

Geometric Guides for Interactive Evolutionary Design

By

Theodora Retzepi

A Doctoral Thesis submitted in partial fulfilment of the requirements
for the award of Doctor of Philosophy of Loughborough University

August 2018

Loughborough University
Wolfson School of Mechanical, Electrical and
Manufacturing Engineering

© Theodora Retzepi 2018

Abstract

This thesis describes the addition of novel 'Geometric Guides' to a generative Computer-Aided Design (CAD) application that supports early-stage concept generation. The application generates and evolves abstract 3D shapes, used to inspire the form of new product concepts. It was previously a conventional Interactive Evolutionary system where users selected shapes from evolving populations. However, design industry users wanted more control over the shapes, for example by allowing the system to influence the proportions of evolving forms. The solution researched, developed, integrated and tested is a more cooperative human-machine system combining classic user interaction with innovative geometric analysis.

In the literature review, different types of Interactive Evolutionary Computation (IEC), Pose Normalisation (PN), Shape Comparison, and Minimum-Volume Bounding Box approaches are compared, with some of these technologies identified as applicable for this research.

Using its Application Programming Interface, add-ins for the Siemens NX CAD system have been developed and integrated with an existing Interactive Evolutionary CAD system. These add-ins allow users to create a Geometric Guide (GG) at the start of a shape exploration session. Before evolving shapes can be compared with the GG, they must be aligned and scaled (known as Pose Normalisation in the literature).

Computationally-efficient PN has been achieved using geometric functions such as Bounding Box for translation and scaling, and Principle Axes for the orientation. A shape comparison algorithm has been developed that is based on the principle of non-intersecting volumes. This algorithm is also implemented with standard, readily available geometric functions, is conceptually simple, accessible to other researchers and also offers appropriate efficacy.

Objective geometric testing showed that the PN and Shape Comparison methods developed are suitable for this guiding application and can be efficiently adapted to enhance an Interactive Evolutionary Design system. System performance with different population sizes was examined to indicate how best to use the new guiding capabilities to assist users in evolutionary shape searching. This was backed up by participant testing research into two user interaction strategies. A Large Background Population (LBP) approach where the GG is used to select a sub-set of shapes to show to the user was shown to be the most effective.

The inclusion of Geometric Guides has taken the research from the existing aesthetic focused tool to a system capable of application to a wider range of engineering design problems. This system supports earlier design processes and ideation in conceptual design and allows a designer to experiment with ideas freely to interactively explore populations of evolving solutions. The design approach has been further improved, and expanded beyond the previous quite limited scope of form exploration.

Certificate of Originality



Certificate of Originality Thesis Access Conditions and Deposit Agreement

Students should consult the guidance notes on the electronic thesis deposit and the access conditions in the University's Code of Practice on Research Degree Programmes

Author...Theodora Retzepl.....

Title...Geometric Guides for Interactive Evolutionary Design.....

I Theodora Retzepl, of Flat 6 Granger Court, Oxford Street, Loughborough, "the Depositor", would like to deposit 'Geometric Guides for Interactive Evolutionary Design', hereafter referred to as the "Work", once it has successfully been examined in Loughborough University Institutional Repository

Status of access OPEN / ~~RESTRICTED~~ / ~~CONFIDENTIAL~~

Moratorium Period.....years, ending...../.....20.....

Status of access approved by (CAPITALS):...DR IAN J GRAHAM.....

Supervisor (Signature)..........

School of...Mechanical, Electrical and Manufacturing Engineering.....

Author's Declaration I confirm the following :

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this thesis, that the original work is my own except as specified in acknowledgements or in footnotes, and that neither the thesis nor the original work therein has been submitted to this or any other institution for a degree

NON-EXCLUSIVE RIGHTS

The licence rights granted to Loughborough University Institutional Repository through this agreement are entirely non-exclusive and royalty free. I am free to publish the Work in its present version or future versions elsewhere. I agree that Loughborough University Institutional Repository administrators or any third party with whom Loughborough University Institutional Repository has an agreement to do so may, without changing content, convert the Work to any medium or format for the purpose of future preservation and accessibility.

DEPOSIT IN LOUGHBOROUGH UNIVERSITY INSTITUTIONAL REPOSITORY

I understand that open access work deposited in Loughborough University Institutional Repository will be accessible to a wide variety of people and institutions - including automated agents - via the World Wide Web. An electronic copy of my thesis may also be included in the British Library Electronic Theses On-line System (EThOS). I understand that once the Work is deposited, a citation to the Work will always remain visible. Removal of the Work can be made after discussion with Loughborough University Institutional Repository, who shall make best efforts to ensure removal of the Work from any third party with whom Loughborough University Institutional Repository has an agreement. Restricted or Confidential access material will not be available on the World Wide Web until the moratorium period has expired.

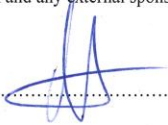
- That I am the author of the Work and have the authority to make this agreement and to hereby give Loughborough University Institutional Repository administrators the right to make available the Work in the way described above.

- That I have exercised reasonable care to ensure that the Work is original, and does not to the best of my knowledge break any UK law or infringe any third party's copyright or other Intellectual Property Right. I have read the University's guidance on third party copyright material in theses.
- The administrators of Loughborough University Institutional Repository do not hold any obligation to take legal action on behalf of the Depositor, or other rights holders, in the event of breach of Intellectual Property Rights, or any other right, in the material deposited.

The statement below shall apply to ALL copies:

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Restricted/confidential work: All access and any copying shall be strictly subject to written permission from the University Dean of School and any external sponsor, if any.

Author's signature..........Date.....06/8/2018.....

user's declaration: for signature during any Moratorium period (Not Open work): <i>I undertake to uphold the above conditions:</i>			
Date	Name (CAPITALS)	Signature	Address

Acknowledgements

This thesis is dedicated to Laura and our son Theodore...!

I would like to thank the people who have helped me through this intensive period, whose support over these three years has been an integral part of my journey.

Firstly, I would like to thank Laura for her love and patience. You have always been there for me, willing to help and motivate me to become better. This work would not have been that good without your unending support and inspiration. I was amazed by your willingness to proof read my thesis, and by your patience with all of the ups and downs of my research. Sharing my life with you has made me become a more responsible person, I really feel my work has benefitted from all of this.

I am especially grateful to my supervisor, Dr Ian Graham, for his advice, responsiveness and his superb guidance in the successful completion of my thesis. I feel our excellent cooperation and teamwork enabled me to exponentially grow and develop my coding ability. In addition, I would like to thank my co-Supervisor, Dr Mey Goh, for her invaluable feedback, guidance, challenging questions and insightful comments – all of which contributed to the development of my thesis and my personal growth.

I would also like to thank my parents and my brother, whose unconditional love and guidance have been a constant – throughout my life, in whatever venture I pursue. Your emotional and financial support enabled me to follow my dreams to move to England and seize the opportunity of a PhD scholarship.

Finally, I would like to thank the EPSRC for providing the funding which allowed me to undertake this research and for giving me the opportunity to attend conferences. This has truly been a life changing experience.

Contents

Abstract	i
Certificate of Originality	ii
Acknowledgements.....	iv
Contents.....	v
List of Figures	ix
List of Tables	xxi
List of Abbreviations	xxiii
List of Nomenclature	xxv
1 Introduction.....	1
1.1 Computer-Aided Design.....	1
1.2 The need for Interactive Evolutionary CAD	3
1.3 Wider research vision	4
1.4 Aims and Objectives.....	4
1.5 Novelty.....	5
1.6 Contribution to Knowledge.....	6
1.7 Thesis Structure	6
2 Literature Review	8
2.1 Overview of generative CAD systems.....	8
2.2 Evolutionary Optimisation	9
2.2.1 <i>Evolutionary Algorithms</i>	10
2.2.2 <i>Hybrid Simulated Annealing and Genetic Algorithm</i>	15
2.2.3 <i>Evolutionary Dynamic Optimisation</i>	17
2.3 Interactive Evolutionary Optimisation.....	18

2.3.1	<i>Pareto-optimal points Interaction Strategies</i>	21
2.3.2	<i>Efficient Interaction Strategies</i>	24
2.4	Interactive Evolutionary Design (IED)	26
2.4.1	<i>ICE-GCAD</i>	27
2.4.2	<i>IDSET</i>	27
2.4.3	<i>Machine Learning in IED</i>	29
2.4.3.1	Aesthetic descriptors in IED	31
2.4.3.2	Human preference in evolutionary art	33
2.4.4	<i>EvoShape</i>	34
2.5	Geometry Mathematics	39
2.6	Pose Normalisation	42
2.6.1	<i>Reflective Symmetry</i>	43
2.6.2	<i>Pairwise Alignments</i>	46
2.6.3	<i>Pose Normalisation for non-rigid shapes</i>	47
2.6.4	<i>Principal Component Analysis Methods</i>	49
2.6.5	<i>Combined Pose Estimation (CPE)</i>	51
2.6.6	<i>Minimum Projection Area (MPA)</i>	53
2.6.7	<i>Planar-Reflective Symmetry Transform</i>	54
2.6.8	<i>Partial Symmetry-Based Alignment</i>	56
2.6.9	<i>The Spherical Trace Transform</i>	58
2.7	Minimum-Volume Bounding Box	59
2.7.1	<i>Approximation algorithms of a point set in three dimensions</i>	60
2.7.2	<i>Shape approximation in robot grasping</i>	61
2.7.3	<i>Iterative approach of an arbitrary solid</i>	63
2.7.4	<i>Finding minimal enclosing boxes</i>	65
2.8	Shape comparison	66
2.8.1	<i>Dimension Amnestic Pyramid Match Kernel</i>	66
2.8.2	<i>Geometric comparison</i>	69
2.8.3	<i>Comparison based on 3D morphing</i>	71
2.9	Technology Review Conclusions	73
2.9.1	<i>Knowledge Gap Identification</i>	75

3	Methodology	76
3.1	Programming Environment.....	76
3.2	Objective testing	76
3.3	Subjective testing.....	79
3.3.1	<i>Number of Test Participants</i>	<i>79</i>
3.4	Summary	81
4	3D Alignment and Shape Comparison Analysis.....	82
4.1	Shape Matching Introduction	83
4.1.1	<i>Introduction to Geometric Guides</i>	<i>83</i>
4.1.2	<i>Introduction to Pose Normalisation</i>	<i>84</i>
4.1.3	<i>Introduction to Shape Comparison</i>	<i>85</i>
4.2	Detailed Analysis.....	86
4.2.1	<i>Rotation</i>	<i>86</i>
4.2.2	<i>Scaling.....</i>	<i>88</i>
4.2.3	<i>Translation</i>	<i>90</i>
4.3	3D Alignment and Shape Comparison Conclusions.....	92
5	User Interaction Strategies	93
5.1	Introduction to user interaction strategies	93
5.2	Detailed Analysis.....	94
5.2.1	<i>Initial GG analysis and benchmarking with standard-sized population</i>	<i>94</i>
5.2.2	<i>Large Background Populations.....</i>	<i>106</i>
5.3	User Interaction Strategies Conclusions.....	119
6	Participant Testing.....	121
6.1	Introduction to participant testing	121
6.2	Geometric Guides with Large Background Populations	121
6.3	Geometric Guides with Sporadic User Interaction	135
6.4	Discussion.....	146
6.5	Participant Testing Conclusions	147
7	Conclusions and Future Work	148

7.1	Conclusions	148
7.2	Future Work	151
References		153
A)	Appendix: Calculate fitness.....	A.1
B)	Appendix: Geometry	B.1
C)	Appendix: Rotation and scaling with BB	C.1
D)	Appendix: Import target	D.1
E)	Appendix: Compare values	E.1
F)	Appendix: Convert to fitness	F.1
G)	Appendix: Subset of shapes to the user	G.1
H)	Appendix: Detailed benchmark Example.....	H.1
I)	Appendix: Participants' Feedback.....	I.1
J)	Appendix: LBP Functions Map	J.7
K)	Appendix: SUI Functions Map.....	K.1
L)	Appendix: Publications	L.1

List of Figures

Figure 2.1. Population, Chromosomes and Genes.	11
Figure 2.2. Crossover operation to generate offspring.	12
Figure 2.3. A chromosome before and after mutation.	12
Figure 2.4. Schematic diagram of natural evolutionary systems (Elbeltagi et al., 2005).	14
Figure 2.5. The red arrow on the left shows the best solution. But if a simple algorithm finds its way to where the green arrow on the right shows, it will get trapped in there: all of the neighboring solutions are worse. The green arrow is a local maximum.	16
Figure 2.6. General IEC system: System optimisation based on subjective evaluation (Takagi, 2001).	18
Figure 2.7. Psychological distance (d) between the target in psychological space and actual system outputs becomes the fitness axis of a feature parameter space where EC searches for the global optimum in an IEC system (Takagi, 2001).	19
Figure 2.8. Progressively interactive approach to handle a two-objective maximisation problem (f_1 , f_2) (Sinha et al., 2014).	21
Figure 2.9. Polyhedral cone in three dimensions (Sinha et al., 2014).	22
Figure 2.10. Illustration of the ROI. It is the preferred part of the Pareto-optimal region from the DM's perspective (Said et al., 2010).	23
Figure 2.11. Contours of the chosen value function (acts as a DM) and the most preferred point corresponding to the value function (Deb et al., 2010).	24
Figure 2.12. Initial shapes (Sato and Hagiwara, 2001).	28

Figure 2.13. Average number of changes that the users make to fitness during different generations (Machwe and Parmee, 2006).	31
Figure 2.14. Architecture of EDS (Li and Hu, 2015).	32
Figure 2.15. Accuracy obtained by low-level and high-level descriptors along with generations (Li and Hu, 2015).....	33
Figure 2.16. Evolution of measure values (Ekart et al., 2011).	34
Figure 2.17. Design concept developed using EvoShape. The image in this figure was created in 2011 as publicity media.	36
Figure 2.18. EvoShape screen-shot.....	37
Figure 2.19. Object construction via Boolean interaction (Graham et al., 2008).	37
Figure 2.20. Edge blending strategies (Graham et al., 2008).	38
Figure 2.21. An example of panoramic images that represent the symmetric planes and the symmetry score graphs. At the bottom is the graphs of rotation (in degrees) giving the maximum symmetry score and at the top is the images of the best rotation, (Sfikas et al., 2014).	44
Figure 2.22. Precision-Recall (%) chart for Princeton Shape Benchmark test dataset. A comparison between SymPan+3D model pose normalisation method on PANORAMA retrieval results and state-of-the-art 3D object retrieval techniques (Sfikas et al., 2014).....	45
Figure 2.23. Pairwise alignments; an autocorrelation descriptor is used to predict their relative alignment (Averkiou et al., 2015).	46
Figure 2.24. Fraction of models (y-axis)-angle threshold (x-axis) chart for 8 datasets. Averkiou et al.'s (2015) results are better than the results of random alignments (with only 8% accuracy for a threshold of 15°). This method suffers from near-symmetries of shapes (Averkiou et al., 2015).	47

Figure 2.25. Translation normalisation using the centroid of the surface of the object, example in an object with and without extruding parts (a). Scale normalisation to the unit cube (b) (Papadakis, 2014).	48
Figure 2.26. (a) Translation and (b) scale normalisation using OCSVM and the centroid and average distance of the surface (Papadakis, 2014).	49
Figure 2.27. PCA's inconsistency in identifying the positive and negative side of an axis (Farrugia and Barbarar, 2015).	50
Figure 2.28. NPCA's inconsistency in identifying the three axes (Farrugia and Barbarar, 2015).	51
Figure 2.29. Diagram of the method (Axenopoulos et al., 2011).	51
Figure 2.30. Diagram of the CPE method (Axenopoulos et al., 2011).	52
Figure 2.31. Canonical orthographic projection views of a car model based on its ideal canonical coordinate frame (Johan et al., 2011).	53
Figure 2.32. Examples showing that one canonical view of a 3D model usually has the MPA. In each row, the first three images are the front, left, and top views of a 3D model and the remaining three images are three arbitrary views of the same model. The number underneath each view is its normalised projection area (Johan et al., 2011).	54
Figure 2.33. The PRST captures the degree of symmetry of arbitrary shapes with respect to reflection through all planes in space. Darker lines represent greater symmetries (Podolak and Funkhouser, 2006).	55
Figure 2.34. A mug with and without handles. The centre of mass and PCA axes are drawn in dotted green (they move depending on the presence of handles). The centre of symmetry and principal symmetry axes using PRST are shown in solid red (they remain constant under transformation of the shape) (Podolak and Funkhouser, 2006).	55

Figure 2.35. (a) B-Rep objects in a violin model. (b) B-Rep object representing a violin body. (c) B-Rep entities within a face of the violin body (Dang et al., 2014).	56
Figure 2.36. Results of symmetry detection (Dang et al., 2014).	57
Figure 2.37. Alignment results: The pink plane is the dominant symmetry plane (first canonical plane), the blue plane is the second plane, and the green is the third plane (Dang et al., 2014).	57
Figure 2.38. The spherical trace transform: The define plane $\Pi(\eta, \rho)$ is tangential to the sphere S_ρ with radius ρ and centre at the origin, at the point (η, ρ) , where η is the unit vector in \mathbf{R}^3 (Zarpalas et al., 2007).	58
Figure 2.39. Rotation of $f(x)$ rotates $F(\eta, \rho)$, without being affected by independent rotations (upper left image). Thus, $F(\eta^2, \rho^2) = F(\eta'^2, \rho^2)$ (Zarpalas et al., 2007).	59
Figure 2.40. Q is a set of n points in \mathbf{R}^2 . s and t are the points in S that define the longest edge of bounding box. s' and t' are two points in Q . μ is the direction perpendicular to st and $s't'$. R is the minimum-area bounding rectangle of Q in the direction $s't'$ and μ . ω is the length of the edge of R in the direction μ . u and v are the two points of Q lying on the two edges of R parallel to $s't'$ (Barequet and Har-Peled, 2001).	60
Figure 2.41. Bounding boxes of two spatial point sets. The diameter of the sets are shown as a nearly vertical line segments (Barequet and Har-Peled, 2001).	61
Figure 2.42. The Stanford bunny model and the root MVBB of its vertices (Huebner et al., 2008).	62
Figure 2.43. (a) A mean cut of a bunny model. (b) They test only the planes parallel to the parent MVBB. (c) A good cut parallel to the root MVBB plane \mathbf{B} (Huebner et al., 2008).	62

Figure 2.44. Bunny sample projections onto the three faces of the root box (Figure 2.43) according to the face-parallel cutting scheme in Figure 2.43 (b) (Huebner et al., 2008).	62
Figure 2.45. Best cuts along the six box directions (Huebner et al., 2008).	63
Figure 2.46. Procedures for finding out the minimum bounding box of a simple object (Chan and Tan, 2001).	64
Figure 2.47. The Armadillo model and its 1 st , 2 nd , 3 rd manifold harmonic bases (a, b, c). It can be seen that the bending-invariance is achieved (a', b', c') (Wu et al., 2010).	67
Figure 2.48. A partial matching example. Note that this partial algorithm does not consider the symmetric shape matching, therefore a query on the right hand (a) may return a partial matching result with maximum similarity on the left hand (d) (Wu et al., 2010).	67
Figure 2.49. A similarity comparison example. The distance value indicates the dissimilarity between the query model #0 and the match model. A smaller distance means a more similar match (Wu et al., 2010).	68
Figure 2.50. Another similarity comparison example (Wu et al., 2010).	68
Figure 2.51. When the objects are similar, only a small fraction of their volumes are outside of the other objects boundary (Novotni and Klein, 2001).	69
Figure 2.52. The bananas are markedly similar but they need to be aligned before the comparative test is applied (Novotni and Klein, 2001).	70
Figure 2.53. The volume of banana with black contour is checked against the offset hulls (Novotni and Klein, 2001).	70
Figure 2.54. Rays (dashed lines) are shot from the centre (white dot) of a canonical sphere (dashed circle) through the object points (black dots) to the sphere's surface. The distance d_i travelled by the ray from a point p_i to the sphere's surface and the number of object surfaces (solid lines; 2, in this case) penetrated by the ray	

since it leaves the sphere's centre are recorded in the feature maps (Yu <i>et al.</i> , 2003).	72
Figure 2.55. Counting number of object surfaces. A ray (dashed arrow) is shot from the centre (grey dot) of the sphere (dashed curve) through an object point (black dot) to obtain a cone (dotted lines) with a small angle. The other object points (white dots) within the cone are considered. Checking the number of times the surface normals (arrows) at the points change direction, compared to the ray's direction, in increasing distance of the points from the sphere's centre, gives the number of surfaces that the ray penetrates (Yu <i>et al.</i> , 2003).	72
Figure 3.1. Objective testing diagram.	78
Figure 3.2. Five users would be expected to find 85% of the development problems (Nielsen, 2012).	80
Figure 4.1. GG (targets) that have been used during shape matching development, and their volumes.	83
Figure 4.2. A population of 12 grey forms and brown GG after PN, compared to the same population beforehand (left).	85
Figure 4.3. From left to right: GG; form; after PN; intersection; non-intersecting volumes...	85
Figure 4.4. Top row – Alignment by human. Bottom row – Alignment using PA.	87
Figure 4.5. A shape's PA aligned with the Z-axis (left) compared to a more intuitive alignment of the shape (right), where the PA of the dominant geometry, the cylinder, is used.	88
Figure 4.6. A comparison of the BB scaling method with the volume scaling method.	89
Figure 4.7. Scaling using BBs (upper): the BBs of the grey forms do not exceed the BB of the brown GG. Scaling by volume (lower) is less satisfactory, and the form BBs generally exceeds that of the GG.	90
Figure 4.8. Centre of BB translational alignment method against CoM.	91

Figure 4.9. A population of 12 grey forms and brown GG after PN, compared to the same population beforehand (inset left).	91
Figure 5.1. Large Background Populations (LBP).	93
Figure 5.2. Sporadic User Interaction (SUI).	94
Figure 5.3. Shape matching with benchmark using GG.	96
Figure 5.4. Three further shape matching examples – evolved populations alongside their GG shapes.	98
Figure 5.5. Forms’ fitness values optimising to broken ice cream.	99
Figure 5.6. Forms’ fitness values optimising to roof.	99
Figure 5.7. Forms’ fitness values optimising to cylinder.	100
Figure 5.8. Forms’ mean (left) and maximum (right) fitness values evolving towards a Cylinder with four different starting populations.	101
Figure 5.9. Forms’ mean (left) and maximum (right) fitness values evolving towards a Broken Ice Cream with four different starting populations.	102
Figure 5.10. Forms’ mean (left) and maximum (right) fitness values evolving towards a Traffic Cone with four different starting populations.	102
Figure 5.11. Forms’ mean (left) and maximum (right) fitness values evolving towards a Roof with four different starting populations.	103
Figure 5.12. Average maximum fitness values of four different starting populations for each of the four GGs.	103
Figure 5.13. Average mean fitness values of four different starting populations for each of the four GG.	104
Figure 5.14. Normalised average mean fitness values of four different starting populations for each of the four GG.	105

Figure 5.15. Average average matches of four different starting populations between the evolved forms and the four GG.	106
Figure 5.16. Shape matching with LBP of 40 shapes approach using GG.....	108
Figure 5.17. Three further shape matching examples with LBP of 40 shapes approach – evolved populations alongside their GG shapes.	110
Figure 5.18. Forms’ fitness values evolving towards a traffic cone.....	111
Figure 5.19. Forms’ fitness values evolving towards a roof.	111
Figure 5.20. Forms’ fitness values evolving towards a cylinder.	112
Figure 5.21. Forms’ fitness values evolving towards a Cylinder with four different starting populations.	113
Figure 5.22. Forms’ fitness values evolving towards a Broken Ice Cream with four different starting populations.....	114
Figure 5.23. Forms’ fitness values evolving towards a Traffic Cone with four different starting populations.	114
Figure 5.24. Forms’ fitness values evolving towards a Roof with four different starting populations.	114
Figure 5.25. Average maximum fitness values of four different starting populations for each of the four GG, using LBP of 40 shapes.	115
Figure 5.26. Average mean fitness values of four different starting populations for each of the four GG.	116
Figure 5.27. Comparison of LBP of 40 shapes and benchmark.	116
Figure 5.28. Comparison of the fitness of the four approaches: benchmark, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations.	117

Figure 5.29. Comparison of the fitness of the four approaches: benchmark, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations. All four approaches represent a comparable number of evaluated shapes.	118
Figure 5.30. Comparison of the variety of shapes of the four approaches: SUI, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations..	119
Figure 6.1. Participants' preferences in evolutionary shape searching.....	124
Figure 6.2. Left: Three out of five participants said that LBP of 80 shapes has slightly bigger variety. Right: Four out of five participants said that LBP of 80 shapes can create at least one shape that has bigger similarity with the bounding volume.	125
Figure 6.3. This figure shows three examples of sofas: a) created by using LBP of 80 shapes, b) created by using LBP of 40 shapes, and c) created by using EvoShape. d) Shows the bounding volume created at the beginning of the process by the participant when using LBP approach.	125
Figure 6.4. LBP of 80 shapes: In both generations (1 and 2) the participant chose three shapes that they found useful and could inspire the form of a sofa. There is a variety of shapes but none of them can provide good ideas for a sofa yet.....	126
Figure 6.5. LBP of 80 shapes: In both generations (3 and 4) the participant chose two shapes that they found useful and could inspire the form of a sofa. There is still a variety of shapes and some of them could already give ideas for a sofa.	127
Figure 6.6. LBP of 80 shapes: In generation 5 the participant chose two shapes that they found useful and could inspire the form of a sofa. Although the shapes have started to converge, there is still a variety and some of them can still give ideas for a sofa. In generation 6 there is at least one shape that the participant can choose to inspire the form of a sofa.....	128
Figure 6.7. LBP of 80 shapes: In both generations (3 and 4) the participant chose four shapes that they found useful and could inspire the form of a watch. There is a variety of shapes and some of them can already give ideas for a watch.....	129

- Figure 6.8. LBP of 80 shapes: In generations 3 and 4 the participant chose three and two shapes respectively that they found useful and could inspire the form of a watch. There is a variety of shapes and some of them give useful ideas for a watch...130
- Figure 6.9. LBP of 80 shapes: In generations 3 and 4 the participant chose three and four shapes respectively that they found useful and could inspire the form of a guitar. There is a variety of shapes and some of them give useful ideas for a guitar. ...131
- Figure 6.10. LBP of 80 shapes: In generation 5 the participant chose three shapes that they found useful and could inspire the form of a guitar. Although the shapes have started to converge and the variety has decreased a lot, some of the shapes can give ideas for a guitar. In generation 6 there is at least one shape that the participant can choose to inspire the form of a guitar.....132
- Figure 6.11. LBP of 80 shapes: In generations 3 and 4 the participant chose four and three shapes respectively that they found useful and could inspire the form of a hand-held 3D mouse. There is a variety of shapes and some of them give useful ideas for a hand-held 3D mouse.133
- Figure 6.12. LBP of 80 shapes: In generation 5 the participant chose three shapes that they found useful and could inspire the form of a hand-held 3D mouse. Although the shapes have started to converge and the variety has decreased a lot, some of the shapes can give ideas for a hand-held 3D mouse. In generation 6 there is at least one shape that the participant can choose to inspire the form of a hand-held 3D mouse.....134
- Figure 6.13. The ‘marked generations’ are just before the shapes start converging. The starting point 10 is the average of all of the nine different starting populations (three starting populations for each of the three GG).136
- Figure 6.14. Left: All participants said that all versions of SUI approach have similar variety. Right: All participants said that only the first version can create shapes that are similar to the bounding volume.138

Figure 6.15. The variety in LBP approach is big enough to give useful ideas of shapes, and therefore to give shapes similar to the bounding volume.	138
Figure 6.16. This figure shows three examples of sofas: a) created by using SUI first version, b) created by using SUI version two, and c) created by using SUI version three. d) Shows the bounding volume created at the beginning of the process by the participant when using SUI approach.....	139
Figure 6.17. SUI: In generations 10 and 16, the participant chose one and three shapes respectively that they found useful and could inspire the form of a sofa. There is a variety of shapes but none of them can give good ideas for a sofa yet.....	140
Figure 6.18. SUI: In both generations (22 and 28), the participant chose three shapes that they found useful and could inspire the form of a sofa. There is still a variety of shapes and some of them can already give ideas for a sofa.	141
Figure 6.19. SUI: In generation 34, the participant chose two shapes that they found useful and could inspire the form of a sofa. Although the shapes have started to converge, there is still a variety of shapes and some of them can still give ideas for a sofa. In generation 40 there is at least one shape that the participant can choose to inspire the form of a sofa.....	142
Figure 6.20. SUI: In generations 10 and 16, the participant chose two and three shapes respectively that they found useful and could inspire the form of a guitar. There is a small variety of shapes but none of them can give good ideas for a guitar yet.	143
Figure 6.21. SUI: In both generations (22 and 28), the participant chose two shapes that they found useful and could inspire the form of a guitar. There is still a small variety of shapes but still none of them can give ideas for a guitar.	144
Figure 6.22. SUI: In generation 34, the participant chose two shapes that they found useful and could inspire the form of a guitar. The shapes have started to converge and there was no big variety of shapes to give ideas for a guitar. The participant stopped the process.	145

Figure 6.23. Comparison of %useful shapes that each of the three approaches provide. ...146

List of Tables

Table 4.1. Forms' average % volume match with GG where human and PN alignment differ. The average match of the forms with the GG when the shapes are rotated intuitively is 30.58%. This percentage decreases to 27.72% when the PA method is applied.....	87
Table 4.2. Selection of the most efficient alignment methods using Pugh matrix.	Error! Bookmark not defined.
Table 5.1. Key stages in shape matching.	95
Table 5.2. Shape matching to a Broken Ice Cream, Roof and Cylinder GG.	97
Table 5.3. Shape matching to a Cylinder GG.	100
Table 5.4. Shape matching to a Broken Ice Cream GG.	100
Table 5.5. Shape matching to a Traffic Cone GG.	101
Table 5.6. Shape matching to a Roof GG.	101
Table 5.7. Key stages in shape matching.	107
Table 5.8. Shape matching to a Traffic Cone, Roof and Cylinder GG.	109
Table 5.9. Shape matching to a Cylinder GG.	112
Table 5.10. Shape matching to a Broken Ice Cream GG.	112
Table 5.11. Shape matching to a Traffic Cone GG.	113
Table 5.12. Shape matching to a Roof GG.	113
Table 5.13. Selection of the most efficient population sizes, without user intervention, using Pugh matrix.....	Error! Bookmark not defined.

Table 6.1. EvoShape: While two participants with moderate design and CAD experience had been searching for shapes that could inspire the form of a sofa and a hand-held 3D mouse; the participants stopped the process in the third generation because they had already lost the shapes they liked.	123
Table 6.2. EvoShape: While two participants with good CAD experience and one with moderate CAD experience had been searching for shapes that could inspire the form of a watch, a guitar and a building; the participants stopped the process in the fifth generation because they had already lost the shapes they liked.	123
Table 6.3. LBP of 80 and 40 shapes: in the last generation the participants are satisfied with the results – at least one of the shapes gives a useful idea of each of the five products.	123
Table 6.4. SUI first version: At the 40 th generation the participants are satisfied with the results – at least one of the shapes gives a useful idea.	137
Table 6.5. SUI version two and three.	137

List of Abbreviations

AABB	Axis Aligned Bounding Box
ACO	Ant-Colony Optimisation
API	Application Programming Interface
BB	Bounding Box
CAD	Computer-Aided Design
CBR	Case Based Reasoning
CoM	Centre of Mass
CoG-PCA	Centre of Gravity Principal Component Analysis
CoS	Centre of Symmetry
CPCA	Continuous Principal Component Analysis
CPE	Combined Pose Estimation
DAPMK	Dimension Amnestic Pyramid Match Kernel
DM	Decision Maker
DM	Distance Map
DOP	Dynamic Optimisation Problem
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EDO	Evolutionary Dynamic Optimisation
EMO	Evolutionary Multi-objective Optimisation
EMO-VF	Evolutionary Multi-objective Optimisation Value Functions
EO	Evolutionary Optimisation
EP	Evolutionary Programming
ES	Evolution Strategy
FEA	Finite Element Analysis
FFT	Fast Fourier Transform
GA	Genetic Algorithm
GG	Geometric Guides
ICE-GCAD	Interactive Co-Evolutionary Garment Computer-Aided Design
ID	Identification number
IDSET	Interactive Design support System using Evolutionary Techniques
IEC	Interactive Evolutionary Computation
IED	Interactive Evolutionary Design
IEO	Interactive Evolutionary Optimisation
LBP	Large Background Population
MA	Memetic Algorithm
MBB	Minimum Bounding Box
MPA	Minimum Projection Area-based
MVBB	Minimum-Volume Bounding Box
NN	Neural Networks
NPCA	Normal Principal Component Analysis
NURBS	Non-Uniform Rational B-Reps

OBB	Oriented Bounding Box
OCSVM	One-Class Support Vector Machines
PA	Principal Axis
PCA	Principal Component Analysis
PMK	Pyramid Match Kernel
PN	Pose Normalisation
PRST	Planar Reflective Symmetry Transform
PSA	Principal Symmetry Axis
PSO	Particle Swarm Optimisation
SA	Simulated Annealing
SAM	Simulated Annealing Mutation
SAR	Simulated Annealing Recombination
SFL	Shuffled Frog Leaping
SM	Surface Penetration Map
SUI	Sporadic User Interaction
RBF	Radial Basis Functions
RoI	Region Of Interest
r-dominance	Reference solution-based dominance
Ufit	User-assigned fitness
3D	Three-Dimensional

List of Nomenclature

A, B, C	Eigenvalues-principal moments of inertia
C_{new}	New cost
C_{old}	Old cost
d	Distance
d_i	Distance from the object point p_i to the surface of the sphere
f	Fitness function of an evolving form
\tilde{f}	Normalised fitness function of an evolving form
I	Moments of inertia
I_p	Inertia matrix
m	Mass of the object
n	Number of objects
p_i	Point on the object's surface
Q	Set of n points
r	Unit vector pointing from the origin to the object
R	Rotation matrix
R	Minimum-area bounding rectangle of Q
RV	Matrix multiplication
S_f	Objective function of an evolving form
S_p	Sphere
T	Temperature
V	Vector
$V_{cylinder}$	Volume of the Cylinder
V_{form}	Volume of form
V_{GG}	Volume of GG
$V_{GG \cap form}$	Intersecting volume of GG
V_i	Unit vectors
α	Acceptance probability
Δ_ρ	Length of the radius segment
ε	Accuracy controlling factor
η	Unit vector in R_3
θ	Angle
$\Pi(\eta, \rho)$	The define plane
ρ	Radius
ω	Length of the edge of R

1 Introduction

1.1 Computer-Aided Design

Early Computer-Aided Design (CAD) systems were representative; they just followed direct instructions to create drawings or models, and therefore were not very useful for conceptual design. As CAD systems became more accessible, they were also used by engineers, architects, mechanics, inventors, and designers to *explore* design ideas and visualise their concepts. CAD became very important in engineering across most of the design, development and manufacturing processes, for consumer products, as well as electronic and mechanical design automation, shipbuilding, aerospace industries, architectural design and many more.

More recently, simulation tools (e.g. Finite Element Analysis (FEA)) and optimisation techniques have been integrated into CAD systems (rather than being separate applications). The use of CAD with FEA offers remarkable added value to the product design process – reduces design costs by allowing many simulations to be performed using computer software in the time that it would take to build and test a single or a few iterations of a concept or prototype. Optimisation is most critical during the initial stages of design – engineers optimise parameters like weight, cost and strength, while considering constraints, loads, material, and manufacturability.

CAD systems are now beginning to include generative functionality (based on engineering properties) – quickly generating high-performance design alternatives, many that users would never think of on their own, from a single idea. More specifically, in generative design, designers or engineers input into the software design goals such as materials, constraints and manufacturing methods and the software explores all the possible solutions by quickly generating design alternatives. There are potentially thousands of great solutions and the designers choose the design that best fits their needs.

There are many types of tools which are used in early design for engineering problems and now integrated with CAD; engineering calculations (*PTC Mathcad*, 2018), sketching (*Catchbook*, 2018), and generative design (*Autodesk*, 2018). There are also many methods which can be used to increase the possibility of a successful form finding in early design; visual affordances, ergonomics, product semantics, emotional factors and biomimicry. However, these methods are still separate to the CAD system – they are not supported very well by the CAD system.

Product appearance methodologies

In the physical world, objects have physical properties like size, shape or weight that provide hints as to what one could do with them. The use of *visual affordances* may inspire the design of products in human-computer interaction systems and assist designers in form finding by turning the physical world into a virtual world (Manandhar, 2017).

Ergonomics uses data to optimise products and environments to help users enjoy a better overall experience and gives to the design special attention to make sure it is the best possible fit for the user, the environment and the task.

Product semantics is the understanding of the ways in which visible features of products (form, colour, size) could communicate additional meaning to users. It is when people act on what the physical qualities of objects mean to them. Krippendorff and Butter (1984) claims that designers could enhance opportunities for self-expression, by using semantics. Product semantics can provide designers with design concepts and an understanding of the meaning of products.

An emotional factor is how people feel when they are using products in their everyday lives. Designers need to understand how the users are affected by the products and how the products can improve the user experience and fit their needs if they are developed carefully (Komninos, 2018).

Biomimicry is the imitation of the elements of nature for solving complex human problems. It studies nature's models and then imitates or takes inspiration from these designs and processes to solve human problems (*The biomimicry Institute*, 2018). Evolution can inspire designers with solutions that nature has produced and that they have been tested by the various forces of nature itself, to find a design solution.

Limitations in CAD

CAD systems do not involve the users early in the design process and do not allow them to experiment with ideas freely; designers do not find inspiration in conceiving form ideas (Graham et al., 2008). CAD tools, where any number of random forms is presented to the designer, are of limited use too; these forms should be useful and inspire the designer to intuitively guide the process of form generation. These limitations lead to the need for interactive evolutionary CAD which can be used to optimise designs by combining evolutionary algorithms and simulation or to allow designers to interactively explore populations of evolving solutions.

Interactive Evolutionary Design

Interactive evolutionary design is driven by human evaluation, commonly, aesthetic selection. Usually, human evaluation is necessary when the result of optimisation should fit a particular user preference. An advantage of interactive evolutionary design is that it can perform the optimisation based on human preference and feeling rather than relying on the evaluation by fitness functions. Interactive evolutionary design can be a tool for improving collaborative discussions between a designer and customer (Tiainen et al., 2008). The advantage of interactive evolutionary CAD is that it generates ideas of concepts that could inspire the designers – the designer does not develop the form manually. It can also involve the users early in the design process.

So although CAD technologies remain firmly established in the later stages of design, the functional aspects of earlier stage product design are starting to be supported quite well by generative design.

1.2 The need for Interactive Evolutionary CAD

Indications of a need for this work come from at least two places:

❖ Trends in CAD

Commercial CAD systems are becoming more generative (SolidThinking, 2018). There is a trend from representative to assistive CAD systems that changes the way product designers and engineers approach design, and allows them to investigate efficient concepts quickly and easily. However, CAD is still focused on the later stages of design. There is still a need for CAD systems to support earlier design processes and better support ideation in conceptual design (Gharib, 2013).

❖ EvoShape

An earlier attempt at meeting this need was launched in 2011 (EvoShape). It has been used in industry to inspire the design of unique consumer electronics products, elegant super-yachts, eye-catching packaging, refined medical devices, award-winning sculpture, and beautiful household and lighting products. Encouraged by these achievements, it is likely that the design approach can be further improved, and expanded beyond the current, quite limited scope of form exploration.

Current users of EvoShape within the design industry have expressed a desire to have more control over the process beyond interacting with the system during the evolutionary process through selecting and scoring shapes.

1.3 Wider research vision

The aspiration of the wider research vision, of which this thesis is a part, is to create smarter and more co-operative human-machine systems that utilise real-time human interaction during geometric optimisation. In a real-time approach, humans would have control over the process by interacting during the optimisation algorithm (they direct the search using their preference) so they do not have to spend a long time setting up optimisation conditions – which saves engineers days or weeks (InstantCAD, 2017).

This wider ambition is distinctive in focussing on CAD methodologies to support early-stage concept generation by combining automatic and human systems through a blend of engineering optimisation and Interactive Evolutionary Computation (IEC).

1.4 Aims and Objectives

The specific aims of this research are to investigate the integration of guiding geometry with IEC, and the interaction strategies that best exploit it. Guiding geometry is simple target geometric primitives that represent the overall proportions sought – it is a simple box model that the user constructs at the beginning of the process and consists of simple 3D shapes (bounding volumes), to which evolving forms are compared. More specifically, it is proposed that the evolving geometries are tested against simple target geometric primitives (e.g. the guiding geometry mentioned above). They require three-dimensional (3D) alignment before a comparative test is applied. Henceforth, guiding geometry is referred to as Geometric Guides (GG).

The specific objectives of this research are:

1. Identify available strategies for bringing users into a generative design process.
2. Develop a representation for the Guiding Geometry – soft geometric constraints (in the form of simple bounding geometry) created at the start of the evolution process.
3. Develop stand-alone programs to investigate the individual operations involved in Pose Normalisation / alignment:
 - ❖ Rotation.
 - ❖ Scaling.
 - ❖ Translation.
 - ❖ Shape comparison.
4. Find efficient alignment methods.
5. Find efficient shape comparison methods.
6. Integrate the most efficient operations with EvoShape Interactive Evolutionary Design (IED) system.

7. Turn the add-in program into an automated shape matching program without human interaction, to objectively test and assess the efficacy of the 3D alignment and shape comparison and the approaches being developed.
8. Test alignment and shape comparison methods efficiency.
9. Develop effective human-system interaction strategies.
10. Add human intervention to the system to subjectively test and assess the efficacy of the interactive evolutionary strategies and the approaches being developed.
11. Test the human-system interaction strategies with users.

The broad methodology that is used to develop these objectives is a Software application development followed by a combination of objective and subjective testing. More specifically, it is a CAD add-in for the Siemens PLM NX11 CAD system (both new test applications and integration with an existing evolutionary design system) followed by numerical analysis and user testing (both during development of new functionalities, and comparisons between the old and the new system).

1.5 Novelty

The novelty of the research is the combination of simple GG with optimised interactive strategies and evolutionary computation in the CAD domain, and their specific application in the field of form ideation of product design. There are generative CAD systems that aid in functional concept ideation.

SolidThinking was the first commercial generative system to use topology optimisation to enable design engineers, product engineers, and architects to create and investigate structurally efficient concepts quickly and easily. This kind of tool can now be found in other commercial CAD systems – but as discussed earlier, the focus is on functional form rather than aesthetic.

GenoForm (2018) does focus on product appearance, and is the only design exploration software that enables the exploration variations of designs – it is an iterative design exploration tool that gives to the users some variations of their design proposal. However, it relies on a designer already having a design concept in which to model in the CAD system.

In Sinclair's research (2013), GenoForm software was used to generate variations of concept designs after the CAD model had been modified in a way that the participant was satisfied with the design. However, GenoForm was not able to suggest new ideas and most of the participants preferred their own modified model to the options generated by GenoForm. When the designer gives fixed restrictions to the participant as to which features of the model might be changed, GenoForm failed to rebuild the model; it is impossible to create a

model that updates. It takes time to build more robust models using modelling practices for GenoForm; it requires a lot of up-front work to set up. So the research described in this thesis – guided evolutionary shape searching – has not been done before; it is unique. The users have some control over the process beyond interacting with the system during the evolutionary process by creating a GG at the beginning of the process, therefore it is different to the original work on which it is based (Graham, 2002). The system generates shapes that can inspire the users, but by allowing GG to guide the evolutionary process, these shapes are more useful to the user.

Also, the optimised interactive strategies in this research are more sophisticated than other interactive evolutionary algorithms; the system has very low inertia – i.e. quicker than GenoForm because the users do not create their own design (they only have to create a simple guide). It also supports earlier design processes and ideation in conceptual design – allowing a designer to experiment with ideas freely and to interactively explore populations of evolving solutions.

1.6 Contribution to Knowledge

The contribution to knowledge in this research is the combination of design automation (EO) and human interaction (IEC).

The IED system, developed in this research, is a co-operative and efficient human-machine system. This system integrates a GA-driven generative design process with human interaction by allowing users to have control to the evolution of 3D form. Users create a simple GG at the start of the process, the GG is aligned and compared with the evolving forms and human interaction strategies (IEC) that best exploit the 3D alignment and comparison (EO) are established.

In this research, the IED system supports early design processes and ideation in conceptual design and generates forms that can inspire the users and that are not based on predefined designs created by skilled designers. It allows the designers to experiment with ideas freely and to interactively explore populations of evolving solutions.

1.7 Thesis Structure

The rest of this thesis is organised as follows: in Chapter 2, a Technology Review of Evolutionary Optimisation (EO), Interactive Evolutionary Optimisation (IEO), Interactive Evolutionary Design (IED), Geometry Mathematics, Pose Normalisation (PN), Shape

Comparison and Minimum-Volume Bounding Box (MVBB) is provided, analysing and comparing the different methods of IEO, PN and Shape Comparison. Conclusions of the Technology Review about the most suitable techniques of IEO, PN and Shape Comparison methods to be applied for this research are drawn. In Chapter 3, the Methodology that has been used to develop the research solutions is described. In Chapters 4 and 5, the 3D Alignment and Shape Comparison Analysis as well as the User Interaction Strategies are presented alongside the experimental results that have been produced. In Chapter 6, the results of the participant testing are presented and evaluated alongside the participants' feedback. Finally, in Chapter 7, the thesis conclusions and future work are provided.

2 Literature Review

2.1 Overview of generative CAD systems

Generative design is the automated algorithmic combination of goals and constraints to reveal solutions (Hauck, 2018). This approach can help designers to capture the generic geometric features of designs and create design variations as well as decreases the time and cost of design.

There are alternative generative CAD systems available that have been employed for bringing users into a generative design approach, e.g. Genoform (Krish, 2018), Rhino plus Grasshopper (Davidson, 2018), SolidThinking (Altair Company, 2018), Dean's et al. (2005) algorithm-driven designs, Autodesk Project Dreamcatcher (Autodesk Research, 2018) and Frustum Generate (Frustum, 2018).

GenoForm (Krish, 2018) is a professional design exploration software, developed by Sivam Krish, that enables designers to help companies optimise their designs. It allows users to create genetic design and generate variations. GenoForm is based on building intelligent models on CAD systems that capture all the important geometric engineering and constraints, and designs can be modified easily. It focuses on product aesthetic but it relies on designers creating their own design concept in the CAD system.

Grasshopper (Davidson, 2018) is a visual programming language and environment, developed by David Rutten, that runs within a 3D CAD application. Programs are created by dragging components onto a canvas. Rhino plus Grasshopper is used to build generative algorithms (e.g. for generative art). Advanced uses of Grasshopper include parametric modelling for structural engineering, architecture, lighting performance analysis for eco-friendly architecture and building energy consumption.

The SolidThinking (Altair Company, 2018) project was started by Alessandro and Mario Mazzardo in 1991. It develops industrial 3D design software that helps designers explore and evaluate new ideas easily and fast, without the limitations of traditional software. It is a simulation driven design that aims to greatly reduce the costs of designing parts by carrying out detailed simulations and optimisation in the early stages of design. After the geometry is imported and the loads, constraints and materials are defined, the users can then run an analysis or optimisation study. To solve the optimisation the users must decide if, for example, they want to maximise stiffness or minimise mass (Hale, 2018). However, it uses topology optimisation and focuses on requirements of the form and not on the aesthetic of the solutions.

Lionel Dean has been exploring the creative potential of digital design and manufacturing technology as well as 3d printing in art and design. In 2002 Lionel Dean founded FutureFactories (Dean et al. 2005), which is entirely focused on 3D printing technologies and developing computational design methodologies which combine CAD with computer programming. It is a digital design and manufacturing concept and a computer based evolutionary design application. Instead of creating a single discrete design solution or a range of options, the designer creates a template. This template defines the functional requirements of the form and the designer creates rules and relationships which maintain a desired aesthetic over a potentially range of solutions. The solutions are found according to the selection criteria that are specified by the designer. Solutions score high on 'fitness' but are not ultimately the 'target'.

Autodesk Project Dreamcatcher (Autodesk Research, 2018), designed by Mark Davis and the Autodesk Design research group, is a generative design system that enables designers to craft a definition of their design problem through goals and constraints. First the users create the essential design geometry such as important aesthetic areas. Then, they apply fixed constraints, such as bolt holes for a bracket, as well as loads. Next the users define the objectives; what they are trying to achieve with the design generation e.g. maximise the stiffness based on the load conditions and factor of safety for a given mass target. Finally, the users choose the materials and the designs are being generated (Hale, 2018).

Frustum Generate (Frustum, 2018), developed by Harry Zhong, is cloud based generative design system that the users can access through a web browser. This greatly reduces the performance requirements of the computer required to do this sort of optimisation. This system empowers engineers to create optimal design faster through a balance of artificial intelligence and human insight. First, the users create the design space which is where the part material can be, which also includes fixing points, bolt holes etc. The next step is to define the design constraints and to set the boundary conditions. The material, resolution and mass target parameters are defined before the optimisation starts (Hale, 2018).

2.2 Evolutionary Optimisation

Evolutionary Computation (EC) is often used for engineering and design problems that are too complex to tackle deterministically, and is generally focused on optimisation and the later stages of design (Testing, Detail Design, Manufacturing and Further Testing, Refinement and Sales). EC is usually either automatic or interactive, with the latter well-suited for messy problems that are hard to model (e.g. the primary use-case of form ideation within Product Design (Graham, 2002)).

Most engineering research uses several optimisation methods which include auto-control, pattern recognition and engineering design. There are many optimisation problems in everyday life. Some of them are so complex that mathematical optimisation is not achievable and fail in solving large-scale engineering problems with many variables and non-linear objective functions (Løvbjerg, 2002). These problems can be tackled by using EC that is a powerful computational method for solving complex real-world problems, and Evolutionary Algorithms (EAs) that researchers have proposed and have gained wide popularity. Their principal characteristic is that they examine many solutions rather than one solution at a time. EAs are techniques that mimic the biological evolution or the social behaviour of species (e.g. how ants find their food by following the shortest route and how birds find their destination during migration). The first evolutionary optimisation technique introduced in the literature was Holland's Genetic Algorithm (GA) (2016).

2.2.1 Evolutionary Algorithms

Many EAs have been introduced in the last 20 years. One of them is the GA. A GA is a search heuristic inspired by Charles Darwin's survival-of-the-fittest philosophy of natural evolution and generates new individuals with selection, crossover and mutation operators. GAs have been applied successfully to a wide range of problems (Labadie, 2004; van Zyl et al., 2004).

The process begins with the creation of a population. Then the fittest individuals from a population are selected. The offspring is produced which inherits the characteristics of the fittest individuals (parents) and is added to the next generation. If parents have better fitness, their offspring may be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals may be found. Five phases are considered in a GA (Mallawaarachchi, 2017).

1. Initial population:

The process starts with a population which is a set of individuals and each individual is a solution to the problem. An individual consists of a set of variables (Genes) which form a chromosome as shown in Figure 2.1. The set of genes of an individual is normally represented by binary values (string of 1s and 0s).

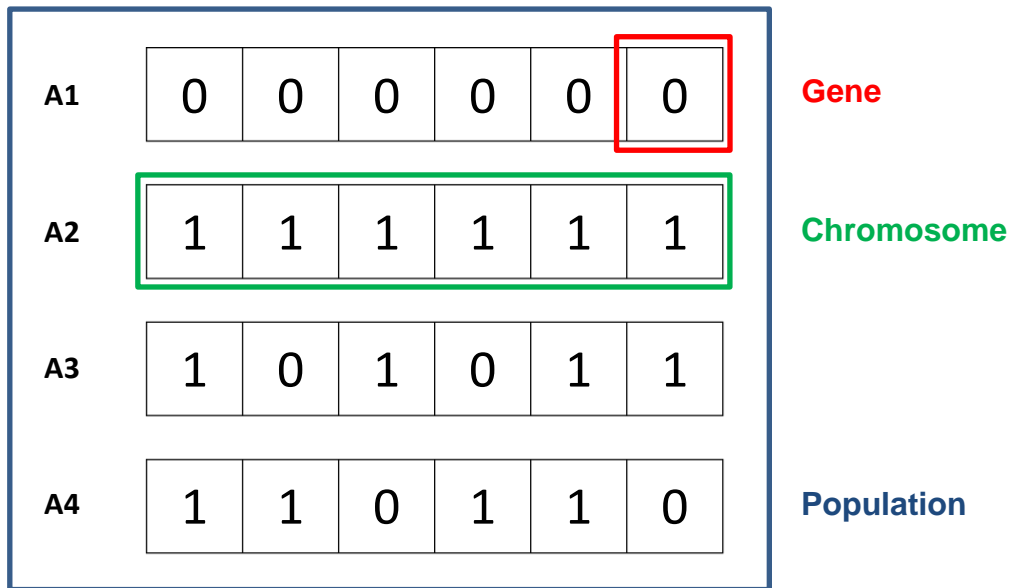


Figure 2.1. Population, Chromosomes and Genes.

