vector product yields total wrench that can be confirmed by visual observation ofFigure 1.3:

$$\begin{bmatrix} F_x \\ F_y \\ T_z \end{bmatrix} = \begin{bmatrix} f_y^1 - f_y^2 \\ -f_x^1 + f_x^2 \\ r \cdot f_x^1 + r \cdot f_x^2 \end{bmatrix}. \tag{1.7}$$

The space that is spanned by all contacts is called the Grasp Wrench Space (GWS) and characterizes the ability of the grasp to balance disturbance forces. A convex hull of vectors from grasp matrix rows with torque values normalized with some reference distance (radius of a sphere with same volume as the grasped object, for example) can be further used to analyze the grasp quality.

Grasp properties:

Given a grasp one can inquire what are the properties of that grasp, how good is it and whether it is suitable for desired task:

Equilibrium grasp is a grasp where fingers can exert forces that will keep the grasped object in the equilibrium state.

Manipulability defines whether arbitrary motions of a grasp object can be generated by applying forces by the fingers.

Immobilization grasp is called immobilizing if applied contacts prevent any motion of the object (translation or rotation).

Grasp redundancy the grasp can consist of number of fingers, while not every one of them is necessary (e.g. for immobilization, or for equilibrium grasp).

Grasp stability Presented byMontana [22].Bicchi and Kumar [6] summarized that the grasp stability depends on the local geometry of the grasp body. One of the definitions of stability is based on the potential function

of the contacts forces, and the grasp said to be stable if for small perturbations from the equilibrium grasp the potential function has positive gradient, and unstable otherwise.
Extension (or reduction) of grasp stability on contact stability can be described as follows: If two bodies are in contact and are held in equilibrium by external forces (e.g. fingers), if for small perturbation of contact position, the contact will stay in that new position or return to the original position than the contact is stable. More detailed explanation presented in thesubsection 2.1.1 .

Force Closure A grasp is a force closure if the space of object wrenches is spanned by the set of finger forces and include the origin surroundings inside the convex hull of applied wrenches.

If a grasp can resist any external wrench applied to the object the grasp is called*force-closure*. One way to ensure that given grasp is force-closure is constructing*convex hull*. If convex hull contains the origin then the grasp is force closure.

Constructing of convex hull is done by using the columns of grasp map: each column represents a vector. Plotting these vectors and closing polygons between them will yield a convex hull shape (polyhedron). Normally tangent forces are result of friction and along with normal forces can be defined by cones, as shown above. For the matter of simplicity of the example, tangent forces are assumed to be independent of normal once. Hence,f_x assumed to be positive or negative, wheref_y is positive only as shown inFigure 1.4 . Once

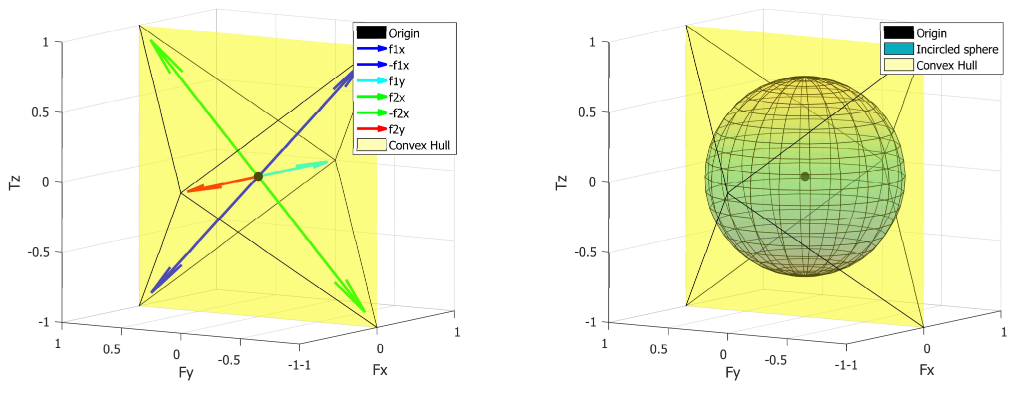


Figure 1.4.: Convex hull (left) and inscribed sphere (right).

convex hull is built, quantitative assessment can be performed. One way of doing that is determining the largest sphere to inscribe in the hull with its center at origin, as shown inFigure 1.4, and measuring it's diameter. If vectors are normalized, the radius of the sphere can serve as***grasp quality measure***(dimensionless value). In case of 3-dimensional space and object convex hull is 6-dimensinoal:$F_x, F_y, F_z, T_x, T_y, T_z$ and it cannot be shown fully in 3-dimensinal domain, but projections such asF_x, F_y, F_z can be presented.

Form Closure A grasp is a form closure if it is a force closure and the fingers are frictionless contacts. The form closure is“geometrical” immobilization of an object, discarding friction forces etc.

Form closure: Constrain the movement of the object

** ***INSERT IMAGES ON FORM AND FORCE CLOSURE, EQUILIBRIUM GRASP, MANIPULABLE GRASP***

Quality Measures

** ***NEED MORE INFO HERE***

[Ferrari and Canny][12] Radius of largest hyper-sphere you can fit in convex hull centered in origin

[Zhu and Wang]: Numerical test which measures the scaling factor for the maximum compact set incribedin the grasp wrench space

Quantifying the quality of the grasp. Task Wrench Set. Inscribed sphere.Borst et al. [8], Roa and Suárez [33]

Grasp planning and construction

When planning a grasp for given object, one has to determine how many contacts and of what kinds are needed.Murray et al. [23] reminds that for spatial object, 4 friction contacts are sufficient for grasp. But, when dealing with robotic grippers, contact locations are limited due to gripper geometry, and forces gripper can apply are limited too. Given an object and a multifingered robot hand (along with robot itself) maximal wrenches an hand can exert can be calculated. Xiong et al. (2007)[39] shows extended grasp capability assessment, but for the purposes of this research it is safely to assume that the objects are light enough and the hand is robust enough to exert desired wrenches. Yet contact locations are limited because of the dimensions of finger and placement of the fingers on the palm (e.g. opposing finger vs symmetric pattern).

Variety of methods for grasp planning was presented during years, yet for the purposes of this research we use geometrical methods as proposed byNguyen [24]. The paper shows that for construction of force closure grasp with 4 planar forces only, the geometric necessary and sufficient condition is that that no more than 2 force directions does not intersect in one point, and remaining 2 force directions result in moments with different signs about the aforementioned intersection point.** ***INSERT NEW IMAGES*** :

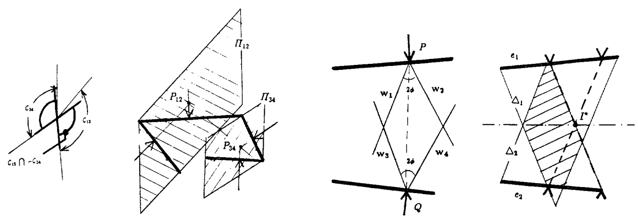


Figure 7: Finding frictionless grasps on four edges.

Figure 8: Finding grasps with friction on two edges.

Figure 1.5.: NGUYEN

The methods uses geometric relations between object's edges and corresponding normal directions to construct polygons and determine edge regions suitable for finger placement. Method can be used both for frictional and frictionless contacts (2 fingers needed for frictional grasp).

Additional method that is used in a research is random grasp generation with evaluation of the resultant grasps, as proposed byBorst et al. [7]. Main idea is to generate random contact points along the object's boundary and evaluate the resultant grasp both for force closure and for given quality measure.

Find optimal set of graphs with independent regions of contact and pick mid points of the regions as optimal grasp points.

1.5.3. Graphs

*** Rethink this section***

The main interest of this research is multiple objects grasping which requires objects interaction classification. Some of the objects can be in contacts with other object(s) and these connections will be partially described by a graph.

Graph is structure describing a set of objects and relations between them. In case when every pair of nodes in graph are connected the graph is called Connected Graph. Undirected graphs alone are the utilized in this work. Adjacency matrix is a way to represent a finite graph. For non-directed graphs the adjacency matrix is symmetric. Matrix elements indicate whether a connection exists between 2 nodes, and also may quantify the connection.

*** EXAMPLE GRAPHS, ADJACENCY MATRICES***

1.6. Assumptions

This work focuses on planar space where objects represented as polygons with finite amount of edges. Knowledge of exact location and orientation of those polygons are assumed, along with the knowledge of the exact shape.

The polygons are assumed to be non-self-intersecting, while for practical purposes the null-thickness regions of self-intersection polygons can be infinitesimally thickened to form a non-self-intersection polygon.

In a given set of objects only connected configurations are of interest. The problem with more than one distinct group of connected polygons are easily divided to subproblems.

The work focuses on construction of equilibrium force closure grasps with frictionless fingers using first order geometry (i.e. form closure).

Finger contacts are point contacts and hence can act on one object at a time, two fingers cannot be placed at same location.

1.7. Problem Statement

*** What should be written here at all? Do I need this section? Probably I do, check Yoav's thesis, and Amir's and Avishay's as well.***

Given: A set of polygonal objects in 2D space.

Desired:

- **A** Configuration of objects in space achieved by translation and rotation only (no reflections).
- **B** Minimal set of finger locations to immobilize all of the given objects.

2. Proposed solution

In a common workspace polygonal objects can contact one another in several ways. Classification of the relations between the polygonal allows more thorough state analysis and further planning. After the classification is presented, an algorithm for finger placement given configuration of polygons is proposed, and then an algorithm for polygons placement is proposed.

** ***The idea can be expanded for various tasks, such as home assistant robot that brings several items in one run. This should go to Motivation.***

Thus, the solution can be subdivided into 2 parts: determining the minimal amount and placement of fingers required to immobilize given set of contacting polygonal objects and a determination of desired configuration (rearranged set of objects) that will require minimal amount of finger to immobilize it. Both parts require examination of inter-object contacts and relations between objects, along with classification for further analysis.

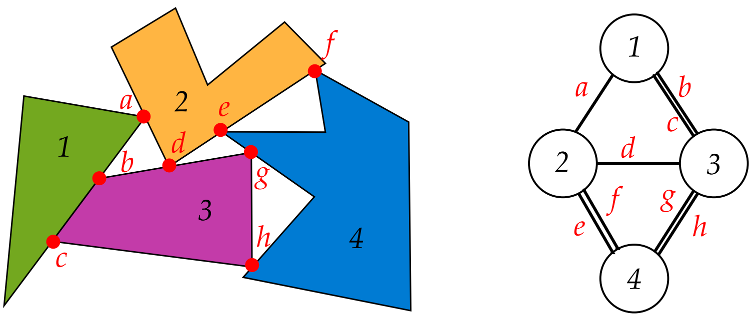


Figure 2.1.: Objects set and a corresponding graph.

Example 2. 4 objects inFigure 2.1 where 1 and 2 touch 3, and 4 touches 2,3. The contacts serve as graph edges, while objects are represented as graph nodes. Graph for the given object configuration is shown below. Adjacency matrix describing given graph is presented:

$$A = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 2 \\ 0 & 2 & 2 & 0 \end{bmatrix}.$$

General idea is to find an amount of missing finger contacts for each object and each set of objects by examining the relationship between them.

2.1. Mathematical description

REWRITE THIS SECTION[1]

Given 2 polygons,p_1, p_2. The twist of polygonp_1 relative top_2 can be defined as$\dot{x}_{12} = (v_{12}, \omega_{12})^T$. Polygons can maintain contact but not penetrate one another.

** ***Let IP denote the set of interaction points common to both polygons. Let $C = \{c_i\}$ be a set of contact descriptions between these polygons.***

Let$\hat{n}_i^j$ denote the inward normal direction of the boundary of polygon$j$ at contact point$i$.

The contact between the polygons can be described using one or more contact points and corresponding contact normal directions.