2. Fitness function:

The fitness function determines how fit an individual is. It is the ability of an individual to compete with other individuals. The probability that an individual will be selected for reproduction is based on the fitness score that the function gives to each individual.

3. Selection:

Two pairs of individuals (parents) are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction. The parents therefore tend to be the fittest individuals, and it is their genes that are passed to the next generation.

4. Crossover:

Crossover is a characteristic that influences the GAs' performance. This is a significant phase in a GA. As shown in Figure 2.2, the crossover exchanging information of parents creates an offspring. A crossover point is normally randomly chosen from within the genes for each pair of parents. For example, consider the crossover point to be 3 as shown below. The genes of parents are being exchanged among themselves until the crossover point is reached and offspring are created.

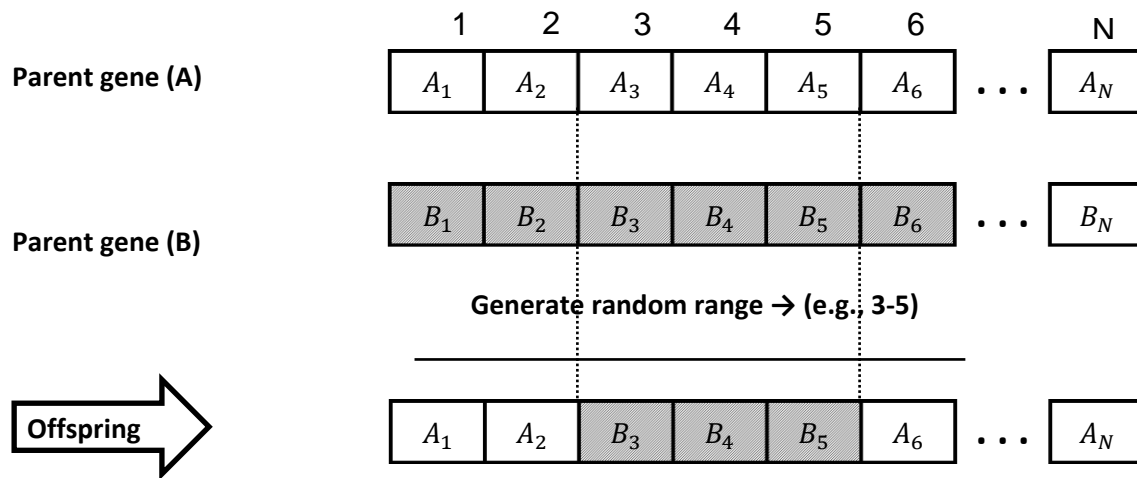


Figure 2.2. Crossover operation to generate offspring.

The new offspring is added to the population.

5. Mutation:

Mutation is a characteristic that influences the GAs' performance. In mutation, a sudden change to an offspring can occur by arbitrarily changing some information of one chromosome from the population, as shown in Figure 2.3. Some of the offspring's genes can be mutated with a low random probability. This indicates that some of the bits in the bit string can be flipped.



Figure 2.3. A chromosome before and after mutation.

Mutation occurs to maintain diversity within the population and prevent premature convergence.

The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation). Then it is said that the GA has provided a set of solutions to the problem.

The population has a fixed size. As new generations are formed, individuals with least fitness die, providing space for new offspring. The sequence of phases is repeated to produce individuals in each new generation which are better than the previous generation. The offspring replaces the population member only if it is better than it. The process goes on with many generations until the best fit solution is achieved.

Other parameters that influence the GAs' performance are the population size and the number of generations that raise the possibilities of achieving an optimum solution, but also increase the processing time.

So that the problem can be tackled and processing time can be reduced, more EAs have been introduced in the last 20 years, as shown in Figure 2.4:

- Memetic Algorithms (MAs) (Moscato, 1989): They are similar to GAs but instead of genes they have memes. They use local search to every population member in order to allow the offspring to gain some experience and achieve local optimum solutions. Like GAs, MAs are influenced by population size, number of generations, crossover rate and mutation rate.
- Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 1995): PSO is based on the social behaviour of migrating birds going to an unknown destination. Unlike GAs, PSO does not produce new birds (offspring). The birds communicate together, find the bird that is in the best location and speed towards it. The process stops when the bird reaches the destination. It is a combination of local and global search while the birds learn from their own experience and from the others' experience around them.
- Ant-colony systems (Dorigo et al., 1996): Similar to PSO, Ant-Colony Optimisation (ACO) algorithms are based on social behaviour of ants trying to find the shortest way from their nest to the source of food. Ants leave pheromone trails to communicate together. The best path is the one with the most pheromone. Over time, all ants will choose the shorter path.
- Shuffled Frog Leaping (SFL) (Eusuff and Lansey, 2003): SFL combines MAs and PSO algorithms. Every frog is divided into subsets (memeplexes) and holds an idea that can be influenced by the ideas of the other frogs. After the memetic evolution, ideas are going among memeplexes in a shuffling process.

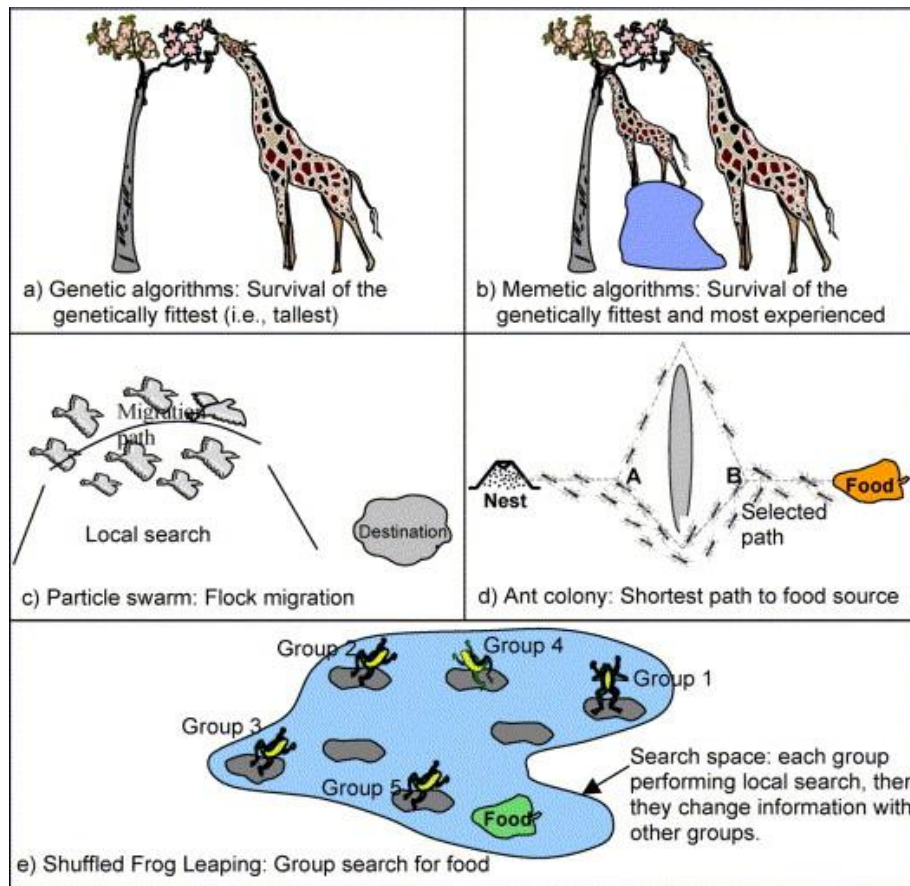


Figure 2.4. Schematic diagram of natural evolutionary systems (Elbeltagi et al., 2005).

According to Elbeltagi et al. (2005), the PSO technique achieves better results than the other methods in success rate and solution quality, and is good in processing time. However, PSO does not integrate well with interaction; it does not produce variety – it is focusing on finding one solution.

EAs provide a framework for effectively sampling large search spaces, are broadly applicable and easily tailored to specific problems. Two more examples of EAs (Jones, 1998) are described below:

- Evolution Strategies (ESs) developed by Rechenberg (2016), where a current population of individuals is randomly initialised, fitness scores are assigned to each of the individuals, new offspring are generated, new fitness scores are assigned to them and the new population becomes the current population.
- Evolutionary Programming (EP) developed by Fogel et al. (1966). Unlike GAs and ESs, in EP there is no *recombination* operator and the evolution depends on the *mutation* operator. Genetic recombination is the production of offspring with combinations of traits that differ from those found in either parent. Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of GA chromosomes

to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence, the GA can come to a better solution by using mutation. Like ESs, EP is more popular in parameter optimisation.

ESs and EP use real numbers while GAs use binary numbers. All three use a selection operator which applies extinctive evolutionary pressure (i.e. in ESs and EP, the operator determines which individuals will be excluded from the new population) or preservative evolutionary pressure (i.e. in the GA the operator selects individuals for breeding). In GAs and EP selection is probabilistic, while in ESs selection is deterministic.

2.2.2 Hybrid Simulated Annealing and Genetic Algorithm

Simulated Annealing (SA) is a method for finding a good solution to an optimisation problem (Geltman, 2014). It is a stochastic based general search tool that mimics the natural process of metals annealing (Kirkpatrick et al., 1983). There are many optimisation algorithms, including hill climbing, genetic algorithms, gradient descent, and more. The SA algorithm has the ability of escaping from local minima that are better than any others neighbouring solutions, but are not the very best and has been applied widely (Dougherty and Marryott, 1991; Marryott, 1996; Goldman and Mays, 1999; Cunha and Sousa, 2001).

An example of SA is the traveling salesman problem where the salesman is travelling to different places in the order that minimises the total number of miles he travels. As the number of places increases, it becomes too computationally intensive to check every possible route. At that point, the SA algorithm is needed.

When an optimisation algorithm searches for the best solution, if a neighboring solution is worse than the current one, then the algorithm stays with the current solution. This can lead the algorithm to get trapped in a sub-optimal place, as shown in Figure 2.5.

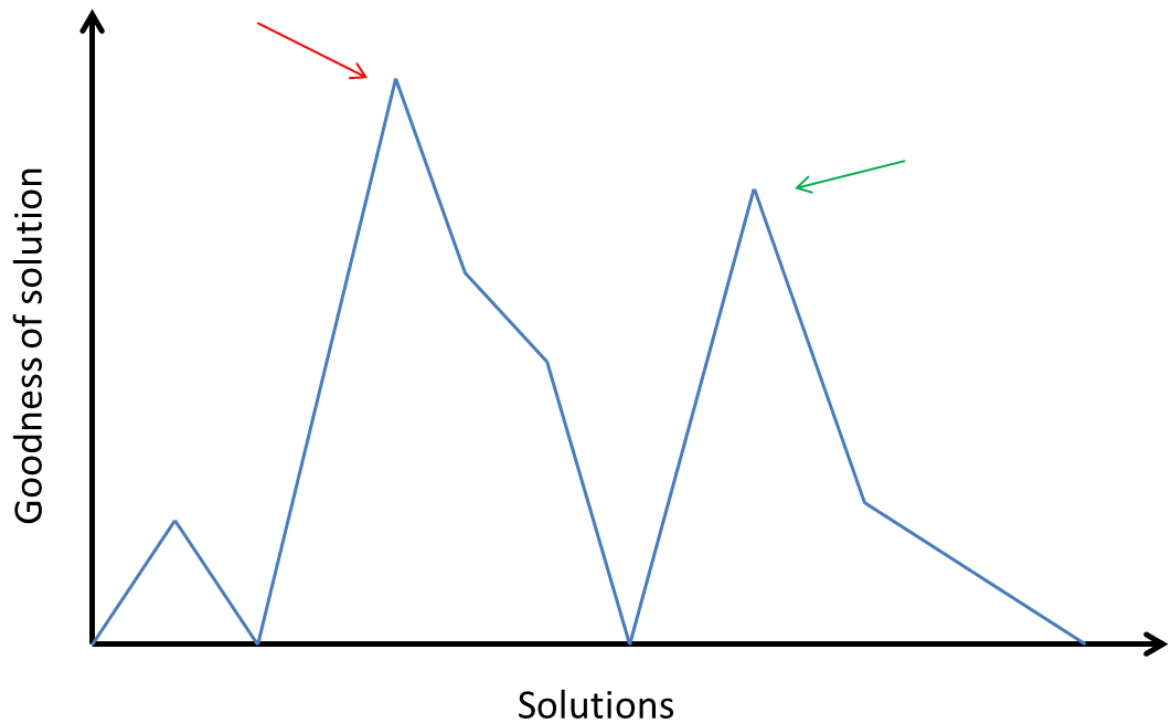


Figure 2.5. The red arrow on the left shows the best solution. But if a simple algorithm finds its way to where the green arrow on the right shows, it will get trapped in there: all of the neighboring solutions are worse. The green arrow is a local maximum.

Here is where SA helps to escape local maxima early in the process:

1. First, generate a random solution.
2. Calculate its cost using some cost function.
3. Generate a random neighboring solution.
4. Calculate the new solution's cost.
5. Compare them:
 - If $c_{\text{new}} < c_{\text{old}}$: move to the new solution.
 - If $c_{\text{new}} > c_{\text{old}}$: *maybe* move to the new solution. To decide, the algorithm calculates the *acceptance probability* and then compares it to a random number.
6. Repeat steps 3-5 above until an acceptable solution is found or a maximum number of iterations is reached.

An important parameter of this algorithm is the "temperature". This name comes from the fact that this algorithm was inspired by a method of heating and cooling metals. The acceptance probability function takes in the old cost, new cost, and current temperature. Once the acceptance probability is calculated, it's compared to a randomly-generated number between 0 and 1. If the acceptance probability is larger than the random number, then the new solution is accepted. The equation 2.1 shows the acceptance probability.

$$a = e^{\frac{C_{new}-C_{old}}{T}}$$

2.1

where a is the acceptance probability, $C_{new}-C_{old}$ is the difference between the new cost and the old cost, T is the temperature, and e is the mathematical constant equal to 2.71828.

This equation was inspired by metalworking and it describes the embodied energy of metal particles as they are cooled slowly after being subjected to high heat. This process allows the particles to move from a random configuration to one with a very low embodied energy. Computer scientists borrow the annealing equation to help them move from a random solution to one with a very low cost.

The hybrid SA and GA is quite a new global optimisation search approach. It can enrich neighbourhood structure in the searching process and improve searching ability in the range of values. Therefore, this hybrid algorithm has some application to the fields of system design (Shieh and Peralta, 2005), system and network optimisation (Ponnambalam and Reddy, 2003), etc.

The hybrid SA-GA algorithm proposed by (Adler, 1993) improves the crossover and mutation based on the acceptance probability rule. The hybrid algorithm replaces the mutation and recombination operators with SA-mutation (SAM) and SA-recombination (SAR). The SAM operator not only mutates a chromosome-solution and returns the new one, but also evaluates the fitness of the solution after mutation and decides whether to accept the new or keep the old one based on the acceptance probability rule. The SAR generates two children solutions based on two parent solutions. Each of the children is then compared with the better parent for acceptance by the acceptance probability rule. The artificial temperature decreases in each generation according to the cooling schedule.

2.2.3 Evolutionary Dynamic Optimisation

Over the last 20 years, evolutionary optimisation in dynamic environments, or evolutionary dynamic optimisation (EDO), has become a challenging research area in the field of evolutionary computation since many real-world optimisation problems have changing conditions over time and need to optimise in a dynamic environment. Changes, e.g. breakdown of machines, new tasks' arrivals or change of some conditions, may influence the objective function and the optimal solution(s) of the optimisation problem.

As is described by Nguyen et al. (2012), Dynamic Optimisation Problems (DOPs) need an optimisation algorithm to find optimal solution(s) and track the optimal solution(s) that are changing over the time. The algorithm needs to consider changes during the optimisation process as time goes by (Branke, 1999), (Jin and Branke, 2005). The most suitable tools that can solve DOPs are EC, and swarm intelligence since they are based on biological evolution which always reacts to changes in the environment.

In non-dynamic evolutionary optimisation environments, the goal is to locate the optimal solution(s) as fast as possible. While the time-changing problems appear, the goal changes to track the changing optimal solution(s). The optimisation algorithm needs to assimilate its previous search experience in order to be effective, since the problem after the changing conditions is related to the problem before the changing conditions.

2.3 Interactive Evolutionary Optimisation

Interactive Evolutionary Computation (IEC) is an optimisation method that affiliates EC based on subjective human evaluation, as described by Takagi (2001). It is simply an EC technique whose fitness function is replaced by a human user. In a general IEC system, as shown in Figure 2.6, EC optimises the target system to achieve the ideal output based on the user's evaluation and the user sees or hears and evaluates system outputs. In regard to this, IEC is a technology that inserts Kansei in the target system. In engineering, Kansei is the development or improvement of products and services by translating the customer's psychological feelings, preferences, intuitions, emotions and needs into the domain of product design. Kansei engineering connects the customer's emotional reactions to the characteristics of a product or service.

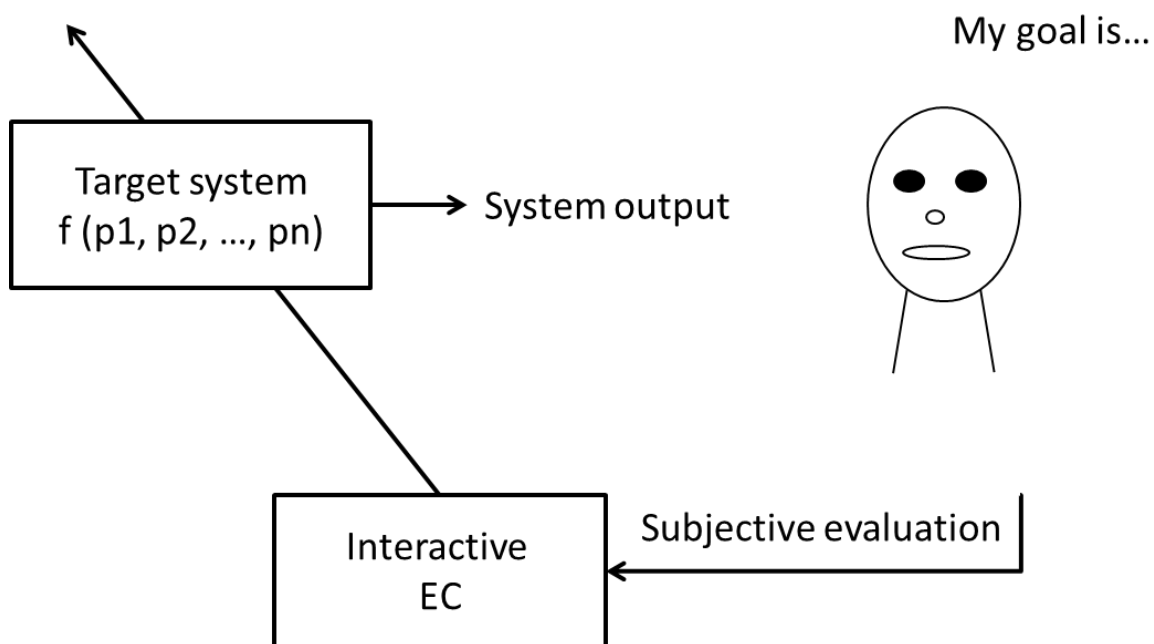


Figure 2.6. General IEC system: System optimisation based on subjective evaluation (Takagi, 2001).

EC is a population-based searching algorithm with many candidate system outputs. These candidates are called individuals. In IEC, these individuals are estimated by the users according to the distance (d) between the target in their psychological spaces and the actual system outputs (Figure 2.7). Corresponding to the psychological distance, EC searches for

the global optimum in a feature parameter space. The global optimum of IEC is an area and not a point. For example, if one wants to find the next model of a car, the aim is to define a number of different car models and not just one point. The candidate car models will be similar to the previous models.

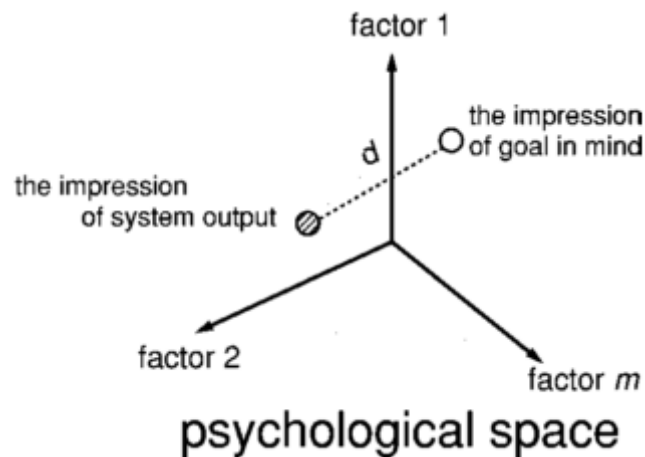


Figure 2.7. Psychological distance (d) between the target in psychological space and actual system outputs becomes the fitness axis of a feature parameter space where EC searches for the global optimum in an IEC system (Takagi, 2001).

The first IEC problem is to reduce human effort. Since humans cooperate with a tireless computer, the IEC generations need to be reduced. The second problem is the small population size and the limited number of generations which cause the reduction of the individuals and make the goal searching difficult. The third problem is how to get the user to evaluate time-variant individuals such as sounds or video with less effort and operation time. The user's psychological fatigue and the operation time increase, since they need to compare the current displayed sounds or video with the previous ones in their memory and to evaluate them. Sounds and video are spatially impossible to display.

The second and the third problems are correlated with the first. Therefore, the fatigue problem which is the biggest IEC problem needs to be solved to make IEC practical.

Human users can decrease psychological fatigue with broad ratings, for instance five or seven rating levels, instead of a higher order of level ratings (Ohsaki et al., 1998). A simulation experiment showed that, when the EC search reaches a number of tens or hundreds of generations, the closer convergence becomes noteworthy. However, the smaller convergence, for example less than the first 10 or 20 generations, is not problematic.

A second way to decrease human fatigue, in a huge population size, is to display a few predicted individuals that have higher fitness values. Takagi (2001) suggested two prediction methods that assimilate the user's evaluation characteristics and predict fitness values: the first uses Euclidean distances in an EC searching space (Nagao et al., 1998), and the second uses Neural Networks (NN) (Biles et al., 1996).

One last method that reduces human fatigue is active intervention, which accelerates EC or IEC convergence. Takagi (2001) proposed two active user interventions: online knowledge embedding (Caldwell and Johnston, 1991) and visualized IEC (Hayashida and Takagi, 1999).

According to Takagi's survey (Takagi, 2001) (Takagi, 2003), IEC is most often achieved in applications such as:

- Music: IEC is applied to music to generate the melody (Biles, 1994). GenJam is an example of a melody generator where a user listens to and evaluates the melody as it is played. If the users give one fitness value for every melody that they hear, it will take too much time and human effort. To tackle this problem, GenJam allows the user to evaluate the melody measure by measure instead of the entire melody. This technique reduces IEC time.
- Face Image Generator: Montage systems combine partial images of facial photos and create a facial image. IEC is applied to montage systems to optimise the combination of partial facial images and produce a face image which is not based on the memory of each facial part but on the observer's total impression (Takagi and Kishi, 1999).
- Image Processing: IEC is applied to image processing to define the sequence of image filtering. Different orders of image filtering outputs different images. For example, an image filtered by A then B is different from an image filtered by B then A. It is not easy for non-experts to decide on a filtering order, but it is easy for them to estimate which image is better.

Other applications that IEC has been used in are virtual reality, control and robotics, graphic arts and animation (Takagi, 2001).

It was predicted that IEC would become more important in design that is correlated with human Kansei factors. As IEC was further developed, it would help unite humanised technology with user-friendly technology. This was Takagi's prediction that this research is now fulfilling.

In the following section, different types of IEC approaches will be covered and some of these technologies are recommended for the proposed research.

2.3.1 Pareto-optimal points Interaction Strategies

Sinha et al. (2014) combined an evolutionary multi-objective technique with the Decision Maker (DM) interaction during the implementation of the algorithm. The evolutionary multi-objective optimisation (EMO) algorithm aims to find the Pareto-optimal¹ points from which the decision maker will select the most preferred point, instead of finding the entire Pareto-optimal front. The algorithm tries to get as close to this point as possible (Figure 2.8) and in as few interactions with the DM as possible.

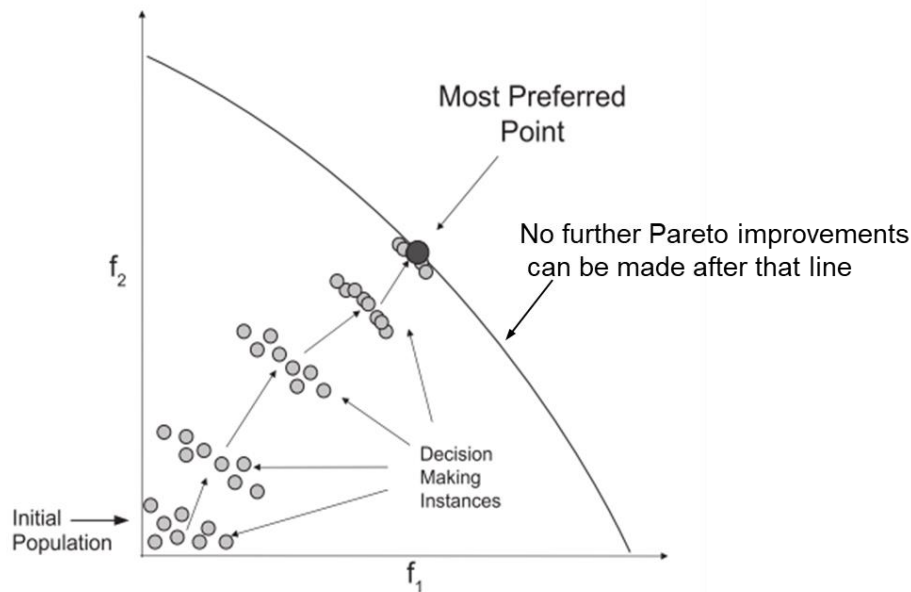


Figure 2.8. Progressively interactive approach to handle a two-objective maximisation problem (f_1, f_2) (Sinha et al., 2014).

There are three different types of approach that a multi-objective optimisation problem can generate:

- In apriori approach, the users make decisions before the EMO algorithm starts. Then, the decisions and the optimisation procedure are united and the most suitable point is found without any further interaction with the DM. However, the solution sometimes is not close to the most suitable solution.
- In the aposteriori approach, the EMO algorithms try to find the entire Pareto-optimal frontier before the user interaction, but this is difficult due to the large number of objectives (number of objectives > 3).
- In the interactive approach, on which they focus, the decision making takes place during the implementation of an optimisation algorithm so that the most preferred point and solution is found.

¹ Pareto optimality is a distribution of resources in which it is impossible to make any one individual better off without making at least one individual worse off. When this is possible, it is called Pareto improvement. So, a distribution is defined as Pareto optimal when no further Pareto improvements can be made.

The method that they describe uses the notion of polyhedral cones to find the most preferred point. They use the best solution in the population to create a polyhedral cone and to use it at each DM call². Then they choose the proper search direction based on the cone and explore the preferred region, as shown in Figure 2.9.

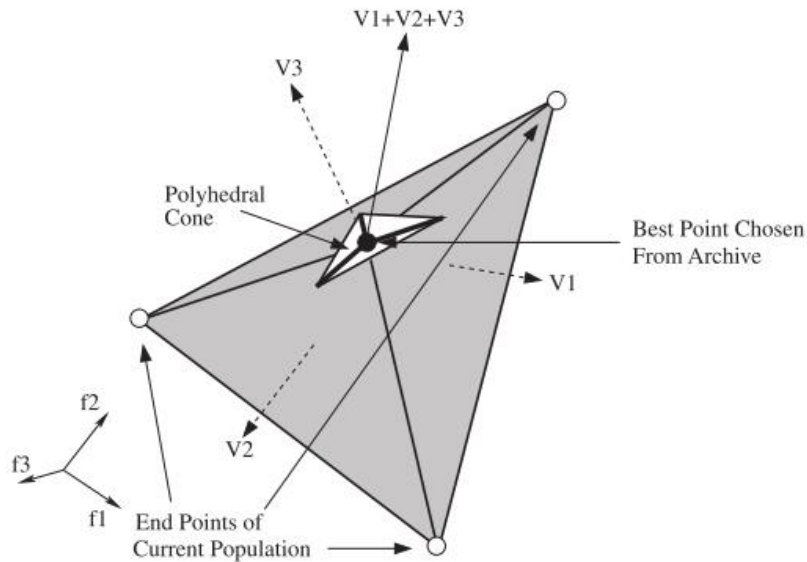


Figure 2.9. Polyhedral cone in three dimensions (Sinha et al., 2014).

The bounding hyperplanes of the polyhedral cone are constructed by:

- The best solution in the population (that forms the peak of the polyhedral cone);
- The end points of the non-dominated part of the parent population at that instant.

Considering that the space is M-dimensional, M different bounding hyperplanes are constructed. Eliciting preferences progressively provides the algorithm information to focus the search on a specific area of the search space. The non-negative normal unit vectors (V_i) of all the M hyperplanes can be summed up ($V_1+V_2+V_3$) to get a search direction.

Before the start of the optimisation, the maximum number of times that the DM will interact has to be provided and the algorithm has to find the best solution within this range. The experiments of Sinha et al. (2014) show that this interactive method is able to efficiently handle optimisation problems with many objectives.

This method should be able to be effectively applied to IED systems and efficiently handle optimisation problems without user's fatigue, since the algorithm uses an interactive approach to get close to the most preferred point in only a few interactions with the user. The maximum number of times the user will interact, that has to be provided before the process starts according to Sinha et al. (2014), would also be useful to be applied to IED

² A DM call is an event where the algorithm seeks preference information from the DM.

systems, since it is critical in defining the accuracy of the optimisation (with higher values leading to better accuracy).

As is described above, the DM has to find the most preferred point and a unique solution instead of being interested in the entire Pareto-optimal front. The algorithm aims for well spread solutions close to the Pareto-optimal front. The advantage associated with seeking the most preferred point, instead of the entire Pareto-optimal front, is that it avoids the intricacies involved in exploring the entire multi-dimensional front. When the DMs use the apriori approach, it is difficult to express their preferences. On the other hand, in the interactive approach, the DMs interact during the execution of the process being able to drive the search toward the preferred parts of the Pareto-optimal region and thus expressing their preferences interactively.

Said et al. (2010) attempt to find a new variant called *reference solution-based dominance* (r-dominance), which is creating a strict partial order between Pareto-equivalent solutions. In this approach, the DM is searching among the Pareto-optimal Region Of Interest (ROI) parts. The ROI is illustrated in Figure 2.10.

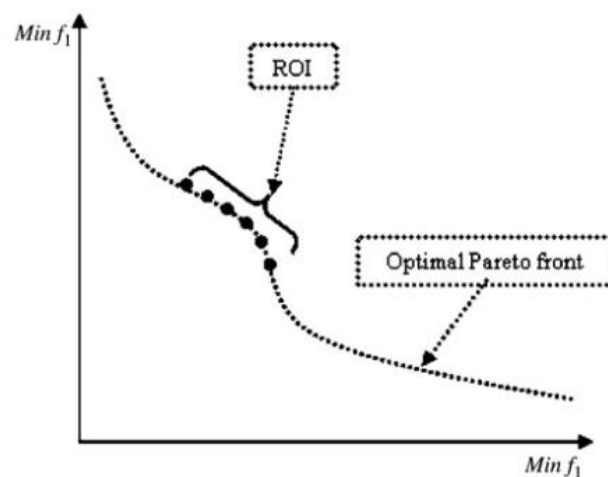


Figure 2.10. Illustration of the ROI. It is the preferred part of the Pareto-optimal region from the DM's perspective (Said et al., 2010).

The r-dominance variant is able to:

- Drive the search toward the preferred parts of the Pareto optimal region;
- Control the number of ROIs and;
- Make the DM interact with the process.

The key feature of this variant is that it tries to get as close to the preferred point as possible and at the same time holds the order that is generated by the Pareto dominance. In order to

estimate how close to the preferred point the solution is, Said et al. (2010) used the *weighted Euclidean distance*³.

Said et al. (2010) have applied the r-dominance method to both apriori and interactive approaches on a set of multi-objective problems. Furthermore, this technique has better results than other Evolutionary Multi-objective approaches.

Since the experiments (Said et al., 2010) have shown the ability to guide the search based on DM's preferences toward the preferred parts of the Pareto-optimal front on a reasonable number of evaluations, this method should be applied to IED systems to accomplish robust results. However, the progressive focus of solutions on a specific region of the search space (i.e., ROI) has led to a population diversity reduction. Therefore, considering that the decreased population diversity should sometimes drive the process to inaccurate results, the IED system would be much more competitive if some strategies, that allow escaping from local optima (mutation), were designed. Especially, some IED systems that provide inspiration – like the IED system proposed in this research – need a very diverse set of solutions and not just traditional convergence.

2.3.2 Efficient Interaction Strategies

Deb et al. (2010) suggest a technique concerning an interactive evolutionary multi-objective optimisation algorithm based on progressively approximated value functions (EMO-VF). The algorithm gradually accepts the DM's preference information after every few generations and leads the DM to the most preferred solution. The preference information constructs a value function which guides the algorithm to search for more preferred solutions, as shown in Figure 2.11.

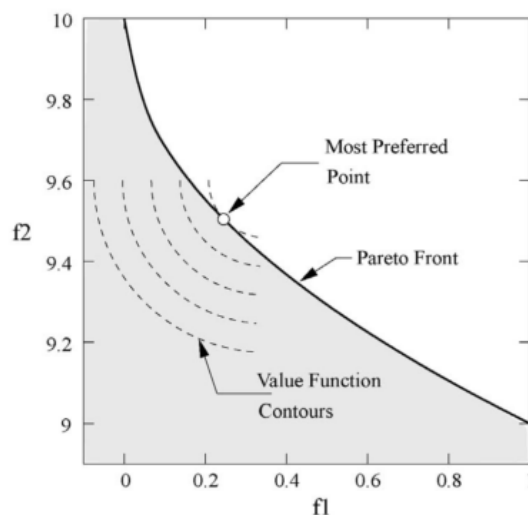


Figure 2.11. Contours of the chosen value function (acts as a DM) and the most preferred point corresponding to the value function (Deb et al., 2010).

³ It is a variant of Euclidean distance used to measure and thereby depict the distances between profile points.

The DM has to rank some non-dominated points from best to worst, this is called a DM call. Until the next DM call, this preference information finds a value function which satisfies the DM's preference information and searches for more preferred solutions by using a preference-based domination principle and utilising a preference-based termination criterion. More specifically, the DM's preference information has been utilised to:

- Find a function value which is maximised between ranked points.
- Redefine the domination principle between the points by using the value function. (The modified domination principle guides the algorithm to search for more preferred solutions).
- Design the termination criterion algorithm using the value function.

This method concentrates on procedures requiring more frequent involvements of a DM with an EMO algorithm and is able to find the most preferred solution in a simple way. With a user's evaluation every few generations and its ability to effortlessly find the best solution, this method should be effective when it is applied to IED systems and should achieve robustness.

Similar to the previous approach, Koksalan and Karahan (2010) also developed a human-based evolutionary algorithm that interacts with the DM during optimisation to recognise their preferences. This method aims to converge to a final preferred region of the DM. The algorithm is tested on multi-objective problems with multiple interactions with the DM to investigate the quality of the solutions. It is observed that the algorithm converges better if the DM interacts many times during optimisation. More interactions also help the algorithm to identify more easily the final preferred region of the DM.

In each iteration stage, a filtered set of individuals are presented to the DM to make it easier for them to choose the best individual. This filtering method uses a modified dominance scheme where an individual is non-dominated if there is no individual that is at least a small amount better than that individual in every objective and at least more than a small amount better in one objective. However, this filtering method may lower the performance of the algorithm. To tackle this problem, first they will expand the amount of individuals presented to the DM and therefore expand the accuracy, and second will perform a better filtering method to make sure that better individuals of the population are selected. The improved method may produce consistent results by being applied to IED systems. However, this method may not be efficiently adapted for the IED proposed in this research as this kind of approach might be hard to implement with subjective fitness assignation – it may cause user's fatigue by interacting many times during optimisation.

2.4 Interactive Evolutionary Design (IED)

In this thesis, research investigating the benefits of combining an interaction system with an automatic system is described. Usable Interactive Evolutionary Design (IED) systems are emerging, but this research's interests are distinctive in focussing on CAD methodologies to support early-stage concept generation by combining engineering optimisation and IED.

Most evolutionary design systems are either interactive (Ventrella, 2008) or automatic (Baluja et al., 1994), (DiPaola and Gabora, 2009). Interactive systems serve aesthetic inspiration because human selection is taking place, but it needs a lot of human effort and this could have negative consequences. On the other hand, automatic systems reduce the human effort but they are not able to generate better aesthetic artworks as the evaluation needs some improvements.

To tackle this barrier, Machado et al. (2005) combine the advantages of both systems and propose partially interactive evolution where the user interaction is reduced but still drives the evolution. More specifically, to address the user fatigue issue, Machado et al. (2005), propose the use of an automatic seeding procedure, phenotype filters and partial automation of the fitness assignment. The experimental results show that these methods are effective to decrease user fatigue without decreasing the aesthetic quality of the results. When applied to IEC systems, the user can select a particular style and let the automatic procedure assign fitness to the individuals according to their similarity to the selected style; the IEC generates individuals that fit the characteristics of the style and then fitness is assigned according to the user preferences and the automatic procedure used as a filter.

This method can efficiently be adapted for the IED system proposed in this thesis as it supports the initial proposals of this research; combine automatic and human systems to allow humans to direct the search using their preference – so they do not have to spend a long time setting up optimisation.

In the following sections, different types of Interactive Evolutionary Design (IED) approaches will be covered and some of these technologies are recommended for the proposed research.

2.4.1 ICE-GCAD

Hu et al. (2008) proposed an interactive co-evolutionary CAD system for garment pattern design (ICE-GCAD) by supporting inexperienced designers to cooperate with experts, though it is very important to include the consumers. Hu et al. (2008) took a leisure shirt as a sample and focus on size fitting.

By using interactive evolutionary systems, users interact with the system constantly by evaluating and choosing alternative designs that are being produced and developed. A co-evolutionary algorithm runs in the background to search and produce the optimal designs by using intelligent techniques like Clonal Proliferation, Hyper Mutation, Co-Evolution and Elimination. The initial designs for the evolutionary algorithm are the common experts' designs. The evolutionary algorithm will then search for optimal designs and the experts will evaluate them. When the termination condition of the algorithm is satisfied, the optimal designs are outputted. If the experts are not satisfied, the system will start a new round of co-design.

In the proposed interactive design, one designer can edit whilst others can only read the design. Designers can invite others and watch the design process by a remote assistant, messaging and adding notes on the specific design. Experts can evaluate a design by scoring and statistical computation approaches sum the scores from all experts.

A challenging problem that must be tackled is the way the design needs to be evaluated. The users can use their experiences or check the size of the design by using some size constraints or logic rules. Experimental results (Hu et al., 2008) prove that ICE-GCAD is a valuable solution to improve the existing GCAD systems. ICE-CAD can be effectively applied to the proposed IED system but the method will be more robust if, for example, geometry constraints are considered.

2.4.2 IDSET

Sato and Hagiwara (2001) propose an artistic shape-creating Interactive Design support System using Evolutionary Techniques (IDSET). They use Stages where shapes of each part are generated (generating-Stages) and Stages where some parts are combined (combining-Stages).

At each Stage, the system generates new shapes using evolutionary techniques, and the shapes are shown to the user. First, the shape of each part is created, and then the entire shape is formed by combining them. To begin with, simple shapes or random shapes are employed which are represented as individuals. Then a population of individuals is generated. All of the individuals are presented on the screen and the user evaluates them. Genetic operators are applied to the individuals in order to produce new shapes in the next generation. The cycle of display, evaluation and genetic operation is called one generation.

Five kinds of initial shapes are used, as shown in Figure 2.12. The evolution begins from these simple shapes. Cutting, expansion and contraction are employed as the genetic operators. Other genetic operators that are applied are addition, deletion, cross over and scale mutation.

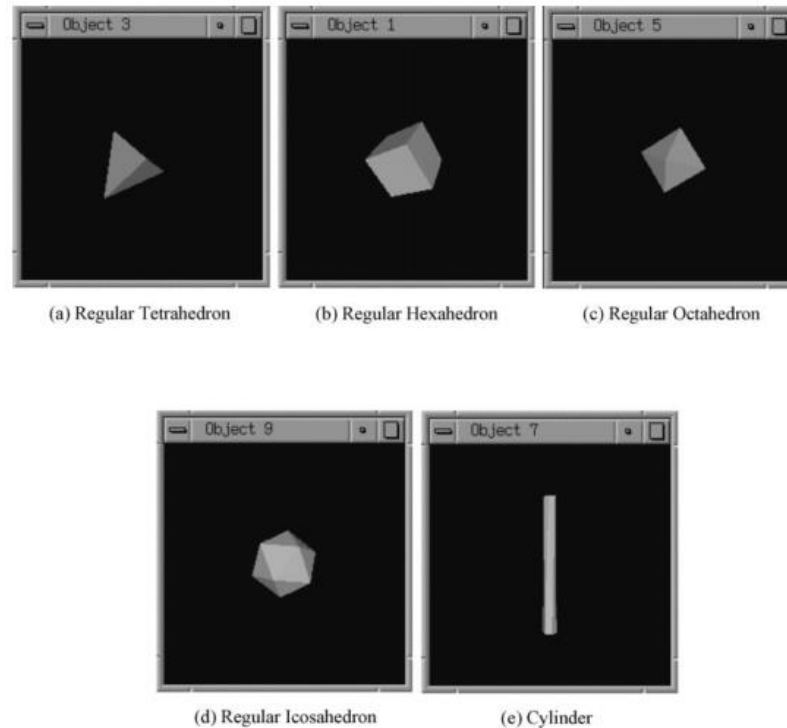


Figure 2.12. Initial shapes (Sato and Hagiwara, 2001).

Since the system begins with creating parts instead of using parts that are prepared in advance, a wide variety of designs can be created. The user only has to evaluate the presented shapes and not create them. Through the interaction between the user and the system, it is possible to make new designs that the user could not have shaped easily. Additionally, the user's creativity may be inspired by watching the evolution process.

The evaluation of individuals is a vital task for an evolutionary technique. Takagi and Unemi (1998) have studied two methods for user evaluation: the *rating-all* method and the *simulated breeding* method. In the rating-all method, the user gives an evaluation value for all of the individuals. Consequently, this technique is best for utilising the human skill of evaluation. However, the psychological and physical user's tiredness is very high because the user has to evaluate all of the individuals through many generations. In contrast, in simulated breeding, a user chooses only the best individual in each generation. Therefore, the user's fatigue is much smaller.

In the suggested IDSET, a user provides only two types of evaluation values: -1 and +1. The value -1 is given to the individual that is useless and then these individuals are destroyed. On the other hand, +1 is given to the individual that is a good one. This individual turns into a parent of all of the children in the next generation. When +1 is given to more than one

individual, one of them is selected as a parent randomly. By giving two types of evaluation values, the population can change based on the user's evaluation while keeping variety in the individuals. It is essential to keep a variety of individuals, specifically in the primary generations, because many designs inspire the user's creativity.

It is found that this technique presents interesting shapes to the user and can support the design of several artistic shapes. The average number of generations required is 57.5 and the user's fatigue to evaluate the various candidates is small. The population size is set to ten individuals in each generation but the maximum number of individuals shown to the user is set to three. If the shape does not exist, the IDSET method will be effective to create the design. And if the designers have some existing parts in their mind, they should be applied the initial shapes.

By applying this IDSET method in the proposed IED system, it should produce inefficient results since the system selects a parent randomly when +1 is given to more than one individual. The user should evaluate the individuals having +1 by scoring them, and the one with the best score should become a parent for the next generation. Apart from the evaluation process, there is also room for improving this technique by improving the initial shapes and the selection of genetic operators for each Stage. Also, by adding some geometric constraints in the evolutionary process, the evolutionary algorithm should achieve more robustness.

On the other hand, Takagi and Unemi's (1998) simulated breeding method should produce efficient results by applying it to the proposed IED system, since a user selects only the best individual in each generation. In addition, the user's effort is small. However, the proposed EvoShape design system (2011) is already a hybrid of the two aforementioned methods, the *rating-all* method and the *simulated breeding* method.

2.4.3 Machine Learning in IED

Machwe and Parmee (2006) introduce Machine Learning within an IED environment. This research concentrates on user-centric intelligent design and on the integration of user evaluation and the machine learning sub-system.

The user interacts with the evolutionary procedure: The system represents the images of the fittest designs and the user defines a User-assigned fitness (Ufit) value to each design. The aim of the Machine Learning is to assimilate on-line the aesthetic preferences of the designer.

To reduce the large amount of solutions that the user has to evaluate, the authors reduced the human interaction gradually from the first generation until the last, and increased the aesthetical solutions in terms of the user preferences. This is a machine-based method which is based on the system that has sufficiently studied the user preferences. The user

ranks the solutions by giving inputs and outputs of an online learning system. The users give inputs according to their preferences by defining an Ufit value to each design, and the inputs become outputs that the Machine Learning assimilates online.

A common issue is that, if the user gives enough information to the system with a good representation, then the algorithm will learn and assimilate the information easier and better and the machine learning system will be overloaded with the instances to be learned. The three techniques that they examined are as follow:

- Fuzzy rule based learning systems;
- Radial Basis Functions (RBF); and
- Case Based Reasoning (CBR).

In the fuzzy system, the input information is characteristic-based variables that are simple and the system fails to work efficiently, especially when the designs are complex. On the other hand, in the case of the RBF and CBR method, the representation is more flexible and the information is based on the geometrical properties.

The CBR is the most efficient technique. The user evaluates at the first population and the library starts to fill with information. The machine learning system ranks the solutions and the user is allowed to change this ranking. According to the experimental results of Machwe and Parmee (2006), once the machine learns the user preferences, the number of changes that the users make will decrease rapidly in the first generations and then slightly until the last generations, as shown in Figure 2.13. With the use of the machine learning sub-system, the user does not need to examine all the solutions.

The CBR technique should achieve efficient results by being applied to the proposed IED system since it is proved (according to Figure 2.13) that it is working effectively within the IEDs and due to the gradual human interaction reduction. Since the proposed IED system uses complex designs, the CBR method should be more flexible than the fuzzy system technique as CBR is based on the geometrical properties while the fuzzy system fails to work efficiently when the designs are complex. However, machine learning should be a challenge that this research may investigate in the future since it is not general enough to be applied outside its specific domain.

Learning Test

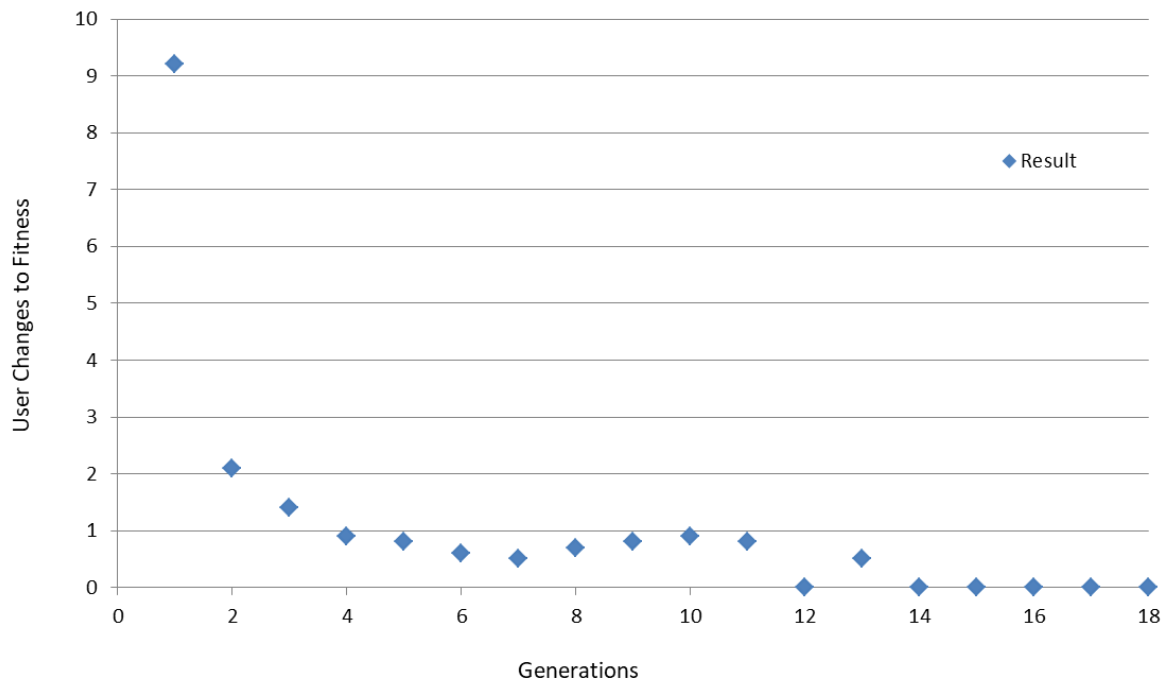


Figure 2.13. Average number of changes that the users make to fitness during different generations (Machwe and Parmee, 2006).

2.4.3.1 Aesthetic descriptors in IED

Li and Hu (2015) describe an approach based on aesthetic learning in evolutionary design. They attempt to make the user aesthetically judge the images and decrease the human interaction by reducing the loops of the process in which the user has to take part. They employ two types of aesthetic descriptors:

- A high-level⁴ descriptor based on pixel distribution; and
- A low-level⁵ descriptor based on aesthetic criteria (it applies computational methods that can make applicable aesthetic decisions in a similar way as humans can (Hoenig, 2005)).

Both descriptors represent the user's preference. The computational learning model is found, by utilising the learning algorithm, and the model is able to drive the following generations as humans do. This indicates that the evolutionary process is automated. There are two models in the learning model: The features extraction from the images by using the above two descriptors to simulate the user's preference and the classifier that is used for ranking the population in each generation.

⁴ It is the kind of features that users employ to describe an image. High-level descriptors include colour harmony, image complexity and image order.

⁵ Low-level descriptors include colour moments, lightness distribution and texture features.

These aesthetic descriptors describe the users' judgements and are applied to every image. The system learns the user's behaviour by using the descriptors' model and so the model is able to help the user in the last evaluation, as shown in Figure 2.14.

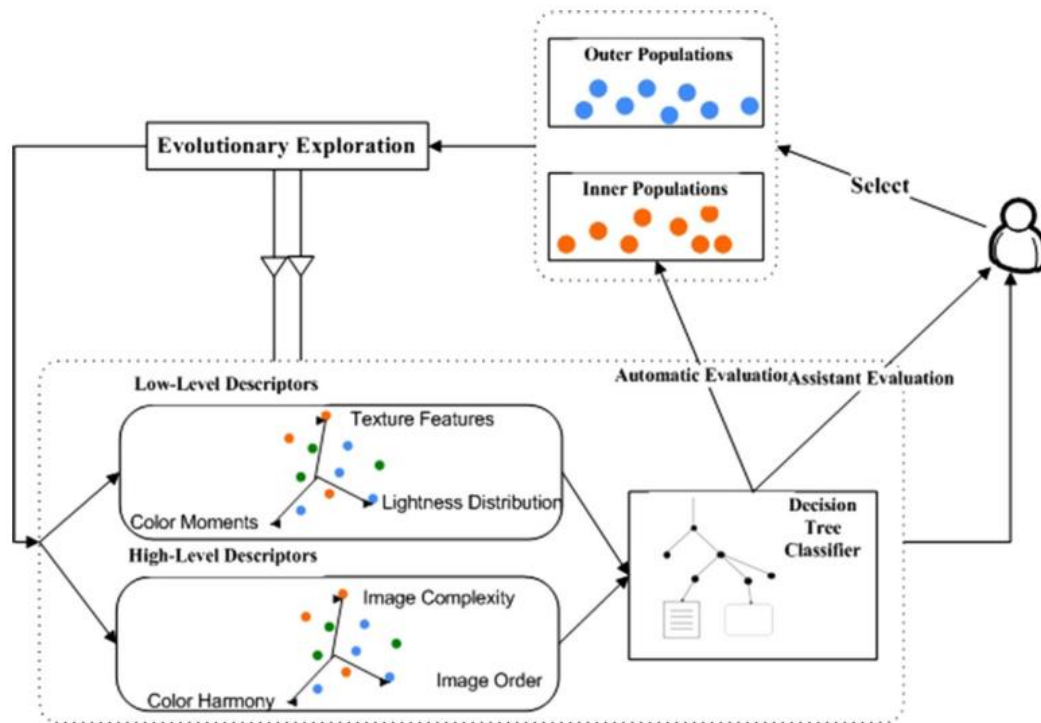


Figure 2.14. Architecture of EDS (Li and Hu, 2015).

The steps that the process of the evolutionary exploration follows are:

- Images are presented to the user on a screen;
- The user chooses their favourite images;
- The images are ranked by the user and those with the highest fitness become parents;
- Descriptors extract features of the images;
- If the number of generations is not fixed, then the user decides when the process will stop;
- The system learns the user's behaviour from the information that is collected from descriptors; and
- The model evaluates the following generations.

Machine learning is adopted to learn the aesthetic judgement in order for the system to have more aesthetic populations and less user interaction. They show that high-level descriptors are useful for predicting fitness in evolutionary design systems with aesthetic quality and that they are more convincing than low-level descriptors (Figure 2.15). Therefore, high-level aesthetic descriptors may be efficiently applied in the proposed IED system, as a future investigation. This aim has been attempted by Karim (2004) with the original EvoShape technology but without the machine learning. However, machine learning

is useful because it means that the system does not have evaluation criteria defined in advance (e.g. geometric properties).

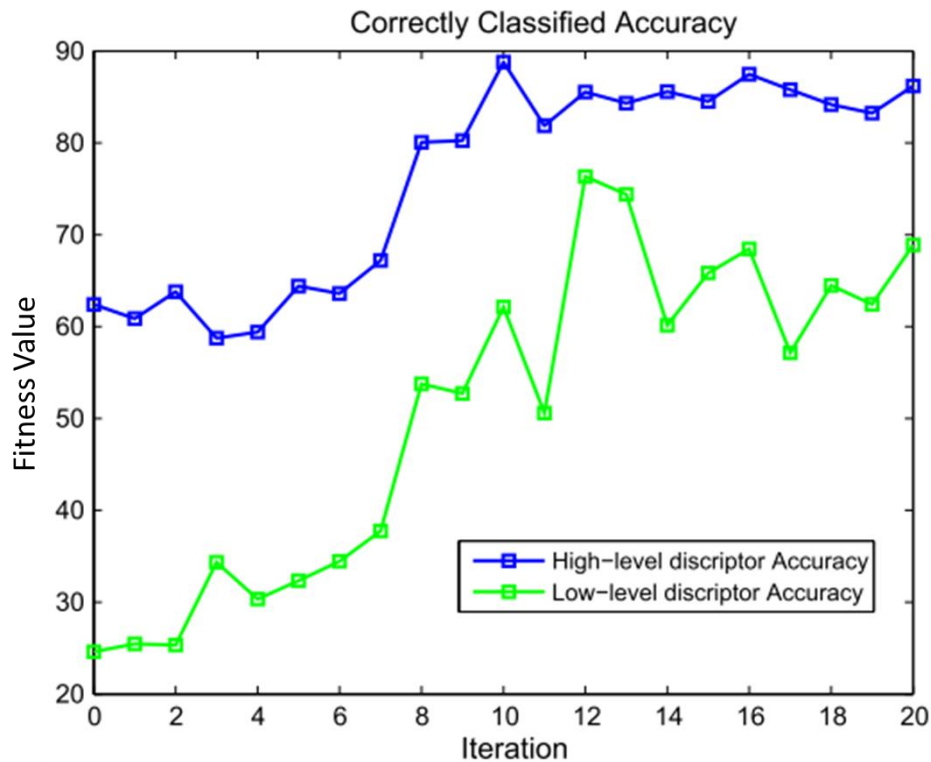


Figure 2.15. Accuracy obtained by low-level and high-level descriptors along with generations (Li and Hu, 2015).

2.4.3.2 Human preference in evolutionary art

Ekart et al. (2011) combine interactive and automatic approaches in evolutionary art by reducing the human contribution, however the user is still able to guide the evolution. This is called partially interactive evolution. They attempt to model human preference by employing four aesthetic measures:

- **Measure R:** This measure is based on a mathematical model that is proposed by Ralph and Ross and Ralph (2006) and finds the colour gradient for each pixel. Then, the response value of the user is computed to each pixel. Measure R ignores images where the user gives very little or very large response (too much information) because it is either not interesting or it is chaotic, respectively. Measure R prefers reasonable values of gradients. For example, very low values give rise to single coloured monotonous areas (which do not interest a viewer) whereas very large values give rise to sharp lines and boundaries separating areas with huge colour differences (which is undesirable).
- **Measure MC:** This measure is based on the theory of Machado and Cardoso (1998) who consider that the value of the aesthetic measure is the ratio of the image complexity to processing complexity. The less the ratio is, the less the complexity of the image is.

- Measure BZ: This measure is based on Birkhoff's (2016) research and computes the entropy H_p . Measure BZ accepts very high and very low lighting values because they both lead to low values of H_p .
- Measure S: In this measure the image is divided into four equal parts to compute how similar these parts are. They manage it by computing Shannon's entropy H_p for each part, as mentioned in Rigau et al. (2016), and they calculate the ratio of the weighted mean of these values to the entropy.

The results of these aesthetic measures are shown in Figure 2.16.

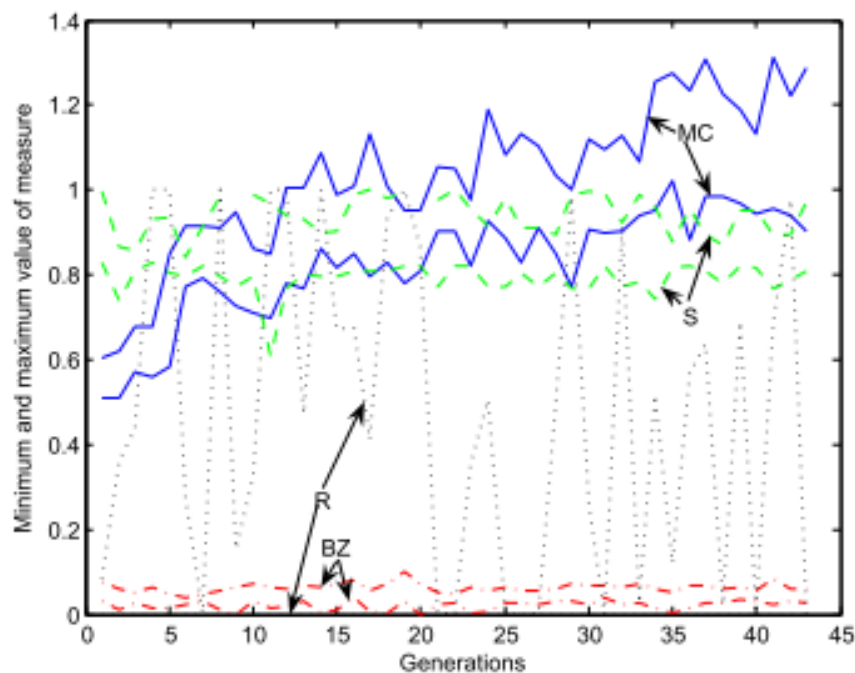


Figure 2.16. Evolution of measure values (Ekart et al., 2011).

In partial automatic evolution the users' previous preferences are examined, modelled and ranked, and then the automatic evolution utilises these measures. This method creates a proper environment to examine user aesthetic preferences but it is very simple and it produces images within a limited set.

2.4.4 EvoShape

EvoShape is a semi-commercial IED system, originally proposed by Graham et al. (2008) and developed and launched in 2011. It was developed from a core technology arising from doctoral research conducted at the Wolfson School of Mechanical and Manufacturing Engineering at Loughborough University (Graham et al., 2001).

The EvoShape application runs within a CAD environment and the geometric modeller of the CAD system interacts with a GA. An example of a product and the evolved geometry which has inspired it is shown in Figure 2.17.

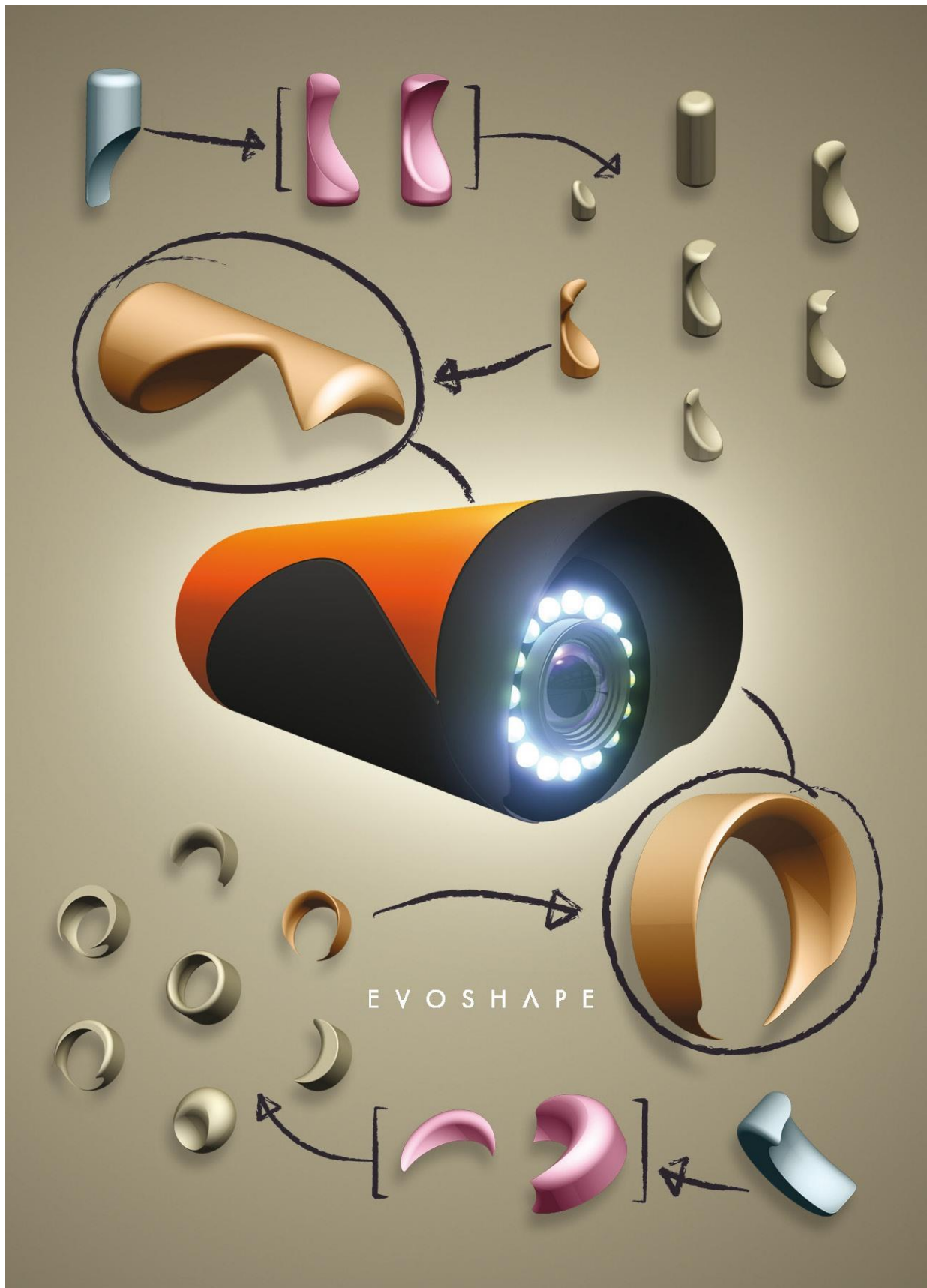


Figure 2.17. Design concept developed using EvoShape. The image in this figure was created in 2011 as publicity media.

Users guide the evolutionary process, from a random starting population of 3D shapes, purely through shape selection or rating. The users interact with the system by selecting or scoring objects from small generations, as shown in Figure 2.18. Using a mouse or touchscreen to choose their favourite objects, the users can work quickly by explicitly choosing parents for the next generation, or scoring objects to influence their probability of being selected by the system as parents.

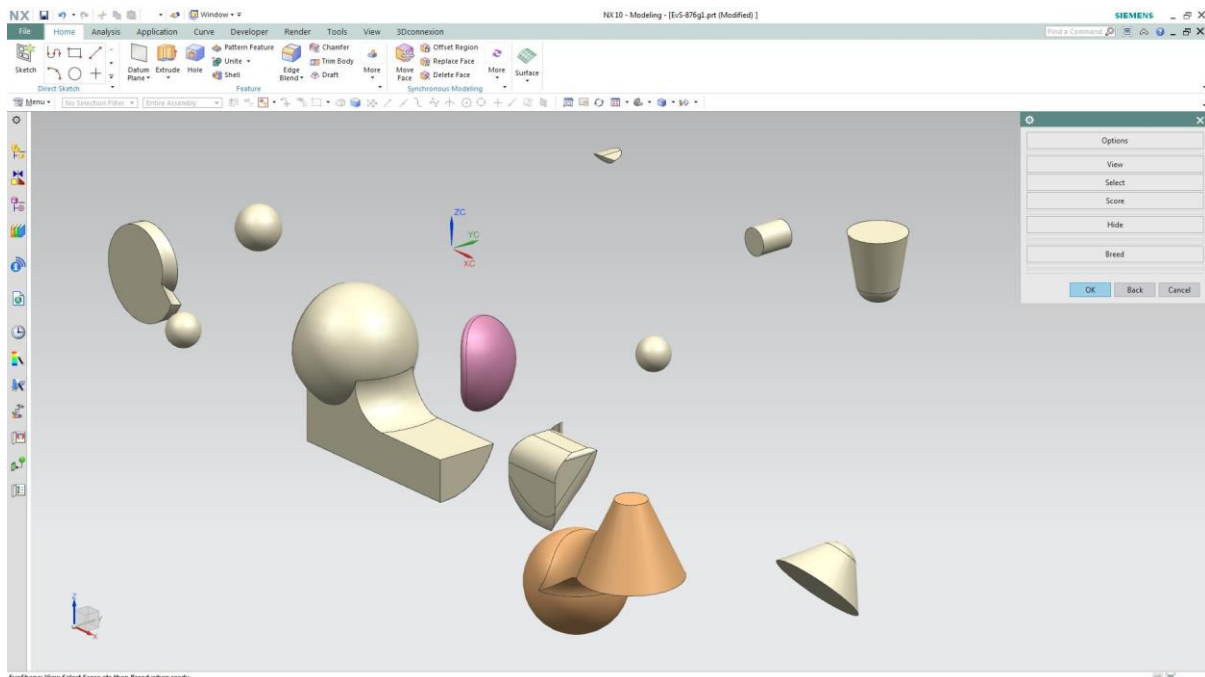


Figure 2.18. EvoShape screen-shot.

In the EvoShape system, each object is built up from five geometric primitives (selected from block, sphere, cone and cylinder) which interface via Boolean operators (unite, subtract and intersect), as shown in Figure 2.19.

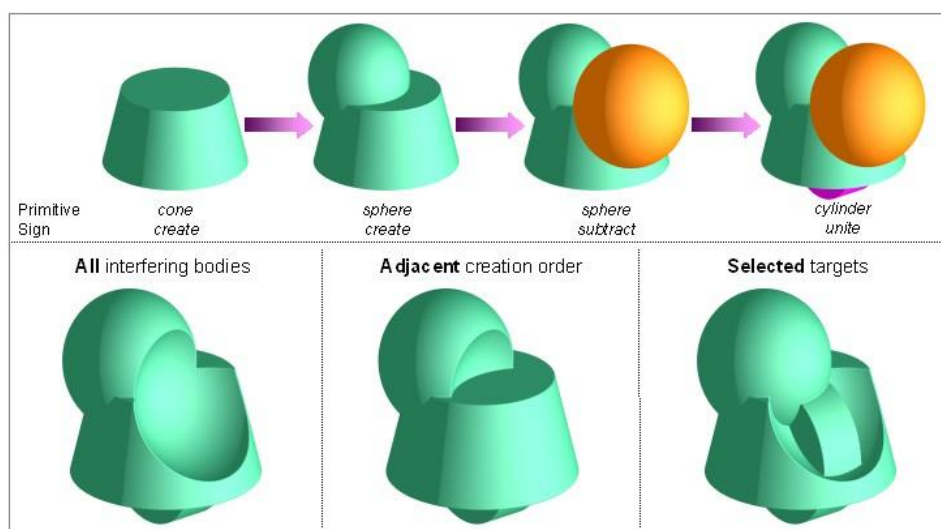


Figure 2.19. Object construction via Boolean interaction (Graham et al., 2008).

An edge blending operator is used to convert the original primitive edges, or those created by the Boolean operators, to smooth or round edges creating new and more interesting shapes as shown in Figure 2.20. This technique contributes greatly to the originality and usefulness of the EvoShape outputs.

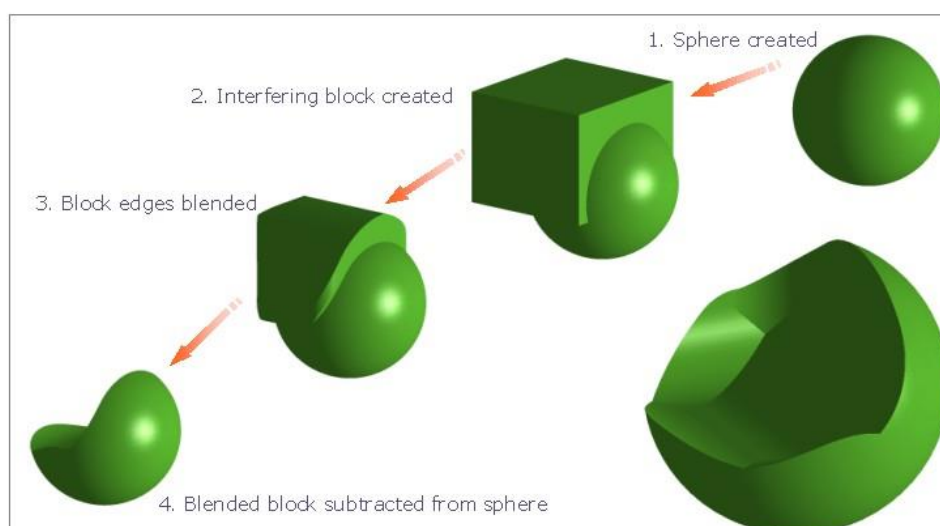
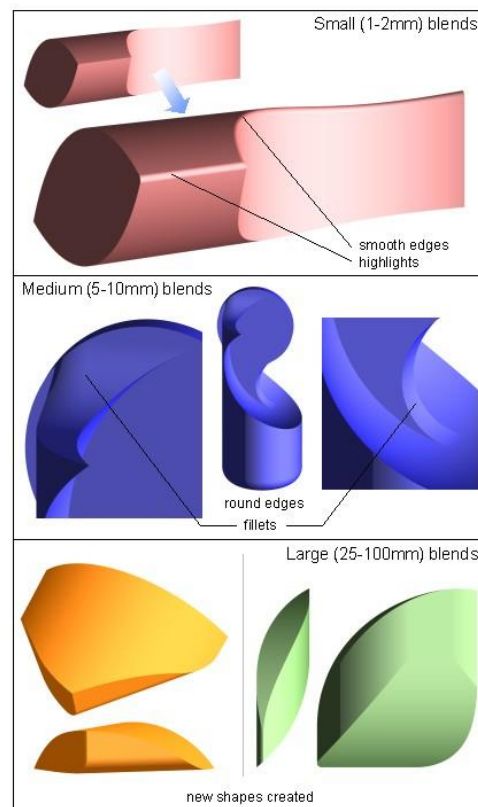


Figure 2.20. Edge blending strategies (Graham et al., 2008).

The EvoShape system is generic and flexible. The users do not need previous experience in using CAD modelling tools and the system is simple and helps users to control the evolutionary process by interacting with the system through selection.

Some industrial users of the system have expressed a desire for more control by setting some geometric constraints or guiding geometry at the onset. This led to the aim of this research, that the user will not only guide the evolution of 3D form by simply selecting or scoring shapes but also by using simple, pre-set fuzzy-constraints in order to create a more cooperative system that combines human interaction and geometric analysis, within an evolutionary design system. This complements the suggestions made for further work by Graham's (2002) thesis.

2.5 Geometry Mathematics

Some mathematical operations have been investigated, and are described in this section, such as centre of mass, bounding box, vectors, moment of inertia and principal axes.

A vector is a mathematical concept that has both magnitude and direction and is often expressed as a series of numbers. For example, in the 2D space of real numbers, the notation (1, 1) represents a vector that is pointed 45° from the x-axis towards the y-axis with a magnitude of $\sqrt{2}$ (Equation 2.2).

$$|u| = \sqrt{x^2 + y^2} = \sqrt{1^2 + 1^2} = \sqrt{2}$$

2.2

For finding the rotation of an object, a matrix of the column vector (x, y) is multiplied by a rotation matrix calculated by the angle θ . A rotation matrix is a matrix that is used to perform a rotation in Euclidean space. For example the matrix in Equation 2.3:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

2.3

rotates points in the xy-Cartesian plane counter-clockwise through an angle θ about the origin of the Cartesian coordinate system. To perform the rotation using a rotation matrix R, the position of each point must be represented by a column vector V, containing the coordinates of the point. A rotated vector is obtained by using the matrix multiplication RV, as shown in Equation 2.4:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = RV$$

2.4

The centre of mass is a position defined relative to an object or system of objects. It is the average position of all the parts of the system, weighted according to their masses. For simple rigid objects with uniform density, the centre of mass is located at the centroid. For example, the centre of mass of a uniform disc shape would be at its centre. Sometimes the centre of mass does not fall anywhere on the object. For example, the centre of mass of a ring is located at its centre, where there is no material. For more complex shapes, there is need of a more general mathematical definition of the centre of mass: it is the unique position at which the weighted position vectors of all the parts of a system sum to zero. A weighted position vector is a vector which points from the origin to an object and has magnitude m , where m is the mass of the object. For a system of n objects, the centre of mass is the point where (Equation 2.5):

$$\sum_{i=1}^n m_i \mathbf{r}_i = 0$$

2.5

where \mathbf{r} is a unit vector pointing from the origin to the object.

Bounding boxes are 3D boxes aligned with the global axes that surround entities within the model. There is a default bounding box for any new model that will surround all entities, including all groups and components. Bounding boxes are only large enough to exactly bind the entities within the model, group, or Minimum Bounding Box (MMB).

Moment of inertia depends on the mass of the object, its shape and its relative point of rotation. It is a measure of an object's resistance to changes in the rotation direction and has the same relationship to angular acceleration as mass has to linear acceleration. Moment of inertia of a body depends on the distribution of mass in the body with respect to the axis of rotation, and is the rotational analogue to mass. The mass moment of inertia about a fixed axis is the property of a body that measures the body's resistance to rotational acceleration. The greater its value, the greater the moment required to provide a given acceleration about a fixed pivot. The moment of inertia must be specified with respect to a chosen axis of rotation. The principal axes of a rotating body are defined by finding values of inertia. The symbols I_{xx} , I_{yy} and I_{zz} are frequently used to express the moments of inertia of a 3D rigid body about its three axes and products of Inertia are given by I_{xy} , I_{xz} and I_{yz} . The Inertia Matrix is shown in Equation 2.6:

$$I_p = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$$

2.6

The three eigenvalues are the principal moments of inertia and are known as A B and C. The three eigenvectors are the principal axes of inertia and are orthogonal. When the axes are aligned with the principal axes, the inertia matrix can be expressed as (Equation 2.7):

$$I_p = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix}$$

2.7

Some of these terms have been investigated in this research and are presented in Chapter 4.

2.6 Pose Normalisation

In recent years, the expanding availability of 3D digital content has created the need for massive 3D model datasets to be generated and the need for methods that can effectively search over large datasets. One of the most common problems that has to be faced is pose normalisation (translation, scaling and rotation) or alignment, which is most often achieved in applications such as:

- Robotics (Kouskouridas et al., 2012)
- Shape matching
- 3D object recognition
- 3D object retrieval
- Visualisation and modelling where 3D shapes need to be positioned in a canonical⁶ frame.

Before any processing, and due to arbitrary scale, position and orientation that 3D objects obtain, the normalisation of the shapes into a canonical coordinate frame is very important. An ideal canonical coordinate frame of a 3D model is the coordinate frame that has axes parallel to the front-back, left-right and top-bottom directions of the model. Pose normalisation includes alignment and scaling. Rotation normalisation is the most difficult part (Chaouch and Verroust-Blondet, 2009; Kazhdan, 2007; Rustamov, 2007), and the traditional technology that is used is the Principal Component Analysis (PCA).

In the following section, methods of assessing similarities will be covered. But before these tests can be done, objects need to be orientated, scaled, and translated. The box-model⁷ needs to be aligned within the evolving forms by allowing optimal geometric analysis. This is most commonly referred to in the literature as “Pose Normalisation” (Sfikas et al., 2011), (Papadakis, 2014), (Farrugia and Barbarar, 2015), (Axenopoulos et al., 2011), (Johan et al., 2011). In the proposed system, Pose Normalisation is necessary to align box-models (created before the evolutionary process is started) with the evolving forms. The aim is to influence the evolution of forms around and within simple pre-set geometry.

For more accurate and consistent results, many alignment and pose normalisation techniques have been proposed. Some of the following techniques may be applicable for this research.

⁶ In mathematics and classical mechanics, canonical coordinates are sets of coordinates which can be used to describe a physical system at any given point in time (locating the system within phase space). A phase space of a dynamical system is a space in which all possible states of a system are represented, with each possible state of the system corresponding to one unique point in the phase space.

⁷ Box modelling is a technique in 3D modelling where a primitive shape (such as a box, cylinder, sphere, etc.) is used to make the basic shape of the final model. This basic shape is then used to sculpt out the final model. The process uses a number of repetitive steps to reach the final product, which can lead to a more efficient and more controlled modelling process.

2.6.1 Reflective Symmetry

The symmetrical properties of a 3D shape can be used to implement rotation normalisation. In this method, the principal axes of the model are the axes that best segment the object into two reflecting parts.

A reflective symmetry computed on panoramic⁸ views was used for a novel pose normalisation method, as described by Sfikas et al. (2014). Due to their perspective that most objects hold a single plane of symmetry, the symmetry plane of a 3D model is detected and the first axis of the model is computed. The remaining principal axes are computed by calculating the variance of the panoramic views. The method is reported as being accurate and with better performance against other approaches.

First, on an arbitrarily chosen cylinder, the surface of the model is projected and aligned with a principal axis in space. A panoramic view representation is created by unfolding the 3D model over a 2D image plane. The 3D model's plane of symmetry is defined by the axis of the projection cylinder and the axis of maximum reflective symmetry. The 3D model's centroid is used to achieve translation normalisation, and scale normalisation is followed so that the 3D model exactly fits inside the unit sphere. A rotation of the symmetry plane takes place by orientating the surface so that it includes the Z axis and the plane of symmetry is detected in the panoramic image where the symmetry score graph is extracted, as shown in Figure 2.21.

⁸ A panoramic view is a wide-angle view in a 3D model.

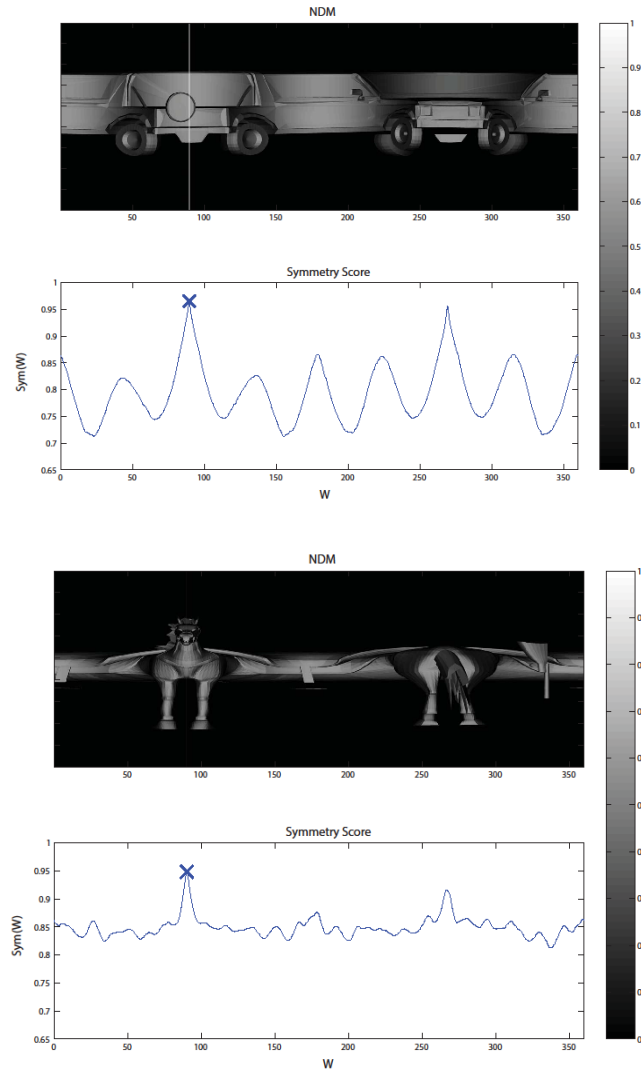


Figure 2.21. An example of panoramic images that represent the symmetric planes and the symmetry score graphs. At the bottom is the graphs of rotation (in degrees) giving the maximum symmetry score and at the top is the images of the best rotation, (Sfikas et al., 2014).

The Principal Component Analysis (PCA)-based pose normalisation methods (Shilane et al., 2004; Theodoridis and Koutroumbas, 2006; Zaharia and Prêteux, 2004) the best-known approach for computing the alignment of 3D objects, can be imprecise, produce poor alignments (Chen et al., 2003) and in most cases fail to capture symmetries and large planar surfaces. On the other hand, experimental results of Sfikas et al.'s (2014) method show that the performance is improved (Figure 2.22) and that the discriminative power of a 3D object retrieval system is enhanced.

However, the symmetry-based techniques are not able to handle complex symmetry 3D objects because symmetry detection either focuses on small fragments or bigger abstract areas of the 3D objects. Also, Sfikas et al.'s (2014) method blindly searches the complete 3D rotational space and aims to decrease the search space through the repeated rotation of the 3D model around one of the principal axes and the computation of the vertical reflective symmetry scores of its projections.

In some IED systems that use simple 3D models, the reflective symmetry algorithm should achieve robustness. But, IED systems with more complex models should produce inefficient alignments by using this method because of the very high computational demands.

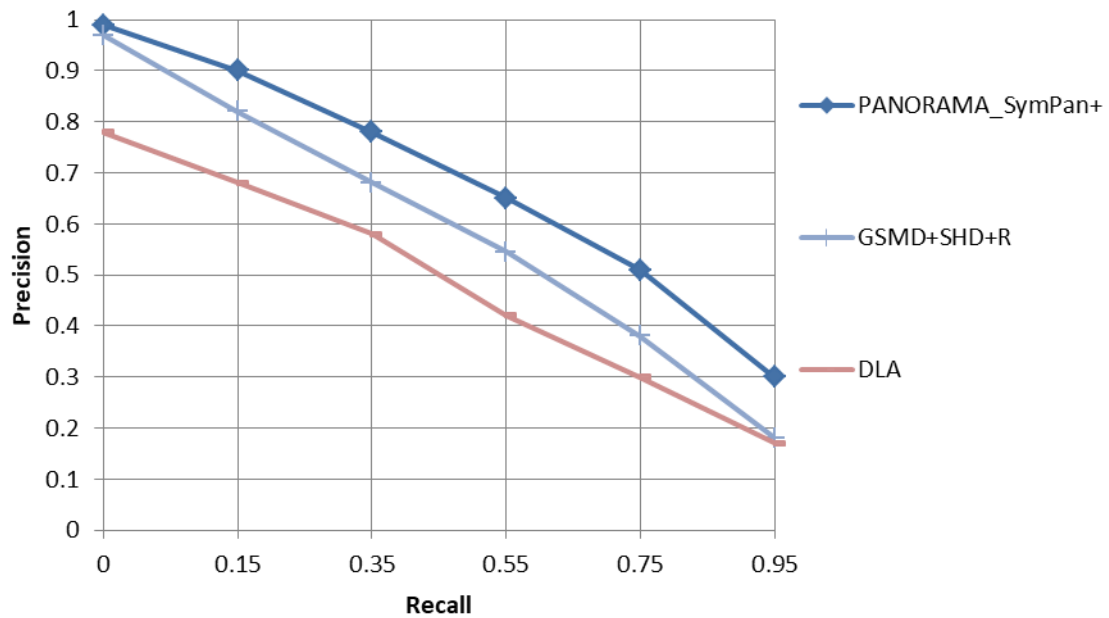


Figure 2.22. Precision-Recall (%) chart for Princeton Shape Benchmark test dataset. A comparison between SymPan+3D model pose normalisation method on PANORAMA retrieval results and state-of-the-art 3D object retrieval techniques (Sfikas et al., 2014).

2.6.2 Pairwise Alignments

Averkiou et al. (2015) focus on resolving rotational autocorrelations of shapes to ease shape co-alignment and therefore to ease rotational normalisation.

This technique is based on co-aligning all the objects jointly without an arbitrarily chosen reference shape. The reference shape is based on the first object that has been selected and therefore lowers the accuracy of the results when there are many different shapes in the database. In order to avoid comparing all the objects, this method is also based first on aligning similar shape pairs and spreading the alignment information to other shapes and second on comparing a shape with itself. They find out if two shapes are similar shape pair by comparing a shape with itself and then comparing this shape with a second shape. If the comparison with the second shape is close to the comparison with itself, then the two shapes are similar shape pair. They compare a shape itself by using an autocorrelation descriptor.

The aim is to find pairs of similar shapes by finding the angles that lead to local minima in order to sample only a small number of possible alignments and therefore to increase the speed of co-alignment. Only a small amount of relative angles leads to local minima, as shown in Figure 2.23.

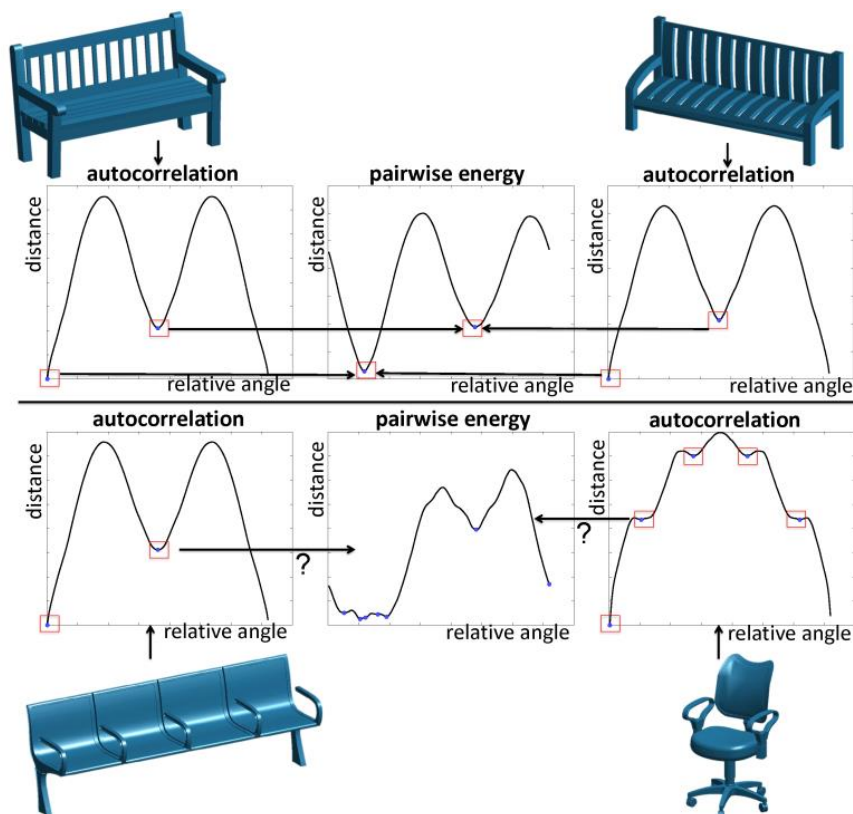


Figure 2.23. Pairwise alignments; an autocorrelation descriptor is used to predict their relative alignment (Averkiou et al., 2015).

Unlike Sfikas et al. (2014), this method does not use rotationally symmetrical shapes, but searches for near-symmetries that may provoke ambiguities in shape matching. These ambiguities are solved with a co-alignment formulation. With respect to state-of-the-art methods, this work results in higher-level performance, leading to efficient and accurate alignments (Figure 2.24), and should be applied to IED systems to produce efficient rotation normalisation. But, according to the authors, if a system contains models with a large number of near-symmetries, it will not be able to produce efficiency.

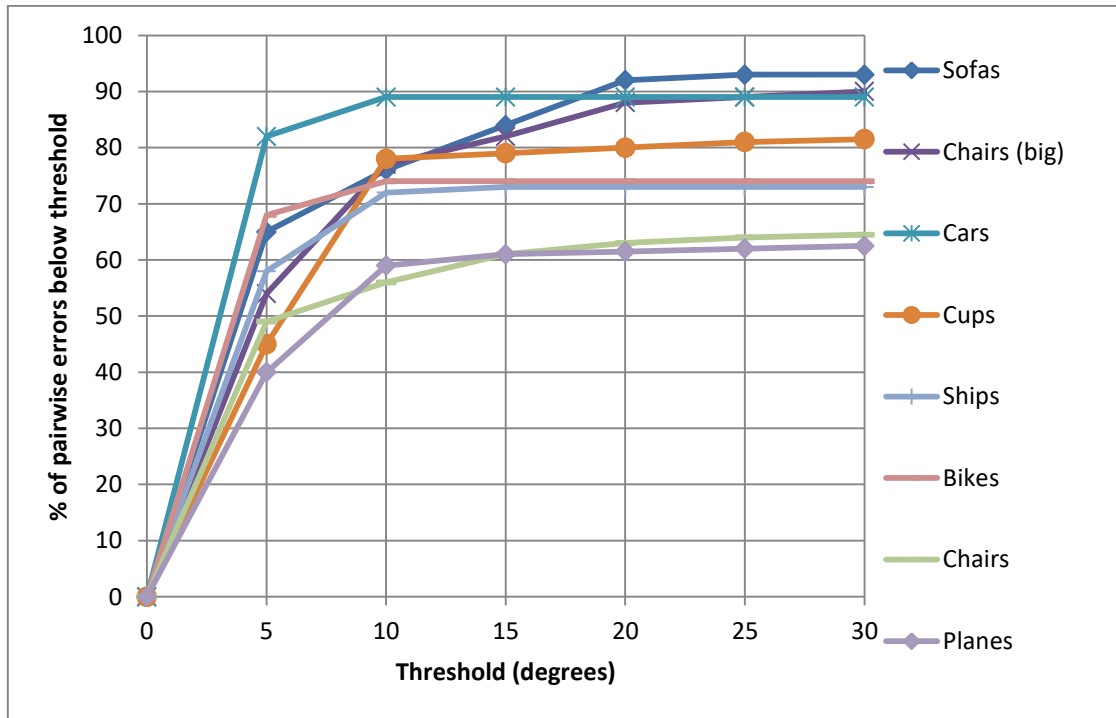


Figure 2.24. Fraction of models (y-axis)-angle threshold (x-axis) chart for 8 datasets. Averkiou et al.'s (2015) results are better than the results of random alignments (with only 8% accuracy for a threshold of 15°). This method suffers from near-symmetries of shapes (Averkiou et al., 2015).

2.6.3 Pose Normalisation for non-rigid shapes

Papadakis (2014) describes the usage of One-Class Support Vector Machines (OCSVM) to increase the consistency of translation and scale normalisation under non-rigid shapes.

In translation normalisation, usually, the centre of mass of a 3D model is computed and located to the coordinates' origin. However, this technique is not effective in 3D objects that are articulated or have extruding parts and outliers (Figure 2.25).

To alleviate this problem, Papadakis (2014) considers the surface of a 3D object as a collection of 3D points and uses OCSVM to compute the decision surface and find the volume constrained within the boundaries of the decision surface. The centre of the object is the centroid of the distribution of the volume. The algorithm is identifying parts that are

extremely small regarding the whole shape and that could derive negative results during the computation of the translation and scale.

In scale normalisation, fitting the scale of a 3D model inside a unit cube is also not an effective technique due to the possible presence of outlying parts (Figure 2.25). To normalise the scale, they calculate the average distance of the distribution of the object from its centre and scale the object in order this distance to be unit, as is shown in Figure 2.26(b).

In the presence of non-rigid transformations, a method similar to the one proposed in Sfikas et al. (2014) or Kazhdan et al. (2004), that depends on the symmetry properties of the objects, becomes less descriptive. The state-of-the-art approaches cannot cope with variation of 3D objects with articulations or extrusions, because the assumption that the centre of a 3D object is the centre of mass of the surface is not robust in the presence of non-rigid transformations. Thus, the OCSVM methodology helps to alleviate this problem and can easily be applied to some IED systems that handle complex 3D objects with extruding and outlying parts, and perform a more consistent translation and scale normalisation.

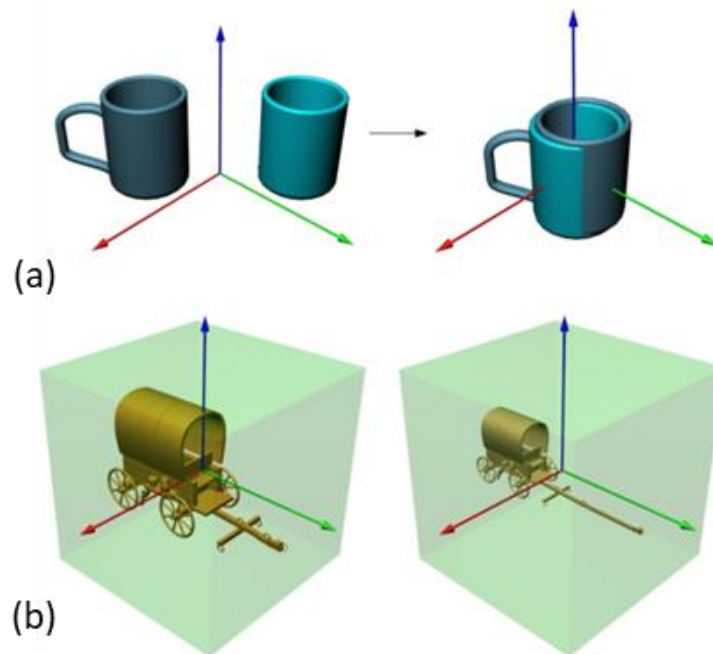


Figure 2.25. Translation normalisation using the centroid of the surface of the object, example in an object with and without extruding parts (a). Scale normalisation to the unit cube (b) (Papadakis, 2014).

The future aim of Papadakis (2014) is to improve rotation normalisation methods that are based on measuring 3D spatial features in order for the consistency of rotation normalisation to be increased.

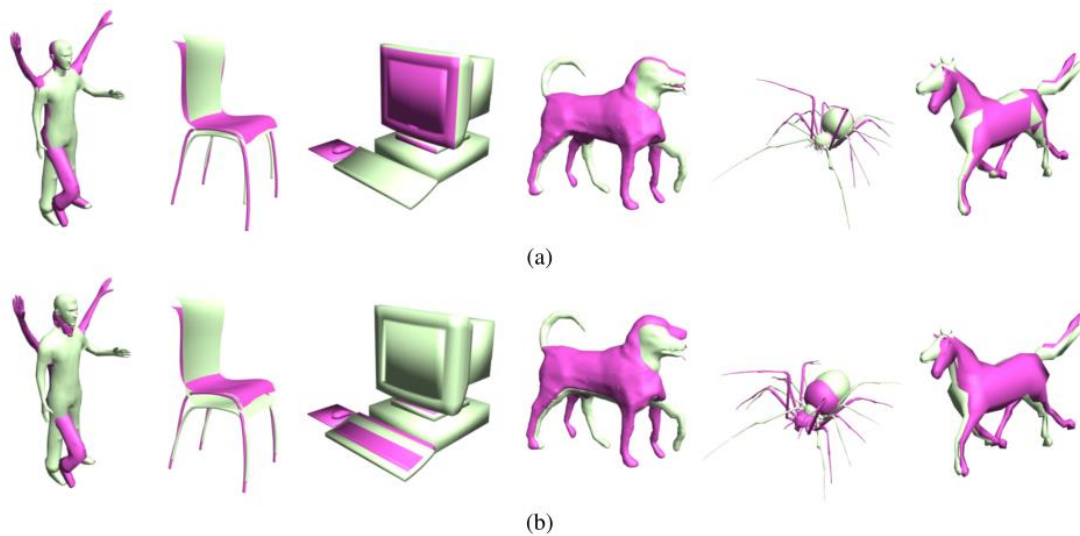


Figure 2.26. (a) Translation and (b) scale normalisation using OCSVM and the centroid and average distance of the surface (Papadakis, 2014).

2.6.4 Principal Component Analysis Methods

Farrugia and Barbarar (2015) focus on pose normalisation for 3D vehicle models using various pose normalisation techniques. There are two main techniques that can be used for rotational normalisation:

- The Principal Component Analysis (PCA) algorithm; and
- Computation of the symmetrical planes of a 3D model which is achieving rotation invariance.

The symmetrical appearance of 3D vehicles should support the use of the symmetrical plane method but there is only one symmetrical plane on each shape of a vehicle. Since PCA based techniques have been proven as the most suitable techniques for 3D vehicle retrieval due to their vertex distribution⁹, Farrugia and Barbarar (2015) tested four different PCA methods on a downloaded database. These four methods are:

- Principal Component Analysis (PCA)
- Centre of Gravity-PCA (CoG-PCA) (Paquet et al., 2000)
- Normal-PCA (NPCA) (Papadakis et al., 2007; Papadakis et al., 2008) and
- Continuous-PCA (CPCA) (Vranic et al., 2001; Vranic, 2004; Vranic, 2005)

The CoG method calculates the centre of the gravity of each mesh face rather than the actual vertices. In the NPCA technique, the principal axes are identified by the covariance of

⁹ In most 3D vehicle models, the coordinate values of the vertices have a predominant variance along three directions. That means that the PCA method is a sufficient technique to align 3D vehicles to a canonical orientation.

the mesh face and not by the vertex points. CPCA is the PCA method applied to an infinite continuous point set rather than to a traditional discrete point set.

All these variances of the PCA differ from the original only in the computation of the covariance matrix and they have been studied because many researchers have shown that the PCA technique may lead to inconsistent results.

For each of the four techniques the authors used two methods to evaluate the pose normalised models:

- In method 1, they record the number of models that aligned with the other models in the same class;
- In method 2, they compare the effectiveness of the four techniques using a scan-line feature descriptor algorithm.

Using the experimental results given by the authors, the CoG-PCA method was more effective regarding computation time. The CPCA method had slightly better performance (Tangelder and Veltkamp, 2008) than the CoG-PCA and noticeably better performance than the PCA and Normal-PCA. Regarding reflection normalisation, PCA was inadequate (Pears et al., 2010) as shown in Figure 2.27, while NPCA had inaccurate results in the three axes' identification as shown in Figure 2.28.

The CPCA-based pose normalisation method, in some cases can be unsuccessful to detect some specific characteristics of 3D models such as symmetries, but they should be more effective when they are applied to IED systems that handle simple geometric shapes.

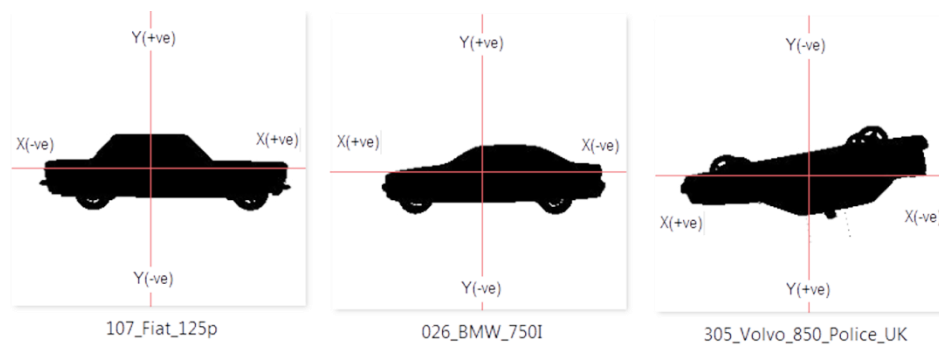


Figure 2.27. PCA's inconsistency in identifying the positive and negative side of an axis (Farrugia and Barbarar, 2015).

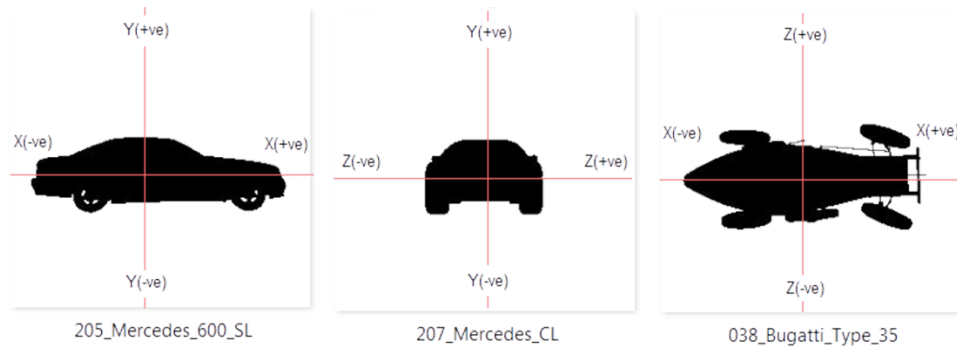


Figure 2.28. NPCA's inconsistency in identifying the three axes (Farrugia and Barbarar, 2015).

2.6.5 Combined Pose Estimation (CPE)

Axenopoulos et al. (2011) combine the plane reflection symmetry and recti linearity¹⁰ to attain a 3D model alignment by using the PCA method. The method starts by locating the 3D object in a coordinate system and the object generates 2D images. A flip-invariant shape descriptor is used for each image. The method is summarised in Figure 2.29.

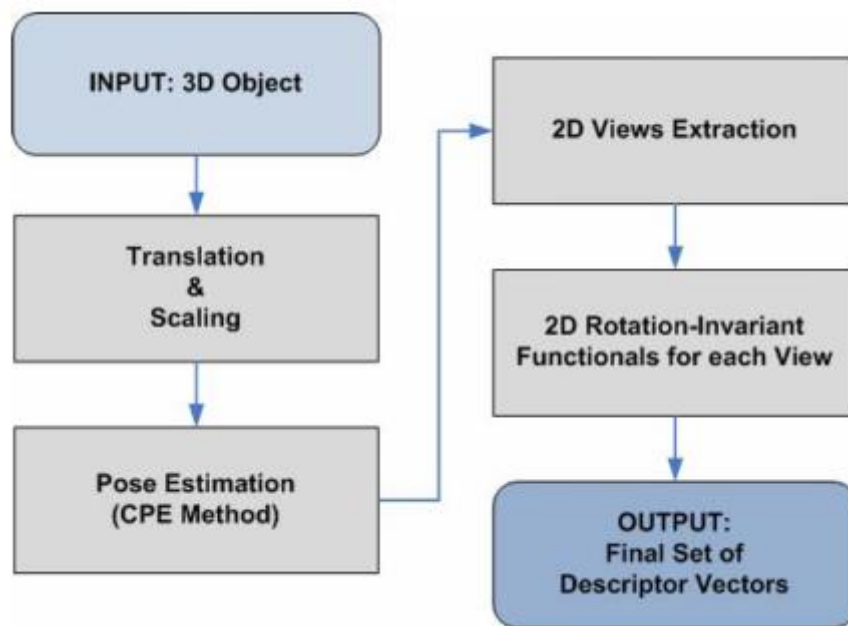


Figure 2.29. Diagram of the method (Axenopoulos et al., 2011).