2.1.1. Relationship between polygons and polygon sets

The examination of given polygon configuration can be subdivided to interaction of each couple of polygons. The polygons can be disconnected - no points belong to 2 polygons simultaneously, or connected - one ore more points belong to 2 or more polygons. The connection between the polygons can be represented by a contact description. A stable contact between 2 polygons can exist in several variants:

Vertex-Edge Vertex to edge contact is the discrete contact where a vertex of the first polygon (p_1) is coincident with an edge of the second polygon (p_2). The contact normal direction is defined by the edge the of the second polygon. The set of contacts for this variant consists of a single contact:$C = \{c_1\}$. Contact restrictions can be described as following: velocity of the contact point on the boundary ofp_1 relative to the velocity of the contact point on the boundary ofp_2 (denotedv_{12}) cannot have component in direction$\hat{n}_1^2$ - polygons cannot penetrate. This can be expressed as follows:$v_{12} \cdot \hat{n}_1^2 \leq 0$. Contact is maintained when the component of the relative velocity in the direction if inward normal of first polygon is 0. Since$\hat{n}_i^2 = -\hat{n}_i^1$,$-v_{12} \cdot \hat{n}_1^2 \leq 0$. Therefore, the contact is maintained when relative velocity of contact points of each polygon are perpendicular to the edge normal.

$$v_{12} \cdot \hat{n}_1^2 = 0 \tag{2.1}$$

Edge-Edge Edge to edge contact is a continuous contact between two bodies along a continuous segment. Since the edges are finite, there exist at least 2 vertices belonging to either one or two polygons that belong to the contact segment. In a way analogous to distributed load concentrated in a point, the continuous distributed contact can be

[1]Treat the contact point and not the whole polygon. Think what do I want to say here at all.

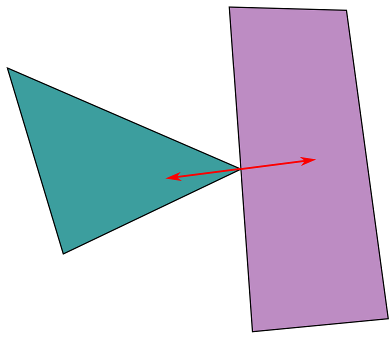


Figure 2.2.: Vertex-to-Edge contact example.

concentrated in 2 different points. Two different vertices are selected to be such points. Since the vertices lie along the same edge, the contact normal directions are the same in this case. The set of contacts for this variant consists of two contacts:$C = \{c_1, c_2\}$, $c_i = \{r_i, \hat{n}\}$, wherer_i stands for contacts' location and$\hat{n}$ stands for the normal direction of the edge. Graphically this can be depicted as follows: Polygon 1, polygon 2, contact line, contact points. Cases with 2 points, case with 3 vertices, case with 4 vertices in contact.

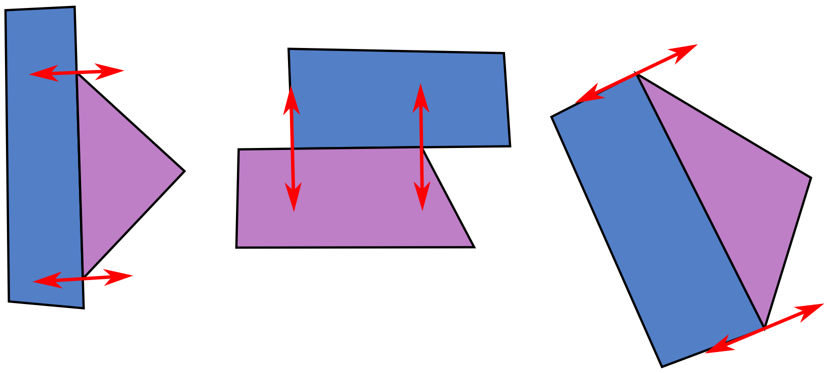


Figure 2.3.: Edge-to-Edge contact examples.

Vertex-Vertex Vertex to inner vertex (in concave polygon) is another type of discrete contact with multiple point-normal parameters. Polygon 1 vertex is in contact with 2 edges ofp_2. In this case, since the movement of polygon 1 relative to polygon 2 is restricted in 2 directions, 2 normal directions exist at same contact point. The set of contacts for this variant consists of two contacts:$C = \{c_1, c_2\}$, $c_i = \{r, \hat{n}_i\}$, where$r$ stands for contact location and$\hat{n}$ stands for edges' normal directions.

Other variants of contact descriptions are variations of the mentioned above. Examples: edge and inner vertex, vertex and vertex, edge and edge. Outer vertex to outer vertex contact assumed to be unstable and is discarded in this work. One polygon can contact several others in distinct points, apart from a case where several vertices are coincident as shown inFigure 2.5, in this case,

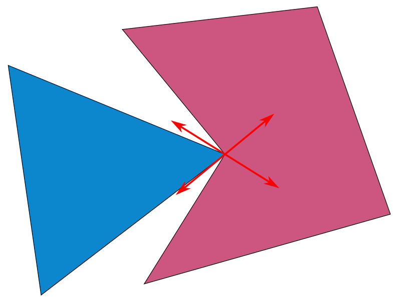


Figure 2.4.: Vertex-to-Inner-Vertex contact example.

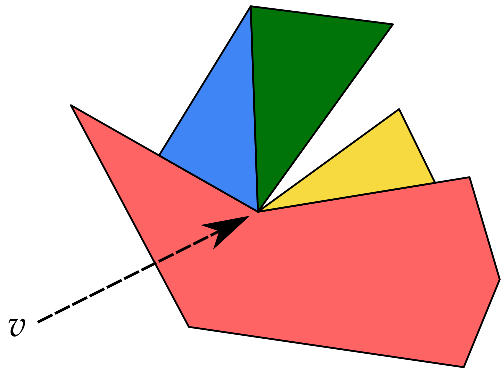


Figure 2.5.: Multi-object vertex$v$.

2.1.2. Contact description

For the purpose of polygon arrangement evaluation, every contact is assumed to be performed by polygon A acting on the polygon B. The following description of a contact is adapted:

$$C = \{id\,(P_A)\,, id\,(P_B)\,, p, \hat{n}\} \tag{2.2}$$

where$id\,(P)$ stands for polygon number/identification entry,p is a contact point location and$\hat{n}$ is the contact normal direction inward the polygon B. An example of inter-object contact description shown onFigure 2.6. The contact is saved as:$C = \{A, B, p, \hat{n}\}$,whereA, B are the names of the polygons,p is the contact location vector in selected coordinate frame and$\hat{n}$ is the inward normal direction of the second polygon at the contact location.

2.1.3. Solution existence for combination of internal and external contacts

Given a set of polygonal objects, where some of the objects are in contact one with another. Each object has at least 4 contacts: some of them are fingers and some are inter-body contacts. The equilibrium equation for each object is written in form of total wrench:

$$G\bar{f} = 0 \tag{2.3}$$

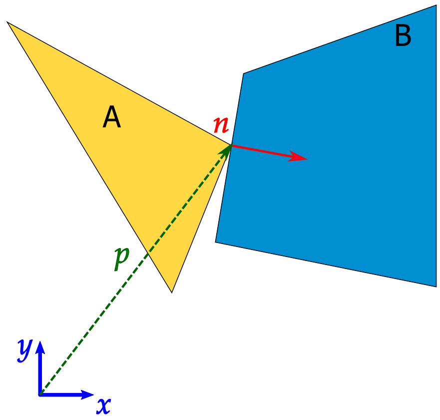


Figure 2.6.: Contact description example.

where:$\bar{f} = \begin{bmatrix} f_{f1} & f_{f2} & f_{p12} & f_{p13} & \dots & f_{pMN} \end{bmatrix}^T$ is a set of forces between fingers and bodies, and between bodies and other bodies in contact.G matrix rows multiplied by the set of forces produce equations for $\begin{bmatrix} f_x \\ f_y \\ \tau \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ for each body. The matrix will be block diagonal with some cross-block columns which correspond to body-to-body contacts.

Lemma 1. *Given an edge of a polygonal object and a frictionless point contact, the generalized (superposition) wrench for all possible locations of contact along the edge can be expressed as 2 dimensional cone (2 vectors) in vertical plane of the configuration space (the plane's normal direction will be in*$f_x f_y$ *plane).*

Proof. All possible contacts on the edge will have the same direction and hence samef_x, f_y components. But for all these possible contacts the wrenches will be different. The positions of the contact points one the edge have limits, and at those limits will define the marginal moments that the contact can apply. These two marginal vectors will form a two dimensional cone in a wrench space. □

Proposition 1. *4 point fingers acting on a 2D object in distinct points along object's boundary may constitute a force closure equilibrium grasp.*

Proof. Caratheodory's theorem shows that a N-dimensional space can be positively span by N+1 vectors. For two-dimensional task space, there are 3 degrees of freedom(x, y, θ) and hence both the configuration space and the wrench space are three dimensional ($\mathbb{R}^3$). Four distinct contacts can be represented be 4 vectors in the wrench space and span it. Because of the configuration/wrench space duality, fully spanned wrench space means fully constricted configuration space: the object will have no freedom of movement. The convex hull of these vectors spans the origin as well, and thus it is an equilibrium grasp. □

Proposition 2. *A set of*N *polygonal objects in*$\mathbb{R}^2$ *in stable (what is stable contact ???) contact one with another can be immobilized by at most*$3N$ *fingers assuming first order non-frictional contacts.*

Proof. Given connected set of objects, for each object at least one contact exist. For an object in$\mathbb{R}^2$ to be first order immobilized minimum 4 constraints required. Knowing that at least one constraint already exists because the set is connected, for each object 3 external constrains are needed, yielding$3N$ fingers for the whole set. □

Proposition 3. *A set of$N$ polygonal objects in$\mathbb{R}^2$ in contact one with another can be immobilized by at least 4 fingers assuming first order contacts and exceptional resulting shape.*

Proof. Assuming optimal configuration of the objects where each of them is immobilized relative to others by form closure conditions and possibly external constraints (EC) the total amount of external constraints for the set immobilization is$\max(4, |EC|)$. Example: jigsaw puzzle which once assembled can be immobilized by 4 fingers. □

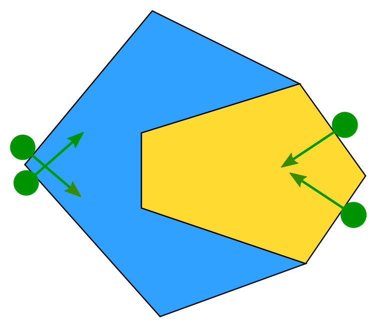


Figure 2.7.: Immobilization by minimum 4 fingers.

Proposition 4. *Given a polygonal object and 3 parallel frictionless contacts on the object boundary (same edge or different edges). It is impossible to construct and first order immobilizing grasp with 1 additional contact.*

Proof. 4 contacts of a frictionless grasp have to not to intersect in one point, as showedNguyen [24]. 3 parallel contacts are“intersecting” at infinity. Moreover, 4-th contact alone cannot span all moments around that intersection point. □

Example 3. Two bodies in contact at one point. Contact forces shown in red, finger forces shown in blue. Each forcef_i acts at point$\mathbf{x}_i$ located at body boundary, with directionθ_i which vector is $\begin{bmatrix}\cos\theta_i \\ \sin\theta_i\end{bmatrix} = \begin{bmatrix}c_i \\ s_i\end{bmatrix}$. The inter-body contact forcef_{AB} selected to act in direction $\begin{bmatrix}c_{AB} \\ s_{AB}\end{bmatrix}$ at point$\mathbf{x}_{AB}$. The objects are immobilized by 3 fingers, according to theProposition 2. Defining a grasp matrix for fingers$m$ to$n$ (note the linear independence of the rows in a matrix):

$$G_{f[m:n]} = \begin{bmatrix} c_m & \dots & c_n \\ s_m & \dots & s_n \\ \mathbf{x}_m \times \begin{bmatrix} c_m \\ s_m \end{bmatrix} & \dots & \mathbf{x}_n \times \begin{bmatrix} c_n \\ s_n \end{bmatrix} \end{bmatrix} \tag{2.4}$$