First, the 3D object is translated by locating the centre of the mass to the centre of the coordinate system and then it is scaled to fit within a bounding sphere. Then, rotation estimation takes place using a novel Combined Pose Estimation (CPE) method which combines the Continuous-PCA with plane symmetry and recti-linearity. The CPE method is shown in Figure 2.30.

¹⁰ Recti linearity is defined as the maximum ratio of the surface area to the sum of three orthogonal projected areas of the mesh.

CPCA is used to produce a first pose estimation and the reflection symmetry is the computed. If there are two or three planes of symmetry, the transformation is kept as it is otherwise the algorithm needs to be corrected using recti linearity. Recti linearity is invariant to scaling, translating and rotating and is also invariant to flipping of the coordinate axes. PCA and recti-linearity happen at the same time and the one that produces the best rotation estimation is chosen.

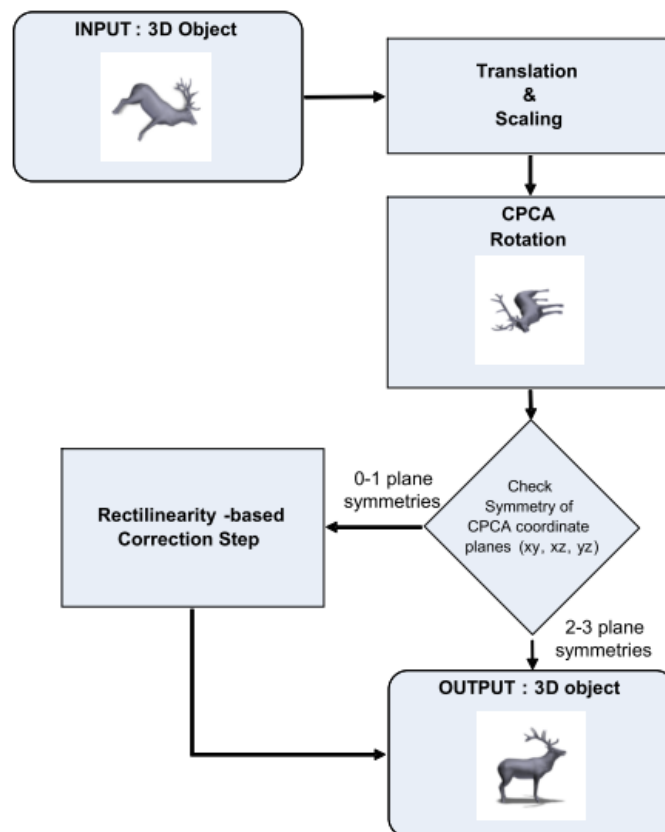


Figure 2.30. Diagram of the CPE method (Axenopoulos et al., 2011).

After alignment, a set of 2D black/white views is extracted and from each 2D view a descriptor vector is also extracted. These vectors are converted into a single descriptor vector which describes the shape of the object. The 2D vectors are flip-invariant which means that even if the orientation is not consistent, the resulting descriptors are the same. View-based similarity is computed by matching each 2D view between 3D models, so that the 3D models can be properly aligned in a coordinate frame.

The experimental results given in Axenopoulos et al. (2011) prove that the CPE technique produces accurate rotation estimation results in comparison with other state-of-the-art approaches.

This technique should achieve robustness in IED systems that use simple models. However, using the centre of the mass for translation normalisation will not produce consistent alignment in more complex objects.

2.6.6 Minimum Projection Area (MPA)

Johan et al. (2011) focus on pose normalisation using a novel Minimum Projection Area-based (MPA) alignment method. They use an optimisation method¹¹ that finds three principal axes to align a model:

- The first axis gives the MPA;
- The second axis is vertical to the first and also gives the MPA;
- The third axis derives from the point that the other two axes intersect.

An object has a MPA by orthogonally projecting it in the direction parallel to one of the axes of the ideal canonical coordinate frame. Figure 2.31 shows that there are three canonical orthographic projection views for every 3D model: front-back, left-right, and top-bottom views.

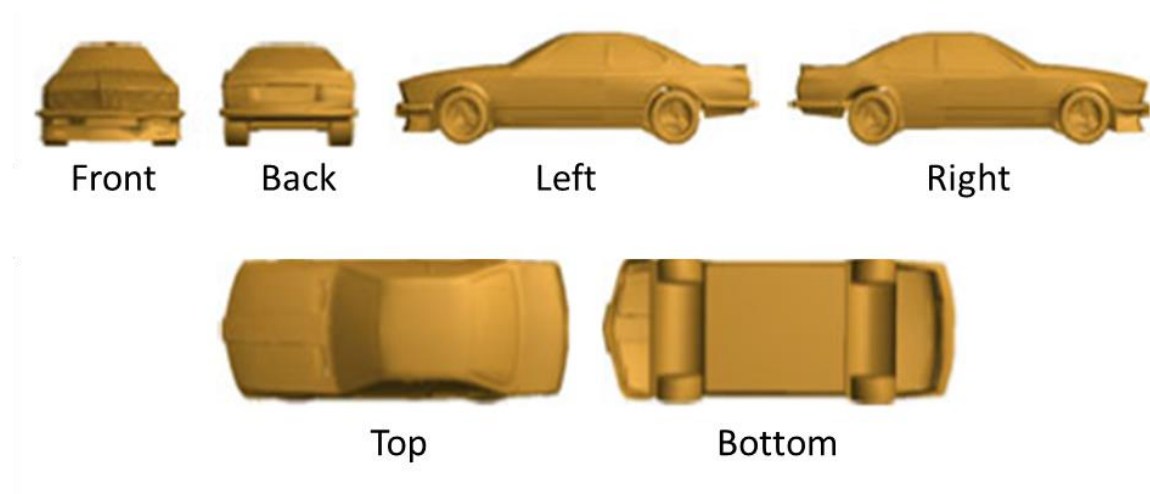


Figure 2.31. Canonical orthographic projection views of a car model based on its ideal canonical coordinate frame (Johan et al., 2011).

Each of the three canonical views has a minimum projection area that is compared to some arbitrary views, as shown in Figure 2.32.

The axes orientate so that similar models can be aligned in similar poses. Johan et al. (2011) use a sphere to find the axis with minimum projection by sampling a set of points on the surface of the sphere. The MPA method can be considered as a view-based approach and Johan et al. (2011) use view point selection techniques to find the best views that describe a 3D model.

¹¹ This method is based on the PSO method which was employed by Johan et al. (2011) to efficiently find the axis with minimum projection area.

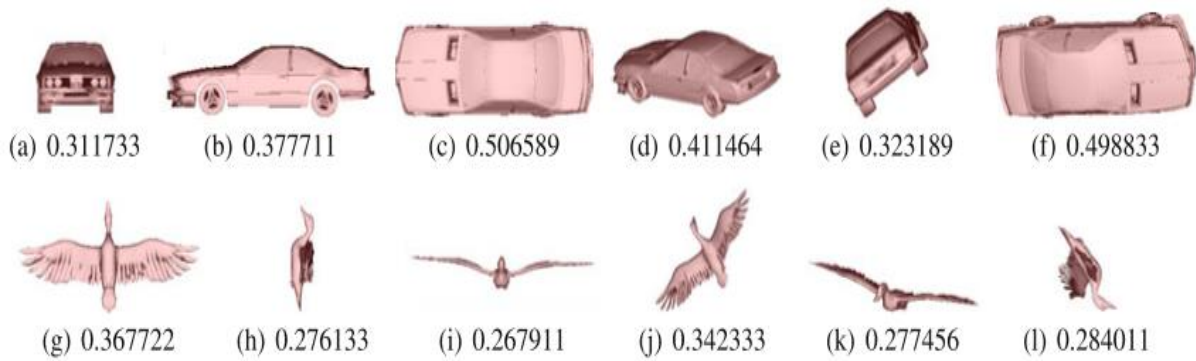


Figure 2.32. Examples showing that one canonical view of a 3D model usually has the MPA. In each row, the first three images are the front, left, and top views of a 3D model and the remaining three images are three arbitrary views of the same model. The number underneath each view is its normalised projection area (Johan et al., 2011).

The experiments, that Johan et al. (2011) performed, prove that the alignment axes are consistent and parallel to the canonical frame and the pose normalisation of similar models is robust under variations and noise. This algorithm is not based on the traditional PCA-based approach but attains better results in 3D object retrieval than CPCA, NPCA and PCA.

In further work, symmetry should be combined with the MPA so that the method can be improved. This should produce efficient rotation normalisation before the 3D comparison between the forms and the GG, when applying it to IED systems.

2.6.7 Planar-Reflective Symmetry Transform

Podolak and Funkhouser (2006) describe a planar reflective symmetry transform (PRST) for 3D models that captures a continuous measure of the reflectional symmetry of a shape respectfully to all potential planes. This transform is computed by a Monte Carlo¹² sampling algorithm which is constant under transformations, and determines the centre of symmetry as well as the principal symmetry axes which are essential for aligning models in a canonical system.

As shown in Figure 2.33, the darkest point (which is the greatest symmetry) is the centre of the object and the main lines are the main axes. In some parts of the objects (the motorcycle wheels for example) the symmetries represent the local maxima while some asymmetric parts (the mug handle for example) do not influence the global maxima.

¹² In computing, a Monte Carlo algorithm is a randomised algorithm whose running time is deterministic, but whose output may be incorrect with a certain (typically small) probability.

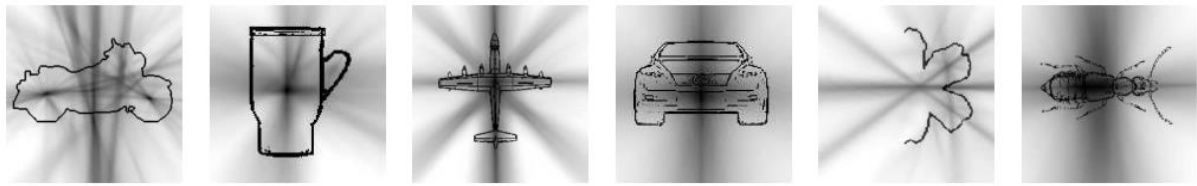


Figure 2.33. The PRST captures the degree of symmetry of arbitrary shapes with respect to reflection through all planes in space. Darker lines represent greater symmetries (Podolak and Funkhouser, 2006).

As it is known, alignments are usually computed with PCA using the centre of mass as the origin and the principal axes as the orientation. However, the results are not always consistent, desirable and similar to what humans would choose. This can be proved considering the mugs in Figure 2.34.

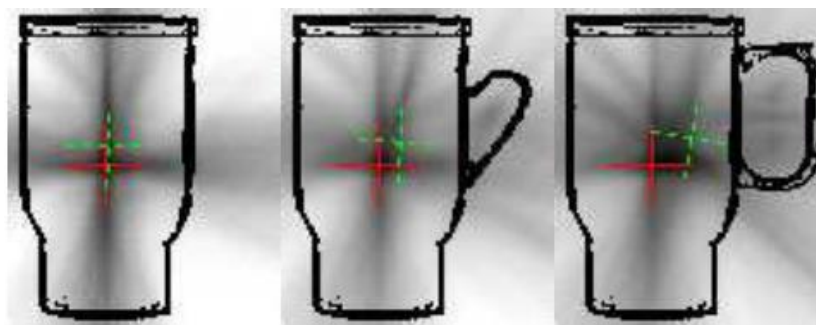


Figure 2.34. A mug with and without handles. The centre of mass and PCA axes are drawn in dotted green (they move depending on the presence of handles). The centre of symmetry and principal symmetry axes using PRST are shown in solid red (they remain constant under transformation of the shape) (Podolak and Funkhouser, 2006).

Podolak and Funkhouser (2006) produce better alignments using the PRST and introducing the Centre of Symmetry (CoS) as well as the principal symmetry axes (PSA). The plane with maximal symmetry is the first PSA, the maximal symmetry vertical to the first is the second PSA and the plane which is vertical to both first and second axis is the third PSA. Finally, the intersection of those three planes is the centre of symmetry.

For the matching process, Podolak and Funkhouser (2006) employ a shape descriptor using the PRST to calculate the diversity between two aligned meshes between their discrete PRSTs. When the one object is symmetrical and the other is not, this measure produces a large distance.

Although this technique captures not only global symmetries of an object but also local symmetries of main parts, it successfully detects only symmetries of an entire object (while Podolak and Funkhouser (2006) are using the segmentation algorithm, local symmetries sometimes fail to be detected). This algorithm should achieve efficient normalisation by being applied in IED systems as the centre of symmetry and principal symmetry axes remain constant under transformation of shapes but it may not be robust for some complex objects.

2.6.8 Partial Symmetry-Based Alignment

Complex objects with multiple symmetric parts corresponding to different sized regions has been researched by Dang et al. (2014) and is called Partial Symmetry-Based Alignment.

Dang et al. (2014) describe a partial symmetry-based alignment for 3D models with Non-Uniform Rational B-Rep¹³ (NURBS) surfaces, as shown in Figure 2.35. The dominant partial symmetric surface is found and used to determine the first canonical plane from which the rest are identified based on the MPA. This method uses robust algorithms to detect partial symmetries and to align the models based on their dominant partial symmetries.

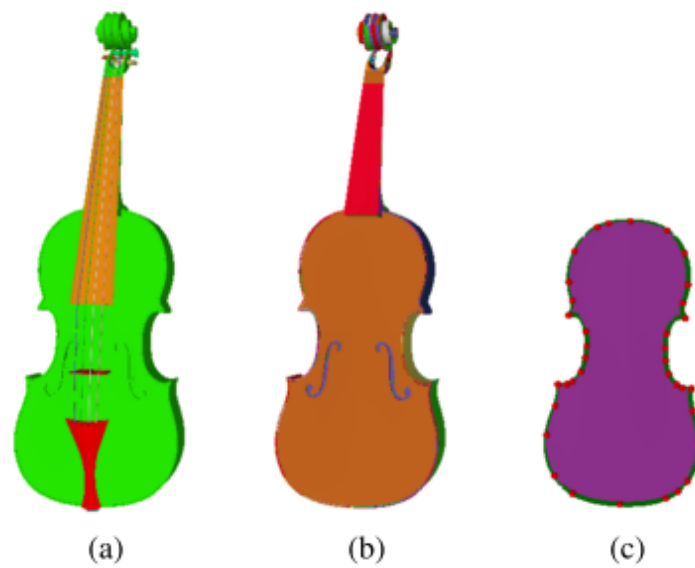


Figure 2.35. (a) B-Rep objects in a violin model. (b) B-Rep object representing a violin body. (c) B-Rep entities within a face of the violin body (Dang et al., 2014).

The two algorithms that are used are as follows:

- The symmetry detection algorithm that detects the dominant symmetry plane, as shown in Figure 2.36; and
- The axes generation algorithm that evaluates the three canonical axes based on orthographic projection.

¹³ B-Rep model consists of multiple B-Rep objects and B-Rep object consists of multiple faces.

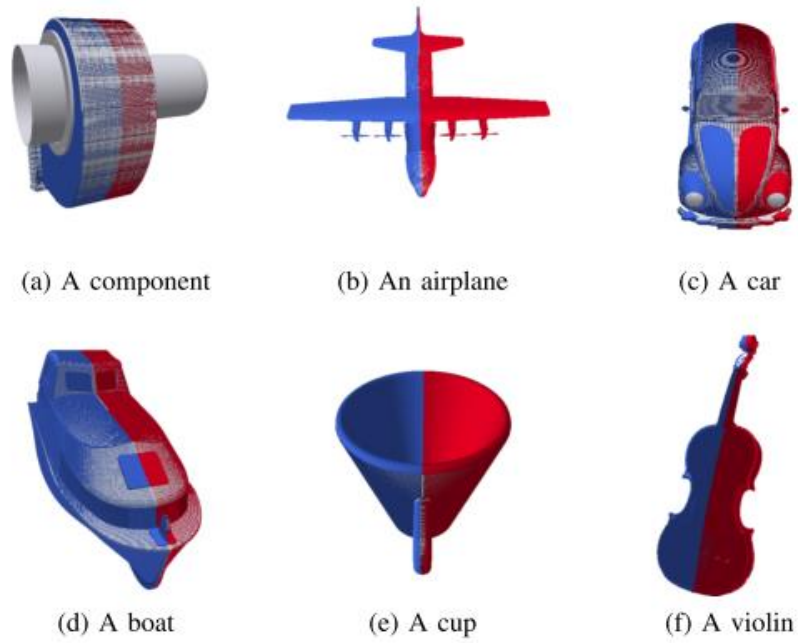


Figure 2.36. Results of symmetry detection (Dang et al., 2014).

After the detection of the symmetrical faces within the model, the dominant symmetry is found and the first canonical plane is estimated. The second axis is vertical to the reference axis and the third is determined by the latter. Examples of alignment are shown in Figure 2.37.

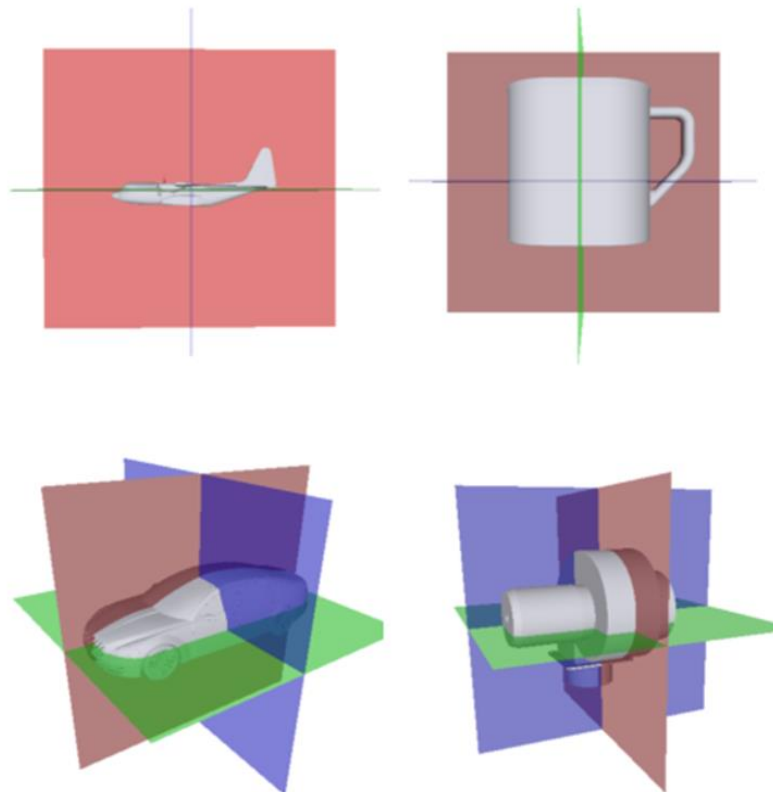


Figure 2.37. Alignment results: The pink plane is the dominant symmetry plane (first canonical plane), the blue plane is the second plane, and the green is the third plane (Dang et al., 2014).

Compared to PCA approaches, this method is closer to human perception and it works better even if the model has a partial symmetry or even if the model is complex, since it is based on matching of trimming loops corners. The precision is higher than the PCA since it does not use global symmetry. The partial symmetry-based alignment method should achieve robustness and effective alignment in IED systems with complex models.

2.6.9 The Spherical Trace Transform

Zarpalas et al. (2007) present a novel method for 3D model search and retrieval using the spherical trace transform. After the 3D objects are translated and scaled, a descriptor vector that is rotation invariant is constructed without the use of rotation normalisation. The spherical trace transform is tracing the volume of a 3D object using:

- 2D planes which are tangential to concentric spheres;
- Radius segments, as shown in Figure 2.38, where Δ_ρ is the length of the radius segment.

Descriptor vectors that are rotation invariant are produced using a set of functions.

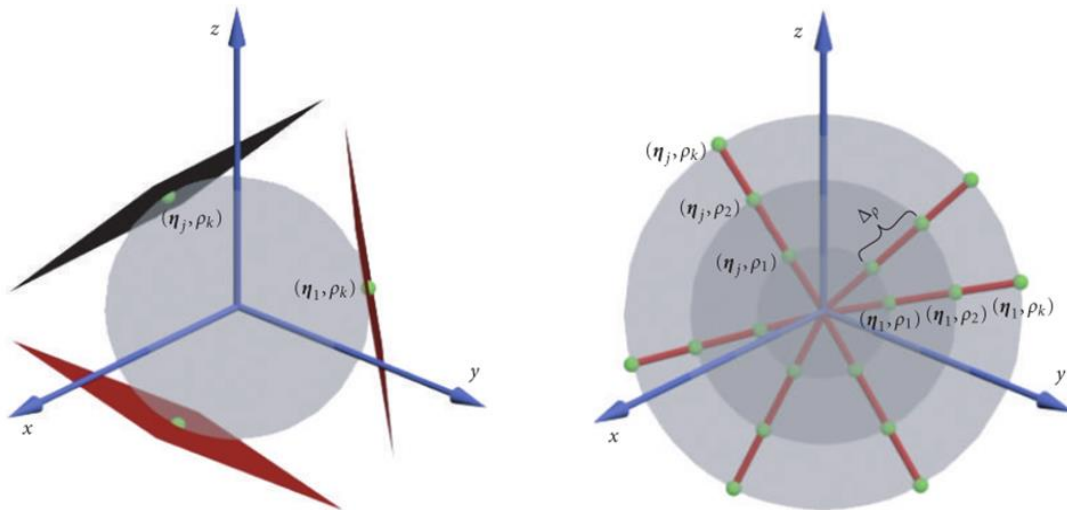


Figure 2.38. The spherical trace transform: The define plane $\Pi(\eta, \rho)$ is tangential to the sphere Sp with radius ρ and centre at the origin, at the point (η, ρ) , where η is the unit vector in \mathbf{R}_3 (Zarpalas et al., 2007).

Specifically, a set of functions is applied to the volume of the object, a new domain of concentric spheres is produced where a new set of functions is applied and finally a descriptor vector which is rotation invariant is constructed.

In this method, a descriptor is produced without the use of rotation normalisation. When rotation normalisation is used for an object at specific radius, a different shaped object will be created and the same descriptors will be produced for these different objects (because all shells of the same radius are integrated). However, Zarpalas et al. (2007) method is not affected by independent rotations (Figure 2.39); the descriptors that are produced are not

identical due to the fact that while tracing the volume of the object, the planes will contain information of the object in different radius.



Figure 2.39. Rotation of $f(x)$ rotates $F(\eta, \rho)$, without being affected by independent rotations (upper left image). Thus, $F(\eta_2, \rho_1) = F(\eta'_2, \rho_1)$ (Zarpalas et al., 2007).

Zarpalas et al. (2007) have shown in their experimental results that this method is precise and these descriptors are used for 3D model matching in an efficient manner and may be efficiently adapted for IED systems.

2.7 Minimum-Volume Bounding Box

The problem of defining the minimum oriented bounding box (OBB) happens in many industrial applications like packing, optimum layout design, image processing and shape analysis. The algorithm can be applied to many other fields like fitting an object into a predetermined container or creating bounding boxes in solid modellers to improve their performance. Three-dimensional boxes can also be applied to some important applications like computer graphics (for fast rendering of a scene or for collision detection), and statistics (for storing and performing range-search queries on a large database of samples). The problem in such applications are divided into two: finding a good split of a given set of points into two (or more) subsets, e.g., in a way that minimises the maximum diameter of the subsets, and computing an optimal box (or another generic shape) that encloses each subset.

2.7.1 Approximation algorithms of a point set in three dimensions

Barequet and Har-Peled (2001) give effective algorithms for solving the following problem: Given a set S of n points in R^3 and a parameter $0 < \varepsilon \leq 1$ which is the accuracy controlling factor, find a rectangular box that encloses S whose volume is at most $(1 + \varepsilon)$ times the minimum volume of a bounding box of S . Barequet and Har-Peled (2001) focus on computing a nearly optimal box that encloses subsets of given sets of points rather than on finding a good splitting of a given set of points into two (or more) subsets (Huebner et al., 2008).

Barequet and Har-Peled (2001) present two $(1 + \varepsilon)$ -approximation algorithms for the Minimum-Volume Bounding Box (MVBB) problem in R^3 . Both algorithms use an approximation of the diameter of S to define a grid. First, a good enough approximation of the diameter of the point set is needed, which is the distance between the two furthest points of S . Second, they compute an approximating box, as shown in Figure 2.40 and Figure 2.41. The diameter of the point set is chosen as an approximate direction for building the first approximate OBB.

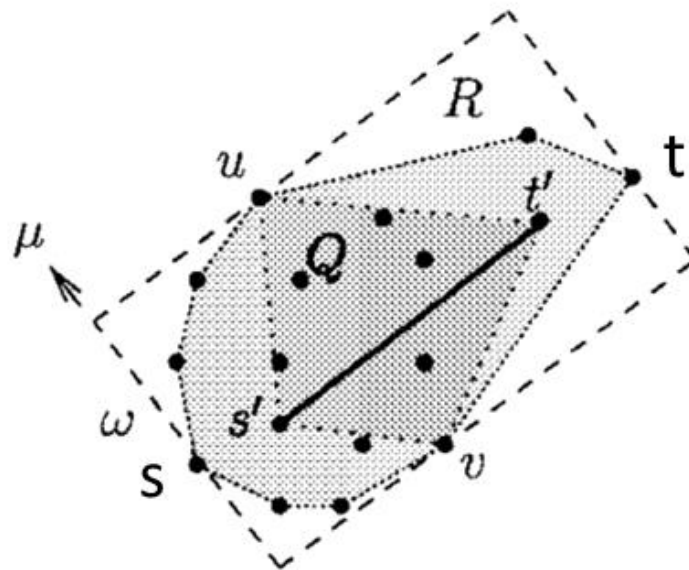


Figure 2.40. Q is a set of n points in R^2 . s and t are the points in S that define the longest edge of bounding box. s' and t' are two points in Q . μ is the direction perpendicular to st and $s't'$. R is the minimum-area bounding rectangle of Q in the direction $s't'$ and μ . ω is the length of the edge of R in the direction μ . u and v are the two points of Q lying on the two edges of R parallel to $s't'$ (Barequet and Har-Peled, 2001).

The first algorithm approximates S by a set of grid points and computes the exact MVBB of the last set. This algorithm is possibly too difficult to implement, the computational time is significant if the accuracy required is high and the number of points is large. So, they propose a second less effective approximation algorithm but easier to implement. The second algorithm computes the box with a minimal volume out of a small number of bounding boxes whose orientations are induced by the grid. Different orientations of the

OBBs of the point set are computed based on the vector directions of two of the grid points. The minimum bounding box is then recorded.

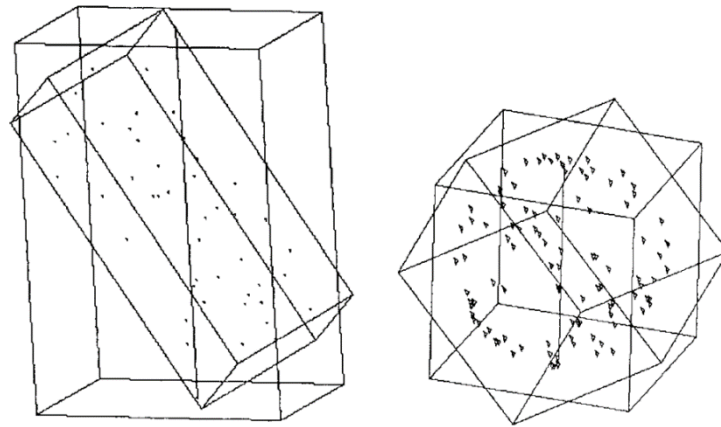


Figure 2.41. Bounding boxes of two spatial point sets. The diameter of the sets are shown as a nearly vertical line segments (Barequet and Har-Peled, 2001).

A MVBB of a scaled-up space is a $(1 + \epsilon)$ -approximation of the MVBB of S . This version of the algorithm requires big running time. However, it is possible to reduce the running time of the first algorithm by scaling down the space so as to transform the bounding box to a unit cube. This method together with an efficient PN technique can effectively minimise the R area between the GG (bounding box) and the forms when adapted for IED systems.

2.7.2 Shape approximation in robot grasping

Huebner et al. (2008) enclose 3D data points into primitive box shapes by using a ‘fit and split’ algorithm that is based on Barequet and Har-Peled’s (2001) effective MVBB algorithm. Huebner et al. (2008) were interested in investigating which primitive shape representations might be appropriate for the task of grasping arbitrary, unseen objects Huebner et al. (2008) chose boxes as a very simple and rough approximation representation.

Huebner et al. (2008) produce a tight-fitting, oriented MVBB that encloses the data points by performing the computation on an arbitrary point cloud, as can be seen in Figure 2.42. The aim is to approximate a shape with as few boxes as possible.

Huebner et al. (2008) decided to test only the planes parallel to the parent MVBB, as Figure 2.43 shows. The relation of the box volume before and after can be a measure of a good split: the less volume that the two resulting child MVBBs will have, the better the split of the parent box is.

To decrease the problem, Huebner et al. (2008) evaluate the best cut by first projecting the data on 2D grids which correspond to the surfaces \bar{A} , \bar{B} and \bar{C} and then splitting a 2D box by an edge-parallel line. The Figure 2.44 shows that there are six valid split directions left, two for each of the surfaces \bar{A} , \bar{B} and \bar{C} . The best bunny cuts that minimise the summed volume of the two partitions are shown in Figure 2.45.

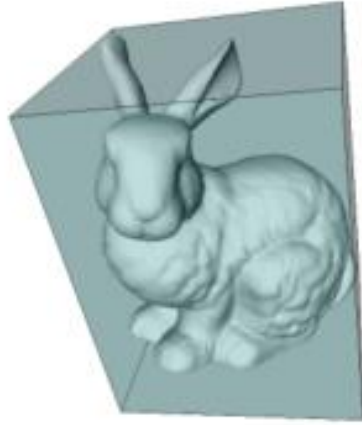


Figure 2.42. The Stanford bunny model and the root MVBB of its vertices (Huebner et al., 2008).

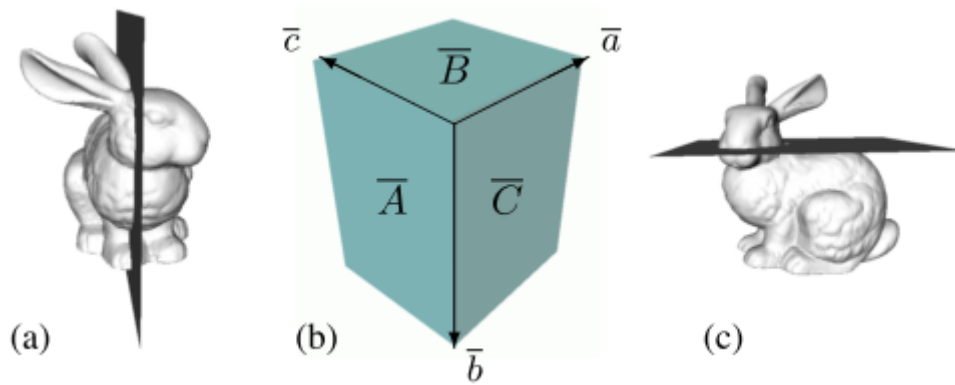


Figure 2.43. (a) A mean cut of a bunny model. (b) They test only the planes parallel to the parent MVBB. (c) A good cut parallel to the root MVBB plane \bar{B} (Huebner et al., 2008).

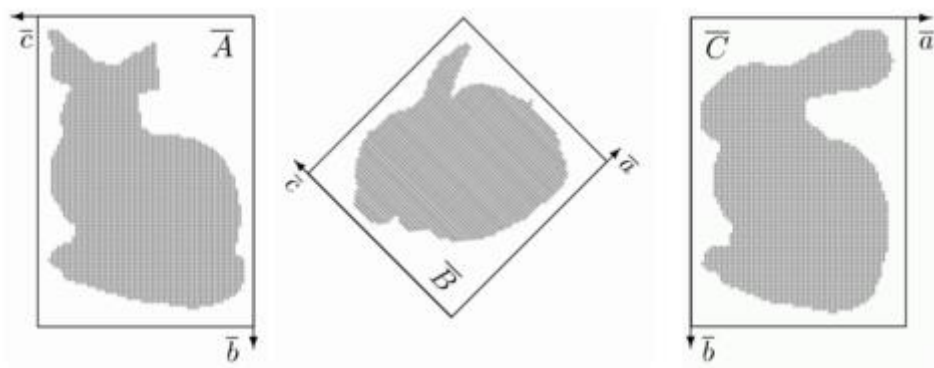


Figure 2.44. Bunny sample projections onto the three faces of the root box (Figure 2.43) according to the face-parallel cutting scheme in Figure 2.43 (b) (Huebner et al., 2008).

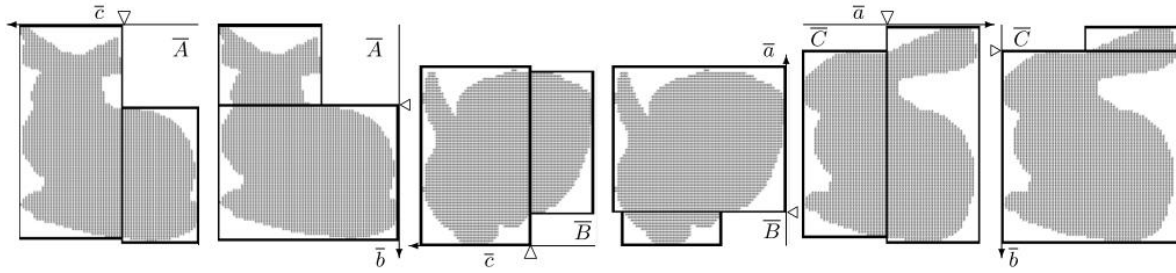


Figure 2.45. Best cuts along the six box directions (Huebner et al., 2008).

Huebner et al. (2008) propose MVBBs as effective box decomposition on a ‘fit and split’ approach. This technique is highly efficient and simple for achieving exact shape approximation. However, exact approximation may not be necessary for grasping tasks. This technique is based on box representation and decomposition with an efficient splitting criterion that offers fast computational methods.

The ‘fit and split’ method can achieve robustness in the IED system proposed in this research – splitting the bounding box of the forms during scaling, can produce more accurate scaling results.

2.7.3 Iterative approach of an arbitrary solid

Chan and Tan (2001) propose a technique for determining the minimum oriented bounding box of an arbitrary object by projecting the object onto the three principal planes and using the projected contours for analysis. The orientation of the contours are determined by rotating them within a specific angle range. These orientations are then used to approximate the orientation of the object so that its bounding box volume is minimised.

According to Chan and Tan’s (2001) first theorem, if an arbitrary object is reoriented in such a way that the bounding boxes of its projected areas to the three principal planes are a minimum, the volume of the bounding box of the reoriented object so derived is also a minimum. So, the axis aligned bounding box (AABB) of the object is equal to its minimum oriented bounding box (OBB_{min}). Hence, the question is how to reorient an object to achieve this theorem.

Chan and Tan’s (2001) second theorem shows that the bounding volume of an object will become a minimum if the model is rotated about an axis passing through the AABB centre (as shown in Figure 2.46) and reoriented according to the minimal projected area of AABB on the three principal planes successively. Then, the volume of $AABB_v$ of the object is equal to the minimum volume of OBB_{min_v} of the object. The steps to achieve this theorem are:

- Find the bounding box of the object as shown in Figure 2.46a.
- Create an axis R_z that is vertical to the XY plane and is passing through the centre of the bounding box (see Figure 2.46a).

- The object is repeatedly rotated about R_z until its projected bounding box area on the XY planes achieves a minimum. This gives the XY-bounding box whose y edge length is smaller than the x edge length. The resultant object is shown in Figure 2.46b.
- The model is rotated about R_y until its projected bounding box area on the ZX plane is a minimum. Axis R_y is chosen because the y edge length does not change during the rotations. This gives the ZX-bounding box with a z edge length smaller than the x edge length as shown in Figure 2.46c.
- Axis R_x , which is vertical to the y and z edge lengths, is chosen. The object is rotated about R_x until its projected bounding box area on ZY plane achieves a minimum (see Figure 2.46d).
- The object is now oriented at a position which gives an approximate minimum bounding box volume. The rotation sequence is $R_z \rightarrow R_y \rightarrow R_x$.

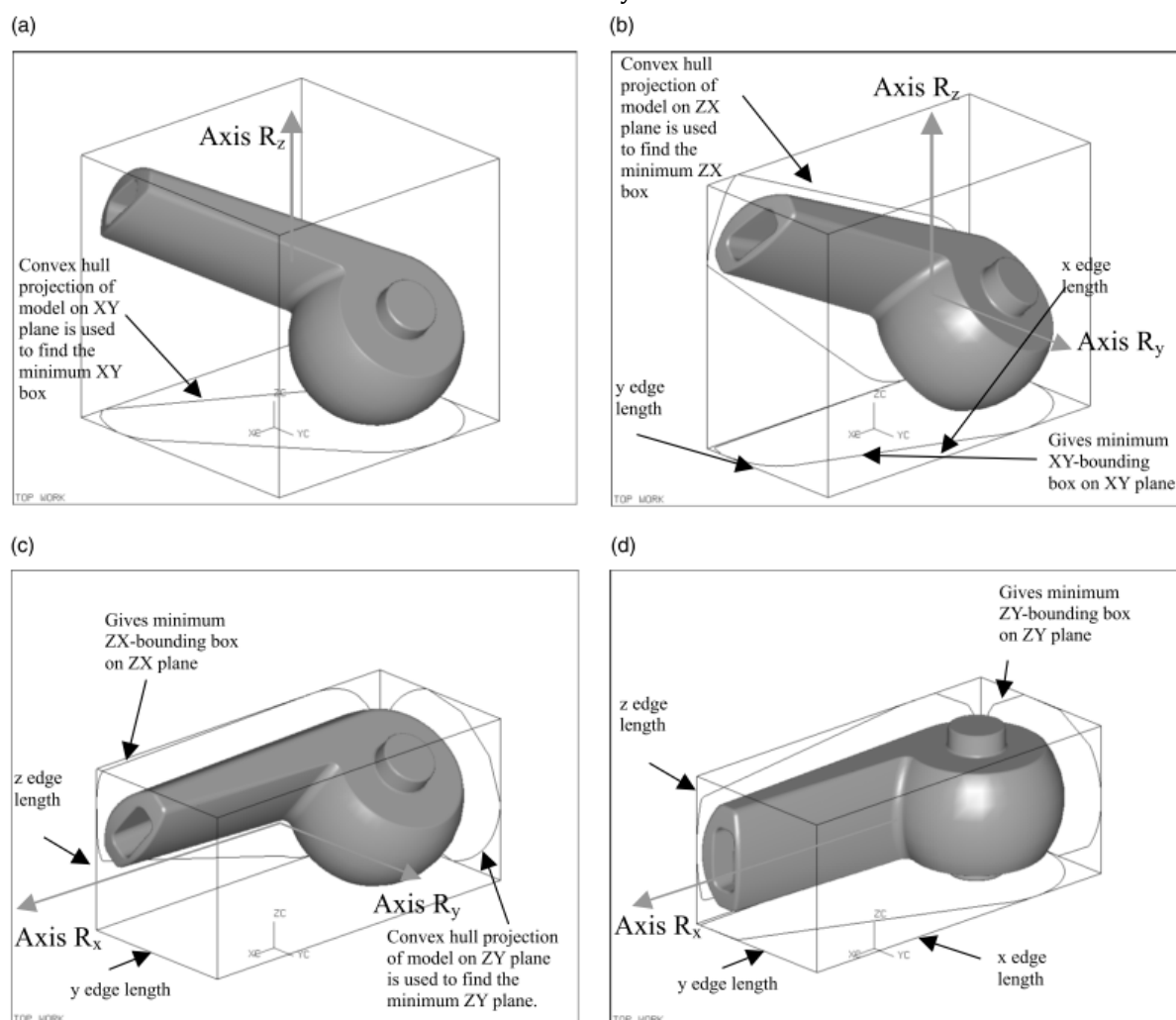


Figure 2.46. Procedures for finding out the minimum bounding box of a simple object (Chan and Tan, 2001).

A 64% reduction in volume and 46% reduction in surface area is achieved. This algorithm contains many rotations of an object. So the run time mostly depends on the selection of angle increment for the rotation of the object. During the iterative process, there may be

more than one angle of rotation that gives an absolute minimal projected area. This problem needs further analysis.

This iterative approach achieves efficient volume and surface reduction and can be efficiently adapted for IED systems. This can be achieved by rotating the form in $R_z \rightarrow R_y \rightarrow R_x$ sequence and increasing (scaling) the volume of the form instead of reducing the volume of the GG (bounding box). The form can be continually scaled until one of the x, y, or z length of the form is equal to one of the x, y or z length of the GG (bounding box) respectively.

2.7.4 Finding minimal enclosing boxes

Rourke (1985) has shown that a box of minimal volume circumscribing a convex polyhedron of n vertices must have at least two adjacent faces flush with edges of the polyhedron. The convex polyhedron is oriented one at a time with each face of the polyhedron lying on, for instance, the XY plane. The polyhedron is projected onto the XY plane. The AABB of the polyhedron and its volume are computed for each orientation. The smallest volume and its corresponding orientation are recorded. In Rourke's (1985) algorithms, the computational time is significant if the number of faces of the polyhedron is large.

Chan and Tan's (2001) algorithm is mainly suitable for complex objects where their minimum bounding box may not be obvious. For objects that contain no more than ten thousand vertices, either Chan and Tan's (2001) algorithm or Rourke's (1985) algorithm can be used if a precise OBB_{min_v} is not required. If the model is very complex and contains more than ten thousand vertices, either Barequet and Har-Peled (2001) algorithm or Chan and Tan's (2001) algorithm is suitable if computational time is important in the consideration.

In fact, all these algorithms can be combined into one method in order to allow the advantages of each of them to be used. For example, if a large complex object is given, the approximate orientation of the object, that its AABB is equal to the OBB, is first determined by either Barequet and Har-Peled's (2001) algorithm or Chan and Tan's (2001) algorithm. The result is recorded and then used for orientation filtering in Rourke's (1985) algorithm.

2.8 Shape comparison

Recent developments in 3D shape modelling have led to an increased accumulation of 3D models in databases and on the Internet and, with the rise in the number of 3D scanned objects, 3D object retrieval is becoming widespread in the fields of computer vision, computer graphics, computer aided design (CAD), molecular biology etc.

The success in elaborating a solution to content based retrieval of 3D models depends heavily on the existence of an efficient and intelligent comparison of 3D objects. Therefore, as a starting point, analysis of and experimentation with geometric similarity was researched and conducted. The secondary research is described in the following sections.

2.8.1 *Dimension Amnestic Pyramid Match Kernel*

Saupe and Vranic (2001) consider 3D object retrieval by using spherical harmonics and moments in which a polygonal mesh serves as a query and similar objects are retrieved from a collection of 3D objects. First, models are normalized into canonical coordinates (pose normalisation). The goal of this procedure is that if one chose a different scale, position, rotation, or orientation of a model, then the representation in canonical coordinates would still be the same. Second, feature vectors are extracted and compared. In the feature vector space, the nearest neighbours are computed and ranked. The features capture the 3D shape of the objects and they range from simple bounding box parameters (Paquet et al., 2000) to complex image-based representations (Heczko et al., 2002). The features are stored as vectors of fixed dimension. Third, a similarity search happens. The features are designed so that similar 3D objects are close in feature vector space. Using a suitable metric, nearest neighbours are computed and ranked. A various number of objects are thus retrieved by listing the top ranking items.

Wu et al. (2010) propose 3D shape comparison and partial matching, using a novel shape descriptor to capture local and global features of 3D objects based on high quality manifold harmonic bases constructed. Then they perform 3D retrieval to compute the similarity between two feature-sets using Dimension Amnestic Pyramid Match Kernel (DAPMK).

More specifically, to construct high-quality manifold-harmonic bases for complex 3D shapes, Wu et al. (2010) first propose a novel symmetric mean-value Laplace-Beltrami¹⁴ representation. Second, Wu et al. (2010) propose a novel global and local shape descriptor which is invariant to rotation, translation, scaling, and non-rigid bending as can be seen in Figure 2.47.

¹⁴ The Laplace operator can be generalised to operate on functions defined on surfaces in Euclidean space. The Laplace-Beltrami operator is a more general linear operator which is used to compare and analyse geometric shapes. It is invariant under isometries and well suited for analysis or retrieval of non-rigid objects.

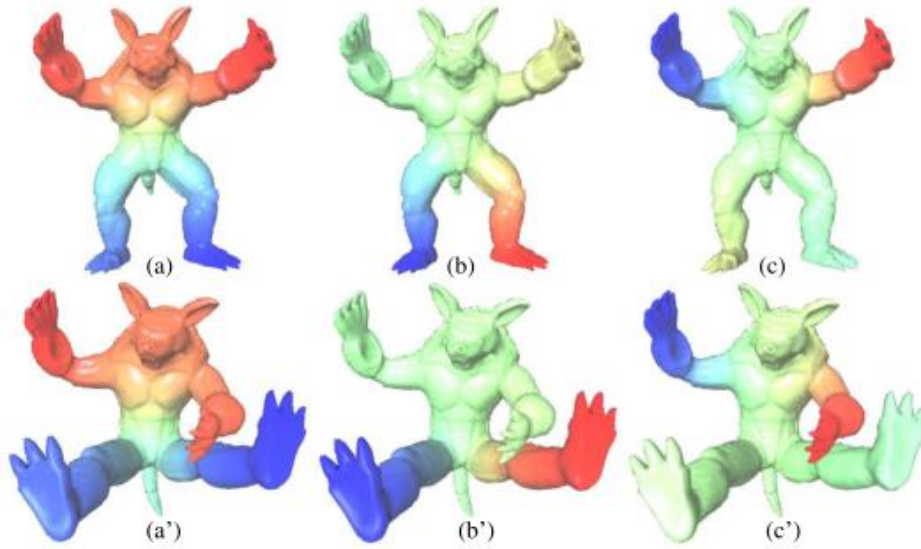


Figure 2.47. The Armadillo model and its 1st, 2nd, 3rd manifold harmonic bases (a, b, c). It can be seen that the bending-invariance is achieved (a', b', c') (Wu et al., 2010).

Third, by using the local and global features captured by the shape descriptor, Wu et al. (2010) perform a similarity comparison and partial matching (Figure 2.48) among 3D shapes by computing the pairwise similarities of 3D models using DAPMK. Pyramid Match Kernel (PMK) is an efficient approach to quantify the similarity between two unordered feature-sets. However, the approximation of PMK to the optimal feature matches deteriorates linearly with the dimension of local features, which prohibits the direct use of many dimensional features. DAPMK is a general, data-independent Kernel which can handle many dimensional features and gives an upper bound of approximation error independent of the dimension of local features (Liu et al., 2008). Wu et al. (2010) find matched regions on other models that are similar to the query region on a given 3D model. Wu et al. (2010) define a match with a similarity measure. A smaller distance means a more similar match, as can be seen in Figure 2.49 and Figure 2.50. To find the matches, Wu et al. (2010) use DAPMK as the distance/similarity function and look for the maximum similarity match.

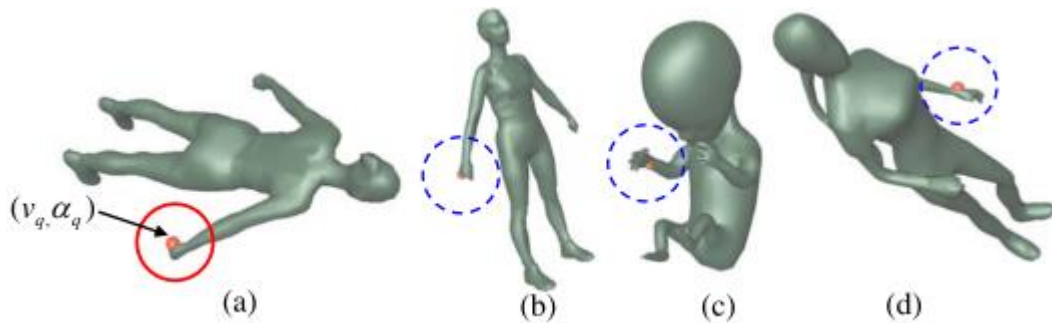


Figure 2.48. A partial matching example. Note that this partial algorithm does not consider the symmetric shape matching, therefore a query on the right hand (a) may return a partial matching result with maximum similarity on the left hand (d) (Wu et al., 2010).





Query # : 0	Match # : 1	Match # : 2	Match # : 3
Distance : 0.000	Distance : 0.038	Distance : 0.157	Distance : 0.679
			

Figure 2.49. A similarity comparison example. The distance value indicates the dissimilarity between the query model #0 and the match model. A smaller distance means a more similar match (Wu et al., 2010).













Match # : 0	Match # : 1	Match # : 2	Match # : 3	Match # : 4	Match # : 5
Distance : 0.000	Distance : 0.028	Distance : 0.185	Distance : 0.418	Distance : 0.626	Distance : 0.682
					
Match # : 6	Match # : 7	Match # : 8	Match # : 9	Match # : 10	Match # : 11
Distance : 0.750	Distance : 0.769	Distance : 0.802	Distance : 0.813	Distance : 0.856	Distance : 0.875
					

Figure 2.50. Another similarity comparison example (Wu et al., 2010).

It is difficult to handle a 3D mesh surfaces by using classical Fourier¹⁵ analysis because this method is not efficient for non-uniform sampling and complex topology. However, manifold harmonics can naturally generalise Fourier analysis or spherical harmonics on a regular 2D domain onto an arbitrary 3D mesh surface. Kernel's method is dimension-free, suitable for handling high-dimensional feature data that complex 3D objects have and can be effectively used among non-rigid 3D objects.

¹⁵ Fourier transform decomposes an image into its sine and cosine components. The input of the image is a spatial domain, while the outcome is a frequency domain. Each point represents a particular frequency contained in the spatial domain image. The Fourier transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression.

2.8.2 Geometric comparison

Novotni and Klein (2001) proposed a geometric approach for 3D object comparison. First, they solved the pose normalisation problem and then they compared the two objects geometrically. Novotni and Klein (2001) generate distance histograms which are basic tools for the determination of geometric similarity of stiff objects like bones, fruits etc.

For the object comparison sub problem, Novotni and Klein (2001) calculate a volumetric error between one object and a sequence of offset hulls of the other object and vice versa. This process gives them two histograms which are used for the similarity measurement.

Novotni and Klein's (2001) aim is to manage and give a quantitative measure on how much the two objects look like each other. When Novotni and Klein (2001) align two nearly identical datasets (as can be seen in Figure 2.51 on the right) in order for their surfaces to fit one onto another as accurately as possible, each point of the one object will either be enclosed by the other object or be in a small distance from the boundary of the other object and vice versa. On the other hand, when they align two different objects (as you can see in the Figure 2.51 on the left) some volume portions of each object will be in large distance from the surface outside of the other object.

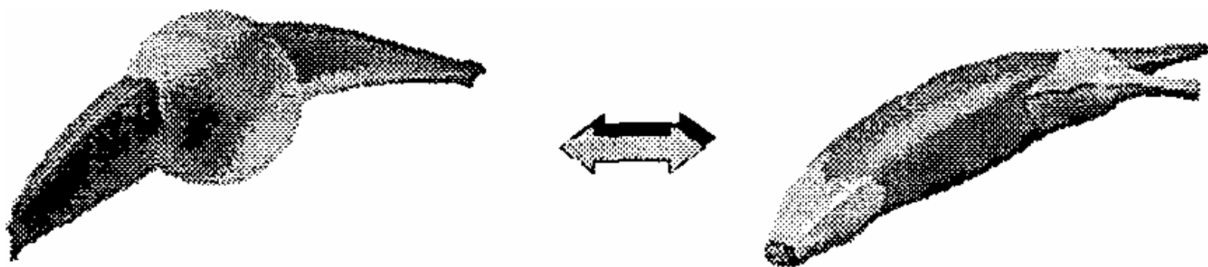


Figure 2.51. When the objects are similar, only a small fraction of their volumes are outside of the other objects boundary (Novotni and Klein, 2001).

In Novotni and Klein's (2001) algorithm, after the pose estimation, they compute how many portions of one object's volume are outside of the other object by using discrete 3D distance fields. For each discrete element of the volumes, they find its distance from the boundary of the other object and create a histogram that contains these values. Finally, based on these histograms Novotni and Klein (2001) compute a quantitative measure of similarity.

Pose estimation is a very important part of the object comparison process (Figure 2.52). In their method, Novotni and Klein (2001) use mass properties to align the objects. Novotni and Klein (2001) first compute the mass of the object, then the centre of mass (CoM) and translate the object so that the CoM is in the origin ($r=0$). After this, Novotni and Klein (2001) compute the principal axes which define an orthogonal coordinate system where the inertia tensor matrix I is diagonal. The principal axes are identified by the eigenvectors of the

inertia tensor. In order to get a correct alignment, Novotni and Klein (2001) match the eigenvectors corresponding to each principal axis of the two objects which eventually results in a rotational transformation. Finally, they scale the objects by normalising the volumes. However, the volumetric method cannot be adapted to the IED system proposed in this research – the Bounding Box (BB) method produces better fitness than the volume method according to the results presented in Section 4.2.2.

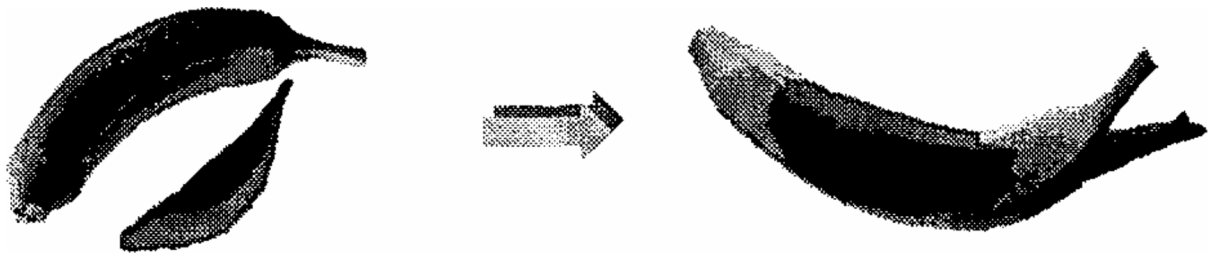


Figure 2.52. The bananas are markedly similar but they need to be aligned before the comparative test is applied (Novotni and Klein, 2001).

After pose estimation Novotni and Klein (2001) use discrete distance fields for object comparison. A distance field is a space where every point has a corresponding value showing its Euclidean distance to the closest feature of the objects existing in the space. They compute 2D distance fields for parallel slices of the examined object and then they convert this data into a 3D raster. They accomplish this by incorporating a new distance composed of the Euclidean distance in the 2D slices and the distance in z-direction. This distance is an estimation for the Euclidean 3D distance. The distance values of each slice are propagated to the next slice by adding or subtracting the distance between two slices which is supposed to be 1 pixel.

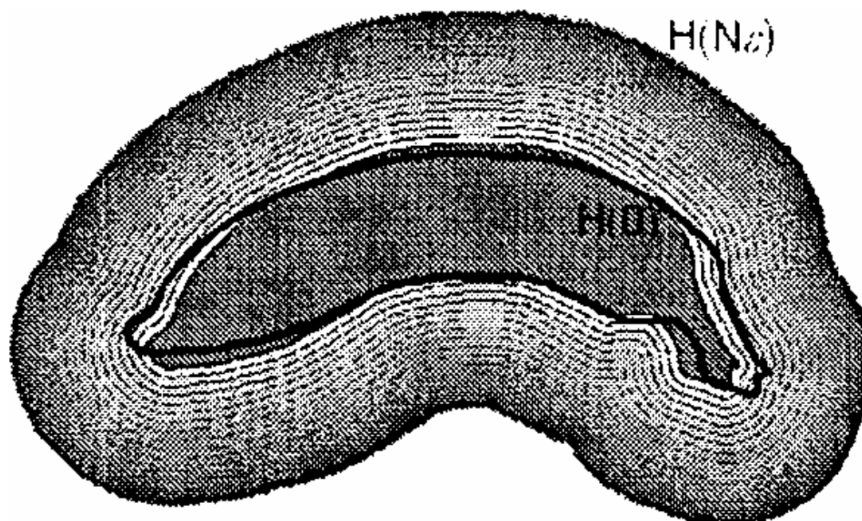


Figure 2.53. The volume of banana with black contour is checked against the offset hulls (Novotni and Klein, 2001).

Successful pose estimation indicates that the volumes of the two examined objects need to be intersected as much as possible. In order to compute the geometric similarity of two

objects, they compute offset hulls of each object based on the distance fields (Figure 2.53). They first determine a step size and consider for both objects the offset hulls. In order for the results of different objects to be comparable the step size has to be uniform. Finally, for each object they compute how much of the volume of the one object is inside the offset hull of the other object and vice versa.

To extract the desired similarity measure, Novotni and Klein (2001) accumulate the volumes remaining outside of each offset hull for both objects. This gives an efficient computable similarity measure which is useful for a wide range of applications.

The strength of this method is proven through results of a number of tests on different data sets. It is applicable for measurement of geometric similarity of 3D objects with stiff structure and their aim is to apply an extended version of the approach in articulated bodies as well. Another future plan is to include texture information and other features in the comparison method.

2.8.3 Comparison based on 3D morphing

Yu et al. (2003) proposed a 3D object retrieval method based on 3D morphing. For each 3D object, this method calculates two spatial feature maps (geometry and topology)¹⁶. The amount of effort required to morph a 3D object into a canonical sphere is calculated by these feature maps, without performing explicit 3D morphing. To achieve retrieval invariant to rotation, reflection, and non-uniform scaling of the objects, Fourier transforms of the feature maps are used for object comparison.

Instead of morphing one object to the other, Yu et al. (2003) morph both of them into a canonical object (sphere or cylinder), and compare the difference between the objects and the canonical object. First, the object is translated in a sphere of unit radius so that the centre of the sphere coincides with the centroid of the object. Second, the object is scaled so that the furthest 3D point on the object is 1 unit distance away from the centroid. Third, PCA is applied on the 3D points on the object to align the axes of the object.

After the alignment process, the distance map (DM) and the Surface Penetration Map (SM) are extracted from the object. A ray is shot from the centre of the sphere through each point of the object to the surface of the sphere. The distance from an object point to the surface of the sphere is recorded in the DM. The DM also captures the spatial locations of the points. It is a measure of the geometrical difference between the object and the sphere. The SM records the amount of the surfaces that the ray penetrates while it is travelling from the centre of the sphere through each point to the surface of the sphere. The amount of

¹⁶ Geometry determines the shapes of the objects and shape features such as size, curvature, and smoothness of object surfaces. Topology determines the structures of the objects such as the number of holes and disconnected components.

surfaces measured must be compressed into a single surface so that the object can be morphed into a canonical sphere (Figure 2.54).

In DM, the length $\|p_i\|$ of a point p_i on the object's surface is the distance of the centre of the sphere to the point. Therefore, length $d_i = 1 - \|p_i\|$ is the distance from the object point p_i to the surface of the sphere along the direction of p_i (Figure 2.54).

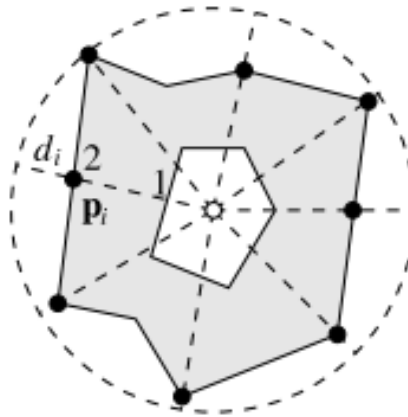


Figure 2.54. Rays (dashed lines) are shot from the centre (white dot) of a canonical sphere (dashed circle) through the object points (black dots) to the sphere's surface. The distance d_i travelled by the ray from a point p_i to the sphere's surface and the number of object surfaces (solid lines; 2, in this case) penetrated by the ray since it leaves the sphere's centre are recorded in the feature maps (Yu *et al.*, 2003).

In the SM, consider a point q on the surface of the object. If a ray is shot from the centre of the canonical sphere through q to the surface of the sphere, the ray may penetrate one or more surfaces depending on the topology and concavity of the object (Figure 2.55). Then, the SM (which is the mean number of the surfaces) will describe the topology and concavity of the entire object.

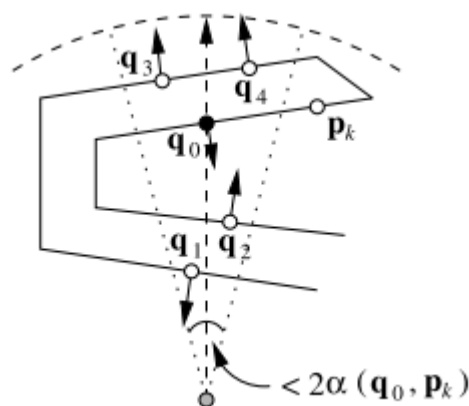


Figure 2.55. Counting number of object surfaces. A ray (dashed arrow) is shot from the centre (grey dot) of the sphere (dashed curve) through an object point (black dot) to obtain a cone (dotted lines) with a small angle. The other object points (white dots) within the cone are considered. Checking the number of times the surface normals (arrows) at the points change direction, compared to the ray's direction, in increasing distance of the points from the sphere's centre, gives the number of surfaces that the ray penetrates (Yu *et al.*, 2003).

To alleviate the problem of possible misalignment, Fast Fourier Transform (FFT) is performed on the feature maps for comparison and retrieval. As spatial information about the objects' features is preserved in the feature maps, FFT of these maps is very accurate and is invariant to rotation, reflection, and scaling. Experimental results show that their method of retrieving 3D models is very accurate but very computationally intensive.

2.9 Technology Review Conclusions

The literature review to date has covered many IEC and PN strategies that are applicable for this research, as well as shape comparison strategies that are required to compare evolving and guiding geometry once they have been aligned.

Involving users early in an EC process to evaluate and guide it has shown to be broadly beneficial. Interactive evolutionary optimisation is highly applicable to the IEC design system being developed. The most suitable interaction evolutionary optimisation strategies candidates for IEC design system research have been presented and compared. The Pareto-optimal points Interaction Strategies method is recommended. This method should get close to the most preferred point and effortlessly find the best solution in only a few interactions with the users and without fatigue. This method should be combined with a maximum number of times the user will interact, that has to be provided before the process starts, to define the accuracy of the optimisation (with higher values leading to better accuracy). Another method that is recommended is the CBR technique. This technique should reduce the human interaction gradually and decrease the human's effort. Since the type of IEC design envisaged generates complex 3D models, CBR should be flexible and robust as it is based on geometrical properties. However, this method concentrates on machine learning sub-system and on assimilating on-line the aesthetic preferences of the designer, which is a challenge that can be investigated as further work beyond the scope of this research project – It was decided that machine learning was out of scope of the research during an early review. Takagi's (2003) method of Euclidean distances in an EC searching space is also recommended. This method can be a useful addition as it improves the variety of solutions. However, it can only be considered as future work because it takes too long to integrate into the CAD system. Also, objects that mathematically are very similar could be subjectively quite different in appearance – a small difference to the geometry can make a big difference.

The main driver for PN research, in the wider research community, is the increasing need to search large databases of CAD geometry (e.g. parts libraries). Conveniently, it is highly applicable to the shape-comparison application in the IEC design system being developed. Here, PN is essential to align the simple pre-set guiding geometry with the evolving forms being generated by the combination of the GA and CAD modeller. Many PN methods have

been identified but the nine most relevant PN technology candidates to this research and suitable for IEC design system research have been presented and compared. In conclusion, since the IEC design systems of interest generate quite abstract and non-symmetrical 3D models with blended edges, the PCA-based PN method is recommended. This can be combined with a robust method similar to OCSVM that does not use the centre of the mass as the centre of the object but finds the best 'centre' of objects for translation and scaling, and with a method similar to Partial Symmetry-Based Alignment for achieving an efficient orientation. Therefore, readily-available deterministic geometric properties such as the centre of the Bounding Box for translation as well as Principle Axis (PA) for rotation and BB for scaling have been investigated. These PN methods are simple and easy to use, and the results offer appropriate efficacy for the application – and objective tests to confirm this have been completed and are presented later in the thesis.

However, before a comparison is made, which assesses the degree to which evolving shapes match the target shapes, the aim is to find the best fitting of the box-model within each object, to utilise geometric analysis to define the remaining volumes. To achieve this, the most suitable MVBB methods for IEC design system research have been presented and compared. Since the IEC design systems of interest generate quite abstract and non-symmetrical 3D models, Barequet and Har-Peled's (2001) approximation algorithm of a point set in three dimensions as well as Chan and Tan's (2001) technique of an iterative approach of an arbitrary solid are recommended. This can be combined with the PN methods recommended above and offer appropriate efficacy for the application.

The most suitable shape comparison technology candidates for IEC design system research have been presented and compared. The idea of the geometric approach (Novotni and Klein, 2001) that accumulates the volumes remaining outside of each offset hull for both objects and the approach based on 3D morphing (Yu et al., 2003) that compares the difference between the objects and the canonical object is efficient for this research. The algorithm used for this research is based on a simplified version of Novotni and Klein's (2001) geometric method – the principle of non-intersecting volumes. This algorithm can be implemented with standard CAD geometric functions, is simple, accessible to other researchers and offers appropriate efficacy for the application – and objective tests to confirm this have been completed.

2.9.1 Knowledge Gap Identification

In this research, the knowledge gap identification is the existing design approach to be further improved, and also employed within Engineering Design; by incorporating intelligent geometric analysis capabilities within Interactive Evolutionary Design. Thereby the knowledge gap identification is to create a co-operative and efficient human-machine system.

This has been achieved by enabling users to guide the evolution of 3D form using simple, pre-set fuzzy-constraints (i.e. GG), by establishing Human-Computer Interaction strategies that gain the most optimisation benefit from human partnership and by researching and developing 3D form-comparison algorithms and the associated orientation, alignment, and scaling of evolving shapes. GG for Interactive Evolutionary Design is boosting creative solutions and generating efficiencies in subsequent design steps.

There is still a need for CAD systems to better support ideation in conceptual design. There is a significant gap in the market for a tool that actually generates forms that are not based on predefined designs created by skilled designers, and better supports form development. To exploit this gap, CAD systems should provide a source of inspiration through a system that evolves forms.

Therefore, simple GG have been combined for the first time with optimised interactive strategies and evolutionary computation in the CAD domain, and applied in the field of form ideation of product design.

Since users of EvoShape within the design industry have expressed a desire to have more control over the process beyond interacting with the system during the evolutionary process¹⁷, the software allows them to create GG at the start of the process to guide the evolution of 3D form.

¹⁷ Purely through rating and selecting forms.

3 Methodology

The research methodology that is used to develop the research solutions is a Software application for objective and subjective testing. Objective testing is used to test geometry mathematics and the efficacy of evolutionary components, whereas subjective testing is user testing of different interaction strategies.

3.1 Programming Environment

NX, formerly known as Unigraphics or UG, is an advanced high-end CAD/CAM/CAE software package developed by Siemens PLM Software. It is used for design, engineering analysis, manufacturing design and throughout the product development process.

NX Open is a collection of API toolkits that allows developers to create custom software applications for NX through an open architecture. The NX Open APIs provide the applications and tools that allow customers to access the NX Object Model and create and edit objects.

Open C API is designed to support an easy interface between NX and the outside world. The Open C API consists of a large set of user callable and readily-available CAD functions. Open C API programs can run in two different environments, depending on how the program was linked. The two environments are: External and Internal.

In this research, an add-in (Internal Open C API program) for the Siemens NX11 CAD system was developed and integrated with EvoShape, and run from inside NX. Internal Open C API programs can only be run from inside of the NX session. The objectives 1-10, which are introduced in Section 1.4, have been achieved through objective and subjective testing.

3.2 Objective testing

In the objective testing, the objectives 1-8 have been achieved to:

- a. Test Geometry mathematics
- b. Test the efficacy of evolutionary components

Objective 1: Develop a representation for the simple GG

Soft geometric constraints (in the form of simple bounding geometry) have been created at the start of the evolution process to investigate the alignment and shape comparison methods.

Objective 2: Develop stand-alone programs to investigate the individual operations

Some stand-alone programs were written to investigate the individual operations:

- ❖ rotation
- ❖ scaling
- ❖ translation
- ❖ shape comparison

Objective 3: Find efficient alignment methods

A program was developed in which alignment (PN) between evolving shapes and GG was achieved and geometry mathematics was tested. The successful 3D alignment methods formed a shortlist of methods which could be tested mathematically for accuracy.

Objective 4: Find efficient shape comparison methods

A program was developed in which shape-comparison between evolving shapes and GG was achieved and geometry mathematics was tested. The successful shape comparison methods formed a shortlist of methods which could be tested mathematically for accuracy.

Objective 5: Integrate the most efficient operations with EvoShape Interactive Evolutionary Design (IED) system

The successful 3D alignment and shape comparison methods were chosen to integrate with the GA.

Objective 6: Turn the add-in program into an automated shape matching program

This add-in program (given in Appendix A), B), C), D), E), F), G)), was turned into an automated shape matching programme – the human interaction was removed and consistent experimental environment was created – in order to objectively test and assess the efficacy of the evolutionary components and the approaches being developed.

Objective 7: Test alignment and shape comparison methods efficiency.

The shortlist of methods consists of two approaches for each stage of PN which were assessed and compared; Principal Axes (PA) or manual alignment for rotation, Volume or Minimised Bounding Box (MBB) for scaling and, Centre of BB or Centre of Mass for translation (see Chapter 4). The most efficient approach for each stage of PN was investigated by combining rotation, scaling and translation and the most efficient combination was established through fitness values. The shape comparison algorithm is based on the principle of non-intersecting volumes, is implemented with standard geometric functions, and is simple and accessible to other researchers.

32 iterations (as described in Chapter 4), with different starting populations were run. A code which generates text files was written. The information from the text files was imported to MS Excel, and graphs were created and analysed.

Objective 8: Develop effective human-system interaction strategies

The efficacy of the PN and shape comparison methods was tested with two broad approaches of user interaction strategies to investigate how best to use the GG capabilities to assist users in evolutionary shape searching (Chapter 5): Large Background Populations (LBP), or Sporadic User Interaction (SUI). For the SUI approach, four different starting populations and four different GG were used for each of the two approaches of each stage of PN, resulting in 32 separate shape matching tests of 240 forms in each test. For the LBP approach, four different starting populations and four different GG were used, resulting in 16 separate shape matching tests of 240-480 forms in each test.

Objective and fitness value graphs based on the shape comparison were produced on MS Excel. A combination of objective tests (Chapters 4 and 5) and subjective user-testing (Chapter 6) were completed. Both the SUI and the LBP approaches need further consideration with user interaction (Chapter 6). The objective testing diagram is shown in Figure 3.1.

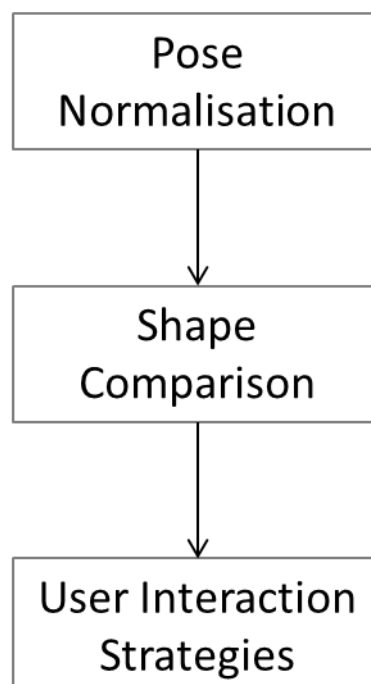


Figure 3.1. Objective testing diagram.

3.3 Subjective testing

In the subjective testing, the objectives 9-10 have been achieved to:

- a. Test to the users interaction strategies by using:
 1. Task observation and statistical analysis (laboratory test).
 2. Interview.

Task observation and statistical analysis was selected for this research as it helps target audiences (students and professional designers) to better understand the evolution process and interact with it. An interview methodology was also selected for this research as it allows developers to ask in-depth open ended questions.

3.3.1 Number of Test Participants

Testing a small number of participants and running many small tests is usually enough to obtain a satisfactory level of understanding (Nielsen, 2012). The data from a single test participant can give developers almost a third of all there is to know about the development of the design.

Whether developers test websites, intranets, PC applications, or mobile apps, with five participants they almost always get close to participant testing's maximum benefit-cost ratio (Nielsen, 2012). When developers test the second participant, they will find out that this person does some of the same things as the first participant. By adding more and more participants, developers learn less and less because they will keep seeing the same things again and again. After the fifth participant, the developers will not learn much that is new (Figure 3.2).

Contrarily, Perfetti and Landesman (2001) after testing five participants, had found only 35% of all development issues. After 18 participants, they were still discovering serious issues and had uncovered less than half of the 600 estimated website or software development problems. Spool and Schroeder (2001) also testified a large-scale website evaluation for which five participants were nowhere near learning 85% of the problems.

The variation is because of the large and complex systems where participants have a larger number of problems available for discovery along the way, so any given problem is less likely to be found.

In this research, the participants are not experts or familiar with the domain. However, the efficacy of the system is not hard to evaluate, the system is not complex and the participants can test the system with small multiple iterations. Therefore, in this research, acceptable results come from testing around five participants and running as many small tests as possible.

Objective 9: Add human intervention to the system

In the subjective testing, human intervention was added in order to subjectively test and assess the efficacy of the interactive evolutionary strategies and the approaches being developed. More specifically, the automated shape matching programme was:

- ❖ Either replaced by user interaction at intervals of between 4 and 11 generations (SUI approach; or
- ❖ Added in every generation and repurposed to select a subset of shapes to present to the user (LBP approach).

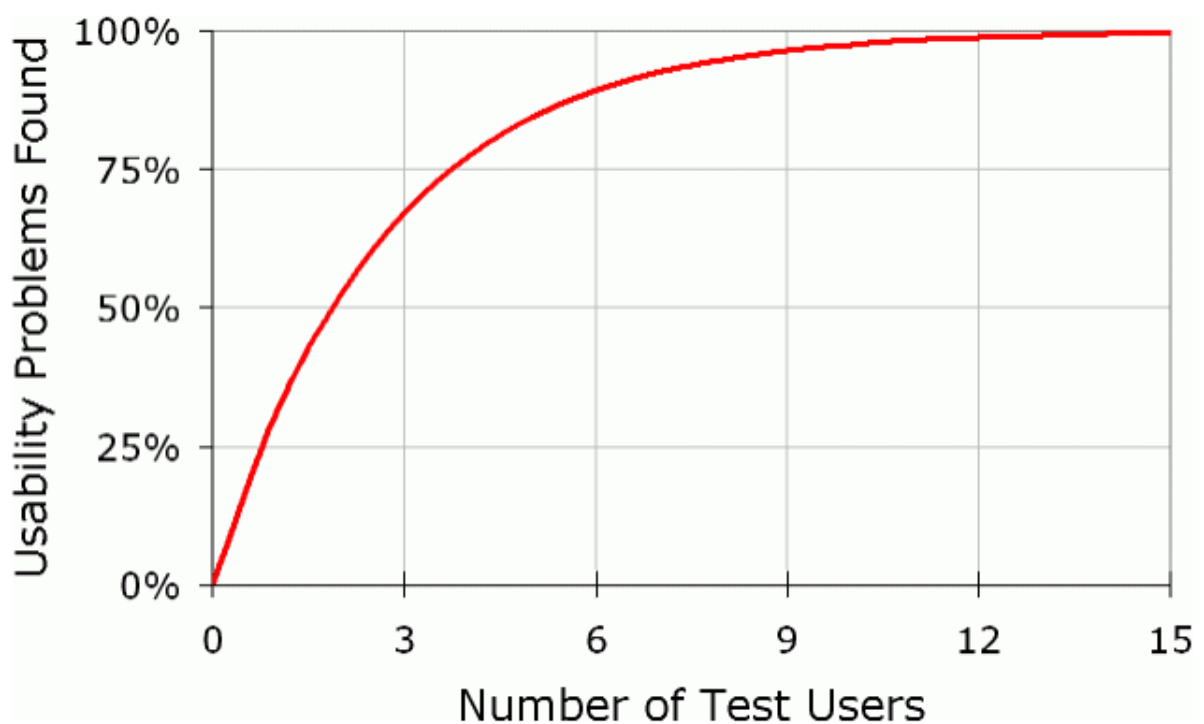


Figure 3.2. Five users would be expected to find 85% of the development problems (Nielsen, 2012).

Objective 10: Test the human-system interaction strategies with users

This was achieved by using task observation (laboratory test) to test the interaction strategies by allowing the users (participants) to guide the evolution of the 3D form by simply creating guiding geometry (GG) at the start of the evolution process (see Chapter 6). This allows participants to have more control during the process.

The participants were selected by contacting people that either work (collaborators or friends) or study (students of the author's PhD supervisor) at Loughborough University. Some users were students in their third year of UG studies in an Engineering Department or Design School, familiar with CAD systems and experienced in design but with little NX CAD system experience; therefore they created their GG with the support of the developer.

A combination of the above two user testing methods (task observation and statistical analysis and interview) was used for this research to develop the research solutions. Three different versions of the system were tested by the users: EvoShape, LBP and SUI. The order can be an important step in the process as the users' fatigue is increasing during the testing, which can affect the results. Some of them were asked to test EvoShape first and then LBP, whereas others were asked to do the opposite. The SUI approach was tested by the users separately on a different day. The users did not work for more than an hour at a time because the fatigue can be increased again and affect the results.

Since the efficacy of the system in this research is not hard to evaluate, five participants have been selected for the first testing process, to test EvoShape and LBP. After they compared the EvoShape and LBP versions, only two (one designer and one non designer that were easily available) participants of these five participants were needed for the second testing process – as they only needed to test SUI and compare it with the two previous methods that were already tested by them.

The ethics procedure that was followed with the participants was: all participants were ensured that their confidentiality and anonymity will be respected, and that their participation in this research is voluntarily – the participants could leave the testing process any time and for any reason.

When interviewed, all users stated a strong preference for versions that included GGs. Observations backed up these subjective findings in several ways, e.g. the time and number of iterations it took to arrive at shapes with which users were satisfied, and the proportion of useful shapes generated in a given time.

Statistical analysis was used to examine how many users managed to complete the task to the end of the evolution process and compare the different users' outputs according to their preferences and their interaction with the process (see Chapter 6).

3.4 Summary

The methodology described in this chapter has established the optimum combination of an automatic and human system, namely EO and IEC. In this research the strategy to investigate this gap and integrate a GA-driven generative design process with human interaction is:

- ❖ To enable users to guide the evolution of 3D form not only by selection but also by creating a simple GG at the start of the process.
- ❖ To develop 3D alignment and comparison algorithms between the GG and the forms.
- ❖ To establish Human-Computer Interaction strategies that best exploit these algorithms – LBP and SUI.

4 3D Alignment and Shape Comparison Analysis

This chapter analyses shape matching by using GGs. 3D alignment is introduced first, which is required before shape comparison, to find the best position of each form to be compared to the GG. Alignment (rotation, scaling, and translation) is a process known as PN in the literature. Once form and GG are aligned, shape comparison follows to calculate how closely the forms and GG match.

Both PN and shape comparison methods have been implemented using geometric functions available within a CAD system via its API, enabling populations of evolving forms to be directed by the GG.

An add-in for the Siemens NX11 CAD system has been developed and integrated with the EvoShape system which has been turned into an automated shape matching programme using GG in order to assess the efficacy of the approaches being developed. Two approaches for each stage of PN have been assessed and compared:

- PA and manual alignment for rotation;
- BB method and volumetric approach for scaling;
- Centre of BB and CoM for translation.

In this chapter, the algorithms and their implementation are presented alongside results and analysis, discussion on limitations and robustness, and their suitability for Interactive Evolutionary CAD.

4.1 Shape Matching Introduction

4.1.1 Introduction to Geometric Guides

The inclusion of GG is an idea introduced by Graham's (2002) thesis that would take the research from the existing aesthetic focused tool towards a system capable of application to a wider range of engineering design problems. In addition, current users of EvoShape within the design industry have expressed a desire to have more control over the process; since it is rare that a designer will have complete freedom over form in a commercial design situation, some have expressed a desire to influence the shapes' overall proportions. The GG is thus created by the user at the beginning of the session. All the GG that have been used during the shape matching research are shown in Figure 4.1: Broken Ice Cream, Traffic Cone, Roof, and Cylinder.

Cylinder is a simple GG represented with single geometric primitive. Both Traffic Cone and Broken Ice Cream are represented with two geometric primitives. Traffic cone is represented with a cone united to a block, and broken ice cream with a sphere subtracted from a cone. However, they can both be approximated with a single geometric primitive. Roof GG is also represented with two geometric primitives (two blocks) but it cannot be approximated with a single geometric primitive and is therefore more complex.

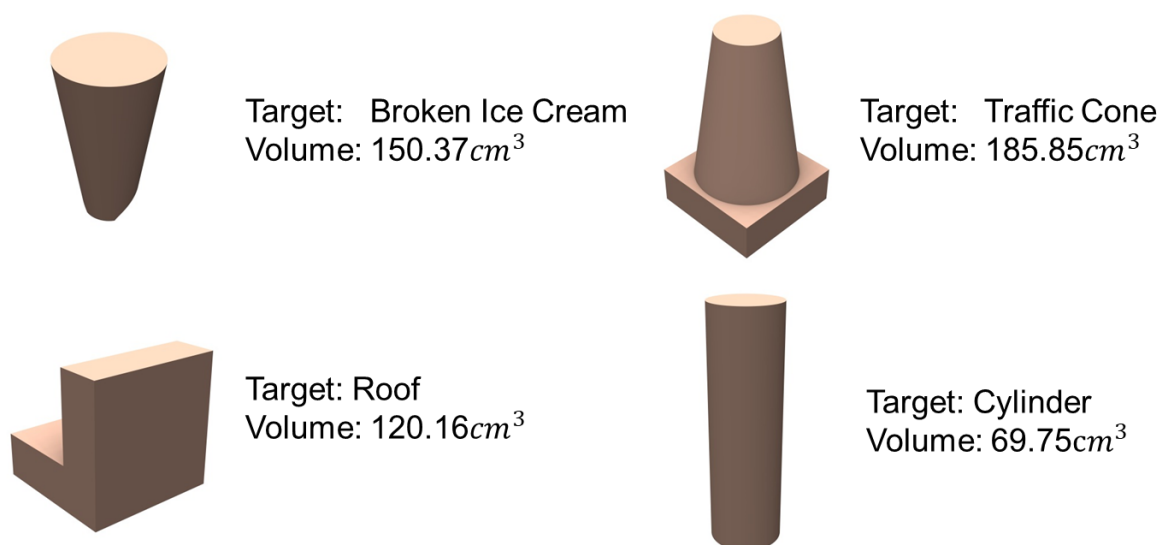


Figure 4.1. GG (targets) that have been used during shape matching development, and their volumes.

4.1.2 Introduction to Pose Normalisation

Before shape comparison can take place, and due to arbitrary 3D orientation, scale, and position, a 3D alignment that finds the best position of each form compared to the GG is required. PN is thus essential to align the GG with the evolving forms being generated by the combination of the GA and CAD modeller.

PN has been achieved using readily-available deterministic geometric properties such as CoM, PA and BB. Rotation, scaling and translation are carried out in that order, and presented in the following sub-sections. The shape comparison technique used to evaluate different PN approaches is based on the principle of sum non-intersected volume, introduced in the next section.

Rotation

In an earlier version of the system, produced for maximum speed and ease of implementation, the GG and all forms' primary PAs were aligned with the Z axis. This had two major advantages: only the major PA needed to be calculated; and the target's BB computed by the CAD system tended to be closer to its minimum, irrespective of the orientation of the user-generated guide geometry. However, one disadvantage was the effect of 'standing all the shapes up' (as seen in Figure 4.2). Early user-testing revealed that users would prefer the original orientation of the GG to be maintained (some products naturally lie down, furniture for instance, and look odd aligned otherwise).

So in the final version, all three principal axes of forms are identified (X, Y and Z), and aligned with the three principal axes of the GG. The centre of the BB of each form is used as the centre of rotation.

Scaling

Firstly, the X, Y and Z lengths of each GG and form's BB are established. Then the ratios (X Y Z lengths of the GG's BB divided by X Y Z lengths of the form's BB) are calculated, and the form is scaled uniformly according to the smallest ratio of the three.

Translation

The centre of the base of each form's BB is used as its local origin within each cell of a 4 x 3 grid, and the centre of the BB of each form and GG is used to co-locate the GG and forms before shape comparison.

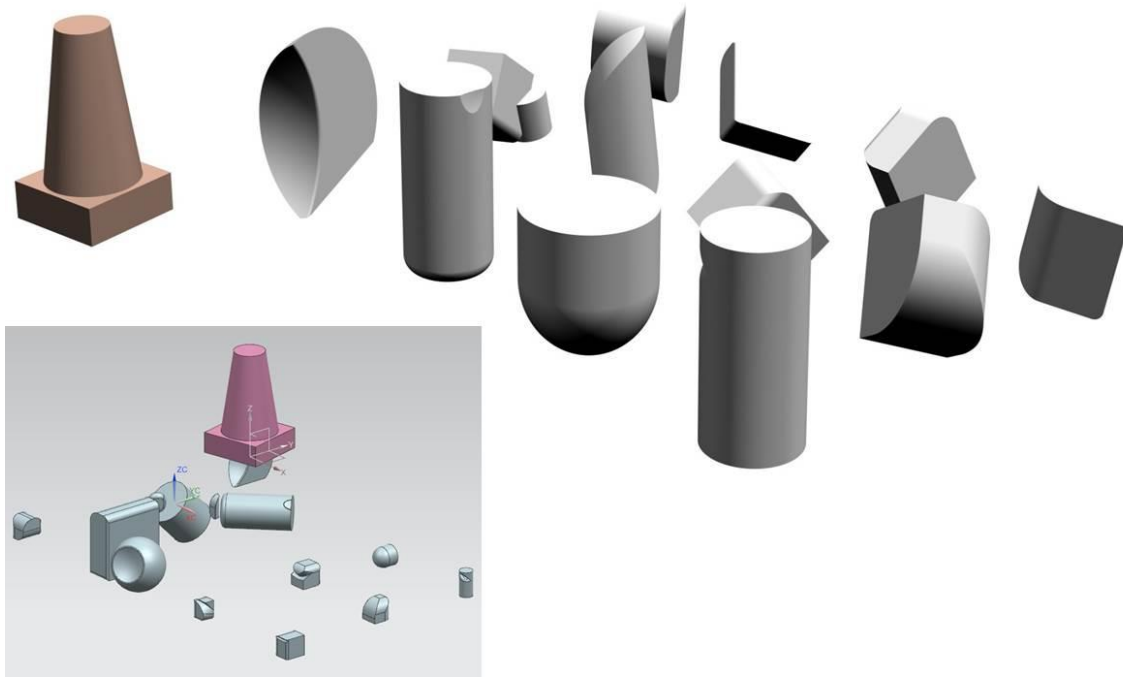


Figure 4.2. A population of 12 grey forms and brown GG after PN, compared to the same population beforehand (left).

4.1.3 Introduction to Shape Comparison

Once form and GG are aligned, Boolean operators are used to calculate how closely they match. In this research, the approach is based on the principle of non-intersecting volumes, i.e. any material volume outside the intersection of the two shapes (form and GG) counts as a penalty. This can arise from GG material not within the shapes' intersection, and material from the form not within the shapes' intersection. The sum of these two volumes acts as an objective function representing how closely the two shapes match, shown graphically in Figure 4.3 and mathematically in Equations 4.1 and 4.2. In the test application the sum non-intersecting volumes tend to zero as the evolving forms more closely match the GG.

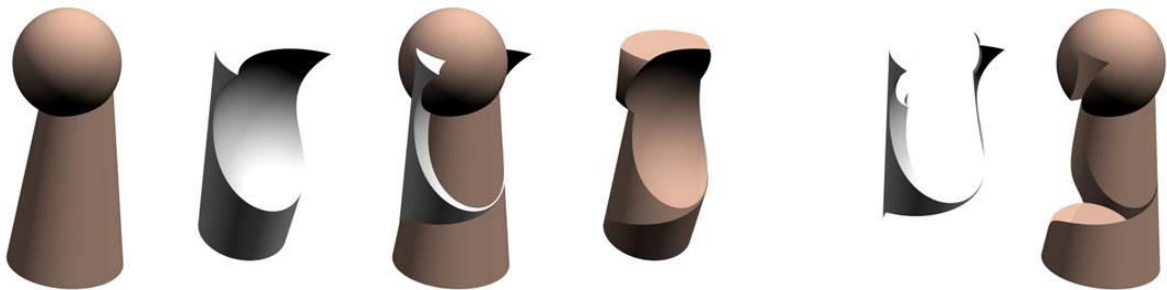


Figure 4.3. From left to right: GG; form; after PN; intersection; non-intersecting volumes.

$$S_f = (V_{GG} - (V_{GG} \cap \text{form})) + (V_{\text{form}} - (V_{GG} \cap \text{form}))$$

4.1

Simplified to:

$$S_f = V_{GG} + V_{form} - 2(V_{GG \cap form})$$

4.2

Where:

- ❖ S_f is the objective function of an evolving form.
- ❖ V_{GG} and V_{form} are the volumes of GG and form respectively.
- ❖ $V_{GG \cap form}$ is the intersecting volume of GG and form.

A form's 'match' with the GG is used throughout this thesis as a normalised and more intuitive expression than fitness function, shown mathematically below in Equation 4.3.

$$\% \text{ 'match' with the target} = 1 - \frac{S_f}{V_{GG}}$$

4.3

4.2 Detailed Analysis

An IEC design system's ultimate aim is not to find a known target shape, but to use a pre-defined shape to *guide* evolving geometry for the user. However, the EvoShape system has been turned into an automated shape-matching programme for this stage of the research using GG to objectively assess the efficacy of the PN and shape comparison.

With the exception of the rotation research, four different starting populations and four different GG were used for each experimental analysis, resulting in 16 separate shape matching runs of 20 iterations (each population has 12 forms, resulting in 240 individual shape-matching operations in total for each run). This combination of 4 x 4 x 20 x 12 is henceforth referred to as a dataset.

4.2.1 Rotation

To test the accuracy of the PA method within the application, a random population of 12 shapes were tested with four different GG ('traffic cone', 'broken ice cream', 'roof' and 'cylinder'), making 48 individual comparisons in total. Most of these comparisons (31) resulted in a difference between rotational alignment using PA and manual alignment. Using the metric '% volume match', a sample of eight comparisons of the PA method with manual rotational alignment have been calculated; each with two different forms, as shown in Figure 4.4.

Where human and PN alignment differs, manual alignment is generally better. According to the experimental results shown in Table 4.1, the average match with the GG when shapes are rotated intuitively is better by 10% than the average match with the GG when the PA method is applied, as calculated by Equation 4.4.

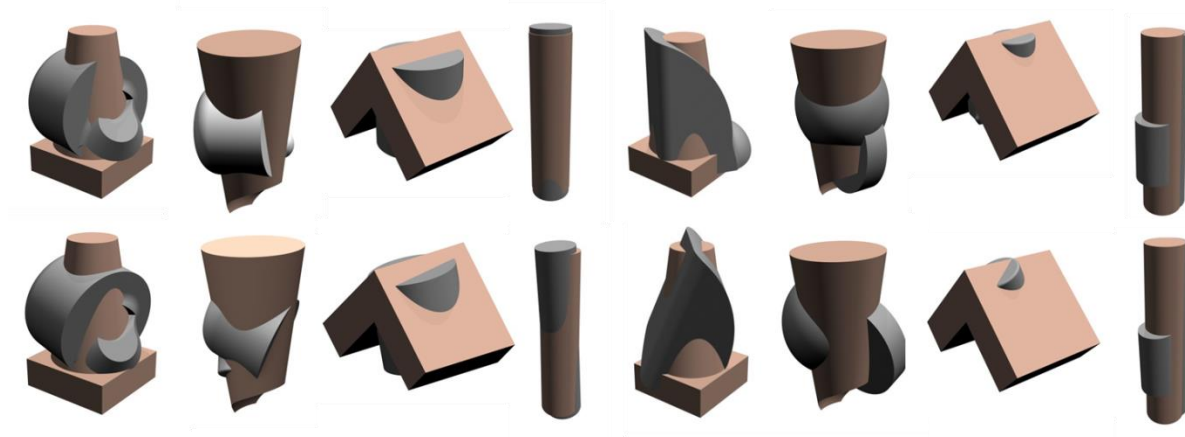


Figure 4.4. Top row – Alignment by human. Bottom row – Alignment using PA.

Table 4.1. Forms' average % volume match with GG where human and PN alignment differ. The average match of the forms with the GG when the shapes are rotated intuitively is 30.58%. This percentage decreases to 27.72% when the PA method is applied.

	Traffic Cone	Broken Ice Cream	Roof	Cylinder	Traffic Cone	Broken Ice Cream	Roof	Cylinder	Average
PA	8.82%	20.21%	10.44%	77.90%	35.12%	8.56%	4.22%	56.48%	27.72%
Human	9.07%	26.87%	10.33%	88.67%	21.80%	29.88%	3.38%	54.67%	30.58%

$$\frac{\text{Average Human match} - \text{Average PA match}}{\text{Average PA match}} = \frac{30.58\% - 27.72\%}{27.72\%} \approx 10.3\%$$

4.4

However, some of the generated shapes in the test had exactly the same alignment with the GG whether the PA method was applied or whether the rotation was made intuitively by a human. For example, in two random generations of shapes, in the case of broken ice cream, three alignments out of twelve were the same. In the case of traffic cone and roof, four out of twelve alignments were the same. And finally, in the case of cylinder, six out of twelve alignments were the same. These results should contribute to the overall assessment of performance deficit when comparing PA alignment to intuitive alignment. So applying these ratios to the 10.3% figure above gives 7.7% for broken ice cream, 6.9% for traffic cone and roof, and 5.2% for cylinder. Combining these gives an indicative overall performance deficit of 6.7%, meaning PA alignment is only around 7% worse than human alignment (Equation 4.5).

$$\frac{\text{Shapes with different alignment}}{\text{Total number of shapes}} = \frac{31}{48} = 64.6\% \rightarrow 64.6\% \times 10.3\% = 6.7\%$$

4.5

Using PAs to align GG and form is computationally efficient – essential for a real-time interactive application. Functionally, however, is not perfect, and can produce counter-intuitive results, as shown in Figure 4.5. These deviations have been shown to have only a minor impact within the context of the guiding application.

The centre of the BB of each form is used as the centre of rotation – this works better than using the CoM, resulting in less of a jump (translational movement). Functionally, this is unimportant as the subsequent translation step accounts for any undue movement of the shape, but it produces a more pleasing result *visually*, if users have opted to watch the analysis on screen.

In conclusion, the PA method, while not perfect, and not as effective as more sophisticated techniques reviewed in the literature, is good enough to be used for *guiding* applications, such as in IEC design systems. This is confirmed by the results of full system validation outlined in Chapter 5.

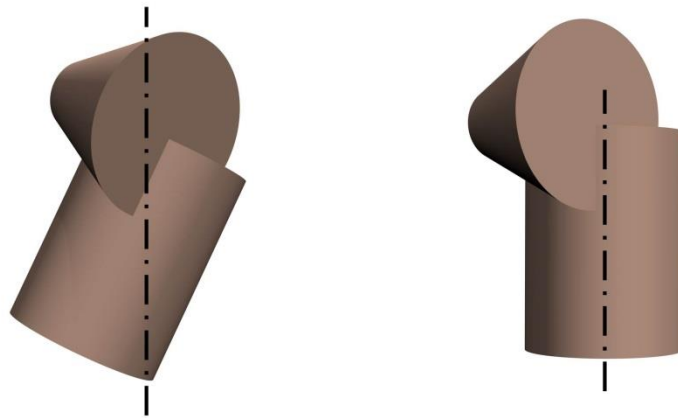


Figure 4.5. A shape's PA aligned with the Z-axis (left) compared to a more intuitive alignment of the shape (right), where the PA of the dominant geometry, the cylinder, is used.

4.2.2 Scaling

The strength of the BB method over a volumetric approach is proven through testing against four different GG (broken ice cream, traffic cone, roof and cylinder) first by scaling them using the BB method and then by scaling them using volume ratio. Results have been generated by averaging a dataset.

The BB method produces a better fit than the volume method (Figure 4.6). Fitness function is inversely proportional to objective function, as shown in Equation 4.6:

$$f = \frac{1}{1 + S_f} \quad 4.6$$

Where:

❖ f is the fitness function of an evolving form.

Objective function is the sum of any material volume outside the intersection of the two shapes (see Section 4.1.3). A form's % match with the GG is used as a more intuitive expression than fitness functions, as shown (full expanded) mathematically in Equation 4.7.

$$\% \text{ match with the target} = 100\% - \frac{(V_{GG} + V_{form} - 2(V_{GG} \cap form))}{V_{GG}} \quad 4.7$$

Simplified to Equation 4.8:

$$\% \text{ match with the target} = \frac{(V_{form} - 2(V_{GG} \cap form))}{V_{GG}} \quad 4.8$$

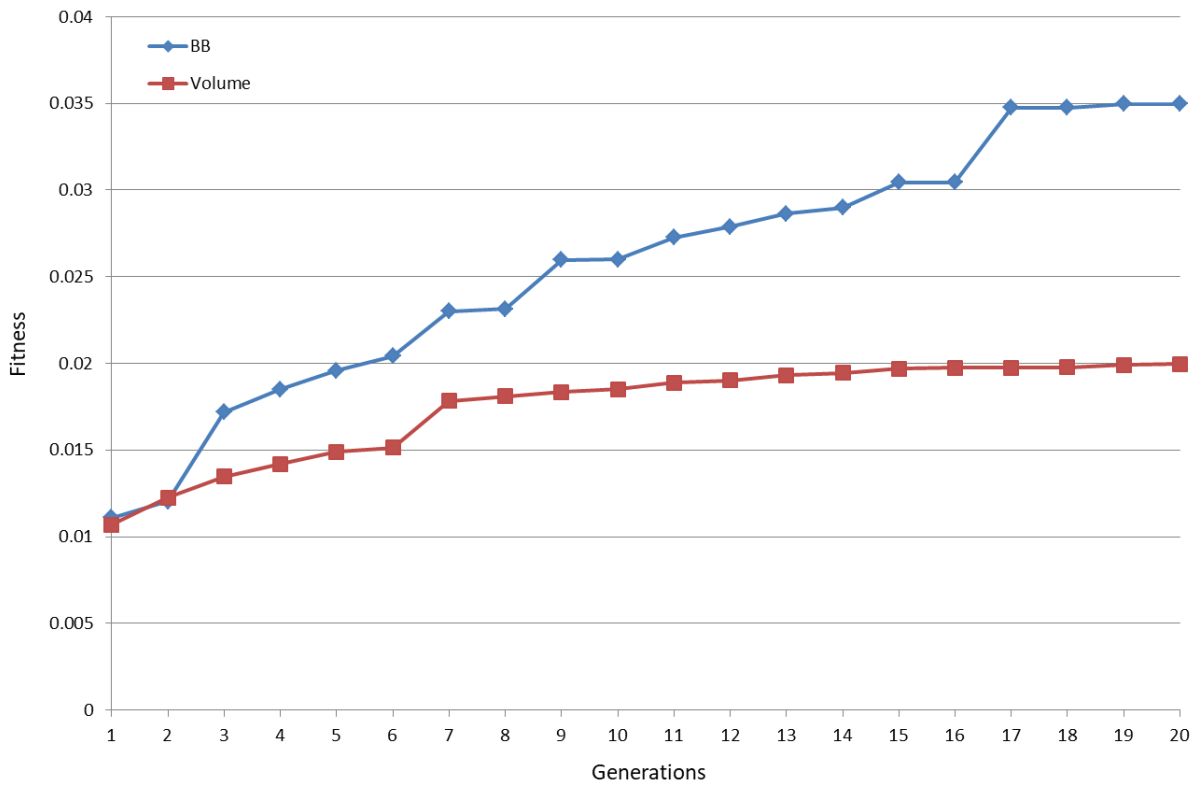


Figure 4.6. A comparison of the BB scaling method with the volume scaling method.

Scaling the shapes by using the BB method has the advantage of keeping the BB of the form within the BB of the GG and the advantage that it is intuitive for users. The BB method produces reasonable scaling estimations, preferred to the results of matching volumes (seen

in Figure 4.7), and achieves robustness with typical geometry of the Interactive Evolutionary CAD system described in this chapter.

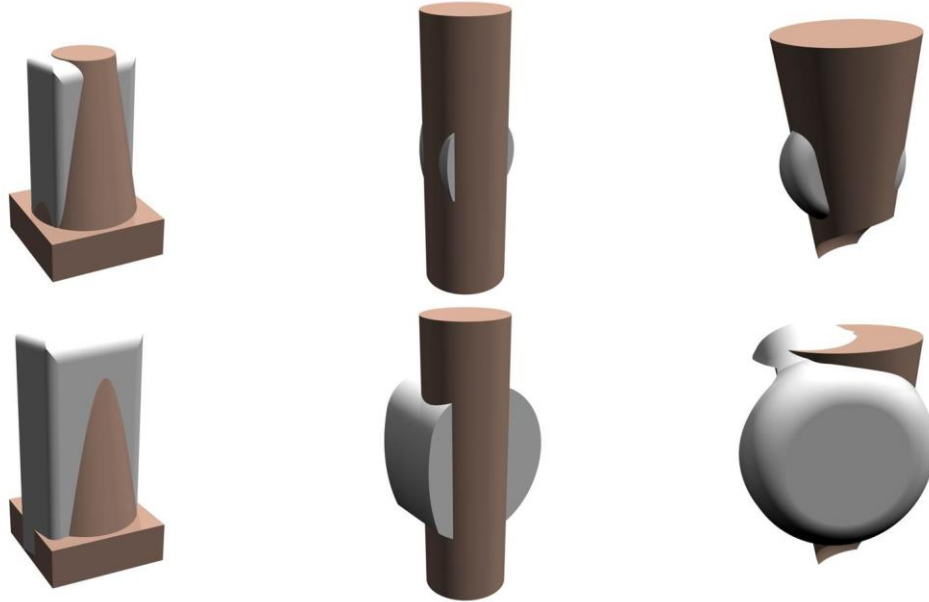


Figure 4.7. Scaling using BBs (upper): the BBs of the grey forms do not exceed the BB of the brown GG. Scaling by volume (lower) is less satisfactory, and the form BBs generally exceeds that of the GG.

4.2.3 Translation

To co-locate the GG before shape comparison, the centre of the GG's BB is co-located with that of the form.

The strength of the 'Centre of BB' method over the CoM method is proven through averaging the results of a dataset. Given that each population has 12 shapes, each data point in Figure 4.8 represents $16 \times 12 = 192$ shape comparisons. The Centre of BB method is clearly better than the CoM method, despite the unfair starting disadvantage caused by randomness in the starting populations.

Translating the forms within each cell of a 4×3 grid by using the centre of the base of each form's BB as its local origin works well visually, producing an even spacing between forms (Figure 4.9).

The centre of BB method also produces the better translation estimations, preferable to the results of using CoM, and achieves robustness with typical geometry of the Interactive Evolutionary CAD system described in this research.