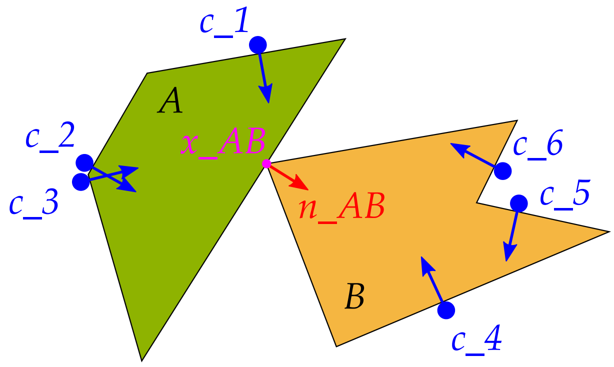


Figure 2.8.: Solution existence example.

such that in case of isolated 2D object

$$\begin{bmatrix} f_{Ax} \\ f_{Ay} \\ \tau_A \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 \\ s_1 & s_2 & s_3 \\ \mathbf{x}_1 \times \begin{bmatrix} c_1 \\ s_1 \end{bmatrix} & \mathbf{x}_2 \times \begin{bmatrix} c_2 \\ s_2 \end{bmatrix} & \mathbf{x}_3 \times \begin{bmatrix} c_3 \\ s_3 \end{bmatrix} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = G_{f[1:3]} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{2.5}$$

When there are 2 or more columns, 3rd row will only be a linear combination of first 2 only if all contacts are in the same point:

$$G_{f[m,n]} = \begin{bmatrix} c_m & c_n \\ s_m & s_n \\ \mathbf{x}_m \times \begin{bmatrix} c_m \\ s_m \end{bmatrix} & \mathbf{x}_n \times \begin{bmatrix} c_n \\ s_n \end{bmatrix} \end{bmatrix} = \begin{bmatrix} c_m & c_n \\ s_m & s_n \\ x_m s_m - y_m c_m & x_n s_n - y_n c_n \end{bmatrix} \tag{2.6}$$

$$\rightarrow \mathbf{x}_m = \mathbf{x}_n$$

Each$G_{f[m:n]}$ block consists of 3 rows and 1:3 columns.When there is one column, third row is a linear combination of first 2:

$$G_{f[m]} = \begin{bmatrix} c_m \\ s_m \\ \mathbf{x}_m \times \begin{bmatrix} c_m \\ s_m \end{bmatrix} \end{bmatrix} = \begin{bmatrix} c_m \\ s_m \\ x_m s_m - y_m c_m \end{bmatrix} \tag{2.7}$$

In case of 1 finger contact on given body, 3rd row of the grasp matrix (torque) will be linear combination of first 2 only if the finger contacts is on the same locations as other (inter-body) contacts (and has same direction), which by definition is neither feasible nor needed.

We can summarize the forces action on 2 bodies:

$$\begin{bmatrix} f_{Ax} \\ f_{Ay} \\ \tau_A \\ f_{Bx} \\ f_{By} \\ \tau_B \end{bmatrix} = \begin{bmatrix} G_{f[1:3]} & \left\{ \begin{matrix} -c_{AB} \\ -s_{AB} \\ -\mathbf{x}_{AB} \times \begin{bmatrix} c_{AB} \\ s_{AB} \end{bmatrix} \end{matrix} \right\} & 0 \\ 0 & \left\{ \begin{matrix} c_{AB} \\ s_{AB} \\ \mathbf{x}_{AB} \times \begin{bmatrix} c_{AB} \\ s_{AB} \end{bmatrix} \end{matrix} \right\} & G_{f[4:6]} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_{AB} \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} =$$

$$= \begin{bmatrix} G_{f[1:3]} & 0 & \left\{ \begin{matrix} -c_{AB} \\ -s_{AB} \\ -\mathbf{x}_{AB} \times \begin{bmatrix} c_{AB} \\ s_{AB} \end{bmatrix} \end{matrix} \right\} \\ 0 & G_{f[4:6]} & \left\{ \begin{matrix} c_{AB} \\ s_{AB} \\ \mathbf{x}_{AB} \times \begin{bmatrix} c_{AB} \\ s_{AB} \end{bmatrix} \end{matrix} \right\} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_{AB} \end{bmatrix} = 0 \tag{2.8}$$

As seen in theEquation 2.8 above, the grasp matrix for the given set of objects consists of a diagonal blocks and off diagonal columns. A system will be in equilibrium if a non-trivial set of forces exist that solves the equation.

** ***Check the following, what is this? Do I need it at all?***

2.1.4. Edge generalized wrench

The evaluation of possible contact positions on an edged can be simplified by introducing an edge generalized wrench:

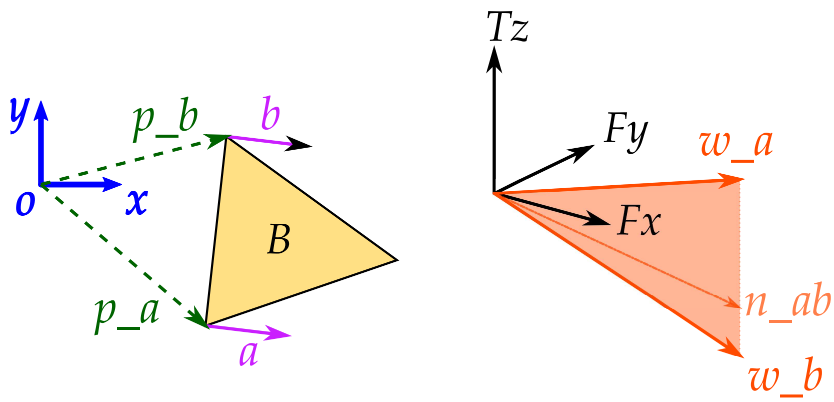


Figure 2.9.: EGW.

Definition 1. Given a polygon's edge$e$ and a reference point$o$, the set of all possible wrenchesw_e can can be defined by a two-dimensional convex cone in a vertical plane of a wrench space.

Denote the edge's normal by$\hat{n}$ and edge's ends by$\vec{e}_1, \vec{e}_2$ relative to$o$, the edge is parameterized by parameter$s \in [0,1]$ the wrench of a contact along the edge can vary between:

$$(1-s)\begin{bmatrix}\hat{n}\\ \vec{e}_1 \times \hat{n}\end{bmatrix} + s\begin{bmatrix}\hat{n}\\ \vec{e}_2 \times \hat{n}\end{bmatrix}, \qquad s \in [0,1] \tag{2.9}$$

which is exactly a linear combination of two marginal wrenches. In a geometrical representation, it is a planar angle contained in a vertical plane (plane's normal direction has no torsional component) in the wrench space. Illustration shows an example of a generalized edge wrench. An edge of polygonal object B has inward normal directionn_{ab}. Two marginal contactsa, b are shown in violet, with locations defined by vectorsp_a, p_b respectively. Wrenches induced by these contacts are calculated by:

$$w_a = \begin{bmatrix}\hat{n}_a\\ \vec{p}_a \times \hat{n}_a\end{bmatrix}, \; w_b = \begin{bmatrix}\hat{n}_b\\ \vec{p}_b \times \hat{n}_b\end{bmatrix}.$$

As can be seen from the FIGURE , contact$a$ induces positive torque about the selected reference point, while contact$b$ induces negative torque. Hence, the 3-rd components of a wrench have opposite signs and are located on different sides of thexy-plane. The expression for the edge generalized wrench can be rewritten to a form of:

$$\begin{aligned}
w_e(s) &= (1-s)\begin{bmatrix}\hat{n}\\ \vec{e}_1 \times \hat{n}\end{bmatrix} + s\begin{bmatrix}\hat{n}\\ \vec{e}_2 \times \hat{n}\end{bmatrix} = \begin{bmatrix}\hat{n}\\ ((1-s)\cdot\vec{e}_1 + s\cdot\vec{e}_2)\times\hat{n}\end{bmatrix}, \qquad s \in [0,1]\\
&= \begin{bmatrix} n_x \\ n_y \\ (x_1 \cdot n_y - y_1 \cdot n_x) + s\left(n_y\left(x_2 - x_1\right) + n_x\left(y_1 - y_2\right)\right)\end{bmatrix}
\end{aligned} \tag{2.10}$$

We can note that only the torque component of the wrench vector changes along the edge, and hence the edge generalized wrench is strictly"vertical" in the wrench space, and that it is linear along the parameter$s$.

2.1.5. Inverted cone test

As stated above, a convex hull of the contact wrenches should contain the origin of a wrench space. Given several contacts, and corresponding wrenches in the wrench space that do not contain the origin, a simple test can show whether additional vector will make total convex hull to contain the origin or not.

Proposition 5. *Given a set of 3 contacts, whose wrenches do not span the wrench space (the origin of the wrench space is not inside the convex hull of these wrenches) while not being in the same plane and additional contact, the contact can complete the force closure grasp of the object if the contact's wrench lies within the inverted cone of the given contacts' wrenches.*

Proof. Each couple of wrenches along with the origin point form a plane, Equilibrium condition for a grasp is that the total wrench for all contacts equals to

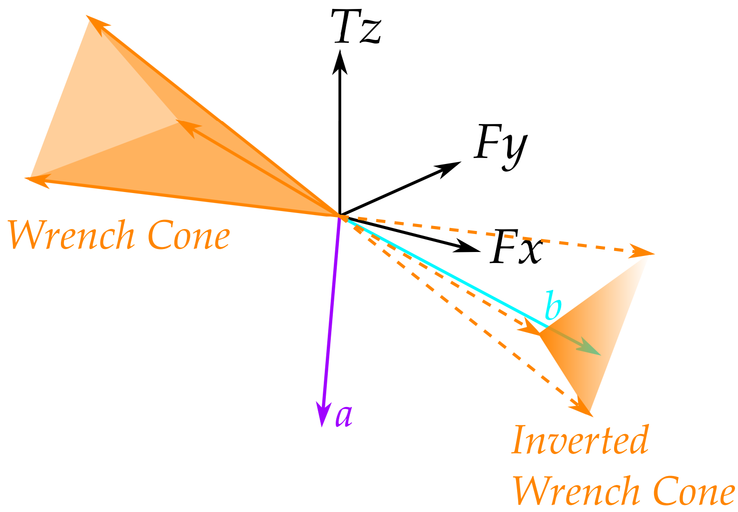


Figure 2.10.: Inverted Cone, with vector$a$ inside and vector$b$ outside.

zero:

$$\sum_{i=1}^{4} G_i f_i = 0 \tag{2.11}$$

WhereG_i is $\begin{bmatrix} n_i \\ p_{c_i} \times n_i \end{bmatrix}$ for each contact. Rewriting theEquation 2.11:

$$G_4 f_4 = -\sum_{i=1}^{3} G_i f_i \tag{2.12}$$

Given the fact that$G$ columns are wrench space vectors, and$f$ are scalar values, this allows the representation of a 4th wrench space vector as a linear combination of other wrench vectors, which is exactly the cone formed by these wrenches. If the 4th vector lies outside the cone then it cannot be represented as a linear combination of other wrenches and hence will not allow equilibrium grasp. □

Example 4. A square object with 3 contacts . Each contact is located 1 unit length from an edge. The contacts are described by the direction and the position as follows:

$$C_a = \left\{ n_a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, p_a = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right\}$$

$$C_b = \left\{ n_b = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, p_b = \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right\}$$

$$C_c = \left\{ n_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, p_c = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \right\}$$

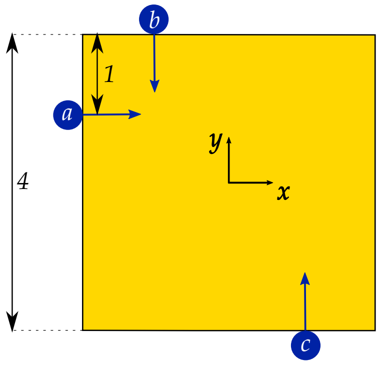


Figure 2.11.: A square object with 3 contacts.