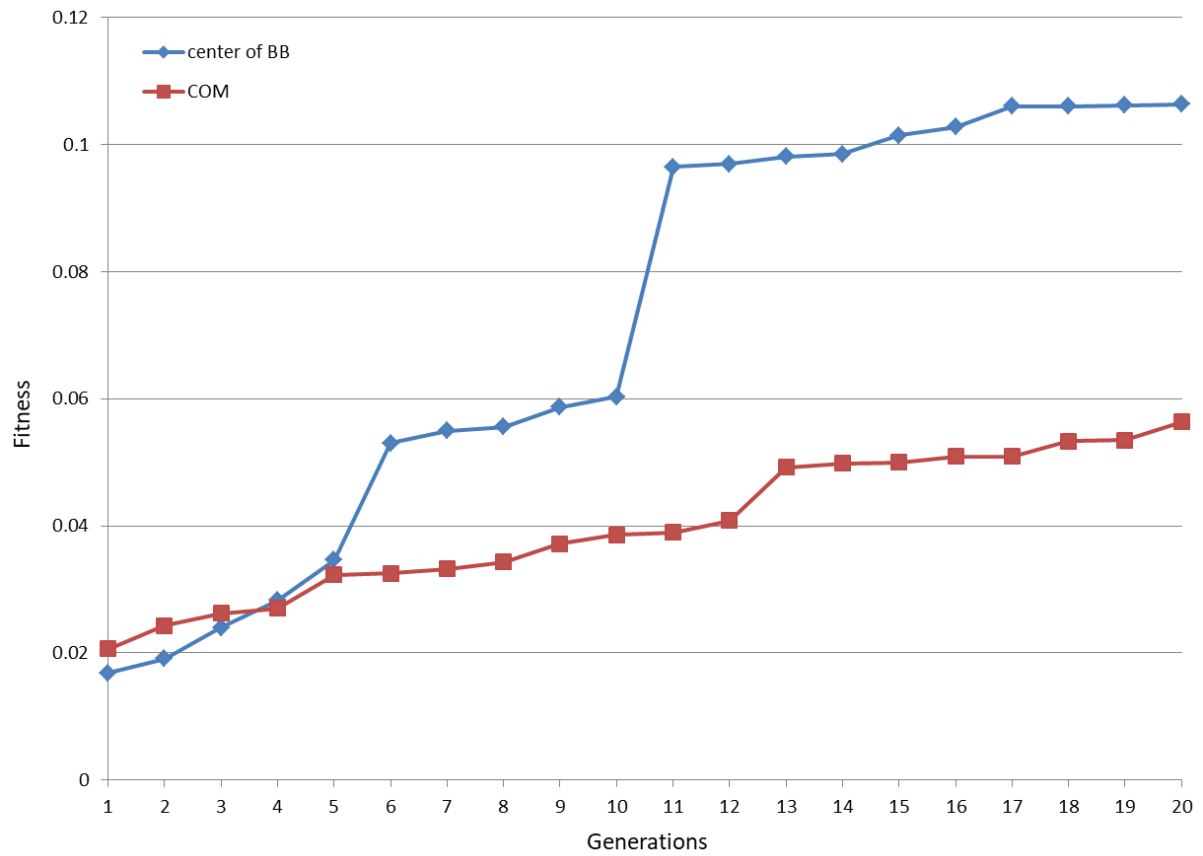


Figure 4.8. Centre of BB translational alignment method against CoM.

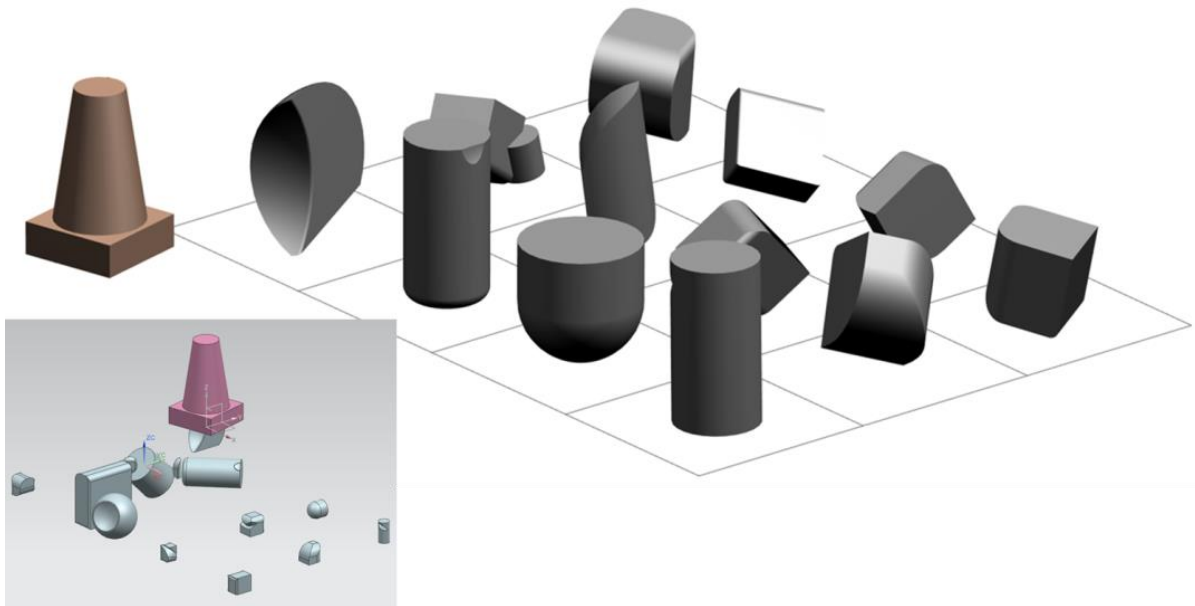


Figure 4.9. A population of 12 grey forms and brown GG after PN, compared to the same population beforehand (inset left).

4.3 3D Alignment and Shape Comparison Conclusions

In this chapter, investigation of shape matching is described which has involved the key aspects listed below:

- Developing a representation for the simple GG. Soft geometric constraints (in the form of simple bounding geometry) have been created at the start of the evolution process to investigate the alignment and shape comparison methods:
 - ❖ GG that are represented with a single geometric primitive or with two geometric primitives but are not complex shapes (Cylinder, Broken Ice Cream, Traffic Cone), work well; produce excellent results, accurate and efficient. On the other hand, GG that are represented with two geometric primitives but are more complex shapes (Roof), do not work well; exhibit a lower match with the forms. However this is partly overcome with the re-introduction of user intervention.
- Develop stand-alone programs to investigate the individual operations:
 - ❖ Rotation: All three principal axes of the form are identified (X, Y and Z), and aligned with the principal axes of the GG. The centre of the BB of the form is used as the centre of rotation.
 - ❖ Scaling: Scaling the shapes by using the BB method has the advantage of keeping the BB of the form within the BB of the GG.
 - ❖ Translation: The centre of BB method is preferable to CoM.
 - ❖ Shape comparison.
- Integrate the most efficient operations with EvoShape IED system.
- Turning EvoShape into an automated shape matching programme using GG in order to assess the efficacy of the approaches being developed.
- Developing PN algorithms between the forms and the GG:
 - Rotating the forms so as to have similar orientation with the GG.
 - Scaling the forms so as to have similar size with the GG. The BB of the form does not exceed the BB of the GG.
 - Co-locating (translating) the forms and the GG at the same place.

5 User Interaction Strategies

This chapter introduces user interaction strategies for evolutionary shape searching. To reintroduce users back into the shape exploration process, two broad approaches of user interaction strategies are examined and presented in this chapter to investigate how best to use the GG capabilities, that have been developed, to assist users in evolutionary shape searching: LBP, or SUI.

5.1 Introduction to user interaction strategies

In the LBP approach (Figure 5.1), populations containing many evolving forms are produced. A subset of the best fitting evolving forms is established by using GG 'behind the scenes' and then presented to the user. The remaining (weakest) evolving forms are deleted.

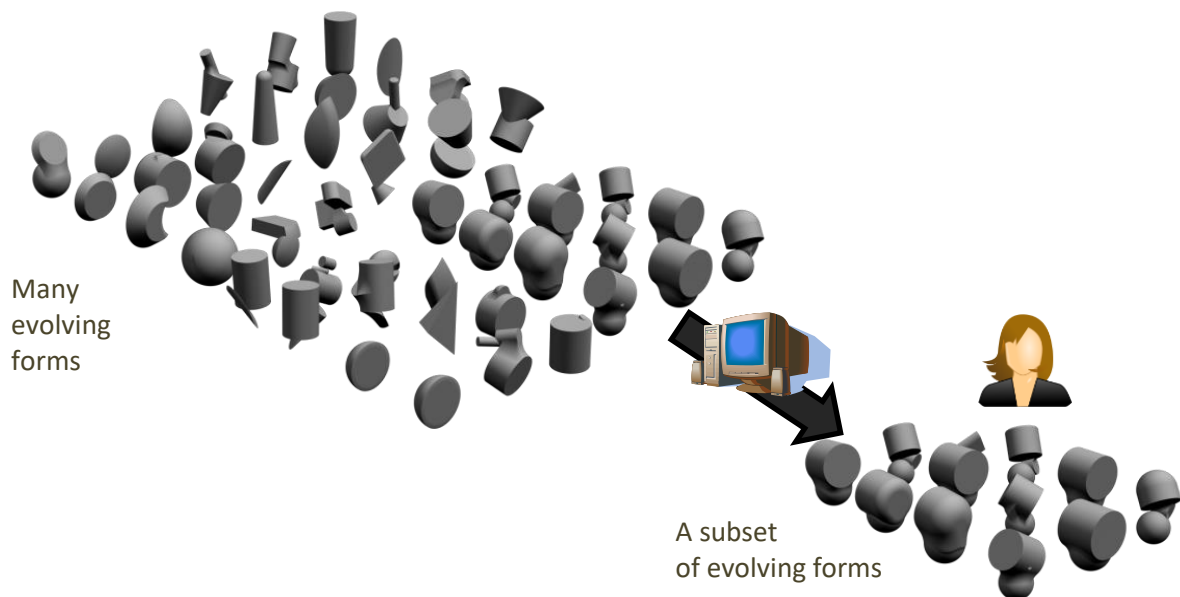


Figure 5.1. Large Background Populations (LBP).

In the SUI approach (Figure 5.2), populations containing a smaller number of evolving forms (same population size of 12 forms as for standard interactive version) are produced. GG are used to evolve the majority of generations 'behind the scenes', with the evolving forms presented to the user every n^{th} generation (sporadically).

During an interactive session, the user is required to select one or more evolving forms or rate one or more evolving forms, from 0 to 10, in the population, providing each form's fitness value. Another generation is then created; the best fitting scores of the last

generation influencing which evolving forms were selected as parents to create this new population.

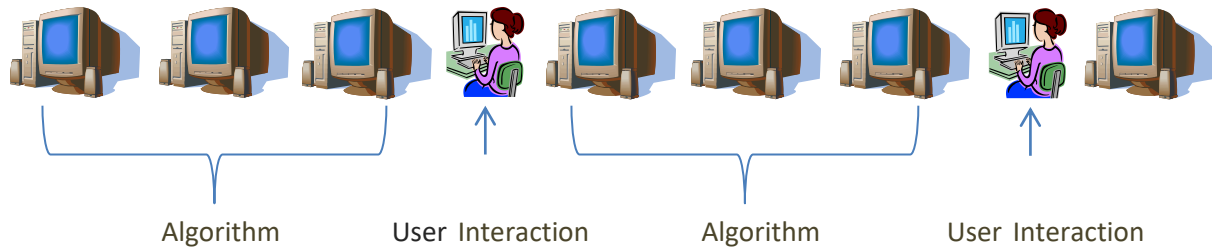


Figure 5.2. Sporadic User Interaction (SUI).

In the SUI approach, the system operates as the original EvoShape program, except the GG algorithm replaces the user for the majority of the populations.

5.2 Detailed Analysis

For the investigation of the interaction strategies, four different starting populations and four different GGs were used, resulting in 16 separate shape matching runs of 20 or 6 iterations (each population has 12 or 80 or 40 forms, resulting in 240 or 480 individual shape-matching operations in total for each run). These combinations of (4 x 4 x 6 x 80) or (4 x 4 x 6 x 40) for LBP and (4 x 4 x 20 x 12) for the benchmark, are henceforth referred to as datasets.

5.2.1 Initial GG analysis and benchmarking with standard-sized population

First, benchmark data have been produced by using 20 generations with 12 phenotypes in each generation, which amounts to 240 forms in total. Random starting populations are identified with an Identification Number¹⁸ (ID) in the following graphs. The first example, shown in Figure 5.3, demonstrates evolving towards a Traffic Cone GG of $185.85cm^3$ volume. The algorithm finds a solution within 11 generations with a percentage of 86% volume match with the GG. The key milestones in this particular experimental run are described in Table 5.1.

¹⁸ a number, 0 – 99999, linked to how many numbers generated by the system's random number generator are discarded before they start to be used to populate the first population of genotypes.

Table 5.1. Key stages in shape matching.

Generation	Stage
1	Initial random population
2	Suitable form type established
8	Population converging to Traffic Cone
10	All forms are now Traffic Cones except mutations
11	Very fit form created
13	Very fit form's genes have increased
14	Approximately one mutation per population
18	Population has converged
18	All forms have 86% match with the Traffic Cone GG

Approach: SUI, 20 generations x 12 phenotypes
Target: Traffic Cone, $185.85m^3$
Match with the target: 86.2%
Achieved in: 11 generations

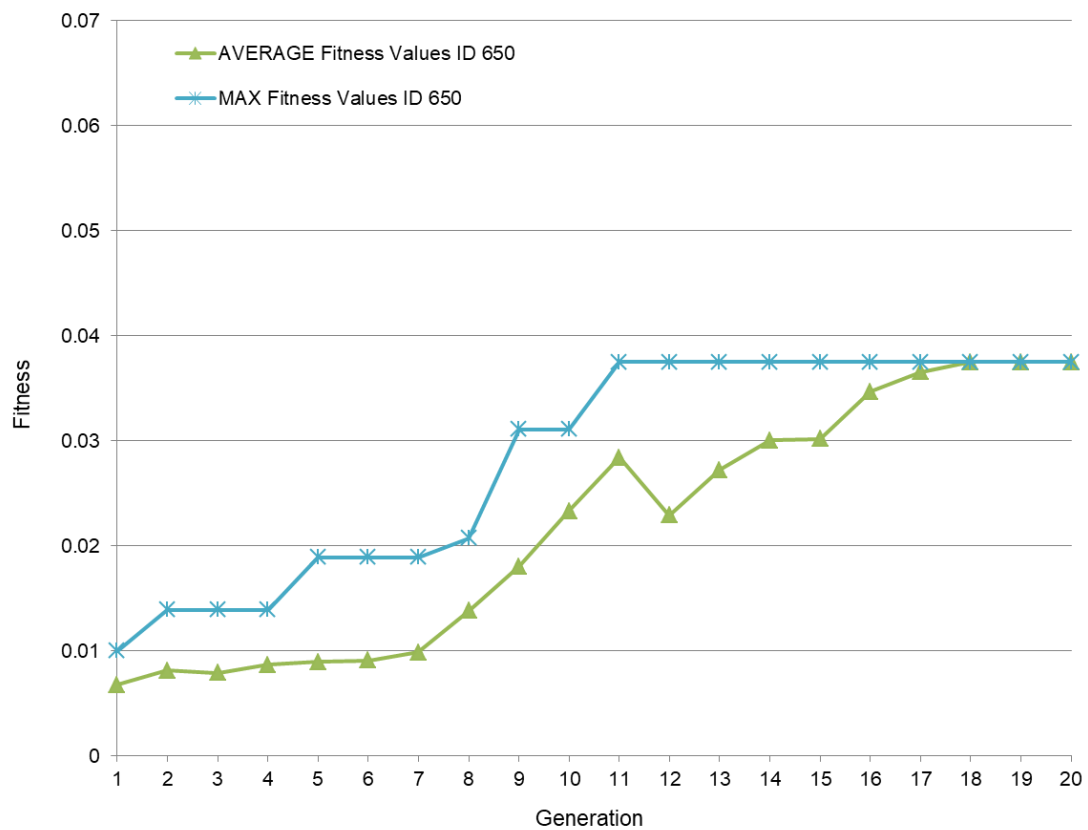
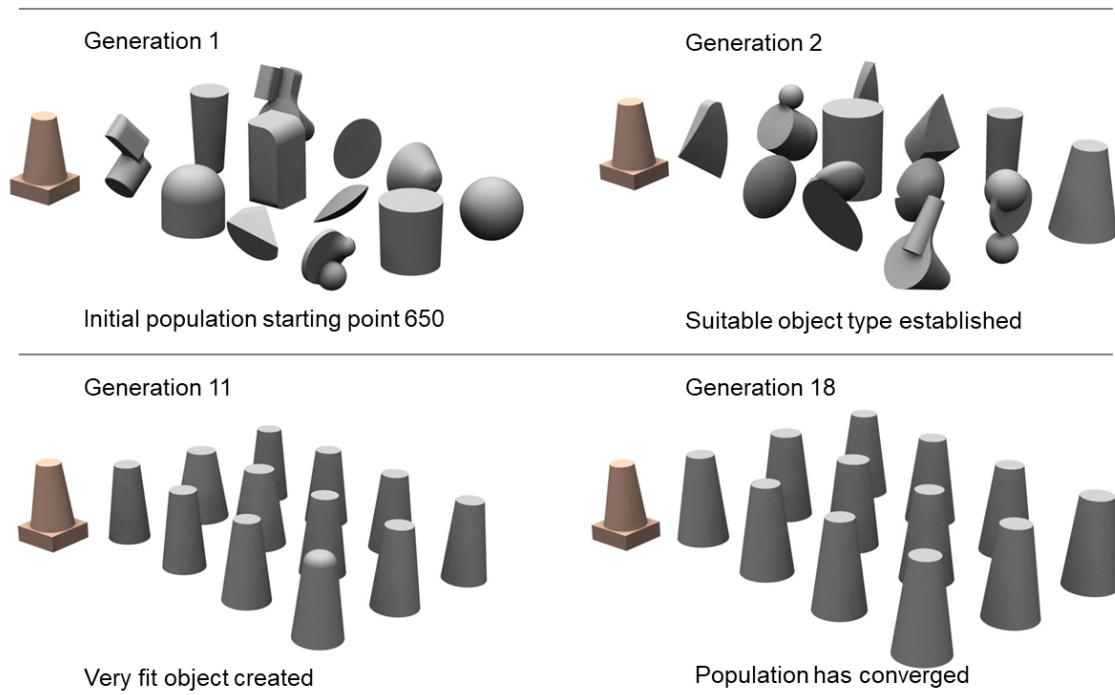


Figure 5.3. Shape matching with benchmark using GG.

Three further examples are summarised in Table 5.2 and in Figure 5.4, with different GGs (broken ice cream, roof, and cylinder).

Table 5.2. Shape matching to a Broken Ice Cream, Roof and Cylinder GG.

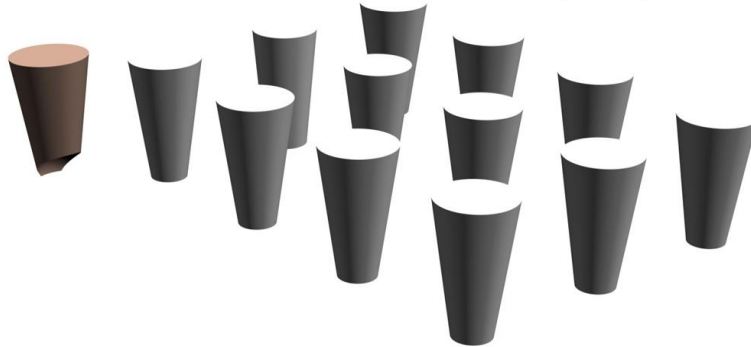
GG	Broken Ice Cream	Roof	Cylinder
Starting populations	228	3900	77682
Volumes	$150.37m^3$	$120.16m^3$	$69.75m^3$
Match with the GG	78%	75%	100%
Achieved in generations	7	9	11
Convergence in generations	19	20	18

The cylinder is a simple GG represented by a single geometric primitive and produces excellent results of 100% match, both accurate and efficient. The traffic cone is represented with two geometric primitives, cone and block, like the broken ice cream which is represented with a cone and a sphere subtracted from the cone. However they both produce good results of 86% and 78% match respectively. On the other hand, in shape matching to a Roof GG, although represented with two geometric primitives (two blocks), is a more complex shape, the probable reason for a lower match with the GG (only 75%).

The forms' fitness values evolving towards these three GGs are shown in Figure 5.5, Figure 5.6, and Figure 5.7 where the starting population is referred to as an ID number. Given the fact that shape matching to a cylinder GG exhibits the highest match (100%), the maximum fitness value in generations 11-20 is 1 (perfect match with the GG).

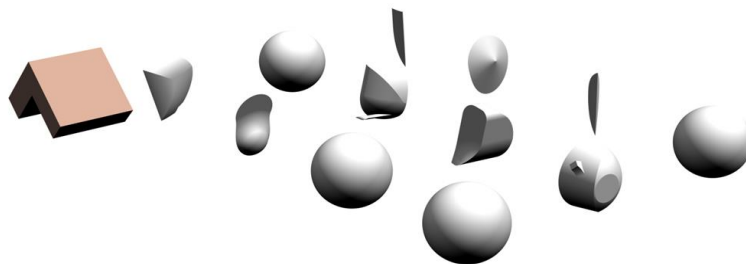
Target: Broken Ice Cream
Starting population number: 228
Volume: $150.37m^3$

Match with the target: 78.81%
Achieved in generation: 7
Convergence in generation: 19



Target: Roof
Starting population number: 3900
Volume: $120.16m^3$

Match with the target: 75.26%
Achieved in generation: 9
Convergence in generation: 20



Target: Cylinder
Starting population number: 77682
Volume: $69.75m^3$

Match with the target: 100.00%
Achieved in generation: 11
Convergence in generation: 18

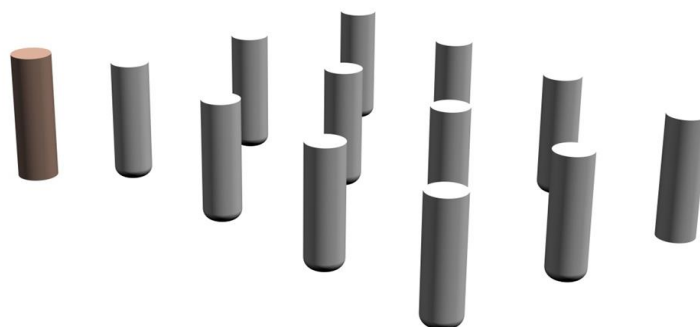


Figure 5.4. Three further shape matching examples – evolved populations alongside their GG shapes.

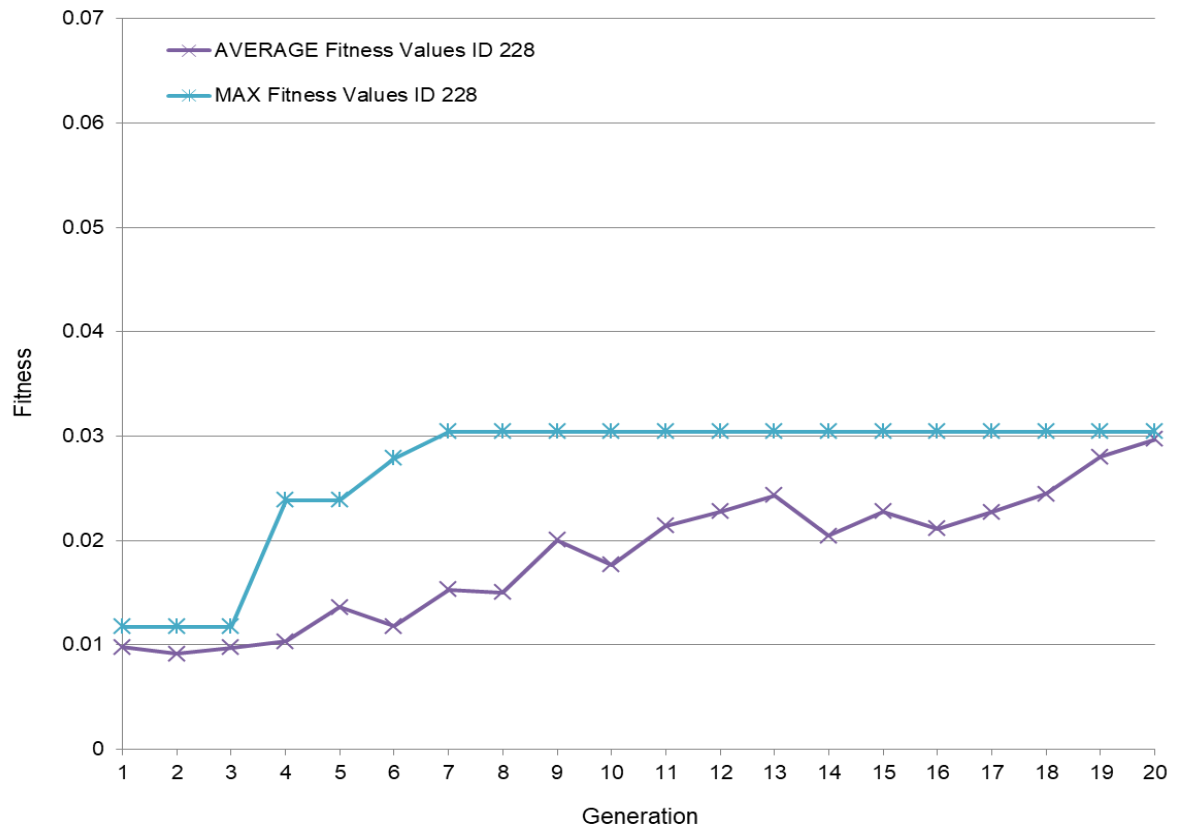


Figure 5.5. Forms' fitness values optimising to broken ice cream.

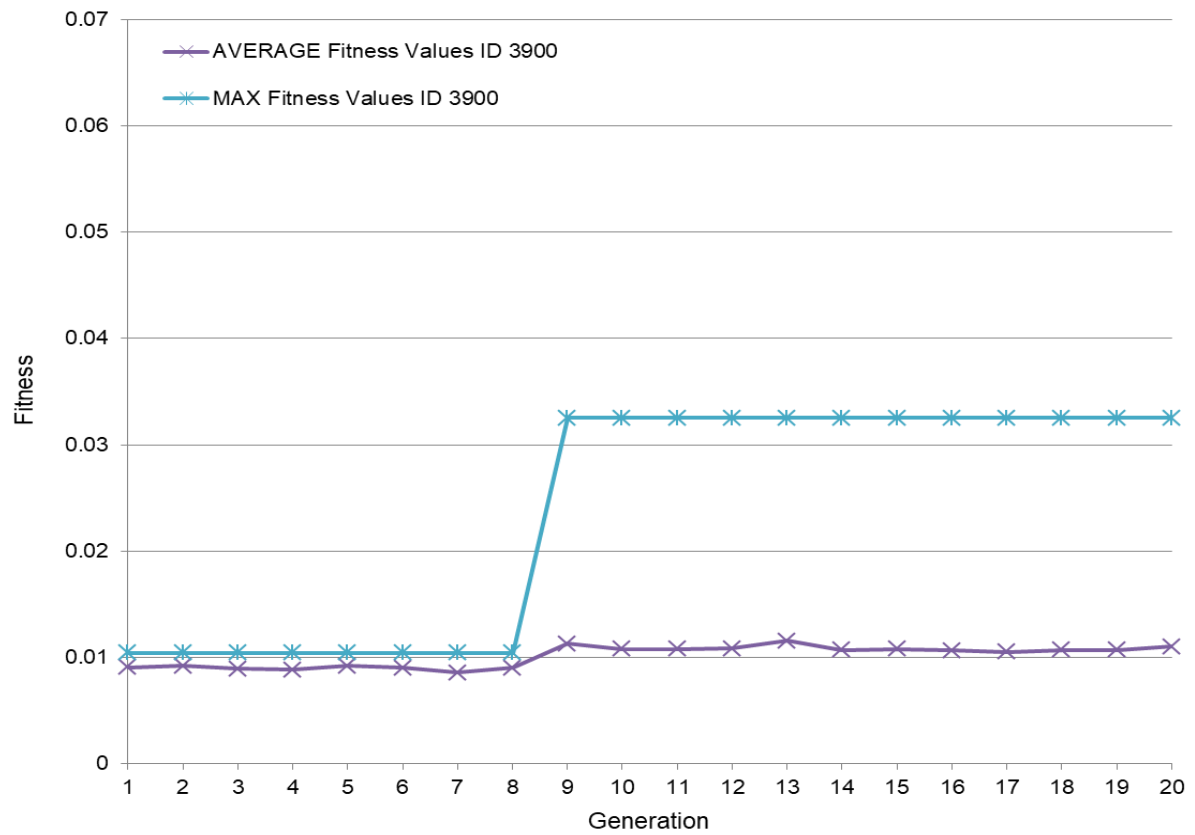


Figure 5.6. Forms' fitness values optimising to roof.

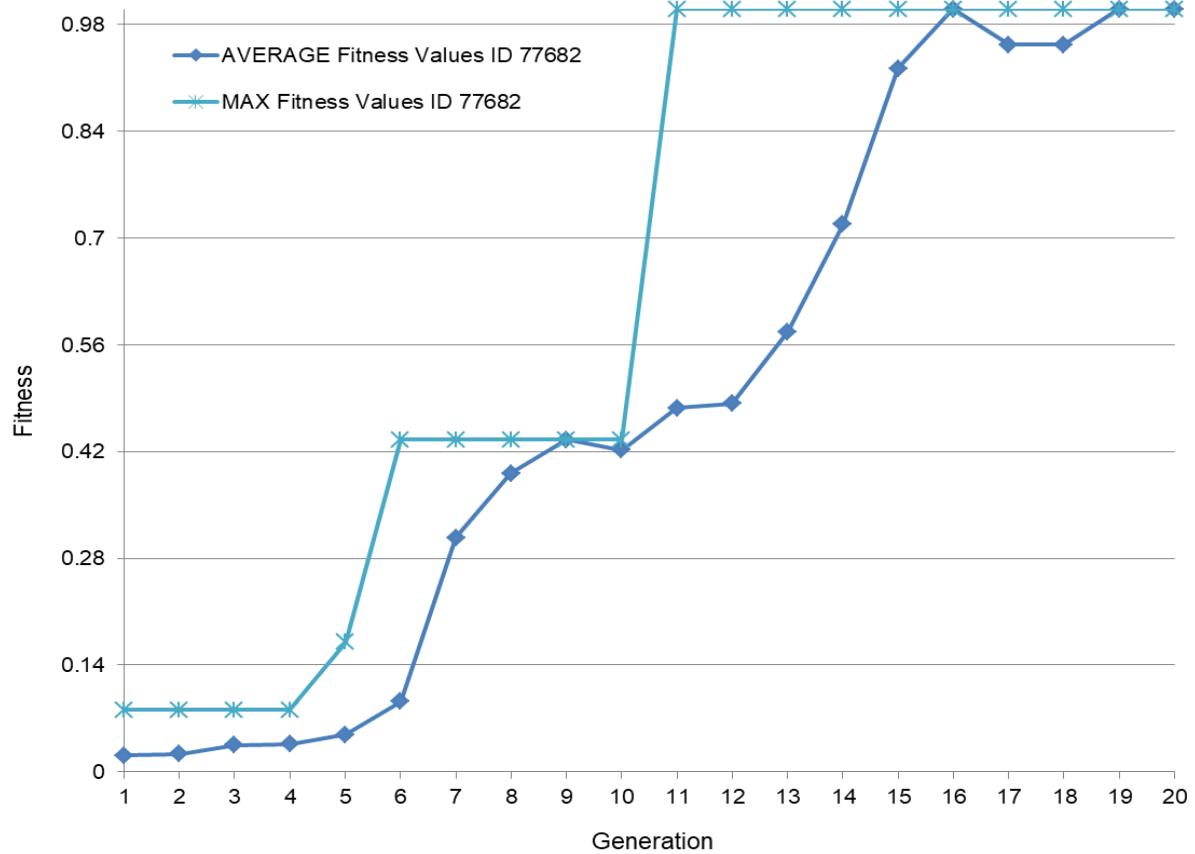


Figure 5.7. Forms' fitness values optimising to cylinder.

For the next 12 tests, the same four GGs with three different starting populations are used. The match with the GG as well as the generations where the match has been achieved are summarised in Table 5.3, Table 5.4, Table 5.5 and Table 5.6. In Figure 5.8, Figure 5.9, Figure 5.10, and Figure 5.11, the graphs show the forms' fitness values with four different starting populations (ID) for each of the four GGs. The average maximum fitness values of these four datasets are shown in Figure 5.12 for each GG.

Table 5.3. Shape matching to a Cylinder GG.

GG	Cylinder		
Starting populations	578	15675	41367
Match with the GG	62%	80%	93%
Achieved in generation	20	13	16
Convergence in generation	20	15	18

Table 5.4. Shape matching to a Broken Ice Cream GG.

GG	Broken Ice Cream		
Starting populations	8766	78901	591
Match with the GG	66%	88%	80%
Achieved in generation	19	15	14
Convergence in generation	20	20	18

Table 5.5. Shape matching to a Traffic Cone GG.

GG	Traffic Cone		
Starting populations	20500	85000	300
Match with the GG	84%	64%	79%
Achieved in generation	13	14	17
Convergence in generation	19	18	19

Table 5.6. Shape matching to a Roof GG.

GG	Roof		
Starting populations	50800	685	870
Match with the GG	84%	88%	55%
Achieved in generation	17	3	5
Convergence in generation	20	20	20

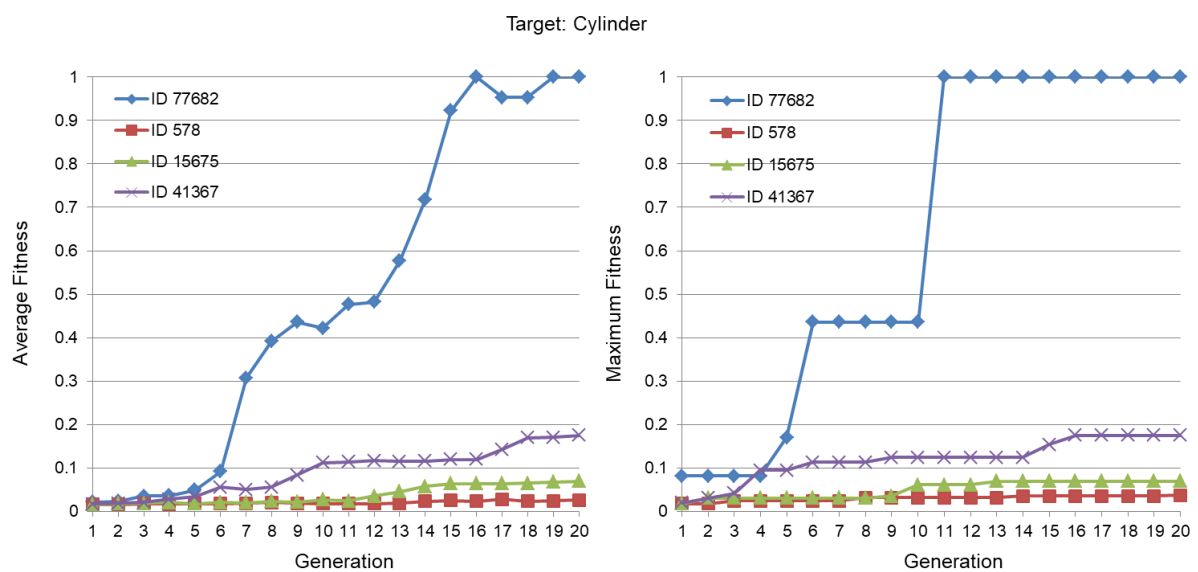


Figure 5.8. Forms' mean (left) and maximum (right) fitness values evolving towards a Cylinder with four different starting populations.

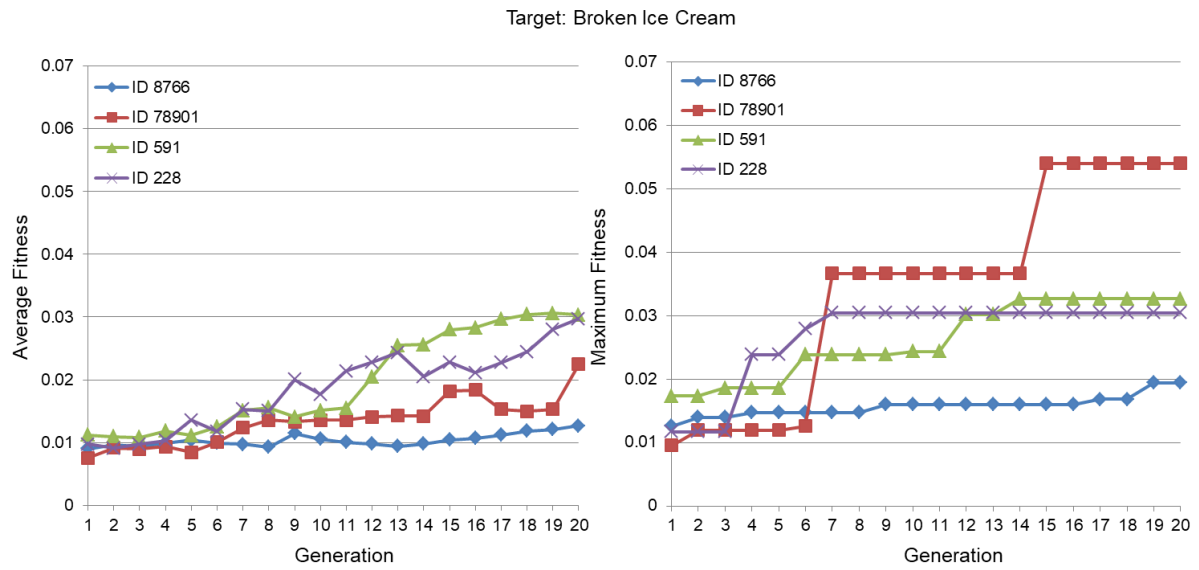


Figure 5.9. Forms' mean (left) and maximum (right) fitness values evolving towards a Broken Ice Cream with four different starting populations.

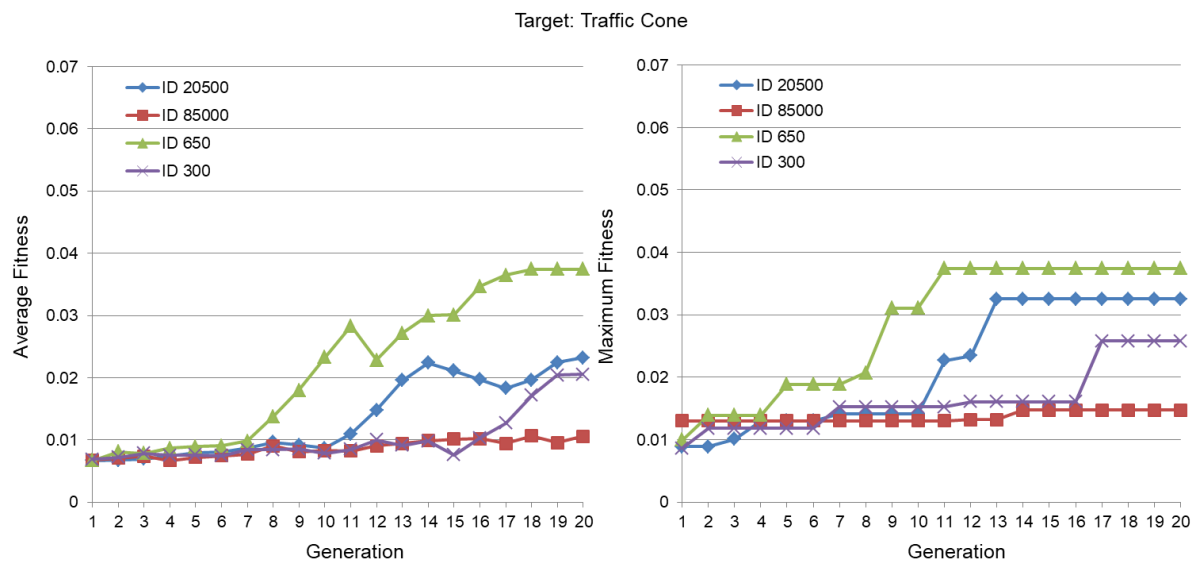


Figure 5.10. Forms' mean (left) and maximum (right) fitness values evolving towards a Traffic Cone with four different starting populations.

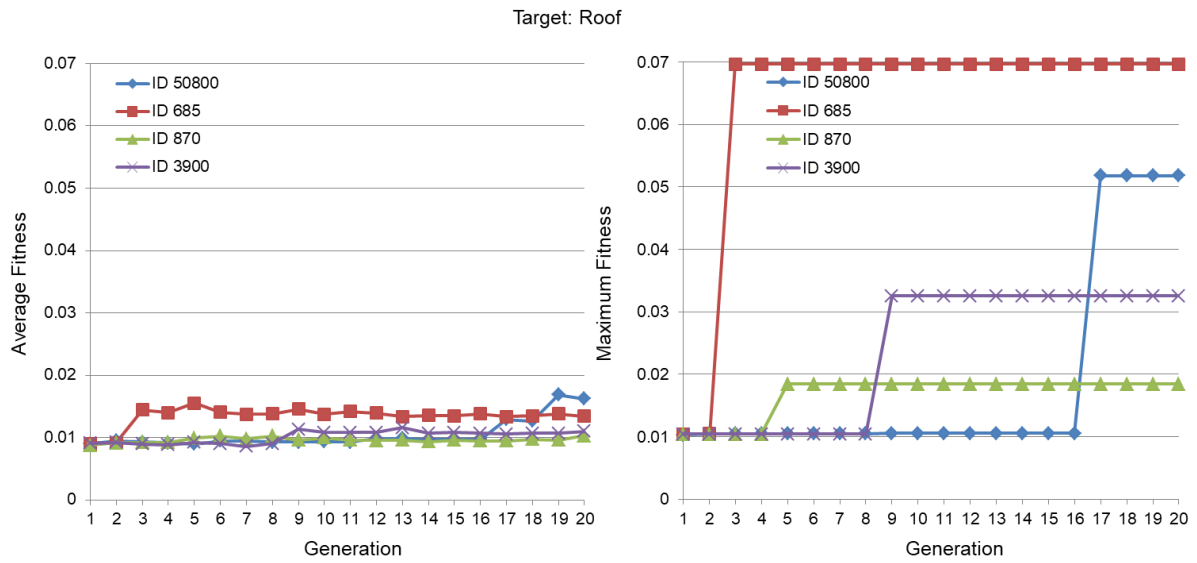


Figure 5.11. Forms' mean (left) and maximum (right) fitness values evolving towards a Roof with four different starting populations.

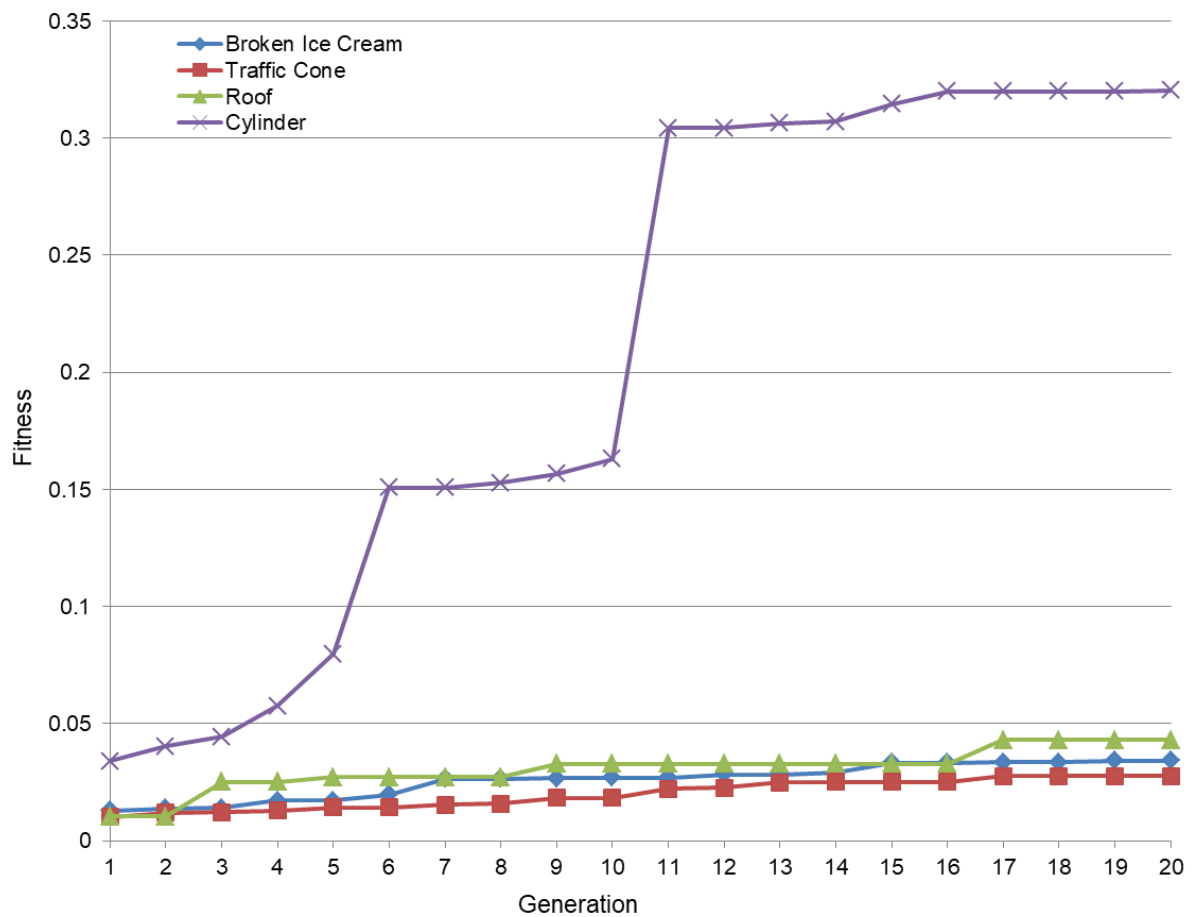


Figure 5.12. Average maximum fitness values of four different starting populations for each of the four GGs.

Again the cylinder produces the best results with the highest fitness values. But, surprisingly, shape matching with a roof does not exhibit lower fitness values, after generation 16, than a broken ice cream or a traffic cone. These unexpected high maximum values are random and the corresponding average fitness values graph shown in Figure 5.13, prove that a roof can give the worst average fitness values because of its shape complexity.

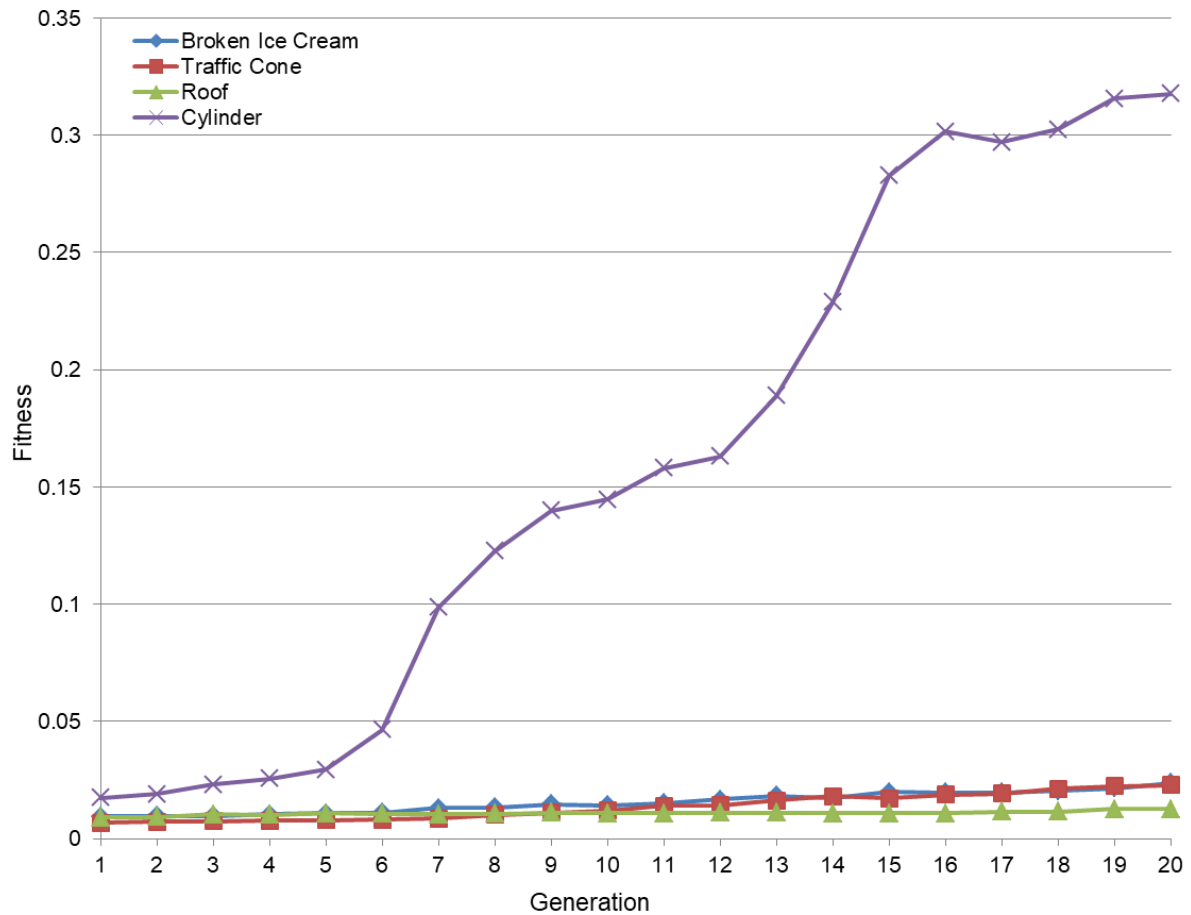


Figure 5.13. Average mean fitness values of four different starting populations for each of the four GG.

However, fitness values depend on the absolute volumes of the GG. For that reason it is not correct to compare fitness values directly when comparing different GGs because the GGs have different volumes. Therefore, fitness values have been normalised as shown mathematically in Equation 5.1 and graphically in Figure 5.14. All the GGs have been normalised according to the 'Cylinder' GG, since it has the smallest volume (69.75cm³).

$$\tilde{f} = f \frac{V_{GG}}{V_{Cylinder}} \quad 5.1$$

Where:

- ❖ \tilde{f} is the normalised fitness function of an evolving form.
- ❖ $V_{Cylinder}$ is the volume of the Cylinder.

A form's % match with the GG is used as a more intuitive expression than fitness functions. Therefore, the graph in Figure 5.15 shows the *average* average matches between the evolved forms and the GG, as calculated using the Equations 4.7 and 4.8. A cylinder can produce efficient results by achieving an 80% match, a traffic cone and a broken ice cream reach 65% and 70% respectively, where as a roof achieves no more than a 20% match.

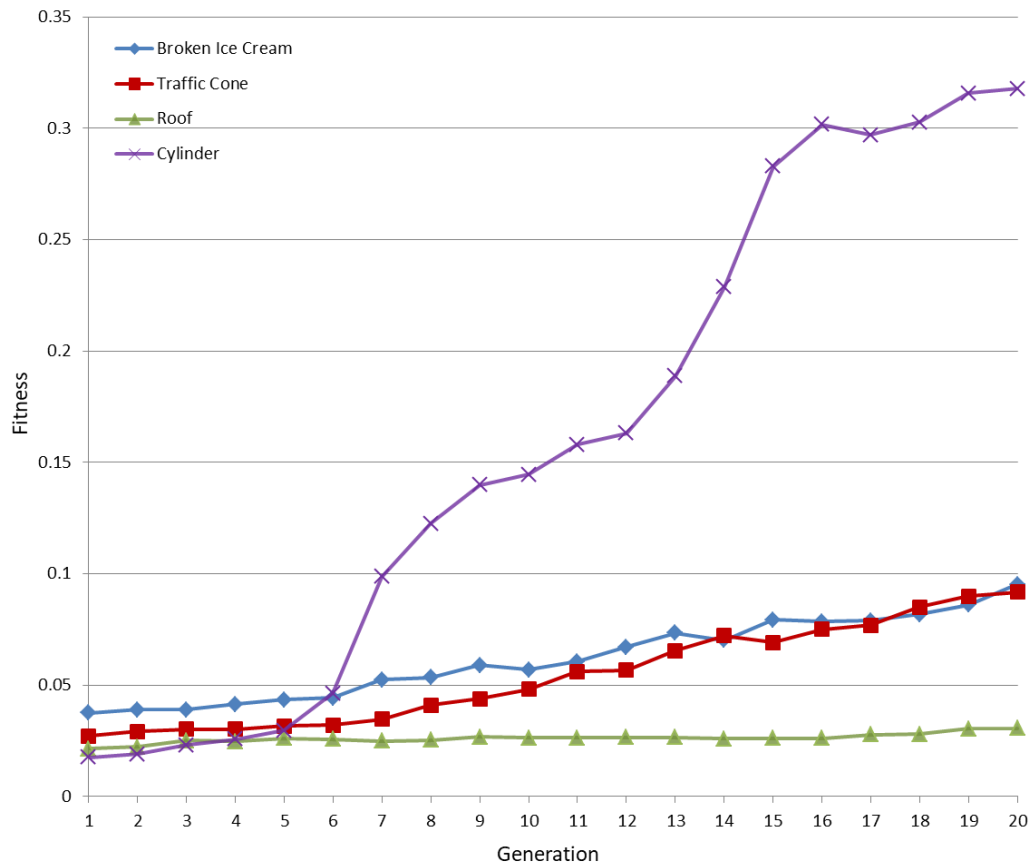


Figure 5.14. Normalised average mean fitness values of four different starting populations for each of the four GG.