Corresponding wrenches are:

$$w_a = \begin{bmatrix} \hat{n}_a \\ \vec{p}_a \times \hat{n}_a \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \begin{bmatrix} -2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$w_b = \begin{bmatrix} \hat{n}_b \\ \vec{p}_b \times \hat{n}_b \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$$

$$w_c = \begin{bmatrix} \hat{n}_c \\ \vec{p}_c \times \hat{n}_c \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Convex cone of these wrenches along with it's inverse are shown inFigure 2.12. The distance of some point from 3 faces of the inverted cone is calculated by projecting the point onto the face normal. Face normal directions are:

$$\hat{n}_{ab} = \frac{w_a \times w_b}{|w_a \times w_b|} = \frac{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}}{\left\| \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \right\|} = -\frac{\sqrt{3}}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\hat{n}_{bc} = \frac{w_b \times w_c}{|w_b \times w_c|} = \frac{\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}}{\left\| \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right\|} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

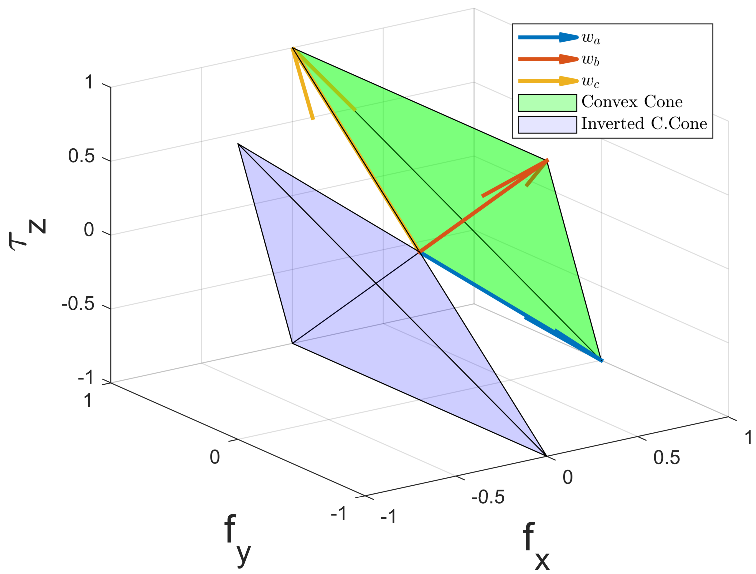


Figure 2.12.: Contact wrenches, convex cone and inverted convex cone.

$$\hat{n}_{ca} = \frac{w_c \times w_a}{|w_c \times w_a|} = \frac{\begin{bmatrix}0\\1\\1\end{bmatrix} \times \begin{bmatrix}1\\0\\-1\end{bmatrix}}{\left|\begin{bmatrix}0\\1\\1\end{bmatrix} \times \begin{bmatrix}1\\0\\-1\end{bmatrix}\right|} = \frac{\sqrt{3}}{3}\begin{bmatrix}-1\\1\\-1\end{bmatrix}$$

Building generalized edge wrenches for each edge: As can be seen fromFigure 2.13, only the eastern edge wrench has the common intersection with the inverted cone. Visual examination of the object confirms intuitively that placement of a finger on the eastern edge only could immobilized the object. It can be seen that only a wrench vector from the eastern edge can complete the convex cone of 3 given contacts' wrenches and form a convex hull which contains the origin in it's interior. Further development shown insection 2.4.

2.1.6. Maximizing the Inscribed Sphere

One of the proposed quality measures is a volume/radius of the sphere inscribed in the convex hull of grasp map basis vectors. Maximizing the sphere radius can be done by increasing the minimal of the distances between the origin end the faces of the convex hull. Since the hull is convex, some of the faces will be tangent to the sphere, and in no case a edge will be tangent. In cases where several contacts act on the object, it is always possible to construct a*convex* hull and thus eliminate redundant wrench vectors.

Proposition 6. *Given a set of k-1 contacts, and their corresponding wrench vectors, convex cone and it's inverse cone; for a generalized wrench that has a*

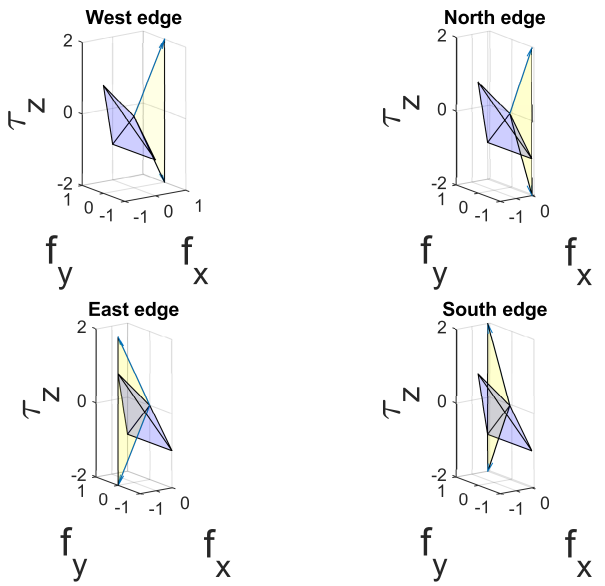


Figure 2.13.: Edge Generalized wrenches for 4 edges of the object.

segment in the interior of the inverse convex cone an optimal k-th contact can be found by examining distances of the convex hull faces from the origin.

Proof. As seen inProposition 5, thew_k has to lie inside of the Anti-Cone. Ifw_k lies on a cone boundary then it and 2 vectors that form that boundary (denoted byw_1, w_2), will form a surface that contains the origin of the wrench space. To maximize the distance of said surface from the origin we first parameterize the wrench vector end point location segment by using theEquation 2.10. For each face of the convex hull, the distance of the face from the origin can be calculated by projecting a point on the plane on the plane's normal. We define a*central axis* of their convex cone as:

$$a = \frac{\sum_{i=1}^{k-1} w_i}{\left|\sum_{i=1}^{k-1} w_i\right|}$$

Assuming that the*k-1* wrenches are sorted in counter-clock-wise direction about the cone axis, meaning that

$$a \cdot (w_i \times w_{i+1}) > 0 \; \forall i \in [1 : k-1],$$

$$a \cdot (w_{k-1} \times w_1) > 0.$$

The equation for the normal direction of a plane formed byw_i, w_{i+1}, w_k is:

$$n_p = (w_i - w_k) \times (w_{i+1} - w_k).$$

If thew_k lies inside the inverted convex cone of first*k-1* contacts, then the normal direction will point from the plane towards the origin. Consequently, the distance of the plane from the origin can be expressed as:

$$d_i = -w_k \cdot (w_i - w_k) \times (w_{i+1} - w_k),$$

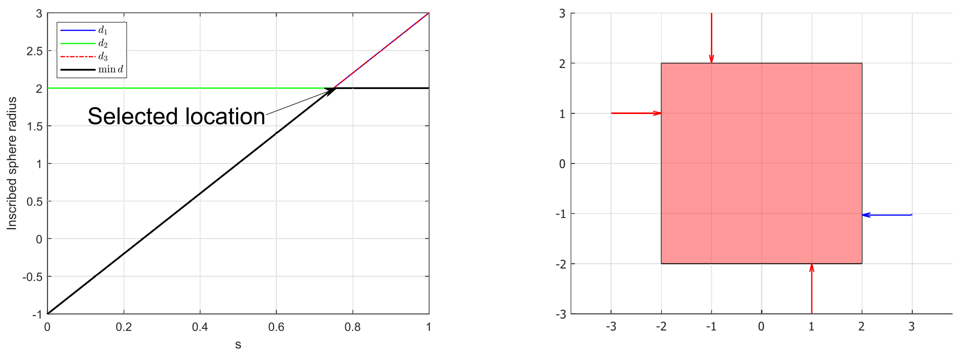
note the"-" sign to obtain positive distance[2]. Substituting the expression for wrench vectors (shown in Appendix) leaves the distance of plane i (contacts i,$i+1$ and parameter s along an edge) to be a function of that parameter only:

$$d_i = f\left(s\right),$$

Ultimately, distance from every face can be expressed as a function of edge parameter s and thus a compound function of minimal distance can be built.

$$d = \min\left\{d_i\right\}\ \ i \in \left[1,\right].$$

Since projection is a linear transformation, and EGW endpoints segment being parameterized uniformly, the distance function will be linearly dependent on on s parameter. Meaning that the distance function will be either constant or monotonically increasing/decreasing along the change in s. Solution can be found by examining k-1 1st order polynomials and finding the maximized minimal value. Since the functions are 1st order polynomials, there either will be 1 such solution or infinite number of solutions (in case where there is a plane parallel to the EGW endpoints segment and hence the distance is constant). □



(a) Distance from convex cone faces. (b) Object with added contact

Figure 2.14.: Selecting the contact based on Inscribed Sphere Radius.

Elaborating on the Example 4, Figure 2.14 presents distances from convex hull faces as function of the parameter s. In this case, due to the symmetry of the object and the contact locations, d_1 and d_3 are equal, while d_2 is constant. In this case one of the distances is constant, and it is minimal at some part of the domain. Contact point selected to be furthest from the brink. Additional examples provided in Appendix.

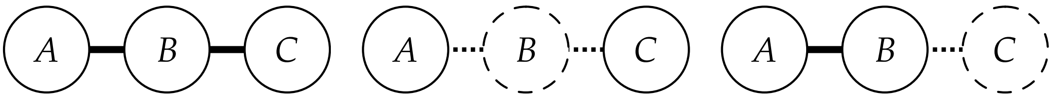
2.1.7. Connectivity graph search

When a object arrangement is given (or obtained) and desired amount of external constraints is found by proposed solution, there is no guarantee that there is no sub-set of objects which is not immobilized.

[2]MATLAB has polyshapes stored with vertices in clockwise direction, hence no minus sign in the code.

This leads to a need of grasp assessment for sub-sets of the given arrangement. Each connected sub-set should be tested and ensured to be first order immobilized. Representing the objects arrangement as a connected graph one can find all connected sub-graphs. Several methods available for sub-graphs extractions based on different criteria, such asUllmann [36], Bron and Kerbosch [9], Gouda and Hassaan [13] but for practical purposes simple algorithm is used:

Recursively excluding nodes and checking whether connected graphs exist. If so - add them to the connected sub-graphs list. This can be implemented both as Depth First Search and as Breadth First Search.



(a) Full graph (b) B node removed no connected subgraphs (c) C node removed AB subgraph is connected

Figure 2.15.: Graph, connected and disconnected subgraphs

2.1.8. Grasp quality measure for a set of objects

Assessment of the Grasp Quality Measure can be performed for given subset. Note that Grasp Quality Measure for single object and for constellation of several object may have different trends at given finger configuration: the grasp may be optimal for the object, but for the constellation it will yield not the highest value.

Example 5. 2 objects in contact. each object is immobilized individually, and the whole constellation is immobilized as well. Examining the grasps by computing GQMs shows that individual GQMs are high (maximal values) but total GQM (due to external contact vectors only) is low due to the directions of the external fingers contacts.

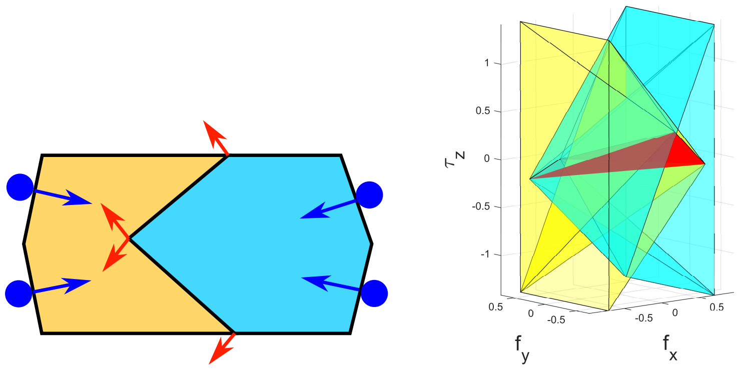
Figure 2.16 shows two polygonal objects immobilized by 4 external contacts. External contact direction vectors have relatively low$y$-components, and hence limit the GQM inscribed sphere radius to be at most as large as the$y$-component of the wrench vectors. The individual convex hulls are significantly larger the the convex hull for the both objects together.

2.2. Determining new configuration for objects

2.2.1. Description

2.2.2. Pseudocode

Given an unordered set of objects, each object is scanned and classified: number of edges, concave vertices, edges' lengths. Scanned and classified objects allow assessment of different constellations for future grasping. Based on the algorithms



(a) Immobilized objects. (b) Convex Hull for each object and for the set (red).

Figure 2.16.: GQMs differences.

and inter-object contacts classification as presented above, several strategies for object rearrangement can be implemented.

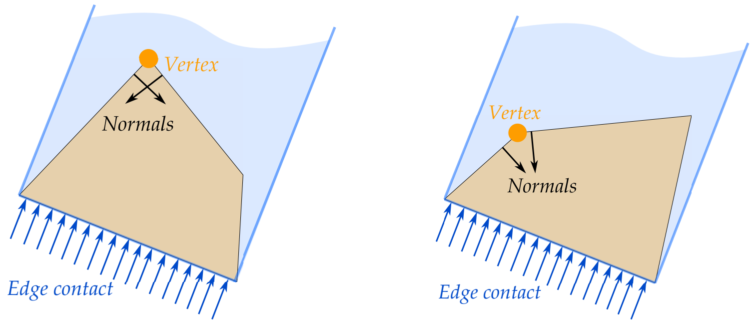
The approach for objects rearrangement is to stack objects in configuration that will require amount of minimal fingers for each object, maximizing number of inter-object effective contacts. Objects in the set in the new configuration can be described by a tree, where the root is some selected object, and all other objects described as children in a way similar to URDF style[34].

First of all, we need to know how many polygons with concave vertices exist, and how many exist with vertex angles less than 90°. Additionally, how many angles are smaller than concave angles if any.

Recapping the inter object contact types along with the additional contacts needed for the immobilization, we assume that the parent object is immobilized and inducing contact forces on a child object, which has to be immobilized by additional fingers. In the following paragraphs, term“fixation” refers to the contacts of the parent object acting on the child object.

Edge-to-edge fixation provides two parallel contacts, their convex cone can be seen as infinite “corridor”, formed by the contact region swept along the contacts’ normal direction. The interaction can be considered useful in cases where some other 2 different edges provide normal directions that are crossing in the parallel contacts’ “corridor”. Simple example would be a case with a vertex inside of the “corridor”, where normal directions of 2 adjacent edges induce opposite wrenches about the edge with two contacts. This requires at least 2 additional external contacts for each object to complete the fixture.

One way of finding such configuration is checking whether there exists an object with a vertex (concave or convex) that is projected on the interior of some non-adjacent edge (base edge), the edge can be used as a base of edge-to-edge contact, and the vertex’s adjacent edges will serve as locations for 2 additional fingers. The grasp will be feasible if normal directions from the adjacent edges and from the base edge will positively span $\mathbb{R}^2$. Example of feasible and non feasible grasps are shown in Figure 2.17.

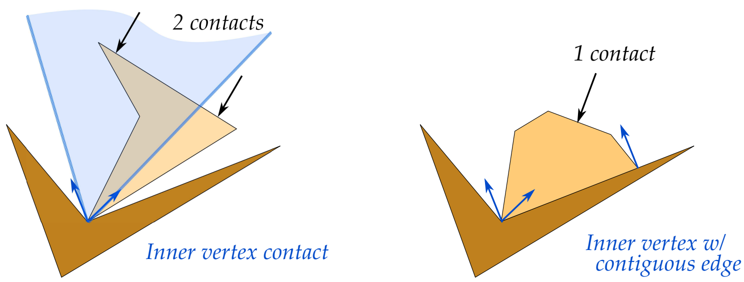


(a) Spanning vectors (b) Not spanning vectors

Figure 2.17.: Edge infinite projection and vertex with adjacent normal vectors.

Vertex-to-inner-vertexfixation provides 2 edges normal directions, which yield a convex cone with distinct vertex (unlike two parallel contacts which can be considered as convex cone with a vertex at infinity).

This fixation allows use of single edge to complete the fixture, given that the edge's inner normal direction is a negative combination of the given contacts' directions, and the edge allows two distinct contacts with opposing wrenches about the vertex (i.e. vertex is projected on the inner of the edge segment), as shown in figure Figure 2.18a. In case no such edge exist, two different edges provide normal directions which form a convex cone that mutually overlap(have a common region) the given convex cone (or their anti-cones), and the line connecting the vertices of the cones is in the interior of the overlapping region.



(a) (b)

Figure 2.18.: Inner vertex immobilization.

Vertex-to-inner-vertex with contiguous edgefixation contributes 3 effective contacts and needs only one external finger contact in the optimal case. For this there should exist an edge which projection along it's normal direction has 2-contact vertex inside it's interior and a there exists an intersecting segment of that projection and the 3rd normal direction that lies inside the convex cone of first 2 normals. Example of such configuration is shown in Figure 2.18b. If no such edge exists, 2 external contacts are required to immobilize this grasp by a method presented above

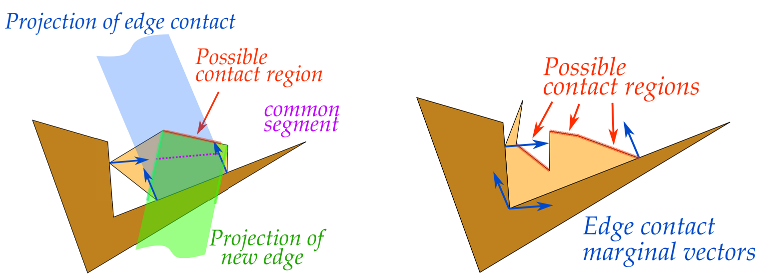
Vertex-to-inner-vertex with 2 contiguous edges fixation provides a optimal configuration which requires sole external contact for the immobilization of the object. Since objects are polygonal objects with non intersection edges and non-zero thickness, there will always exist an edge with a normal direction that will complete the form closure.

Vertex-to-inner-vertex with a contiguous edge and a contact point fixation is similar to the previous one except for the normal direction, and hence in certain cases can be self immobilizing.

Vertex-to-inner-vertex with additional contact points are equivalent to the vertex-to-inner-vertex with contiguous edges formations. The provide same amount of contacts, but the normal directions can differ.

Contiguous edge with vertex point contact fixation forms a case with 3 contacts, which allows fixation with one additional finger under certain conditions: there should exist an edge, which projections common segment with the point contact's normal has a part lying inside the edge contact's infinite projection and also the new edge, point and given edge normal directions should span $\mathbb{R}^2$.Figure 2.19a shows an example of possible contact locations for this fixation.

Example on Figure 2.19b describes 2 contiguous edges, which can be seen as sub case for the described fixation and any contact with normal direction which is negative linear combination of the normals of 2 edges will immobilize the object.



(a) Contiguous edge with point contact.

(b) 2 Contiguous edges.

Figure 2.19.: Example of preferred edge contacts.

Given all these considerations, presented set of objects can be arranged to require the minimal amount of external fingers for each object and hence minimal amount of fingers in total.

The order of the stacking is suitable for the one-by-one object manipulation. Possible inter-object fixations can be ordered by amount and restrictions on additional contacts that are to be applied.Figure 2.20 describes the order of preferable inter-object relations, based on amount of external fingers needed for the immobilization of child object relative to the parent.

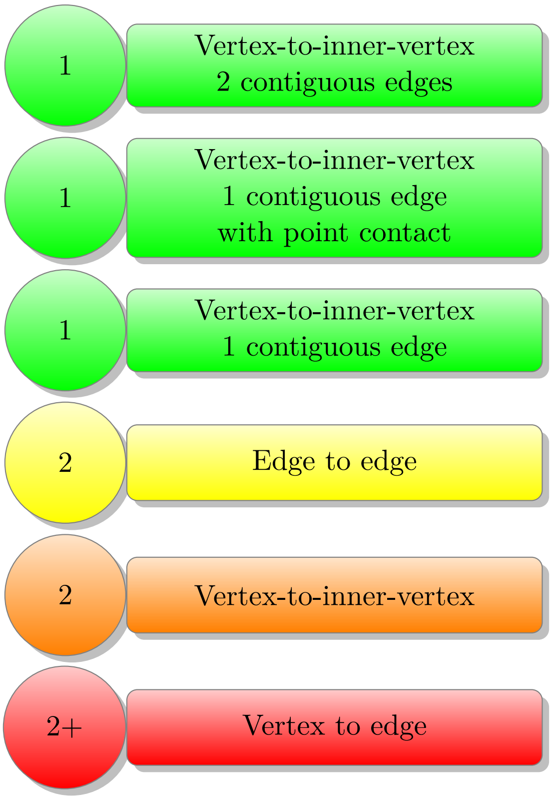


Figure 2.20.: Object placement priority.

Basically, last option (vertex to edge) will never be selected intentionally since if there is an option to place an vertex on an edge, there also is an option to place edge on edge. Vertex-to-inner vertex option also exists but for stability it is preferred to rotate the child object so it will have more parent contacts - vertex-to-inner-vertex with a contiguous edge.

First step is to define a root object, which will serve as base for further object stacking. This is done either by selecting a polygon with concave vertex, or stacking together two objects to form a concavity.

Algorithm 1: Defining the root object

Data: Set of polygons $P = p_i \; i \in [1, K]$

Result: Tree of objects including the root object and possibly one child; defining the root concavity.

if $\exists$ *1+ concave vertices* **then**
    Root object $= p_i$ with widest concave vertex;
else
    if $\exists$ *2 vertices with inner angles* $\geq 90°$ **then**
        Root $= p$ with greatest angle;
        Root.Child(k) $= p$ with second greatest angle;
        Align vertices to conctact, child leans on longest adjacent edge of the parent;
    else
        Root $= p$ with longest edge;
        Root.Child(k) $= p$ with second longest edge;
        Align longest edges to be collinear, vertex on child's longest edge coincident with midpoint of parent's longest edge.;
    end
end

When root object is determined, further stacking can be done. If there are objects with vertex angles less than the concavity angle, they are stacked together inside the concavity, possibly forming multiple concavities. If no such object exists, an object that will form a new concavity is stacked inside the root concavity.

Algorithm 2: Objects stacking

Data: Set of polygons $P = p_i \; i \in [1, K]$, Object tree T

Result: Complete tree defining the configuration of the set of of objects.

$C =$ list of concavities;
initialize C with the root concavity;
L $= \{v\}$ sorted list of polygon vertices;
foreach $v \in L$ **do**
    if $\angle(v) \leq \angle(c_i \in C)$ **then**
        Stack the child object in the c_i, *vertex to innter vertex*, edge aligned preferrably to a edge with non equal length;
        move $L.v$ to $T.v$ and save child location and orientation;
        $C.c_i =$ reduced concavity;
        $C =$ new formed concavities; `/* Upd. concavities list */`
    else
        Find smallest concavity that allows *contiguous edge with vertex contact* interaction, stack object there.
        $C =$ new formed concavities;
    end
end

** ***What if there are no concavities after several stacking steps?***

After the desired configuration is obtained, finger contact locations can be obtained using the algorithms in section 2.4. The locations that are found should be tested to complete a form closure grasp,

** ***What about:***
Description
Completeness
Complexity

2.3. Contacts search

Given a set of polygonal objects, first step is to find the contacts between the objects. The problem is subdivided into assessment of polygon pairs. The polygon interactions assessed by simple algorithm that checks whether vertices of one polygon lie on edges or on vertices of another one. Due to the complexity of the process, it is split to 2 parts: specific point on polygon evaluation and contact search for the polygons. ** ***Add part where regions allowed for finger placement are saved as well.***

The algorithm # detects whether a given point belongs to a boundary of given polygon, whether along the edge or on one of the vertices.

Algorithm 3: `p_on_Polygon`

Data: Point p, polygon P
Result: Boolean True if the point lies on polygon boundary

for e_1, e_2 = *adjacent vertices of* P **do**
    if p_i *is on segment* e_{12} **then**
        return *True*;
    end
end
return *False*;

The algorithm # is used to perform an inspection of contact points for given 2 polygons. The algorithm finds contacts between 2 polygons, returning a set that contains all contacts both the location and direction of the contact and the polygons IDs (numbers) that the contact relates to.