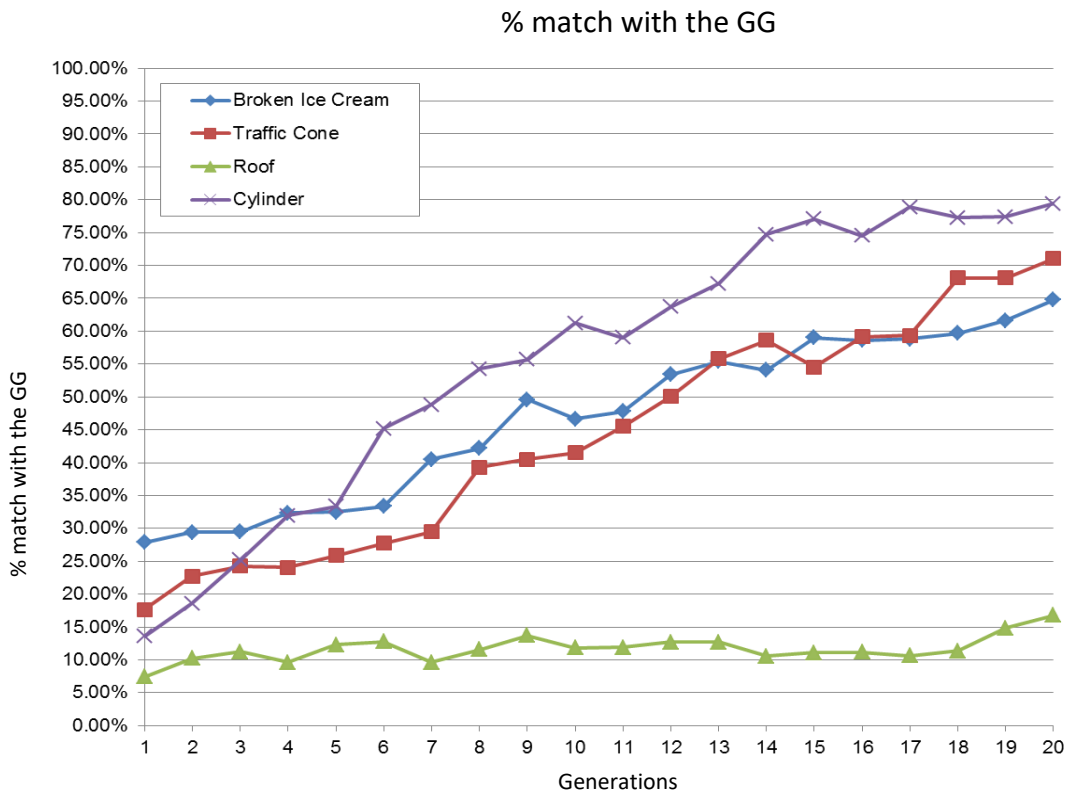


Figure 5.15. Average average matches of four different starting populations between the evolved forms and the four GG.

Ultimately, the IEC design system’s aim is not to ‘find the GG’, but this task is used here to assess the efficacy of the PN and shape comparison. A form’s % match with the GG is sometimes used as a more intuitive expression than fitness functions. Detailed benchmark examples are presented in Appendix H).

5.2.2 Large Background Populations

Due to the large effect of the starting population, and that small populations are well known to be detrimental to GA performance, a method working with LBP was developed which uses the Shape Matching with the GG to present a subset of fitter shapes to the user.

LBP of 40 shapes

Experimental results have been produced by using the LBP of 40 shapes approach where six generations are created with 40 phenotypes in each generation, which amounts to 240 forms in total as an equitable comparison to the benchmark. The strength of the LBP method is proven through the results of a dataset. This dataset of forms includes the GGs broken ice cream, traffic cone, roof and cylinder.

The first example, shown in detail in Figure 5.16, demonstrates evolving towards a Broken Ice Cream GG of volume $150.37m^3$ and the starting population is referred to as an ID number. The software achieves a solution within six generations with 96% volume match

with the GG. The key milestones in this particular experimental run are described in Table 5.7.

Table 5.7. Key stages in shape matching.

Generation	Stage
1	Initial random population
2	Suitable form type established
3	Population converging to Traffic Cone
5	All forms are now Broken Ice Creams except mutations
6	Very fit form created

Approach: LBP, 6 generations x 40 phenotypes
Target: Broken Ice Cream, $150.37m^3$
Match with the target: 96.7%
Achieved in: 6 generations

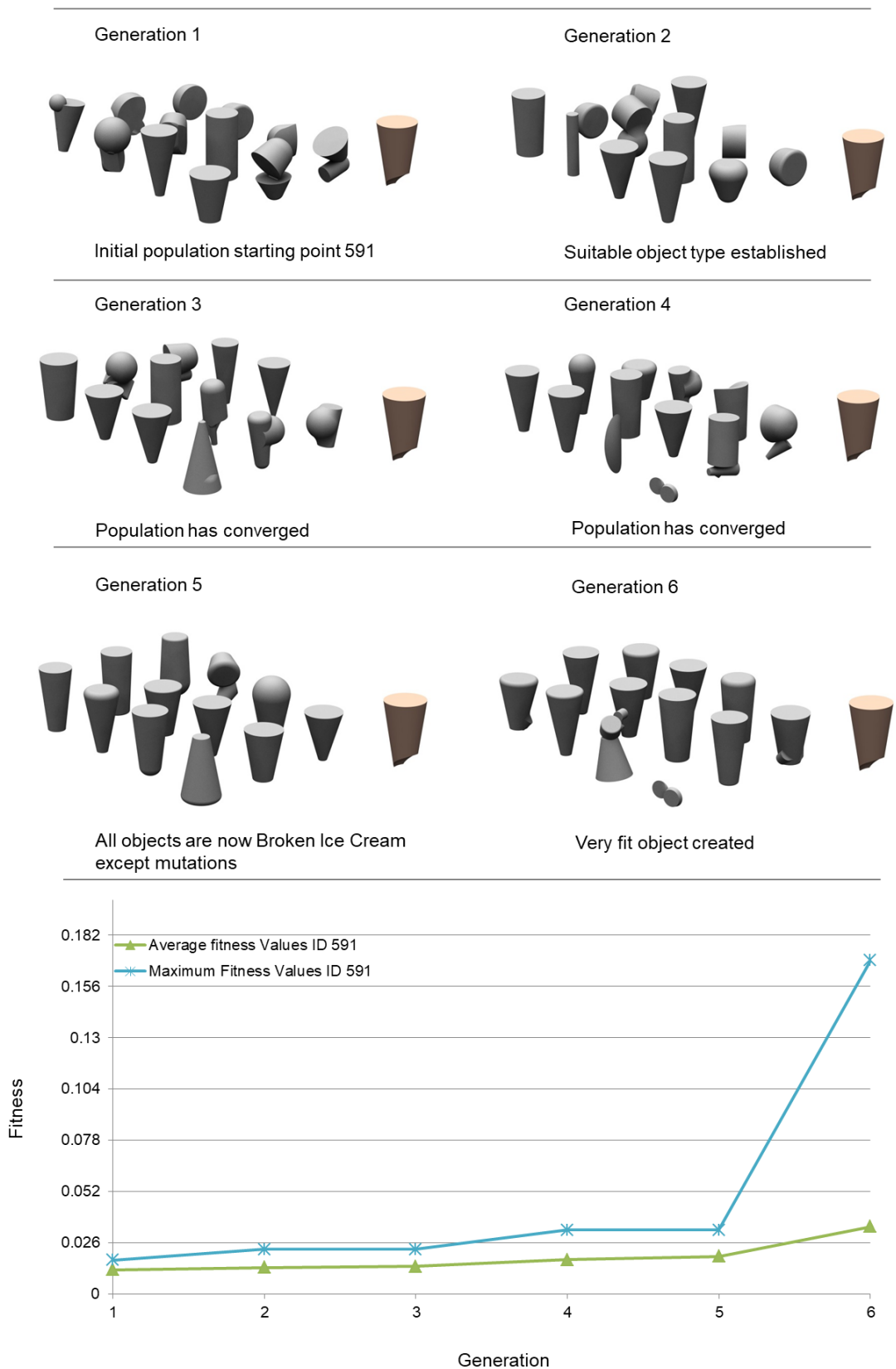


Figure 5.16. Shape matching with LBP of 40 shapes approach using GG.

Three further examples are summarised in Table 5.8 and in Figure 5.17 with different GGs (broken ice cream, roof, and cylinder).

Table 5.8. Shape matching to a Traffic Cone, Roof and Cylinder GG.

GG	Traffic Cone	Roof	Cylinder
Starting populations	650	3900	77682
Volumes	$185.85m^3$	$120.16m^3$	$69.75m^3$
Match with the GG	73%	21%	96%
Achieved in generations	5	1	4
Convergence in generations0.	6	6	4

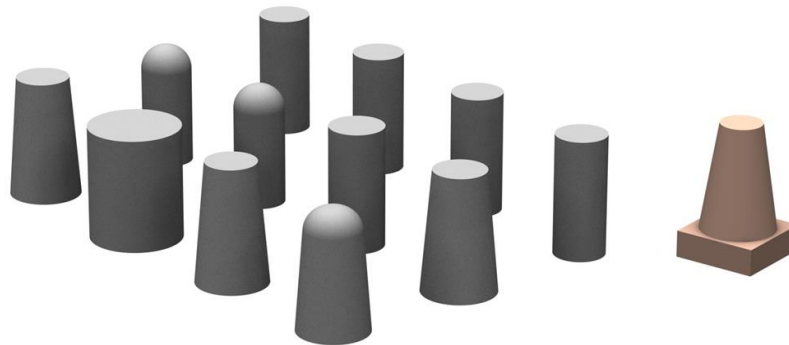
In LBP of 40 shapes approach:

- The cylinder produces good results of 96% match, very close to the results in the standard-sized population approach (benchmark).
- The traffic cone produces slightly lower match (73%) than the benchmark.
- The broken ice cream produces much better results (96% match) than the benchmark.
- And, the shape matching to a roof GG, with increased complexity, exhibits an even lower match with the GG (only 21%).

Forms' fitness values evolving towards these three GGs are shown in Figure 5.18, Figure 5.19, and Figure 5.20 and the starting population is referred to as an ID number. Given the fact that shape matching to a cylinder GG exhibits a high match of 96%, the maximum fitness value in generations 4-6 is very close to 1.

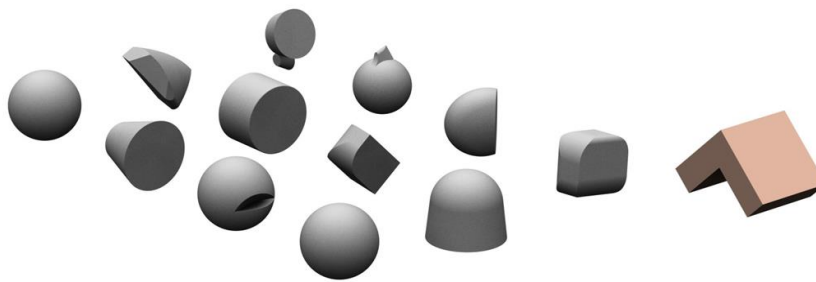
Target: Traffic Cone
Starting population number: 650
Volume: $185.85m^3$

Match with the target: 73.66%
Achieved in generation: 5
Convergence in generation: 6



Target: Roof
Starting population number: 3900
Volume: $120.16m^3$

Match with the target: 21.00%
Achieved in generation: 1
Convergence in generation: 6



Target: Cylinder
Starting population number: 77682
Volume: $69.75m^3$

Match with the target: 96.20%
Achieved in generation: 4
Convergence in generation: 4

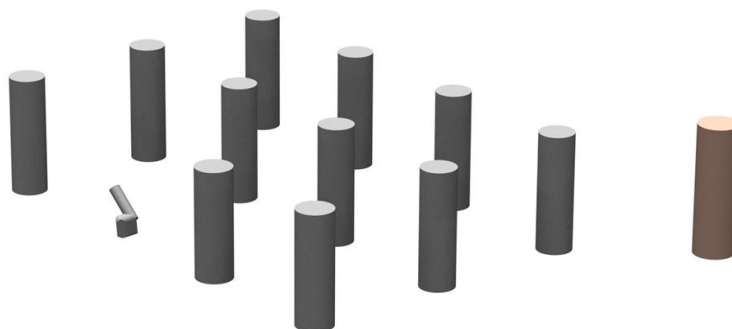


Figure 5.17. Three further shape matching examples with LBP of 40 shapes approach – evolved populations alongside their GG shapes.

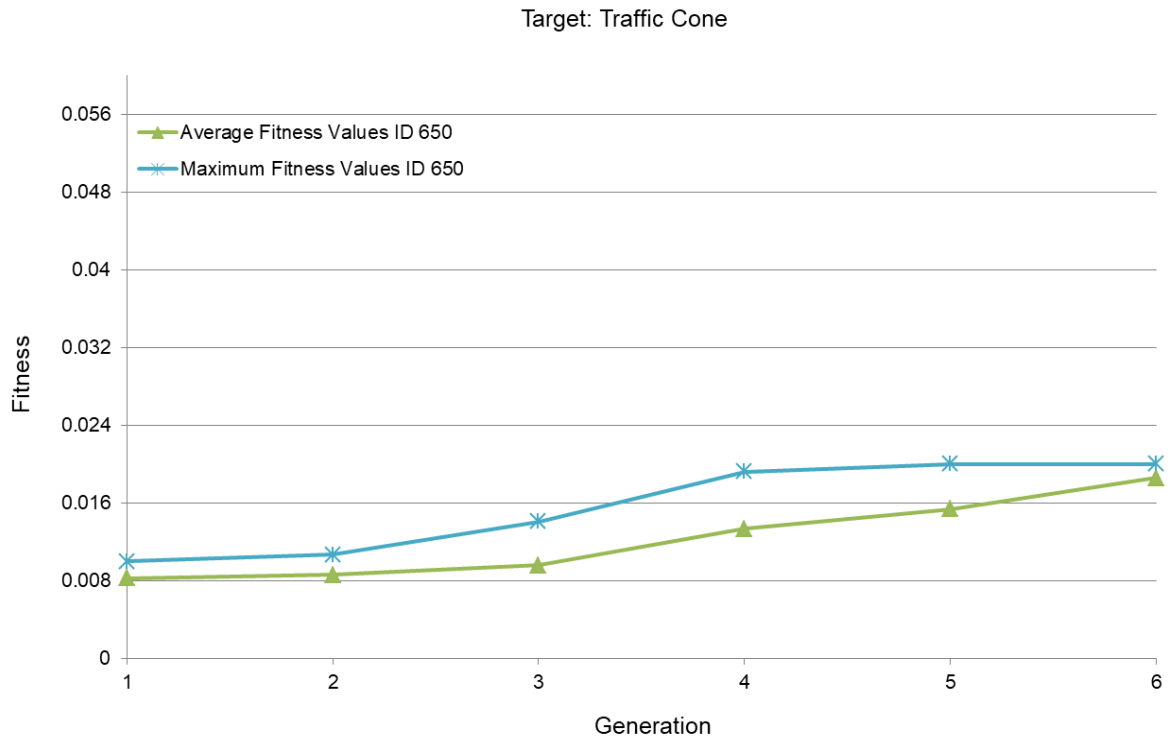


Figure 5.18. Forms' fitness values evolving towards a traffic cone.

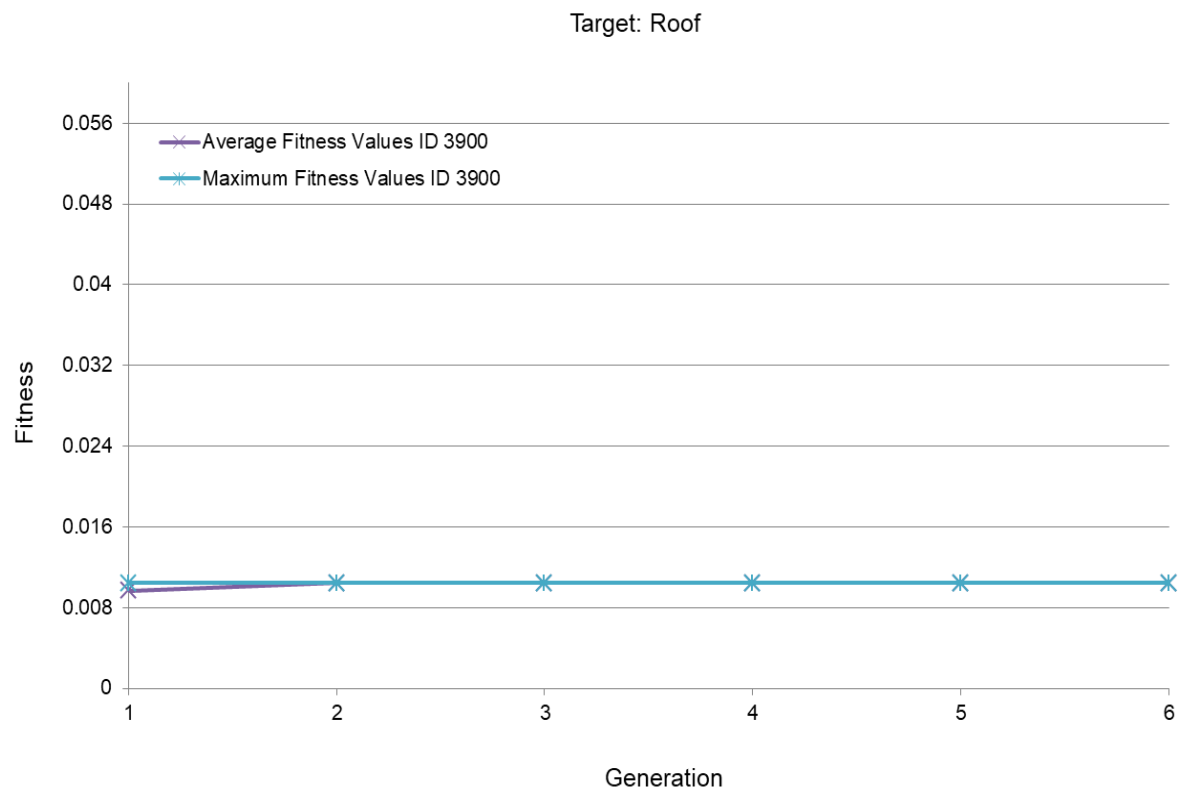


Figure 5.19. Forms' fitness values evolving towards a roof.

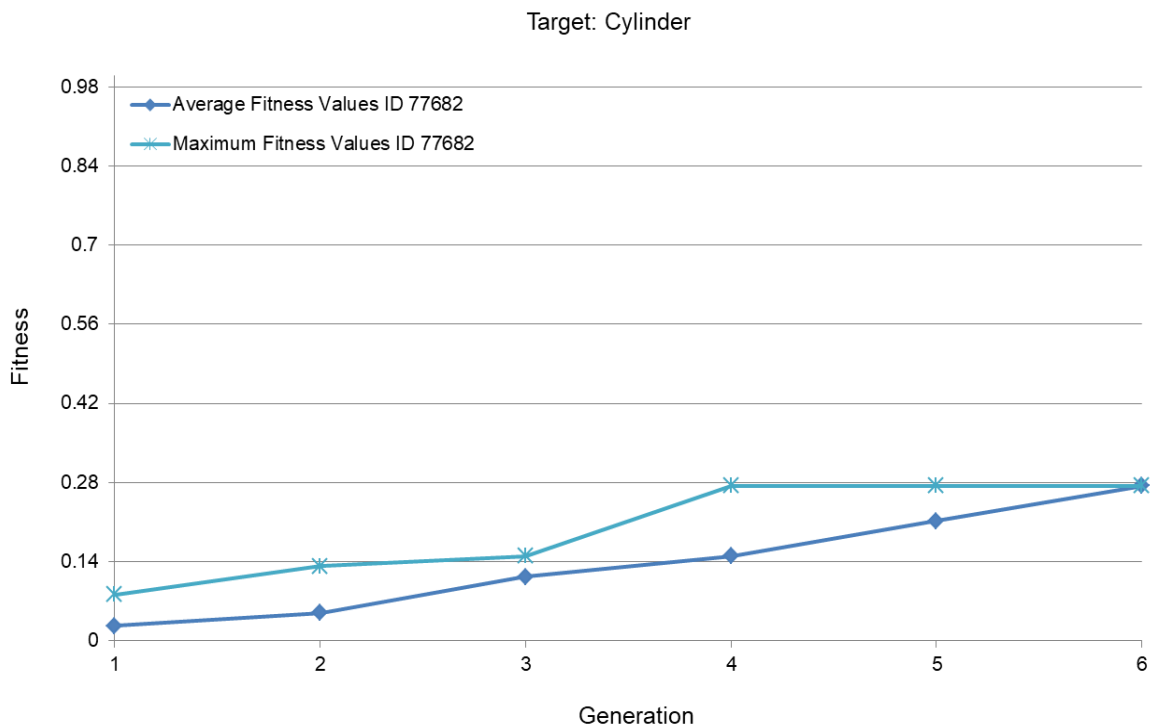


Figure 5.20. Forms' fitness values evolving towards a cylinder.

For the next 12 tests, the same four GGs with three different starting populations are used. The match with the GG as well as the generations where the match has been achieved are summarised in Table 5.9, Table 5.10, Table 5.11 and Table 5.12. In Figure 5.21, Figure 5.22, Figure 5.23, and Figure 5.24, the graphs show the forms' fitness values of four different starting populations (ID) for each of the four GGs. The average maximum fitness values of these four starting populations are shown in Figure 5.25 for each GG.

Table 5.9. Shape matching to a Cylinder GG.

GG	Cylinder		
Starting populations	578	15675	41367
Match with the GG	100%	100%	100%
Achieved in generation	6	3	4
Convergence in generation	6	3	4

Table 5.10. Shape matching to a Broken Ice Cream GG.

GG	Broken Ice Cream		
Starting populations	8766	78901	228
Match with the GG	90%	76%	65%
Achieved in generation	2	2	2
Convergence in generation	6	3	4

Table 5.11. Shape matching to a Traffic Cone GG.

GG	Traffic Cone		
Starting populations	20500	85000	300
Match with the GG	71%	76%	60%
Achieved in generation	3	1	6
Convergence in generation	5	3	6

Table 5.12. Shape matching to a Roof GG.

GG	Roof		
Starting populations	50800	685	870
Match with the GG	93%	97%	30%
Achieved in generation	4	5	2
Convergence in generation	5	6	6

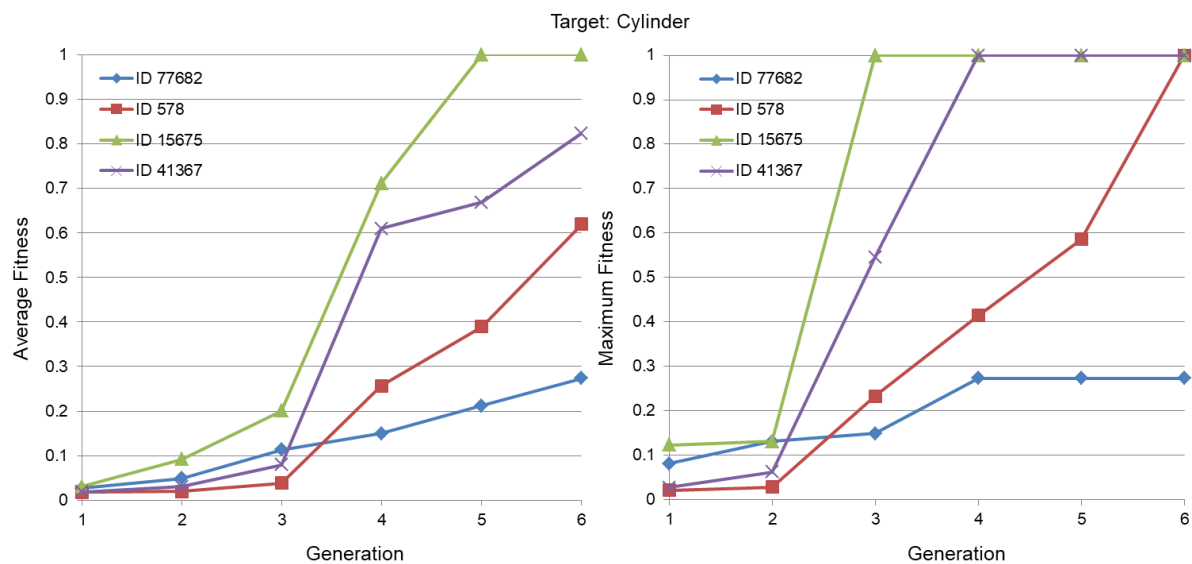


Figure 5.21. Forms' fitness values evolving towards a Cylinder with four different starting populations.

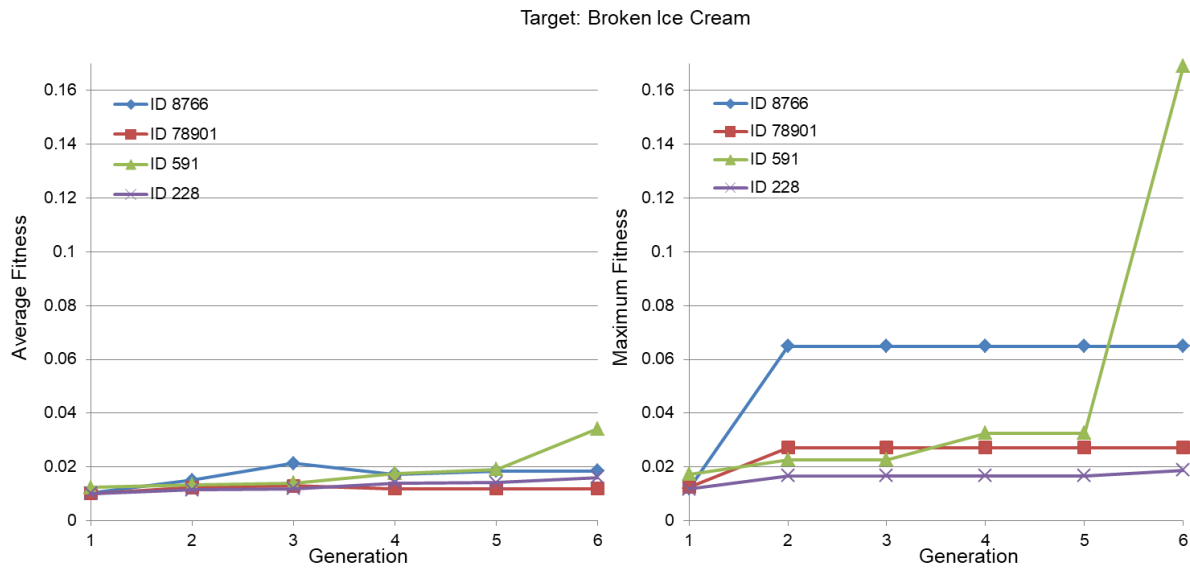


Figure 5.22. Forms' fitness values evolving towards a Broken Ice Cream with four different starting populations.

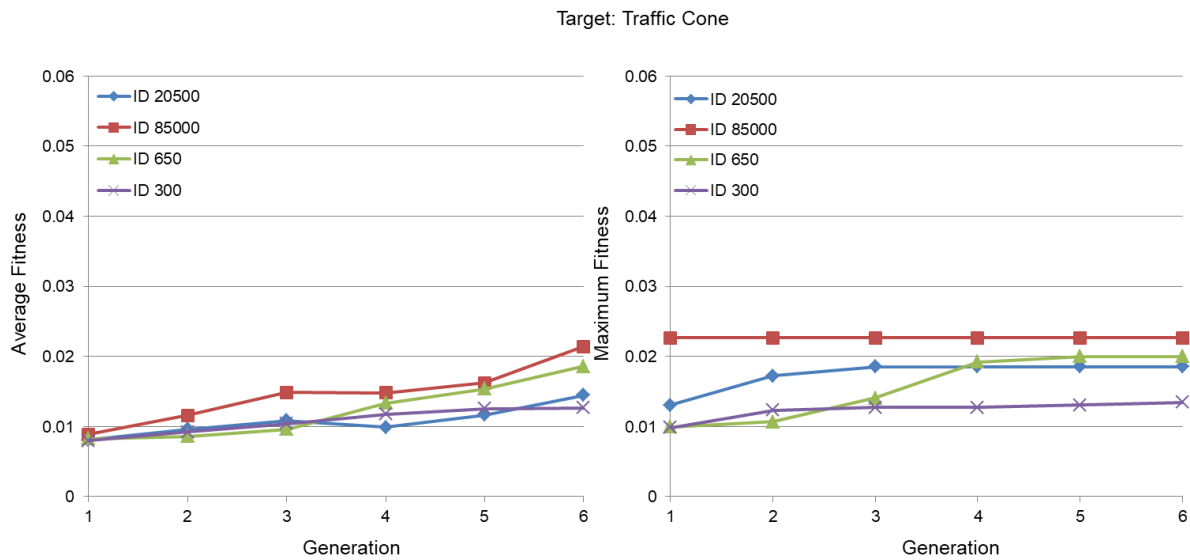


Figure 5.23. Forms' fitness values evolving towards a Traffic Cone with four different starting populations.

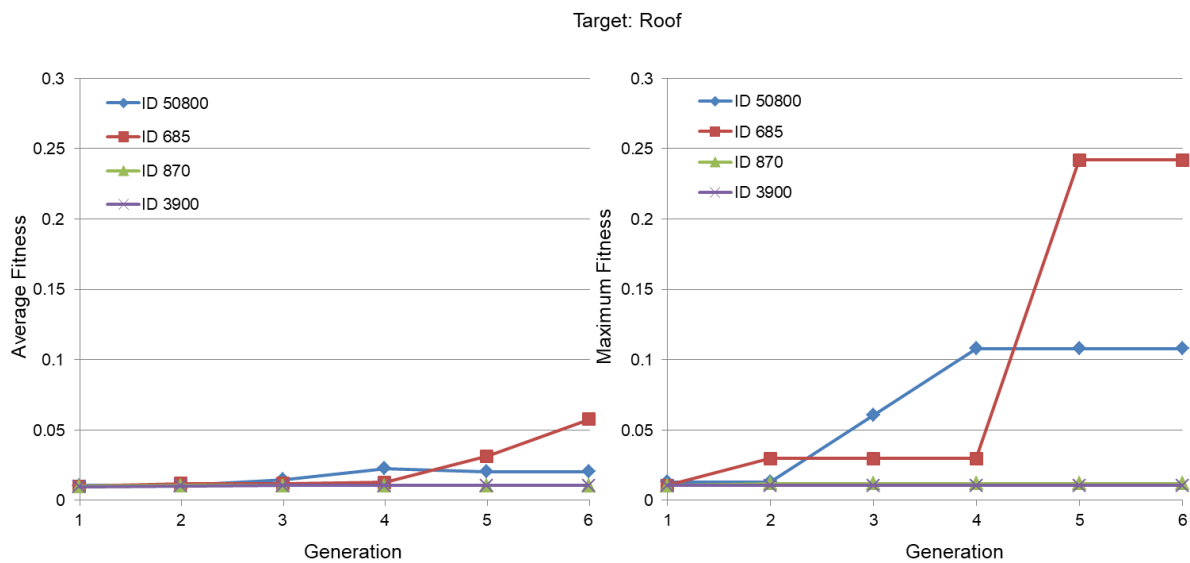


Figure 5.24. Forms' fitness values evolving towards a Roof with four different starting populations.

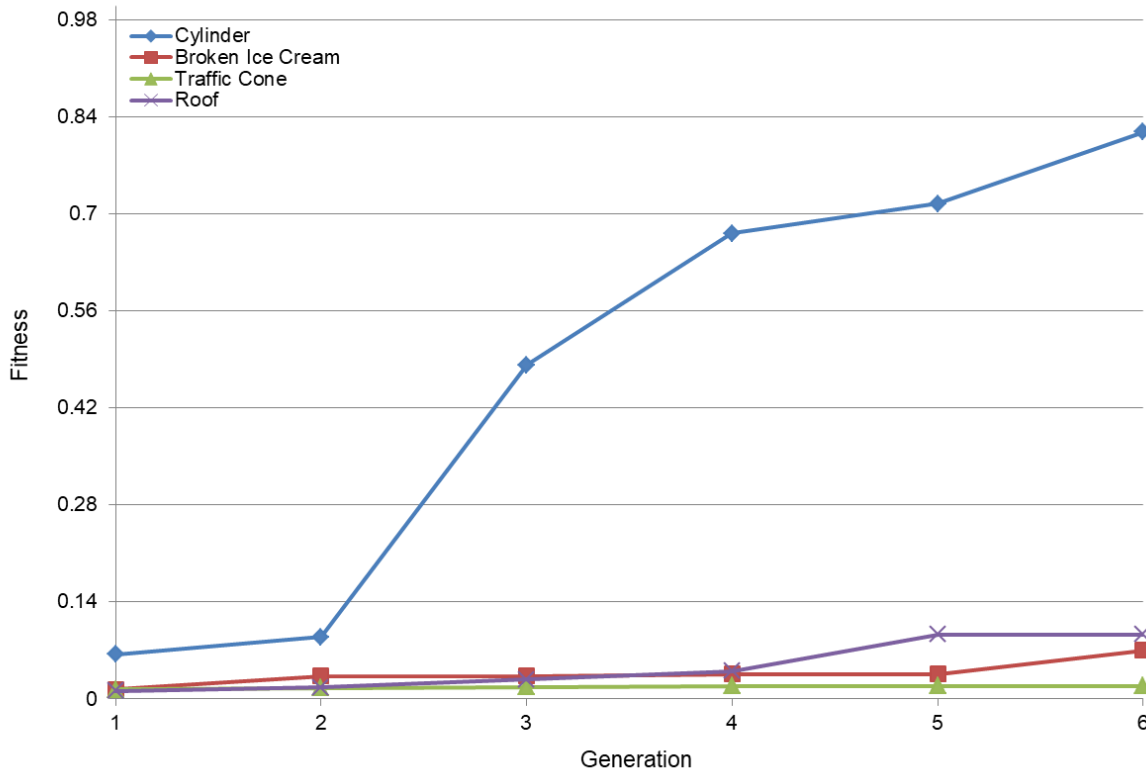


Figure 5.25. Average maximum fitness values of four different starting populations for each of the four GG, using LBP of 40 shapes.

The cylinder produces the best results with the highest fitness values while shape matching with a roof, broken ice cream, or traffic cone exhibit low fitness values during the whole process. The average fitness values graph shown in Figure 5.26, shows that a traffic cone can give the worst average fitness values.

The graph in Figure 5.27 compares the average maximum fitness values between the LBP of 40 shapes and the benchmark. The LBP of 40 shapes approach has better fitness than the benchmark. However, this is not a fair comparison because the two approaches do not represent a comparable number of solutions analysed, shape comparisons or computing time. A fairer comparison of all the approaches is shown in Figure 5.29.

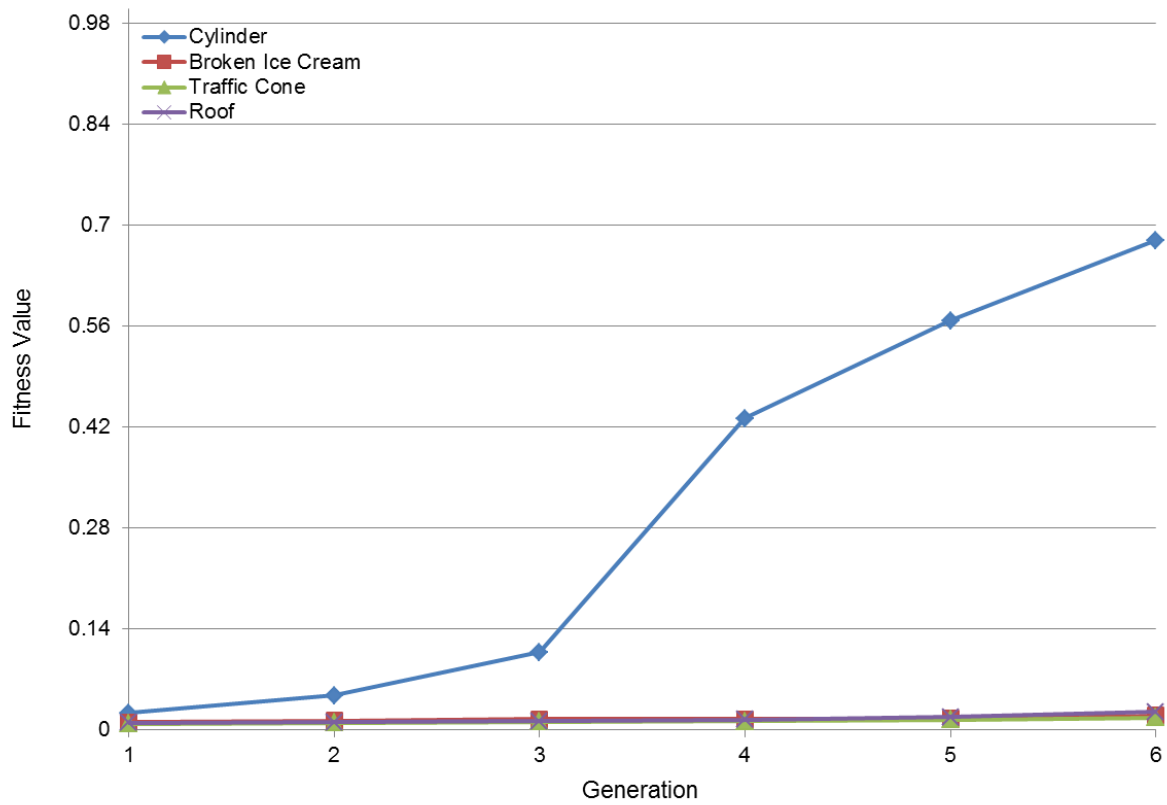


Figure 5.26. Average mean fitness values of four different starting populations for each of the four GG.

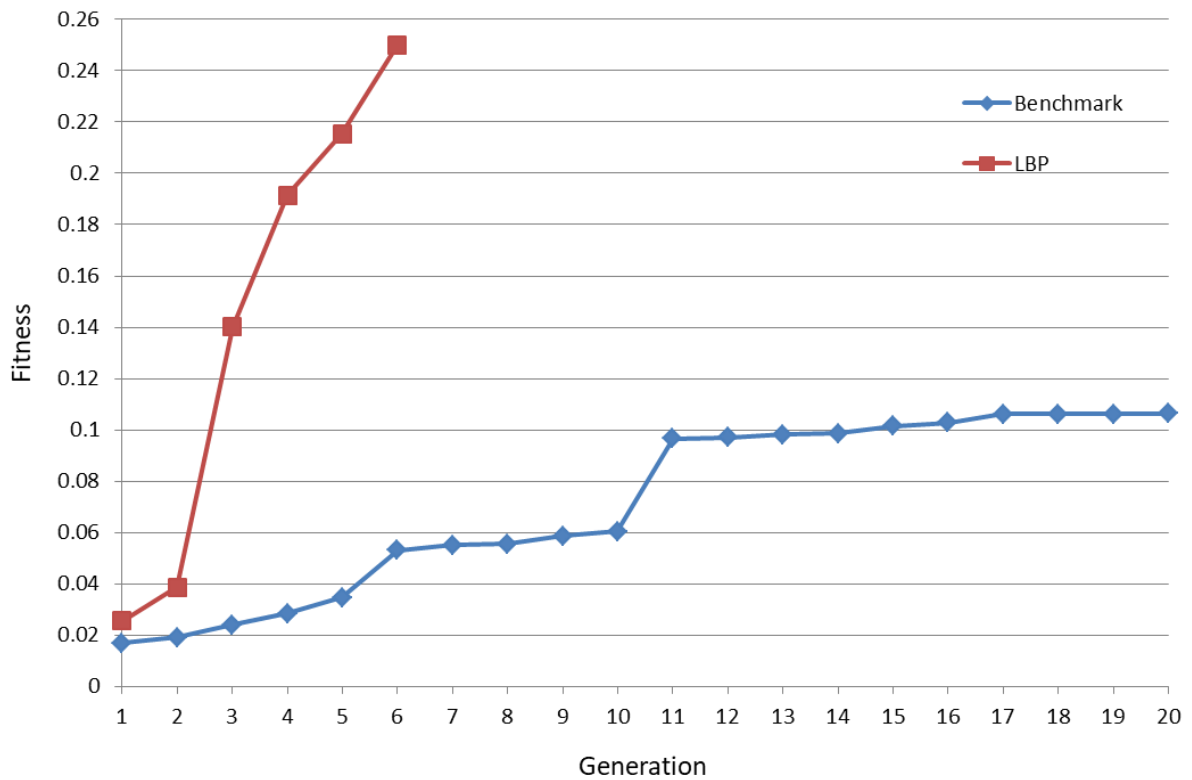


Figure 5.27. Comparison of LBP of 40 shapes and benchmark.

The PN and Shape Matching algorithms that have been developed have been used to direct the evolutionary process during the automated shape-matching task. As a pre-cursor to investigating user interaction strategies, an objective comparison of LBP with a benchmark was carried out that aims to provide better fitness without sacrificing form variety. Results indicated at this stage of development that more tests were needed to produce better fitness and variety, and to investigate the human-machine interaction strategies which best complement the evolutionary design process.

LBP of 80 shapes

For that reason, the number of phenotypes in each generation was increased from 40 to 80. Experimental results have been produced by using the LBP of 80 shapes approach where six and three generations are created with 80 phenotypes in each generation, which amounts to 480 and 240 forms in total, respectively. The strength of the *LBP of 80 shapes* method is proven through experiments based on these datasets. These datasets of forms include four different GGs, broken ice cream, traffic cone, roof and cylinder. Finally, the graph in Figure 5.28 compares the average maximum fitness values between the four approaches: benchmark, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations.

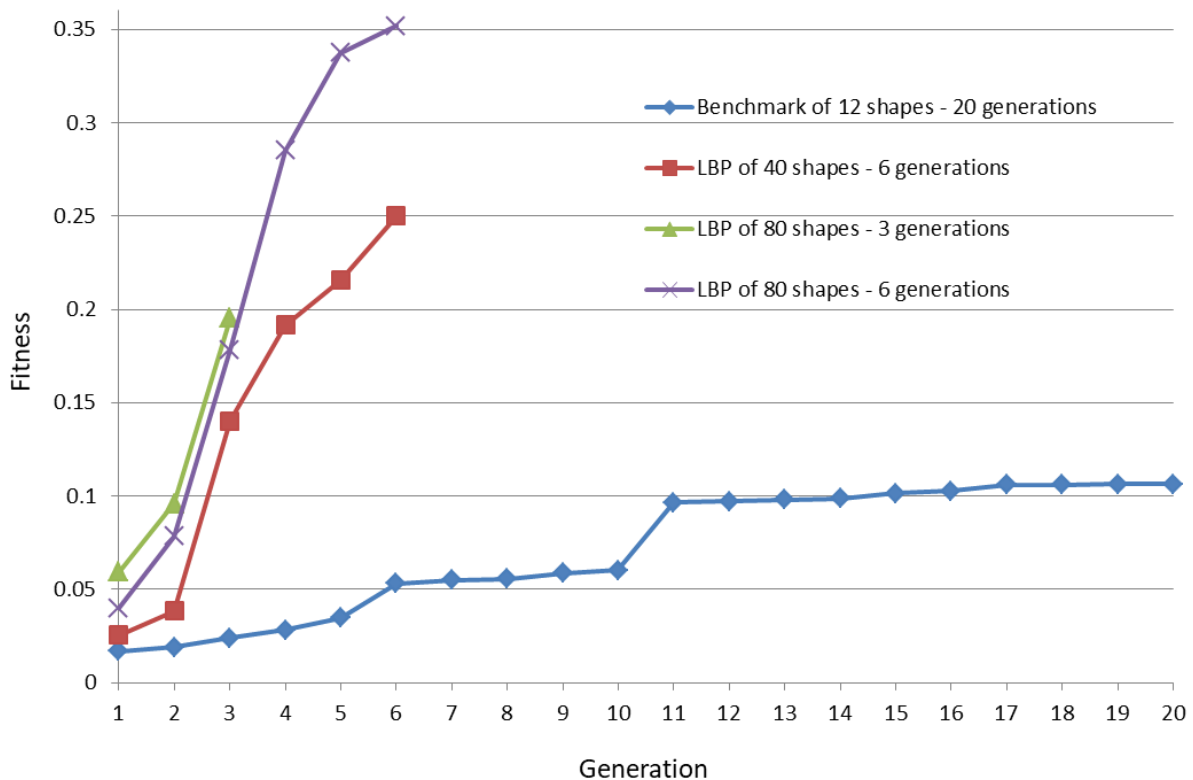


Figure 5.28. Comparison of the fitness of the four approaches: benchmark, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations.

It is not surprising that larger populations perform better, not only because more shapes are being tested in total, but because GA's always perform better with larger populations. A fairer comparison of the four approaches that represents a comparable number of evaluated shapes is shown in Figure 5.29. The starting points of all four approaches are as expected – the LBP of 80 shapes approach has better fitness than the benchmark. After around 240 shapes have been evaluated, the fitness of the LBP of 80 and 40 shapes approach is still increasing whereas the fitness of the benchmark is levelling. After 472 shapes have been evaluated, the fitness of the benchmark is never going to be as high as the fitness of the LBP approach (90% match with the GG) because it converged at a lower fitness than that of the LBP approach. The results confirm the widely-known properties of GA that large populations work better. The LBP of 80 shapes approach is better than the benchmark.

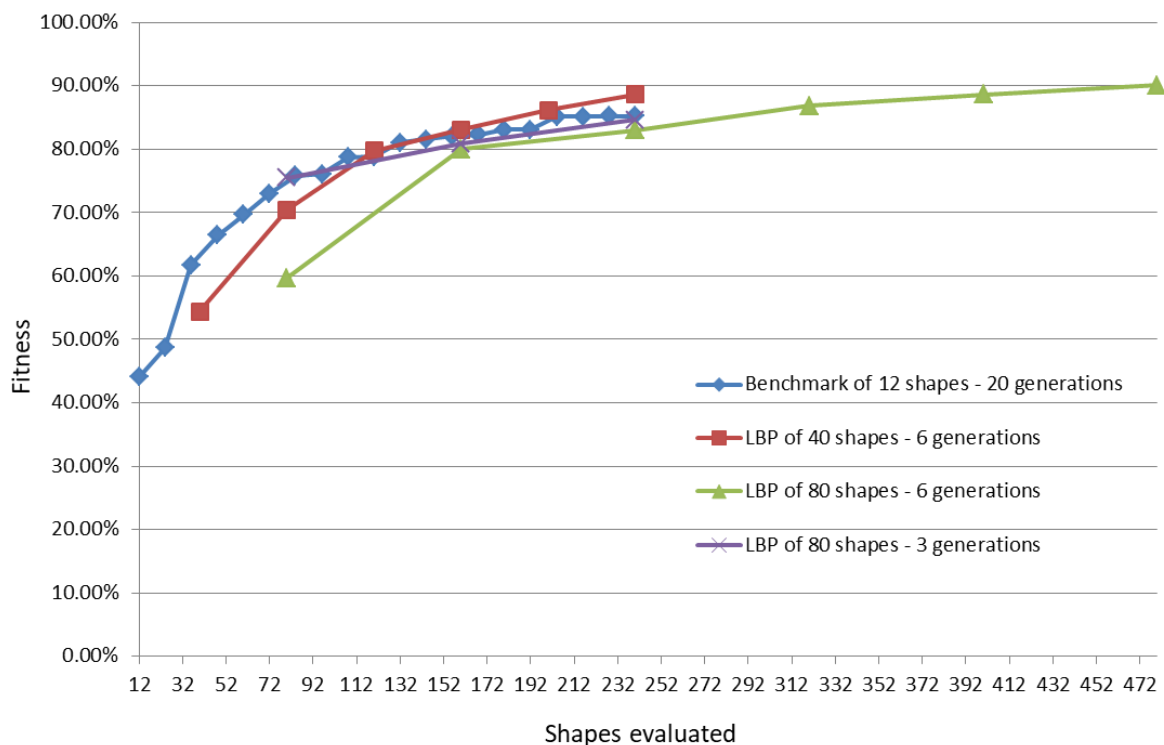


Figure 5.29. Comparison of the fitness of the four approaches: benchmark, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations. All four approaches represent a comparable number of evaluated shapes.

Not only fitness but the variety of the forms is also important when investigating the interaction strategies that best complement the evolutionary design process. Therefore, twelve objective tests using three starting populations and four GGs have measured the variety of the forms by calculating the standard deviation of the fitness values of the twelve fittest forms in each generation.

The results are shown in Figure 5.30, where in SUI approach the variety of the forms is smaller than the variety of the forms in the LBP approaches. The biggest variety is achieved by the LBP of 80 shapes-6 generations approach because it generates more forms (more

options) than the other approaches in each generation. However, after the fifth generation the variety drops as the forms start converging.

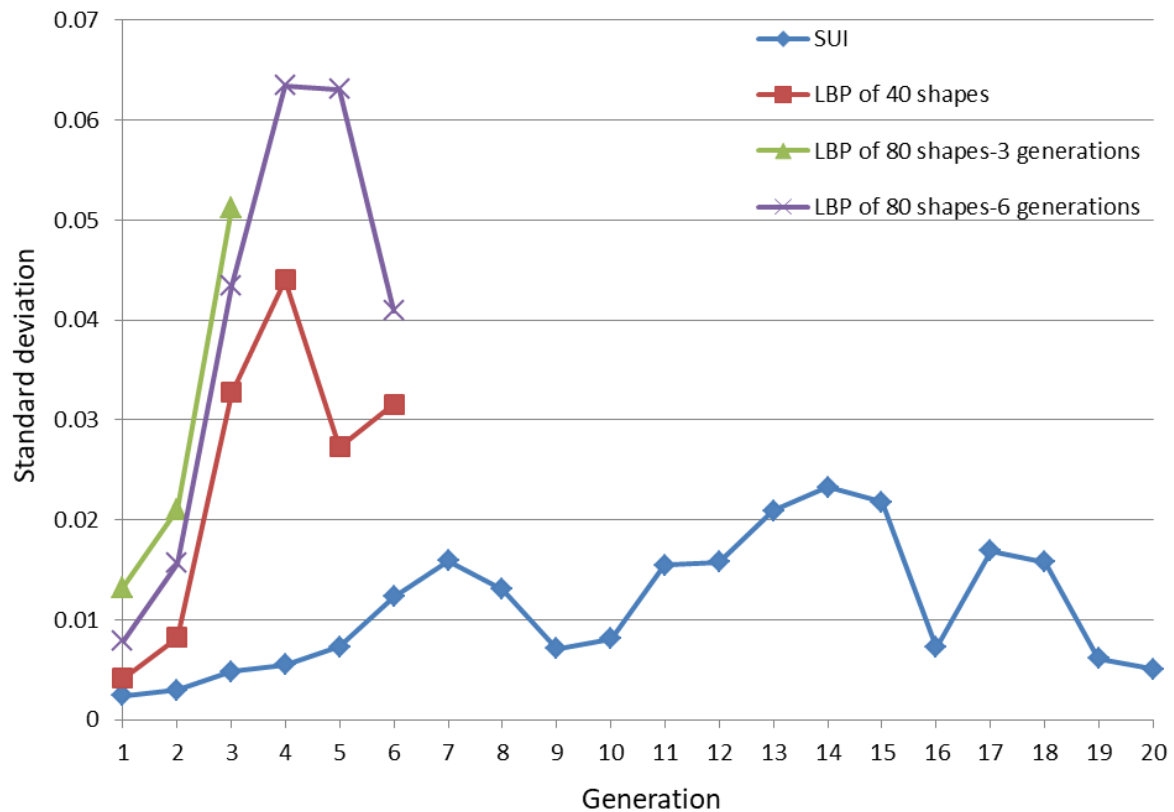


Figure 5.30. Comparison of the variety of shapes of the four approaches: SUI, LBP of 40 shapes, LBP of 80 shapes-6 generations and LBP of 80 shapes-3 generations.