Algorithm 4: `C_from_2P`

Data: Polygons P_1, P_2

Result: C: Set of contacts between 2 polygons

```
C ⟵ {∅};                                  // Initialize empty contact set
for p_i = vertex of P_1 do
    if p_on_Polygon(p_i, P_2) then
        if p_i is a vertex of P_2 then
            if p_i is inner vertex of P_2 then
                V_1 = normal at p_i to adjacent edge#1 of P_2;
                V_2 = normal at p_i to adjacent edge#2 of P_2;
                C ↞ {id(P_1), id(P_2), p_i, V_1};
                C ↞ {id(P_1), id(P_2), p_i, V_2};
            else if p_i is inner vertex of P_1 then
                V_1 = normal at p_i to adjacent edge#1 of P_1;
                V_2 = normal at p_i to adjacent edge#2 of P_1;
                C ↞ {id(P_2), id(P_1), p_i, V_1};
                C ↞ {id(P_2), id(P_1), p_i, V_2};
            end
            /* append the contact to contact set TWICE, for each
               normal direction, along with with polygons' IDs */
        else
            V = normal at p_i to P_2;
            C ↞ {id(P_1), id(P_2), p_i, V};
            /* append the contact to contact set, along with
               polygons' IDs                                  */
        end
    end
end
for p_i = vertex of P_2 do
    if p_on_Polygon(p_i, P_1) and p_i is not a vertex of P_1 then
        V = normal at p_i to P_1;
        C ↞ {id(P_1), id(P_2), p_i, V};
        /* append the contact to contact set, along with
           polygons' IDs                                      */
    end
end
```

Once all contacts between all polygons are determined, remaining needed fingers placement can be done for each polygon.

Given a polygon, and a set of contacts acting on that polygon, a remaining amount of fingers needed can be determined.

The methods relies on a fact that 4 frictionless contacts is a minimum for immobilization of a two dimensional object.

2.4. Fingers placement

2.4.1. Description

Given a configuration of objects in contact (single group, each object has at least one contact), the configuration is parameterized by contact types. Each inter-object interaction is classified according to the variants defined above insubsection 2.1.1. When all contacts are found, each object is checked and an amount of additional contacts for first order form closure is derived. For every object all missing fingers are found.

Several methods for single object finger placement were developed (presented insection 1.4). For the purposes of this work 2 (??) methods were compared: moment-labeling based contact point derivation and grasp quality measure assessed random contact point placement with constraints.

Next step is to perform a check, whether any group of objects is not immobilized. A connectivity graph is build for given configuration. Connected sub-graphs are extracted from the given graph, and corresponding object constellations are tested for 1st order immobilization. If the constellation is not immobilized (see examples #ADDFIG) additional fingers are placed while the constellation is treated like a single object.

After all the variants are checked we can conclude that the group of objects is immobilized.

Steps of the algorithm.4

** *** What about masking the edges that are not available for grasp? Formal description? Flowchart***

The global finger placement process is divided to several cases of polygon contact combinations. Cases of existing 3,2 and 1 inter-object contacts are addressed. When possible, contact locations selected to yield highest grasp quality measure, namely origin centered inscribed sphere radius.

2.4.2. Pseudocode

Finding 1 finger contact given 3+ contacts on an object Given 3 contacts on an object, the viable regions for the 4th finger can be determined by evaluating for each edge whether it is possible to achieve a force closure grasp by placing a finger on that edge. When existing contacts represented by wrench space vectors, it is possible to perform assessment of the convex cone and convex hull. For a convex hull of 4 wrenches to span the origin of the wrench space, any 3 of the vectors should be linearly independent (not lying in one plane) and the 4th vector should be a negative combination of first 3, as shown insubsection 2.1.5. If there exists such wrench, corresponding contact location can be selected to be a complementary finger contact. Since on every edge could exist variety of possible contacts, simple quality measure presented insubsection 2.1.5 is used to select the most promising contact location. The algorithm is presented below:

Algorithm 5: `c4_for_P_given_3c`

Data: Polygon: P, 3 contacts: C_i , $i = 1, 2, 3$

Result: $CP_4 = \{p_i, GQM_i\}$: a set of locations for 4th contact to form desired grasp

$AntiCone = -1 \cdot Cone\left(\{w_{C_i}\}, i = 1, 2, 3\right)$;

$ConeAxis = -\sum_{i=1}^{3} w_{C_i}$;

`/* Form a cone of given contacts */`

for e_i = *edge of* P **do**

  $GW = GeneralizedWrench(e_i)$;

  `/* Create a generalized wrench for the edge */`

  $CS = CommonSegment(GW, AntiCone)$;

  `/* Find a part of the generalized wrench that is inside of the Anti Cone */`

  if $CS \neq \varnothing$ **then**

    `/* If there are wrenches that lie inside of Anti Cone determine the loci of possible contacts that will form a grasp */`

    $w_{optimal} = Projection(ConeAxis, GW)$;

    $GQM_i = GraspQualityMeasure\left(w_{C_k}|_{k=1:3}, w_{optimal}\right)$;

    `/* A vector that yields highest GRASP QUALITY MEASURE is the one that has highest projected length on the central axis of the AntiCone */`

    $CP_4 \twoheadleftarrow \{\texttt{locationFromWrench}\left(w_{optimal}\right), GQM_i\}$;

  end

end

The algorithms gives a solution for cases where 3 given contacts' wrench space vectors are linearly independent. If one of the contacts is a combination of other two - 2 marginal vectors that form the convex cone that includes the 3rd one should be chosen and passed to algorithmc3, `c4_for_P_given_2c`. This method is complete and guaranteed to find a solution if there exist one. The method is computationally inexpensive - O(n), it iterates over the number of edges at most one time. NEED SOME IMAGE HERE #ADDFIG

Set some error margins for placement uncertainty. #CHECK again the logic in this sections, see if it matches the one presented in the class on July 29.

** ***Given 3 distinct contacts on a polygonal object a search is done whether it is possible to immobilize the object by one finger.***
If 3 contacts are parallel or intersect at one point - it is impossible to form an immobilizing grasp by 1 additional finger as mentioned above (REF).
Assuming that 3 contacts are valid, the possible areas for placing a 4-th contact can be done in the following way:
Selecting two (non-parallel) contacts and form a cone along with anti-cone. Check whether a 3rd contact has a segment inside the

cone or the anti-cone or both. Check whether that segment(s) has normal projections on edges of the polygon. For every edge that got a real projection of the segment on it form a

Finding 1 finger contact for symmetric 4+ contacts In multi-object configurations, in particular in those that generated for minimal finger grasping, cases with more than 4 contacts can act on a given object. In such cases possible position of one or more additional fingers can be determined in a way similar to the algorithm presented above.

Given inter-object contacts will form a convex cone of wrench vectors. Each edge can be tested to have a segment of generalized edge wrench inside of the anti-cone of given contacts. If such segment exists, optimal finger placement location can be selected and saved along with the region allowed for finger placement. If given contacts yield a zero-volume convex cone - 2 fingers are tested to complete grasp according to the algorithm #.

Example:Figure 2.21 shows 2 bodies in contact, 4 distinct contact forces act on each body. Adding one finger to each object will immobilized it but the whole

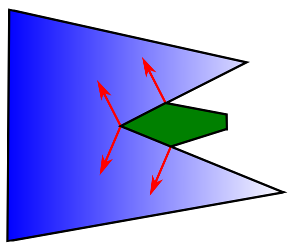


Figure 2.21.: 4 non-immobilizing contacts.

configuration will no be immobilized having only 2 external contacts. Assessment of such cases is discussed insection 2.5.

Finding 2 contacts for a case of 2 given contacts or for a flat wrench convex cone of 3+ contacts. ** $O(nlog(n))$ ***or*** $O(n^2)$ ***?*** Given 2 contacts, or 3 contacts that form a flat wrench cone, additional contacts can be found be examining wrench space vectors. For a case with 3 degenerate contacts, 2 marginal vectors are selected. Wrench space vectors and the origin form a plane (from now on“the plane”). Once such plane is determined, possible complementary contacts can be tested. First, simple one-edge solutions are proposed and evaluated. For each edge a corresponding generalized wrench is built and tested to be on both sides of the plane. If such edge found, two marginal wrenches are tested to get grasp quality measure, which is saved along with contact locations.

Algorithm 6: `c3_c4_for_P_given_2c_A`

Data: Polygon: P, 2 contacts: C_1, C_2
Result: CP_{34}: a set of location pairs for 3rd and 4th contacts to form desired grasp

$w_1, w_2 = \text{WrenchesOf}\,(C_1, C_2)$;
for e_i = *edge of* P **do**
  if $\texttt{span}\,\{\hat{n}_{C_1}, \hat{n}_{C_2}, \hat{n}_{e_i}\} = \mathbb{R}^2$ **then**
    `/* Create a generalized wrench for the edge */`
    $GW = GeneralizedWrench(e_i)$;
    `/* Find marginal wrenches of the generalized edge wrench */`
    $w_a = \min\limits_{w}\,(GW)$;
    $w_b = \max\limits_{w}\,(GW)$;
    `/* If marginal wrenches are on different sides of the plane formed by w3,w4 */`
    if $((w_1 \times w_2) \times w_a) \cdot ((w_1 \times w_2) \times w_b) < 0$ **then**
      `/* If the wrenches are on different sides they will form a convex hull around the origin, save them as potential contacts */`
      $GQM = GraspQualityMeasure\,(w_1, w_2, w_a, w_b)$;
      $CP_{34} \twoheadleftarrow \{\texttt{locationsFromWrenches}\,(w_a, w_b)\,, GQM\}$;
    end
  end
end

If no such edge found, or if higher quality measures are desired, variants of 2 edge combinations are selected to check whether they complete the convex hull of wrench vectors. Generalized wrenches are built and for each pair assessed to have a components on opposite sides of the plane. When such pair is found, corresponding regions of the edges are valid places for finger placement. A selection of contact locations can be done by extracting contact location from wrenches that have largest grasp quality measure. The algorithm is presented below:

Algorithm 7: `c3_c4_for_P_given_2c_B`

Data: Polygon: P, 2 contacts: C_1, C_2
Result: CP_{34}: a set of location pairs for 3rd and 4th contacts to form desired grasp

$w_1, w_2 = \text{WrenchesOf}\,(C_1, C_2)$;
$\hat{n}_{plane12} = w_1 \times w_2$;
for e_i, e_j = *edge pairs of P 2-edge variants list* **do**
  if $\mathtt{span}\,\{\hat{n}_{C_1}, \hat{n}_{C_2}, \hat{n}_{e_i}\} = \mathbb{R}^2$ **then**
    `/* Create a generalized wrench for each edge */`
    $GW_i = GeneralizedWrench(e_i)$;
    $GW_j = GeneralizedWrench(e_j)$;
    if *GW_i is fully on one side of the plane* **then**
      `/* Select first wrench the farthest from the plane */`
      $w_a = \underset{w}{\text{argmax}}\, |w \cdot \hat{n}_{plane12}|\,,\ w \in GW_i$;
      `/* Select second wrench the farthest from the plane from the other side */`
      $w_b = \underset{w}{\text{argmax}}\, |-\mathtt{sign}\,(w \cdot \hat{n}_{plane12})\, w \cdot \hat{n}_{plane12}|\,,\ w \in GW_j$;
      $GQM = GraspQualityMeasure\,(w_a, w_b)$;
      $CP_{34} \twoheadleftarrow \{\mathtt{locationFromWrenches}\,(w_a, w_b)\,, GQM\}$;
    else
      `/* Both GW's are on both sides of the plane */`
      $w_a = \underset{w}{\text{argmax}}\, (\, w \cdot \hat{n}_{plane12})\,,\ w \in \{GW_i, GW_j\}$;
      $w_b = \underset{w}{\text{argmax}}\, (\, -w \cdot \hat{n}_{plane12})\,,\ w \in \{GW_i, GW_j\}$;
      `/* Select maximally distant wrench on each side */`
      $GQM = GraspQualityMeasure\,(w_1, w_2, w_a, w_b)$;
      $CP_{34} \twoheadleftarrow \{\mathtt{locationFromWrenches}\,(w_a, w_b)\,, GQM\}$;
    end
  end
end

Finding 3 finger contacts for 1 inter-object contact This case is the most undefined and require more decisions as to finger placement locations. Some of the proposed methods for object immobilization can be applied for this case with minor adjustments. The method presented below is influenced by methods proposed byWu [38] for IRC evaluation and[25] compliance and grasp polygons.

First, we define a geometric 2 dimensional convex cone: a planar angle between two vectors. The cone can be fully defined by a vertex point and 2 vectors.Total force applied by 2 contacts can be defined to lie in such cone that is formed by two contact normal directions. This is analogous to friction cone of single frictional contact (seeNguyen [24]). The main idea is to form 2 imaginary convex cones each formed by 2 normal vectors. The line connecting the cones' vertices should lie inside of the intersection of 2 cones (or intersection of anti-cones of these 2 cones). If the connecting line lies inside - the grasp will be force closure.

For given problem, one contact is given and non movable, other 3 can be placed anywhere desired. Contact 2 is placed on arbitrary edge in a position that ensures that the cone ($\triangleq ConeA$) formed by it and so$ConeA$ will have the most area of intersection with the polygon. Afterwards other edges tested to form a cone that will have a common intersection with $ConeA$, and this way it will complete the grasp. If there is no intersection between two cones, intersection between anti-cones is tested.

Given a polygon with one inter-polygon contact:

Two variants of the algorithm presented, where first one is testing simple combinations with lower computational complexity, and second one is more robust but requires more computations.

Algorithm 8: `c2, c3, c4_for_P_given_c  Variant_A`

Data: Polygon: P, contact: C_1

Result: CP_{234}: a set of location triplets for 2nd, 3rd and 4th contacts to form desired grasp

for e_i = *edge of* P **do**
    $p =$ `projection_a_on_b_along_c`$(e_i, \hat{n}_{C_1}, \hat{n}_{e_i})$;
    for e_j = *edge of* P, $j \neq i$ **do**
        if $\vec{0} \in ConvexHull\left(\hat{n}_{e_j}, \hat{n}_{e_i}, \hat{n}_{C_1}\right)$ **then**
            `/* The edge can possibly provide 2 contacts to form a force closure grasp. */`
            $p34 =$ `proj_a_on_b_along_c`$(e_i, \hat{n}_{C_1}, \hat{n}_{e_i})$;
            $p2 =$ `proj_a_on_b_along_c`$(p, e_j, \hat{n}_{e_j})$;
            if $|p2 \cup p34| > 1$ **then**
                `/* The projections intersection is more than one point */`
                $m =$ `mean`$(p2 \cup p34)$;
                $l_2 =$ `proj_a_on_b_along_c`$(m, e_i, \hat{n}_{e_i})$;
                $s1, s2 =$ `segment_divided_by_point`$(p34, m)$;
                $l_3 =$ `proj_a_on_b_along_c`$\left(\texttt{mean}(s1), e_j, \hat{n}_{e_j}\right)$;
                $l_4 =$ `proj_a_on_b_along_c`$\left(\texttt{mean}(s2), e_j, \hat{n}_{e_j}\right)$;
                $GQM = GraspQualityMeasure\left(w_{C_1}, w_{l_2}, w_{l_3}, w_{l_4}\right)$;
                $CP_{234} \twoheadleftarrow \{l_2, l_3, l_4, GQM\}$;
            end
        end
    end
end

If no suitable combinations found by the algorithm presented above, following algorithm can be used:

Algorithm 9: `c2, c3, c4_for_P_given_c Variant_B`

Data: Polygon: P, contact: C_1

Result: CP_{234}: a set of location triplets for 2nd, 3rd and 4th contacts to form desired grasp

```
for e_i = edge of P do
    p = projection_a_on_b_along_c(e_i, n̂_{C_1}, n̂_{e_i});
    for e_j = edge of P, j ≠ i do
        for e_k = edge of P, k ≠ i and k ≠ j do
            if 0⃗ ∈ ConvexHull(n̂_{e_i}, n̂_{e_j}, n̂_{e_k}, n̂_{C_1}) then
                /* Since 3 edges along given contact can form a
                   convex hull that containts the origin, we can
                   proceed to determine preferred locations along
                   these edges.                                   */
                p_{e_i} = argmin_{p∈e_i} (p · n̂_{C_1});
                p_{e_j} = argmin_{p∈e_j} (p · n̂_{e_k});
                p_{e_k} = argmin_{p∈e_k} (p · n̂_{e_j});
                /* Find bounding points that yield maximum cone
                   intersection area                              */
                v_1 = intersection(p_{C_1} + n̂_{C_1}, p_{e_i} + n̂_{e_i});
                v_2 = intersection(p_{e_j} + n̂_{e_j}, p_{e_k} + n̂_{e_k});
                ConeA = ConvexCone(v = v_1, n_a = n̂_{C_1}, n_b = n̂_{e_i});
                ConeB = ConvexCone(v = v_2, n_a = n̂_{e_j}, n_b = n̂_{e_k});
                if (v_1 − v_2) ∈ (ConeA ∩ ConeB) ∪ (−ConeA ∩ −ConeB)
                 then
                    /* The line connecting cone vertices is inside
                       the cone or anti-cone intersection means
                       that the grasp is force closure / torque
                       closure.  Now contact locations can be
                       selected.                                  */
                    l_2 = proj_a_on_b_along_c(0.3 · (v_1 − v_2), e_i, n̂_{e_i});
                    l_3 = proj_a_on_b_along_c(0.3 · (v_2 − v_1), e_j, n̂_{e_j});
                    l_4 = proj_a_on_b_along_c(0.3 · (v_2 − v_1), e_k, n̂_{e_k});
                    GQM = GraspQualityMeasure(w_{C_1}, w_{l_2}, w_{l_3}, w_{l_4});
                    CP_{234} ↞ {l_2, l_3, l_4, GQM};
                end
            end
        end
    end
end
```

Once all polygons are treated, desired contacts should be selected. Since all contact positions are saved along with grasp quality measure values, it is possible to select the contacts that provide better grasp. While maintaining

Flowchart for object treatment given contacts.

What about a restrictions on finger placement that arise from close objects? Should it be discussed and treated?? #

** ***DEAL WITH THIS PART LATER***

2.4.3. Completeness

t

2.4.4. Complexity

t

2.5. Multi object grasp evaluation and grasp securing

Once every given object is immobilized by other objects and fingers, the constellation still might be not fully immobilized. Example:Figure 2.22 shows 3 objects: black object is immobilized by 4 contacts from red and blue objects while red and and blue are immobilized by two additional fingers each one. The contacts that act on blue and black objects (shown as red arrows) intersect at same point and hence do not immobilize these two objects together – namely, these two objects can rotate together about the contact intersection point. Total immobilization

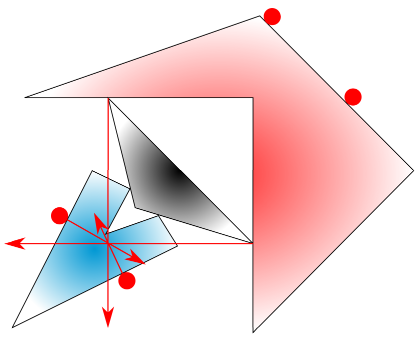


Figure 2.22.: 2 of 3 objects are not immobilized.

can be achieved by ensuring that every possible subset of objects is immobilized. This requires extraction of all possible connected subsets from the given set object and evaluation whether they are or are not immobilized. Each constellation ought to have at least 4 external contacts. If there are less - additional contacts required to immobilize it. If the amount of external contacts is higher than 4, assessment of intersection of contacts external to the subset can show whether the subset is immobilized or not. If the contacts intersect at one point, contact

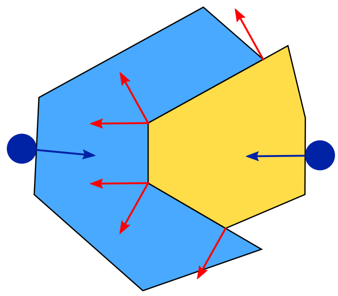


Figure 2.23.: 2 objects set is not immobilized.

positions of added fingers can be altered in a allowed regions that were found for that contact.

Algorithm 10: Multi object grasp evaluation

Data: Set of polygons $P = p_i \; i \in [1, K]$ including locations and orientations, Set of contacts $C = \{c_i\}$ with allowed regions
Result: Set of contact adjustments if needed, set of additional contacts if needed

foreach *subset of object configuration* **do**
    L = $\{lc_i\}$ – set of external contact lines;
    if $|L| < 4$ **then**
        Treat the subset as one object;
        if $|L| = 3$ *and* $\exists x \in lc_i \quad \forall lc_i \in L$ **then**
            Move 1 finger in allowed region;
        end
        Add missing contacts (max 2);
    end
    if $\exists x \in lc_i \quad \forall lc_i \in L$ **then**
        Intersection point between contact lines exists;
        Build 2 EGWs for 2 non-collinear contacts and adjust positions;
    else if $\exists x_1 \in lc_j \quad j = 1, 2, 3$ **then**
        3 of the contacts are intersecting;
        if $\exists x_2 \in lc_k, \notin lc_j \quad |\{lc_k\}| > 1$ **then**
            Exists another point where two or more contact lines intersect;
            continue;
        end
        Build 2 EGWs for 2 non-collinear lc_j and adjust positions;
    else
        continue;
    end
    ]
end

2.6. Objects rearrangement

Once desired configuration is found the rearrangement planning is needed. Many algorithms for single object manipulations were developed, but in this case due to several objects to be moved. Objects placement algorithms are designed in a way that will allow stacking the objects in the defined order.

Objects can be moved individually by executing any planning algorithm for single object.Akella and Mason [1], Bernheisel and Lynch [4], Lynch [18, 19] present methods for moving objects in assemblies in plane. Simultaneous manipulation of several objects is out of the scope of this research, and it is assumed that there is a way to rearrange the objects from given configuration to the desired one.

For the purposes of simulations and experiments, objects are scattered in the workspace at distances that allow for each object to be repositioned at desired location. Each manipulation is defined by base object and moved object. Objects are grasped and moved individually while maintaining finger positions that will not interfere with adjacent objects. Alternatively, if no such option exists, the object can be positioned at a pre-pose and pushed from there.

* **** Should I mention this ?:*** Objects are moved by stable pushing which allows translation and rotation of the object. Geometric center assumed to be the center of rotation.

Base object is an object or a set of objects that serve as a base for a moved object to be placed in contact to.

2.7. Complexity

* **** What in hell do I write here?***

2.8. Limitations

* **** What can I write here?***

Limitations of the finger placement? Finger sizes, proximity to the vertex – dealing with uncertainty. While rearranging:

Limitation of the multiobject evaluation? Object placement?

Quality measure contradictions: what's best for one polygon not necessary is best for polygon constellation.

3. Results

The chapter presents the results of the algorithms performance.** ***Duh!***

3.1. Simulations

Geometrical simulations were performed in MATLAB environment. The simulations included assessment of polygonal objects, rearrangement algorithm evaluations and finger placement algorithms execution. The obtained results are tested for consistency, and mathematically assessed for the completeness of the solution. Several methods are compared and comparison results are presented.

3.1.1. Software

MATLAB computing environment was used for the purpose of geometrical simulations. Specialized graphical user interface (GUI) programs were created for evaluation and presentation of the algorithms[1].

3.1.2. Setup

V-REP environment was used for physical simulation of the performance of the algorithms. The software incorporated several common physics engines which allow testing of different scenarios of computer simulations.

Delta-robot model of && was used to perform as the main manipulator, custom end effector was added to the model to perform grasping of desired object sets.

The end-effector consists of 10 individually actuated fingers, which can move in the“palm” plane freely and have limited movement in Z direction. Depending on the found solution, the desired number of fingers can be extended to execute the grasp.

Objects’ location and orientation is obtained programmatically, while in real life scenario can be done by different means (e.g. RGB or 3D cameras, tracking cameras, resistive or capacitive surfaces etc.)

Control of the robot is done in closed loop by internal means of the V-REP software, while the control of the algorithm steps is done in the specialized GUI. The simulation is divided into several steps:

1. Obtaining environment information:
    a) Number of objects

[1] The programs are available athttps://github.com/yossioo/MSc-Research.

   b) Positions
   c) Shapes

2. Obtaining the desired constellation variants
   a) variants are shown in preview pane along with quality measures
   b) desired variant can be selected for the execution

3. Rearranging the objects
   a) objects are moved by pushing algorithms to obtain the desired configuration

4. Immobilizing the objects
   a) Required amount of fingers are selected and extended to perform the grasp
   b) manipulator moves above the objects
   c) Manipulator and the end effector are lowered down to constrain the constellation

5. Immobilization tests
   a) moving in plane
   b) moving to vertical plane and rotating

Executions of the steps can be controlled from the UI.

3.1.3. Results

A simple case of 3 equilateral triangles presented in the *SIMULATIONS. The designed configuration and designed grasp are presented above. 3D printed end effector is presented on the FIGURE. Due to*

A case of 6 equilateral triangles

A case of 3 squares

A case of 4 squares

A case of 3 hexagons

Cases of 3-4-5-... irregular triangles.

A case of 3-4-5 octagons?

3.2. Experiments

Bring a 3D printed set of objects and designed gripper to the defense. Include the results in the Appendix, and in the presentation

The results of the algorithms' performance were tested in experimental setups. Robotic manipulator with 6DOF was equipped with an end effector designed according to the algorithm's output. The robot is controlled by a laptop with Robot Operating System (ROS) hosted in Ubuntu. Several polygonal objects were selected to be stacked, and after the verification by the algorithm a custom-made end effector with stationary fingers was manufactured by 3D printing. The end effector was tested and confirmed to immobilize the desired set of objects both in simulation and in experiments. Due to the real world imperfections, the fingers of the designed end effectors are made with chamfers, this allowed grasping of the arranged set of items despite small perturbations of item locations and robotic arm performance.

3.2.1. Experimental Setup

For several examined cases of polygon sets,

3.2.2. Experimental Results

A simple case of 3 equilateral triangles presented in the SIMULATIONS. The designed configuration and designed grasp are presented above. 3D printed end effector is presented on the FIGURE. Fingers - 1 joint finger - touches 2 triangles.

A case of 6 equilateral triangles

A case of 3 hexagons

Cases of 3-4-5-... irregular triangles.

4. Discussion

Written before Conclusions

5. Conclusions

Written before the Introduction

Bibliography

[1] Akella, S. and Mason, M. T. (1998). Posing polygonal objects in the plane by pushing. *The International Journal of Robotics Research*, 17(1):70–88.

[2] Asada, H. and Kitagawa, M. (1989). Kinematic analysis and planning for form closure grasps by robotic hands. *Robotics and Computer-Integrated Manufacturing*, 5(4):293 – 299. Special Issue Simulation Software for Robotics.

[3] Ball, R. (1998). *A Treatise on the Theory of Screws*. Cambridge Mathematical Library. Cambridge University Press.

[4] Bernheisel, J. D. and Lynch, K. M. (2004). Stable transport of assemblies: pushing stacked parts. *IEEE Transactions on Automation Science and Engineering*, 1(2):163–168.

[5] Bezak, P., Bozek, P., and Nikitin, Y. (2014). Advanced robotic grasping system using deep learning. *Procedia Engineering*, 96:10 – 20. Modelling of Mechanical and Mechatronic Systems.

[6] Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: A review. pages 348–353.

[7] Borst, C., Fischer, M., and Hirzinger, G. (2003). Grasping the dice by dicing the grasp. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 4, pages 3692–3697 vol.3.

[8] Borst, C., Fischer, M., and Hirzinger, G. (2004). Grasp planning: how to choose a suitable task wrench space. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 319–325 Vol.1.

[9] Bron, C. and Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577.

[10] Bunis, H. A., Rimon, E. D., Allen, T. F., and Burdick, J. W. (2018). Equilateral three-finger caging of polygonal objects using contact space search. *IEEE Trans. Automation Science and Engineering*, 15(3):919–931.

[11] Corke, P. (2011). *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer.

[12] Ferrari, C. and Canny, J. (1992). Planning optimal grasps. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2290–2295 vol.3.

[13] Gouda, K. and Hassaan, M. (2010). A fast algorithm for subgraph search problem.

[14] Landau, L. and Lifshitz, E. (1982). *Mechanics*. Number v. 1. Elsevier Science.

[15] Le, Q. V., Kamm, D., Kara, A. F., and Ng, A. Y. (2010). Learning to grasp objects with multiple contact points. In *2010 IEEE International Conference on Robotics and Automation*, pages 5062–5069.

[16] Lin, Q., Burdick, J., and Rimon, E. (1997). A quality measure for compliant grasps. In *Proceedings of International Conference on Robotics and Automation*, volume 1, pages 86–92 vol.1.

[17] Lozano-Perez, T. (1983). Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120.

[18] Lynch, K. (1992). The mechanics of fine manipulation by pushing. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 3, pages 2269–2276. Publ by IEEE.

[19] Lynch, K. M. (1999). Locally controllable manipulation by stable pushing. *IEEE Transactions on Robotics and Automation*, 15(2):318–327.

[20] Mishra, B., Schwartz, J., and Sharir, M. (1987). On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2(1-4):541–558.

[21] Mishra, B. and Silver, N. (1989). Some discussion of static gripping and its stability. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(4):783–796.

[22] Montana, D. J. (1988). The kinematics of contact and grasp. *The International Journal of Robotics Research*, 7(3):17–32.

[23] Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

[24] Nguyen, V. D. (1986a). Constructing force-closure grasps. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1368–1373.

[25] Nguyen, V. D. (1986b). The synthesis of stable grasps in the plane. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 884–889.

[26] Pajarinen, J. and Kyrki, V. (2017). Robotic manipulation of multiple objects as a pomdp. *Artificial Intelligence*, 247:213 – 228. Special Issue on AI and Robotics.

[27] Pollard, N. S. (1996). Synthesizing grasps from generalized prototypes. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2124–2130 vol.3.

[28] Quillen, D., Jang, E., Nachum, O., Finn, C., Ibarz, J., and Levine, S. (2018). Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. *CoRR*, abs/1802.10264.

[29] Reuleaux, F. (1876). *The Kinematics of Machinery: Outlines of a Theory of Machines. German original (1875). Translated by A. Kennedy.* MacMillan and Co., London.

[30] Rimon, E. and Burdick, J. (1996). On force and form closure for multiple finger grasps. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1795–1800 vol.2.

[31] Rimon, E. and Burdick, J. W. (1995). New bounds on the number of frictionless fingers required to immobilize 2d objects. In *Proceedings of the 1995 International Conference on Robotics and Automation, Nagoya, Aichi, Japan, May 21-27, 1995*, pages 751–757.

[32] Rimon, E. and Burdick, J. W. (1998). Mobility of bodies in contact. i. A 2nd-order mobility index for multiple-finger grasps. *IEEE Trans. Robotics and Automation*, 14(5):696–708.

[33] Roa, M. A. and Suárez, R. (2015). Grasp quality measures: Review and performance. *Auton. Robots*, 38(1):65–88.

[34] ros.org (2009). Urdf.

[35] Siciliano, B. and Khatib, O. (2016). *Springer Handbook of Robotics*. Springer Publishing Company, Incorporated, 2nd edition.

[36] Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42.

[37] Wren, D. and Fisher, R. B. (1995). Dextrous hand grasping strategies using preshapes and digit trajectories. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 1, pages 910–915 vol.1.

[38] Wu, Y. (2006). A geometric approach to the four-finger grasping planning of polygonal objects. In *2006 International Technology and Innovation Conference (ITIC 2006)*, pages 2144–2148.

[39] Xiong, C., Ding, H., and Xiong, Y.-L. (2007). *Fundamentals of Robotic Grasping and Fixturing*, volume 3. World Scientific.

[40] Xydas, N. and Kao, I. (1999). Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results. *The International Journal of Robotics Research*, 18(9):941–950.

[41] Zeng, A., Song, S., Yu, K., Donlon, E., Hogan, F. R., Bauzá, M., Ma, D., Taylor, O., Liu, M., Romo, E., Fazeli, N., Alet, F., Dafle, N. C., Holladay, R., Morona, I., Nair, P. Q., Green, D., Taylor, I., Liu, W., Funkhouser, T. A., and Rodriguez, A. (2017). Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *CoRR*, abs/1710.01330.

Appendix

A. Algorithms

This section presents more elaborate representation of the algorithms presented in the paper.

B. Software

Source code is available athttps://github.com/yossioo/MSc-Research.

Additional videos are available at Youtube channel: &&&

The software used for the research is:

MATLAB r2018a

V-REP 3.5.0 Educational Version

ROS Kinetic on Ubuntu 16.04

C. Examples of calculated grasps

3 fingers - 1 found

D. Examples of rearranged object sets

E. Examples of finger combinations found for sets

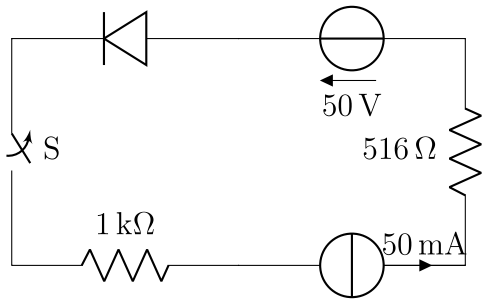
F. Simulations

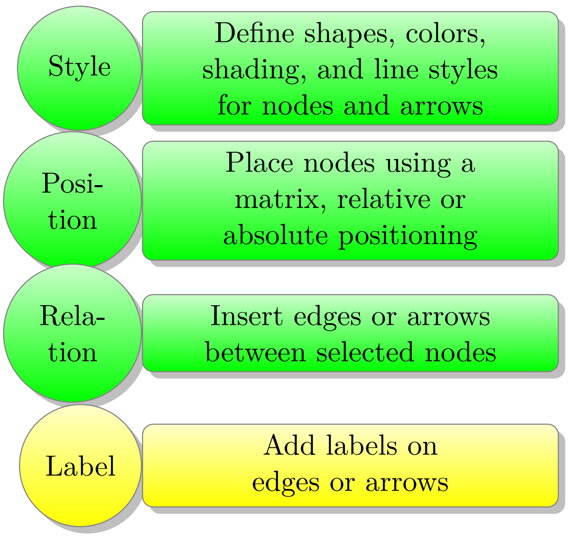
V-REP

G. Experiments

Real robot

H. Junk





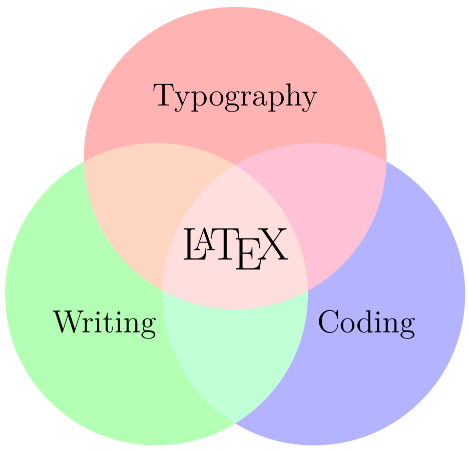
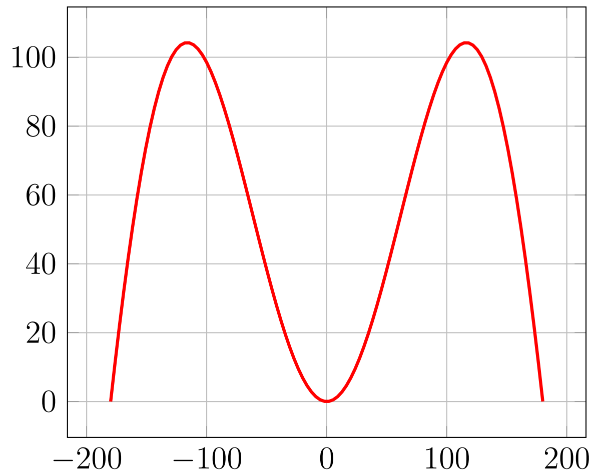


Figure H.1.: 222



XY example

$$\begin{tikzcd}
\text{Some text here} \arrow[drr] & \cos(\theta) & e \\
c \arrow[ur] \arrow[rr, "Over", "Under"'] & & f
\end{tikzcd}$$