The next step in the research, described in Chapter 6, was to combine GG with user-selection; to investigate and develop the human-machine interaction strategies which best complement the evolutionary design process.

5.3 User Interaction Strategies Conclusions

Investigation of several population sizes were described, which examine how best to use the GG capabilities to help users in evolutionary shape searching. Objective and fitness values based on shape comparison have been produced.

The experimental results imply that the LBP of 80 shapes approach has the greatest benefit as it produces significantly better fitness than the benchmark. The variety of the forms is also important to investigate the interaction strategies that best complement the evolutionary design process. Standard deviation is used as a crude proxy for shape variety.

The LBP of 80 shapes approach appears to produce a larger variety of shapes due to the big amount of forms (80) in each population.

However, the results were not strong enough to discard the benchmark as a user interaction strategy. The initial experimental results of this research shows the drawback of the automated shape matching programme – GG with more complex shapes are not successful in guiding the evolution without user intervention. It may be a problem of the representation of the system (GA, PN or scaling representation). Therefore, both the benchmark and the LBP approach needed further consideration with user interaction.

GGs that are represented with a single geometric primitive or with two geometric primitives but are not complex shapes, work well, produce excellent results, both accurate and efficient. On the other hand, GGs that are represented with two geometric primitives but are more complex shapes, do not work well; exhibit a lower match with the forms. However, this is partly overcome with the re-introduction of user intervention.

6 Participant Testing

6.1 Introduction to participant testing

After the objective testing, where experiments with both standard and larger populations were conducted, this chapter introduces participant development testing for evolutionary shape searching.

Due to the large effect of the starting population, and because small populations are well known to be detrimental to GA performance, a method of working with 'larger populations' was developed which uses Shape Matching with a GG to present a subset of fitter shapes to the user. In investigating user interaction strategies, a development comparison of LBP with SUI is carried out that aims to provide better fitness and variety.

To evaluate the efficacy of the system, the participants have tested four different versions of the algorithm: EvoShape, LBP of 80 shapes, LBP of 40 shapes and SUI.

In this chapter, the results of the participant testing are presented and evaluated. The participants' feedback statements are reproduced in Appendix: Participants' Feedback I).

6.2 Geometric Guides with Large Background Populations

The LBP approach has been tested with two groups of five different participants who do not differ significantly in expertise.

The participants, that were asked to evolve shapes that could inspire the form of a sofa, a watch and a hand-held 3D mouse, were students in their third year of students in an Engineering Department or a Design School. These students were familiar with CAD systems and experienced in design but had little NX CAD system experience. The participants, that were asked to evolve shapes that could inspire the form of a building and a guitar, were both Lecturers in Engineering. Some of these products were selected by the participants from a list of products that were provided by the author, while some others (etc. building, guitar) were suggested by the participants as a personal choice. They were asked to test three different versions of the system:

- EvoShape
- LBP of 80 shapes
- LBP of 40 shapes

For the last two versions they were asked to create a GG just before the beginning of the process. Each of the participants was also asked to test the three versions of the system in a different order. The order can be an important step of the process as the participants' fatigue is increasing during the testing and it can affect the results. Some of them were asked to test EvoShape first and then LBP of 80 and 40 shapes, where some others were asked to do the opposite.

During an interview held after the trials, all the participants stated that although the EvoShape process has a large variety of shapes and gives new ideas of useful shapes, by the end of it, the participants will lose the shapes they like because there are no GGs to guide the process (Appendix I), Participant 4, Comment 3 (P 4.3), P 4.4, P 7.3, P 10.4, P 10.5). However, this is not necessarily because of the lack of a GG – it could just be a property of the system as a whole.

In Table 6.1, while participants had been searching for shapes that could inspire the form of a sofa and a hand-held 3D mouse, in the first generation EvoShape gave 16.6% of useful shapes, in the second generation this percentage increased to 25%, while in the third generation the percentage reduced dramatically to 0%. The participants stopped the process in the third generation because they had already lost the shapes they liked. In Table 6.2, while participants had been searching for shapes that could inspire the form of a watch, a guitar and a building, in the first generation EvoShape gave 25% of useful shapes, in the second generation this percentage increased to 33.3%, in the third and fourth generation the percentage reduced to 16.6%, and in the fifth generation EvoShape again had no useful shapes that could inspire the form of a watch, guitar or a building. The participants stopped the process in the fifth generation because they had already lost the shapes they liked. However, some of them mentioned that this approach can be useful when the participants have a general idea on what they want to create.

Since all of the participants had a specific bounding volume in their mind, they found LBP of 80 and 40 shapes more useful – it guides the process and gets rid of shapes that are not similar with the bounding volume after a shape comparison. Although the shapes usually start converging in generation four, it is still enough to give useful ideas of shapes that the participants would not have thought of; in Table 6.3, while participants had been searching for shapes that could inspire the form of a sofa, watch, building, guitar or hand-held 3D mouse, in the first generation GG gave 25%-33.3% of useful shapes, the same as in the second generation; in the third, fourth and fifth generations the percentage reduced to an average of 16.6% of useful shapes, and in the last generation the participants are satisfied with the results – at least one of the shapes gives a useful idea for each of the five products. Convergence happens when all the shapes of a population start looking very similar to each other and therefore the variety of shapes is decreased.

GGs help participants to find the shape easier; ‘the concept has evolved nicer and the process is not random any more’ (P 6.6). The LBP approach is finally useful; it is a shape exploration that gives new ideas of useful shapes. Although this participant stated that EvoShape is random, it is not – it probably appeared random to this participant because they are not experienced with EvoShape.

However, there was one participant who mentioned that a combination of both EvoShape and LBP could give an option to the participants to decide on which generations during the process they would like to use a GG (i.e. it might be useful to stop the convergence and increase the variety of shapes by turning GG off at around generation four). Figure 6.1 shows the participants’ preferences analytically; four out of five participants said that LBP is more useful and has value. One participant said that a combination of both EvoShape and LBP could also be useful. None of the participants preferred EvoShape.

Table 6.1. EvoShape: While two participants with moderate design and CAD experience had been searching for shapes that could inspire the form of a sofa and a hand-held 3D mouse; the participants stopped the process in the third generation because they had already lost the shapes they liked.

Generation	% Useful Shapes of Sofa and 3D Mouse using EvoShape
1	16,6%
2	25%
3	0%

Table 6.2. EvoShape: While two participants with good CAD experience and one with moderate CAD experience had been searching for shapes that could inspire the form of a watch, a guitar and a building; the participants stopped the process in the fifth generation because they had already lost the shapes they liked.

Generation	% Useful Shapes of Watch, Guitar and Building using EvoShape
1	25%
2	33.3%
3	16.6%
4	16.6%
5	0%

Table 6.3. LBP of 80 and 40 shapes: in the last generation the participants are satisfied with the results – at least one of the shapes gives a useful idea of each of the five products.

Generation	% Useful Shapes of products using GG's
1	25% - 33.3%
2	25% - 33.3%
3	16.6% - 25%
4	16.6%
5	8.3% - 16.6%
6	The participants are satisfied with at least one shape that the sixth generation produces

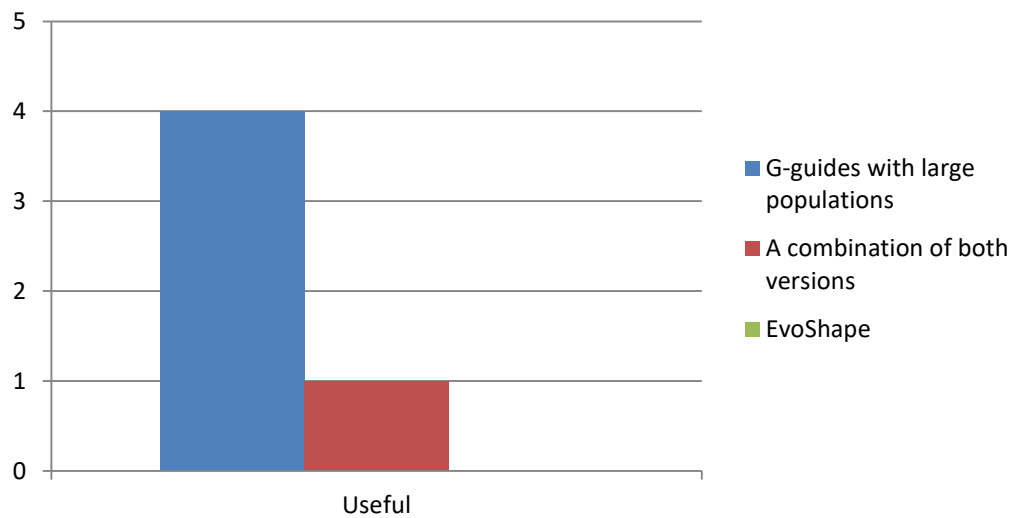


Figure 6.1. Participants' preferences in evolutionary shape searching.

It is also important to mention that both LBP of 80 and LBP of 40 shapes have very similar variety of shapes but the LBP of 80 shapes approach tends to create at least one shape that looks more like the bounding volume and gives better ideas of useful shapes, as shown in Figure 6.2. The Figure 6.3 shows an example of three different sofas using the three system approaches, how much the final products look like the bounding volume created by the participants at the beginning of the process when using LBP, and how the LBP of 80 shapes approach tends to create shapes that give better ideas of useful shapes than the other approaches do.

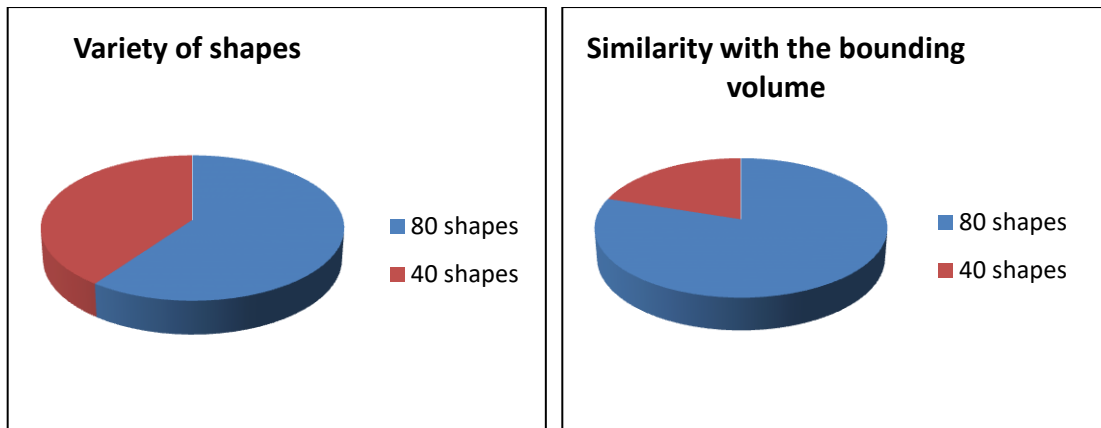


Figure 6.2. Left: Three out of five participants said that LBP of 80 shapes has slightly bigger variety. Right: Four out of five participants said that LBP of 80 shapes can create at least one shape that has bigger similarity with the bounding volume.

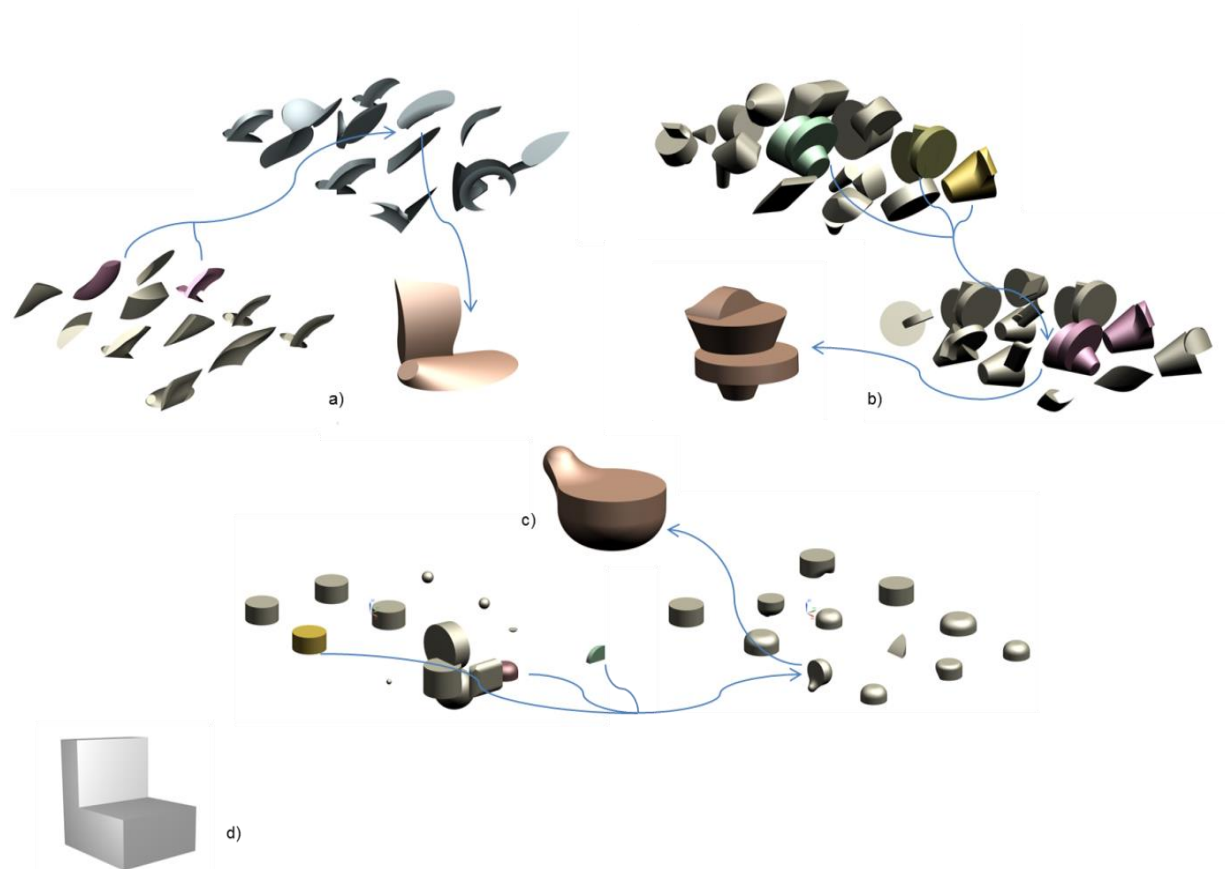


Figure 6.3. This figure shows three examples of sofas: a) created by using LBP of 80 shapes, b) created by using LBP of 40 shapes, and c) created by using EvoShape. d) Shows the bounding volume created at the beginning of the process by the participant when using LBP approach.

Figure 6.4, Figure 6.5, Figure 6.6, Figure 6.7, Figure 6.8, Figure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12 show the final products of a sofa, watch, guitar and hand-held 3D mouse respectively as well as the ancestors' generations of these shapes when using the LBP approach. The LBP Functions Map of the software is presented in Appendix J).

Generation 1

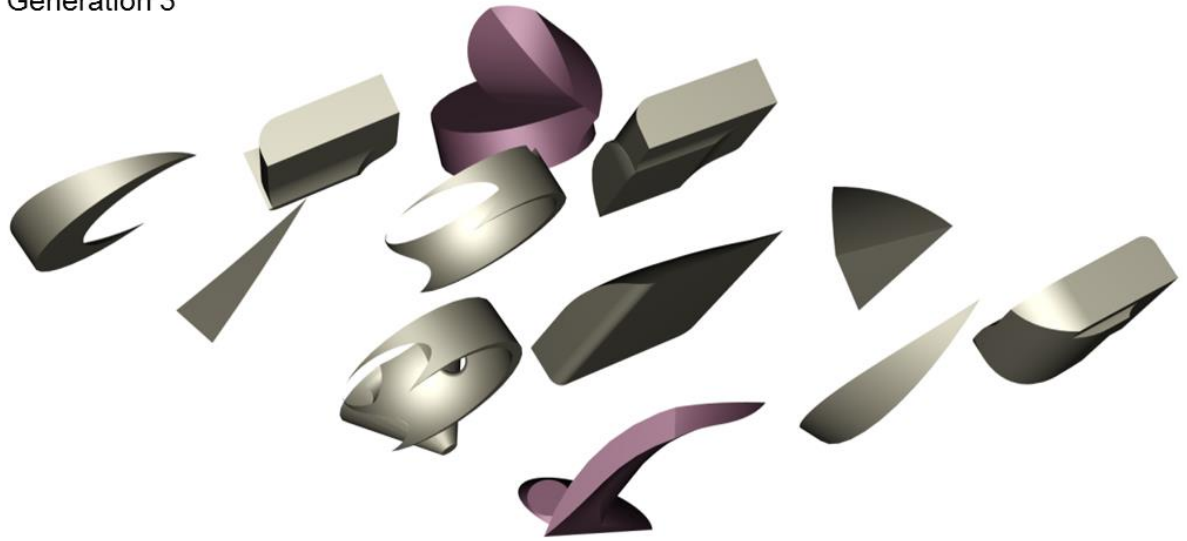


Generation 2



Figure 6.4. LBP of 80 shapes: In both generations (1 and 2) the participant chose three shapes that they found useful and could inspire the form of a sofa. There is a variety of shapes but none of them can provide good ideas for a sofa yet.

Generation 3



Generation 4



Figure 6.5. LBP of 80 shapes: In both generations (3 and 4) the participant chose two shapes that they found useful and could inspire the form of a sofa. There is still a variety of shapes and some of them could already give ideas for a sofa.

Generation 5



Generation 6

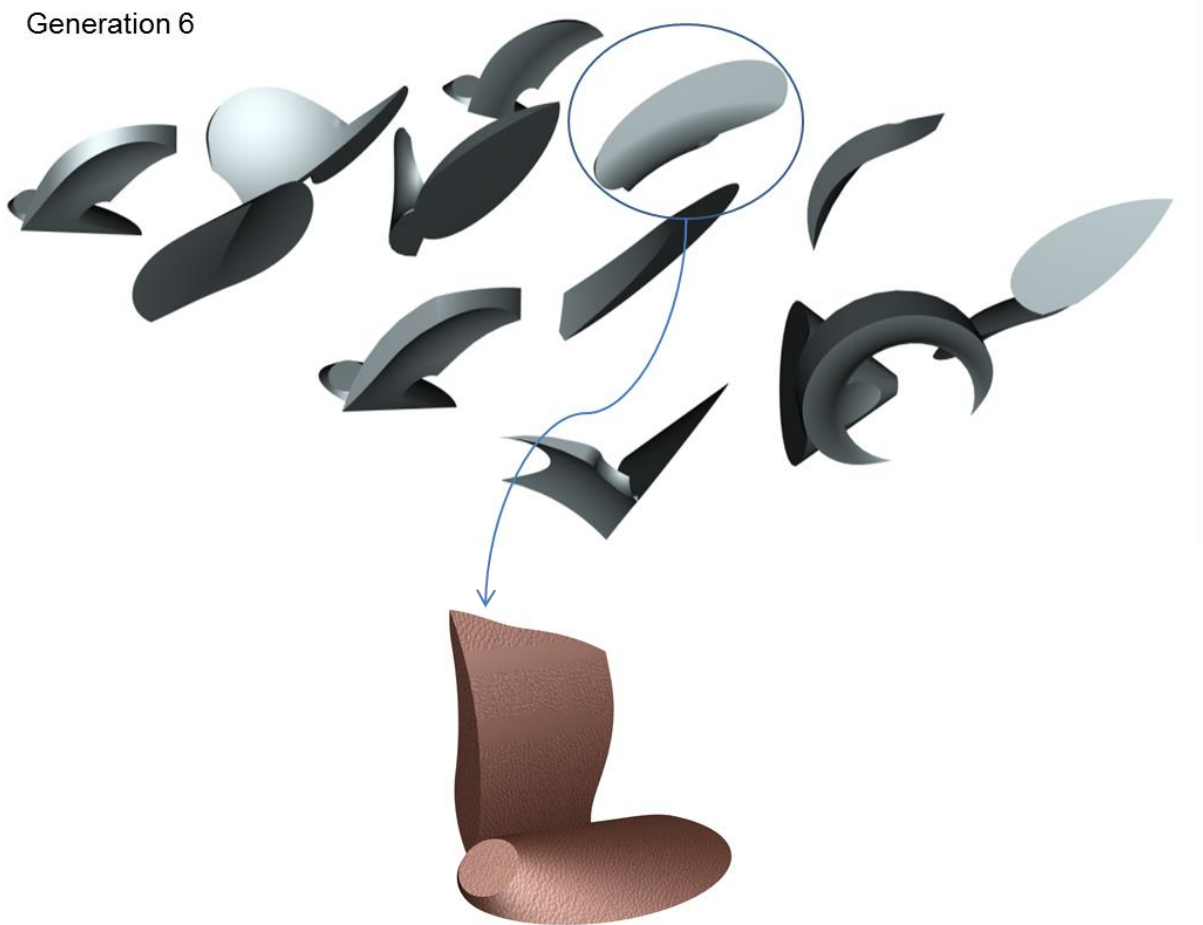
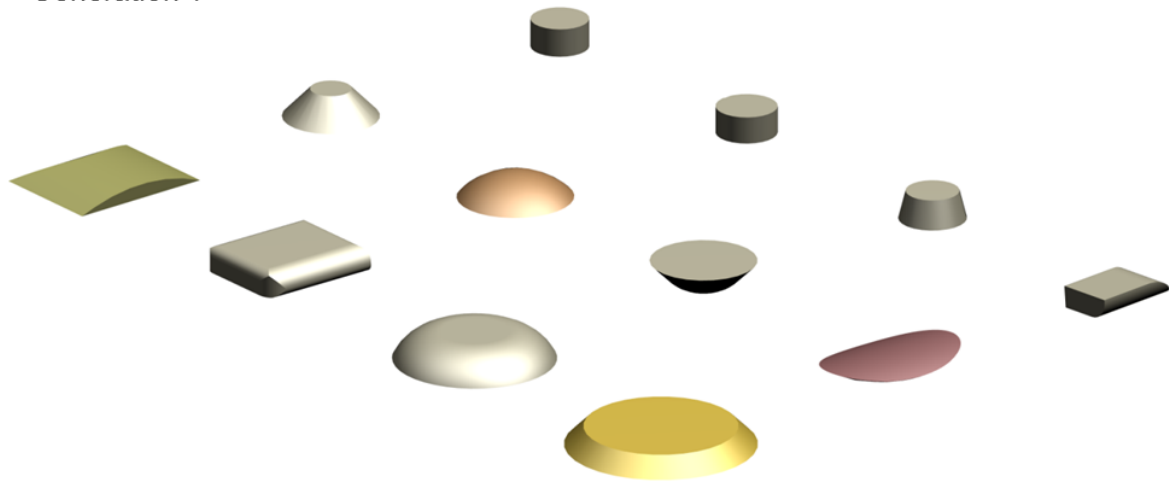


Figure 6.6. LBP of 80 shapes: In generation 5 the participant chose two shapes that they found useful and could inspire the form of a sofa. Although the shapes have started to converge, there is still a variety and some of them can still give ideas for a sofa. In generation 6 there is at least one shape that the participant can choose to inspire the form of a sofa.

Generation 1



Generation 2

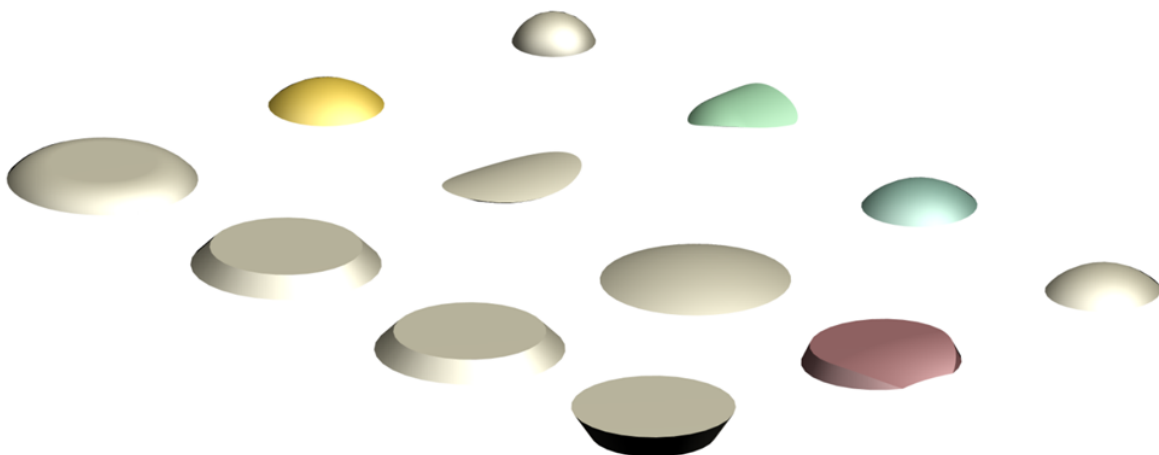
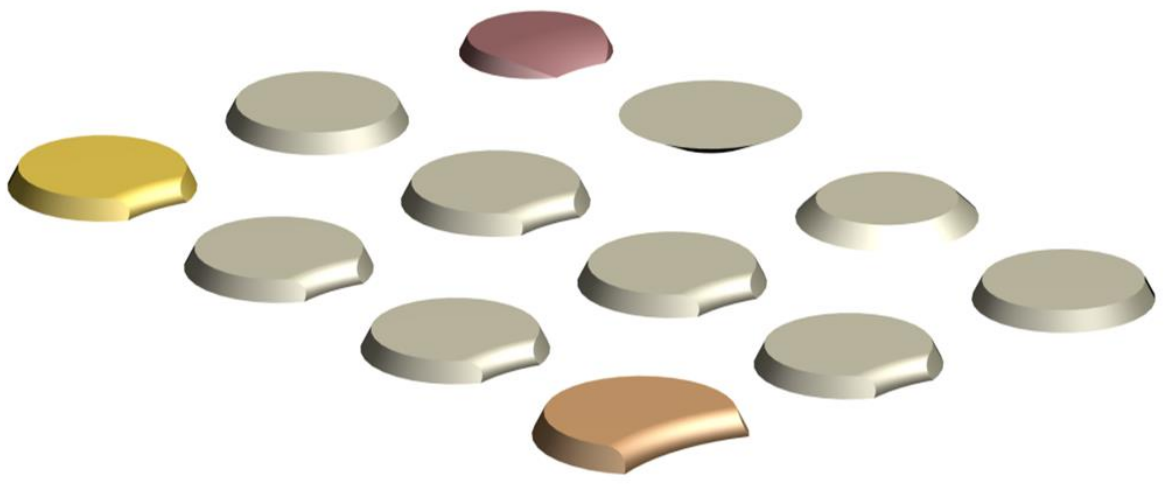


Figure 6.7. LBP of 80 shapes: In both generations (3 and 4) the participant chose four shapes that they found useful and could inspire the form of a watch. There is a variety of shapes and some of them can already give ideas for a watch.

Generation 3



Generation 4

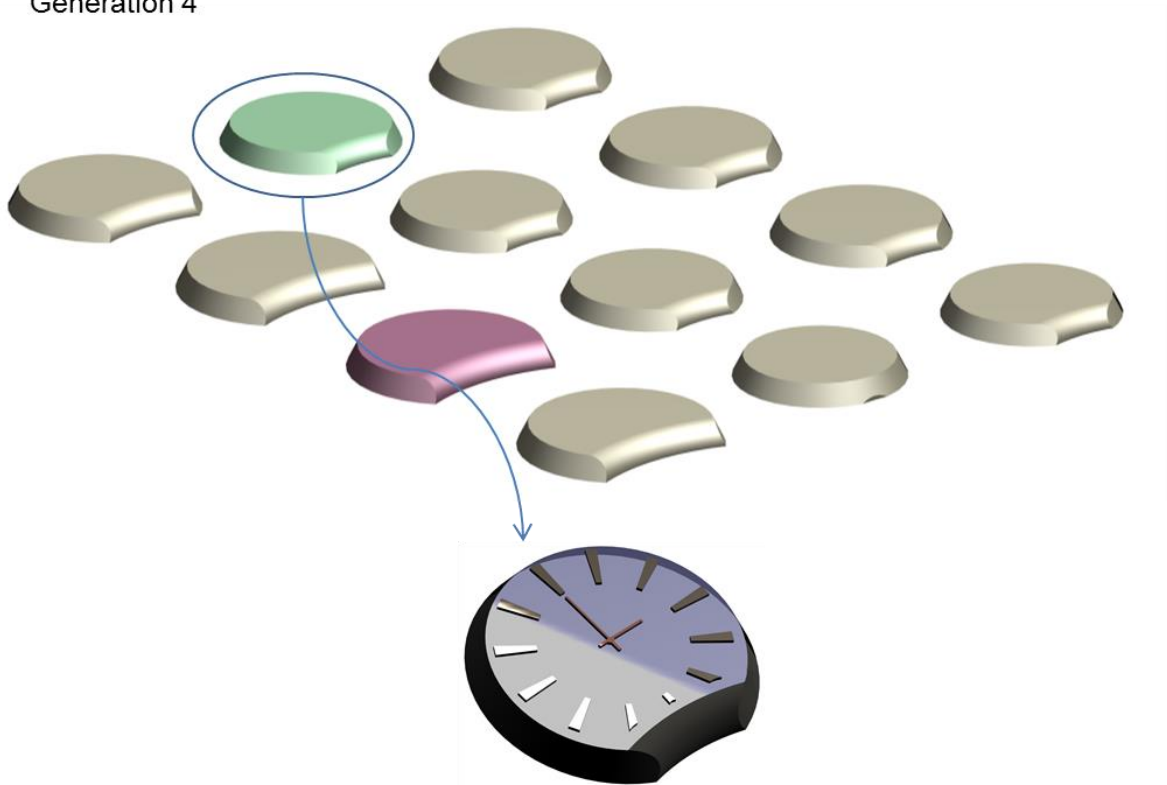
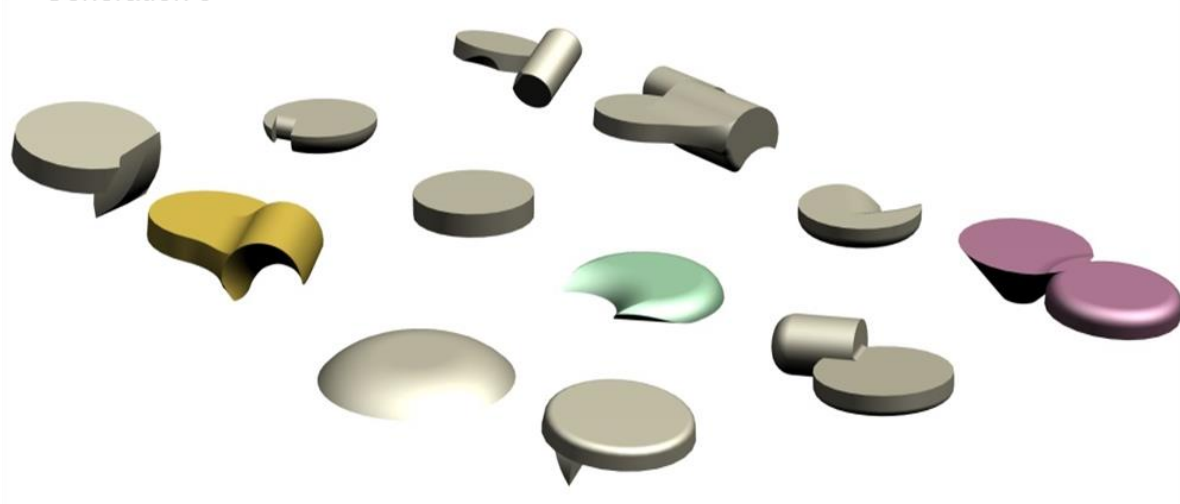


Figure 6.8. LBP of 80 shapes: In generations 3 and 4 the participant chose three and two shapes respectively that they found useful and could inspire the form of a watch. There is a variety of shapes and some of them give useful ideas for a watch.

Generation 3



Generation 4

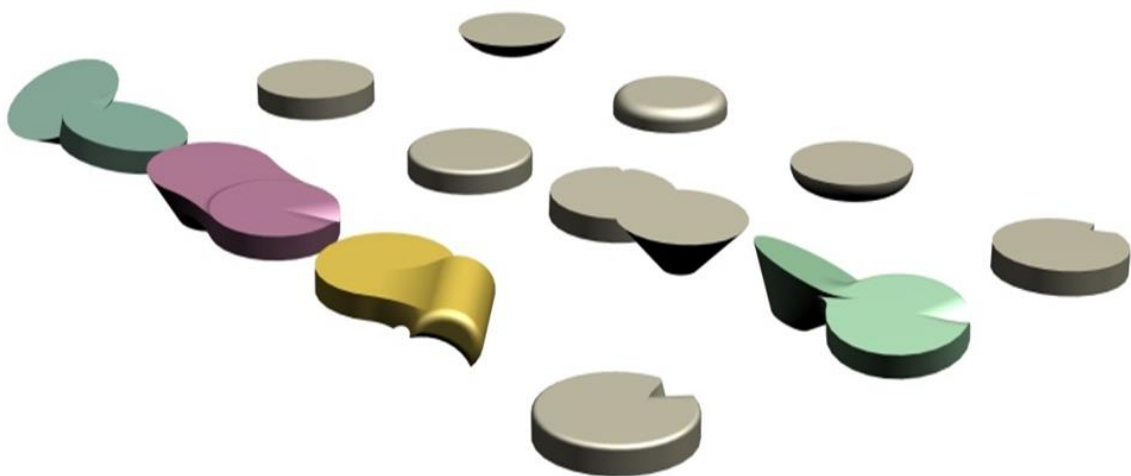


Figure 6.9. LBP of 80 shapes: In generations 3 and 4 the participant chose three and four shapes respectively that they found useful and could inspire the form of a guitar. There is a variety of shapes and some of them give useful ideas for a guitar.

Generation 5



Generation 6

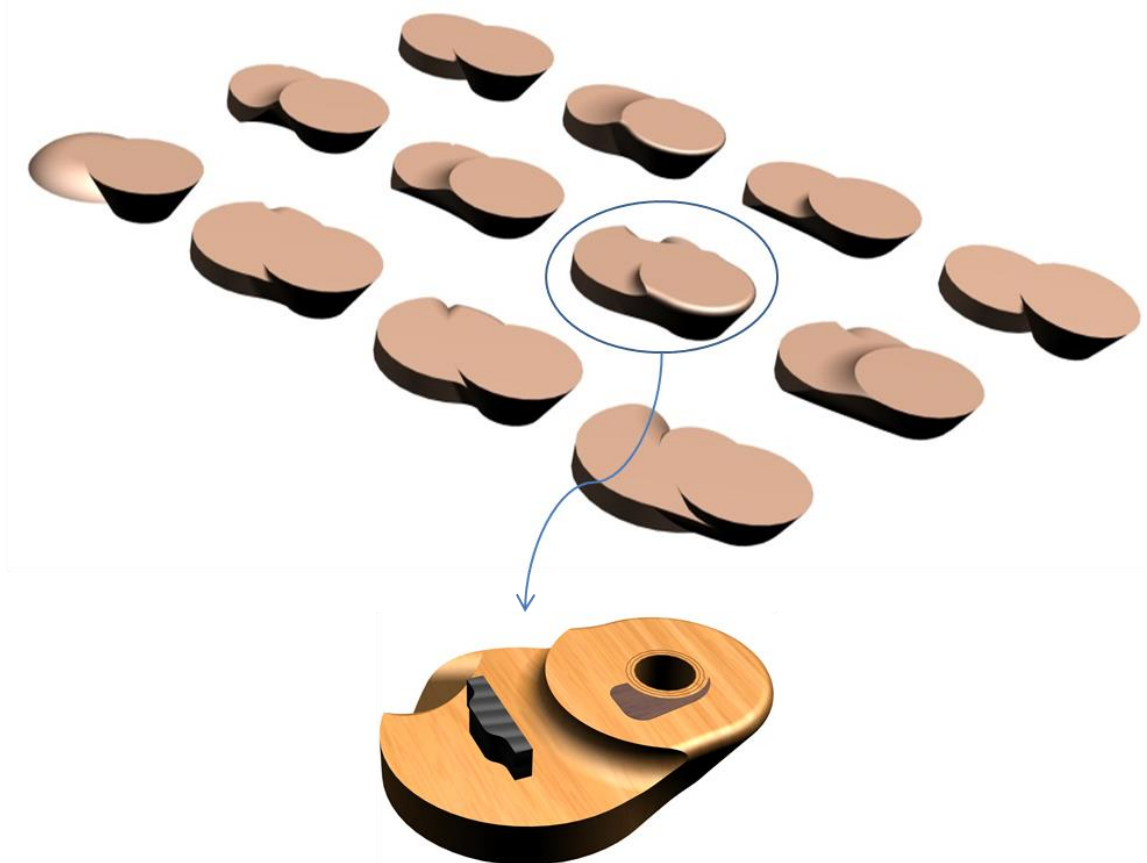
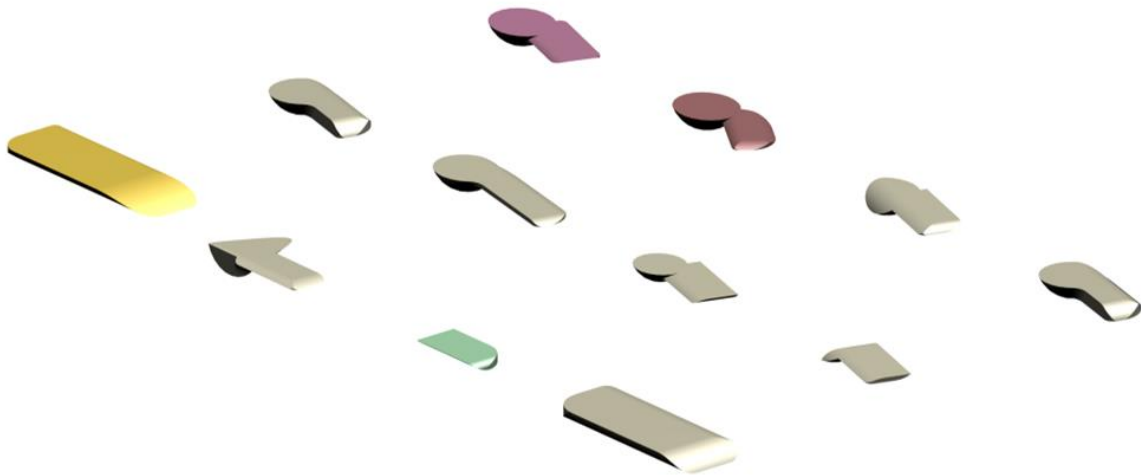


Figure 6.10. LBP of 80 shapes: In generation 5 the participant chose three shapes that they found useful and could inspire the form of a guitar. Although the shapes have started to converge and the variety has decreased a lot, some of the shapes can give ideas for a guitar. In generation 6 there is at least one shape that the participant can choose to inspire the form of a guitar.

Generation 3



Generation 4

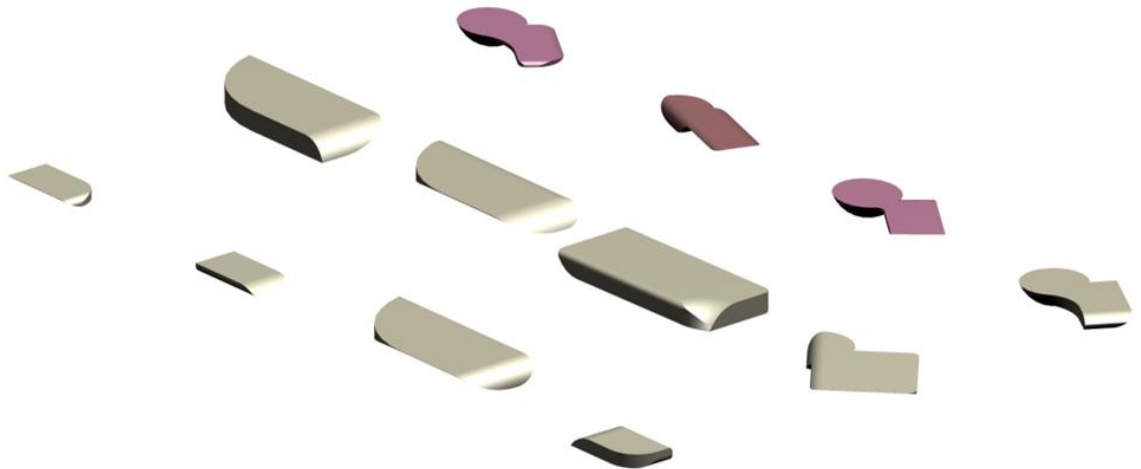
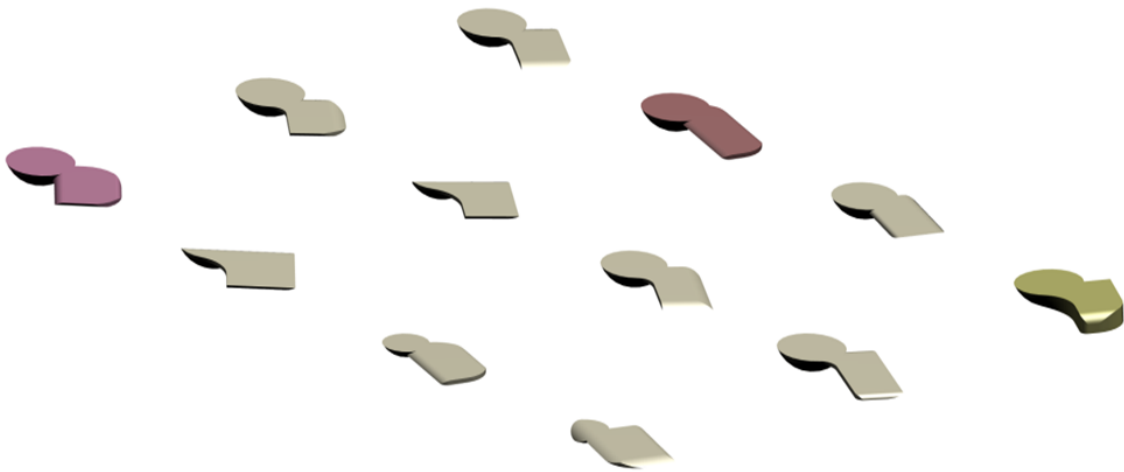


Figure 6.11. LBP of 80 shapes: In generations 3 and 4 the participant chose four and three shapes respectively that they found useful and could inspire the form of a hand-held 3D mouse. There is a variety of shapes and some of them give useful ideas for a hand-held 3D mouse.

Generation 5



Generation 6

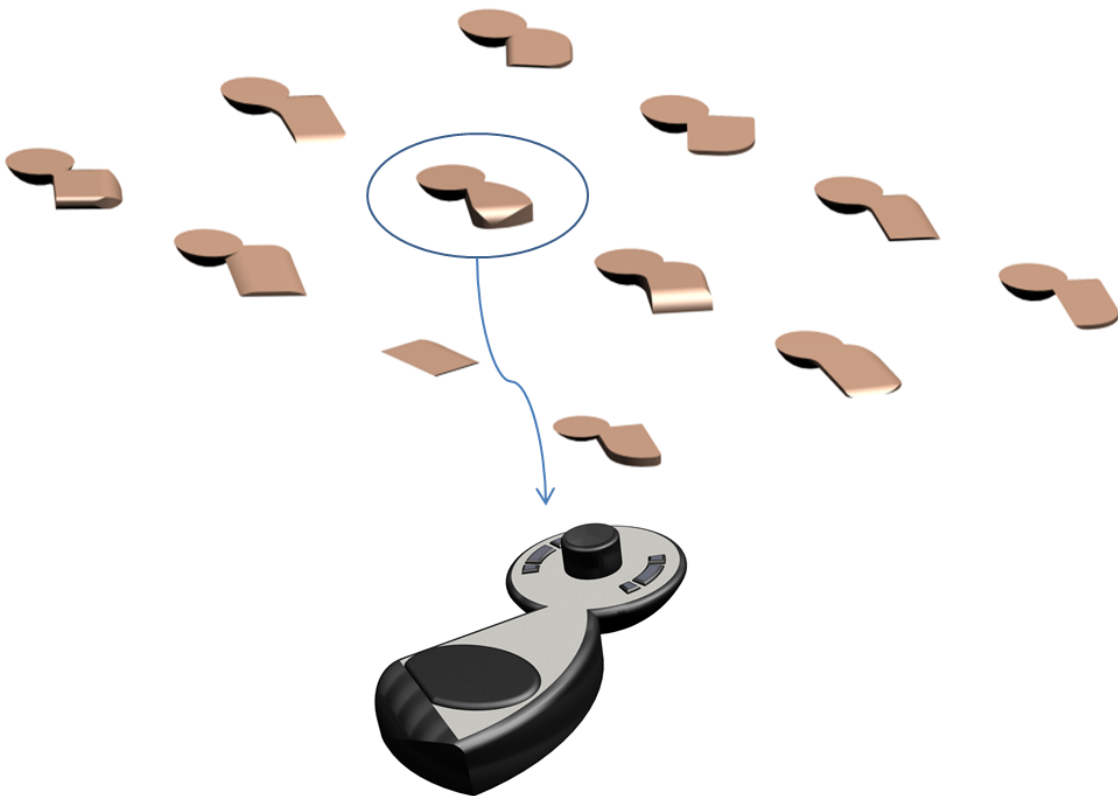


Figure 6.12. LBP of 80 shapes: In generation 5 the participant chose three shapes that they found useful and could inspire the form of a hand-held 3D mouse. Although the shapes have started to converge and the variety has decreased a lot, some of the shapes can give ideas for a hand-held 3D mouse. In generation 6 there is at least one shape that the participant can choose to inspire the form of a hand-held 3D mouse.

6.3 Geometric Guides with Sporadic User Interaction

SUI has been tested by two of these five different participants who do not differ significantly in expertise and belong to the same participant group.

These participants had been searching for shapes that could inspire the forms of a sofa and a guitar. They were asked to test three different versions of SUI that have different participant interaction frequency. These frequencies have been chosen by the author intuitively – the bigger the intervals between the generations the more influence the GG will have and the inheritance is more likely to be lost. The three different frequencies are:

- Interaction in generations 10, 16, 22, 28, 34
- Interaction in generations 10, 20, 28, 34, 38
- Interaction in generations 10, 14, 20, 28, 39

In all three versions, the participant interaction starts in generation 10 (starting point). The starting point is where the participant is starting to interact with the system and happens just before the shapes start converging. This starting point is proven through experimental results of nine tests on three different starting populations of shapes using three different GGs (Traffic Cone, Roof, and Cylinder), as shown in Figure 6.13. To avoid a high interaction effort, the participants did not start interacting at the beginning of the process or in early generations; they started interacting before the shapes start converging where there is still a variety of shapes.

For all three versions, the participants were asked to create a bounding volume just before the beginning of the process. Each of the participants was also asked to test the three versions of the SUI approach in a different order, as before.

Although the shapes usually start converging after generation 10, the participants stated that the first version is still enough to give useful ideas of shapes that they would not have thought of (Table 6.4); while participants had been searching for shapes that could inspire the form of a sofa, in the tenth generation the first version gave 8.3% of useful shapes, in the 16th, 22nd and 28th generation this percentage increased to 25%, and in the 34th generation this percentage reduced to 16.6%. At the 40th generation the participants are satisfied with the results – at least one of the shapes gives a useful idea.

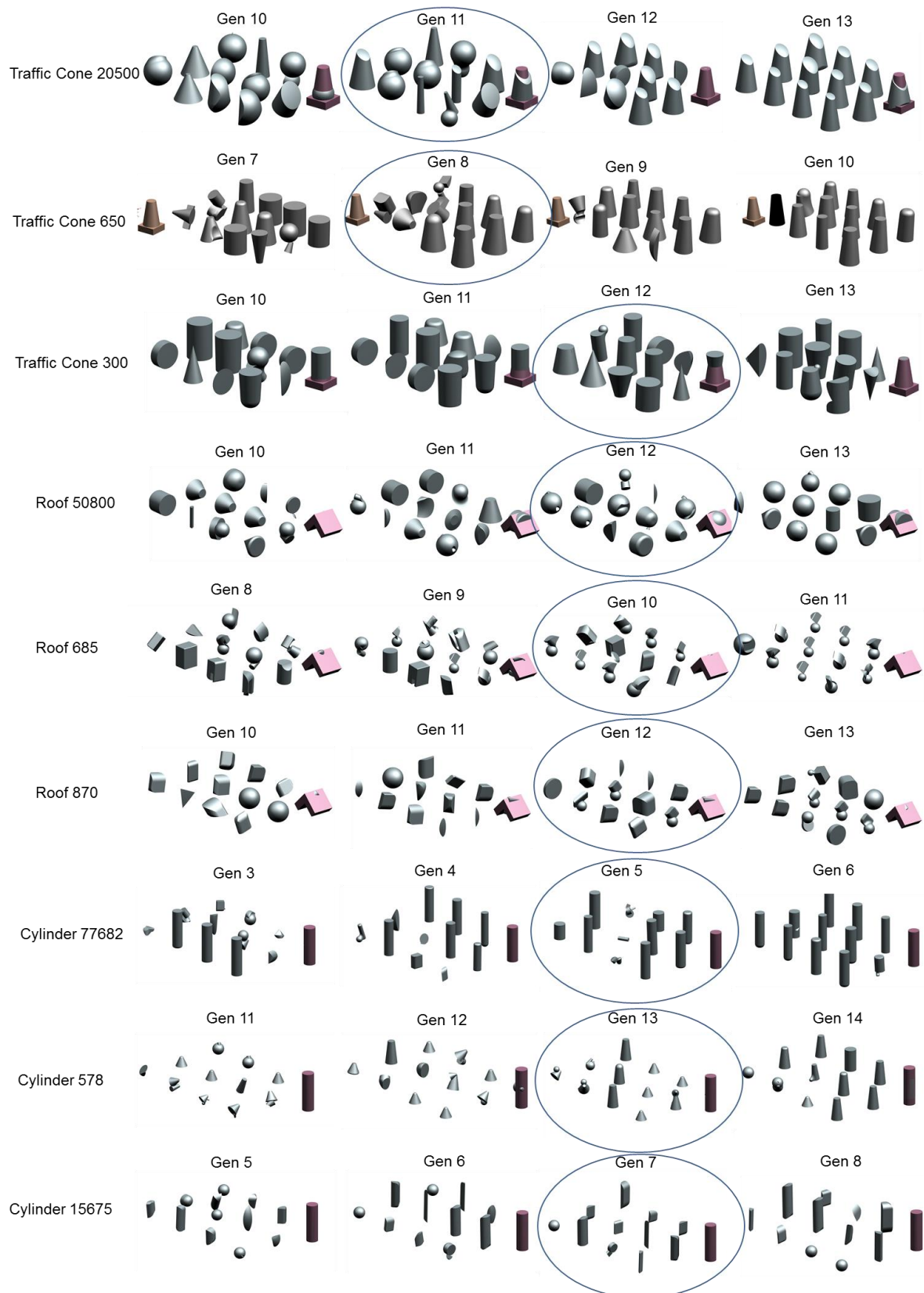


Figure 6.13. The 'marked generations' are just before the shapes start converging. The starting point 10 is the average of all of the nine different starting populations (three starting populations for each of the three GG).

Table 6.4. SUI first version: At the 40th generation the participants are satisfied with the results – at least one of the shapes gives a useful idea.

Generation	% Useful Shapes of Sofa using SUI Version 1
10	8.3%
16	25%
22	25%
28	25%
34	16.6%

In the first version of the SUI approach, the participants interact with the system in a sequence of generations where the difference between the consecutive terms is constant (every time, the difference between the generations is six), whereas in version two and three, this difference is increased and reduced respectively. According to the experimental results after testing the system with participants (Table 6.5), the version two and three of the SUI approach do not give as many useful shape ideas as the first version does; while participants had been searching for shapes that could inspire the form of a sofa, in the tenth generation the version two gave 8.3% of useful shapes, where in the 16th generation this percentage reduced dramatically to 0%. The participants stopped the process at the 22nd generation because they had already lost the shapes they liked. Almost similar results were given from version three where the participants stopped the process at the 28th generation.

Table 6.5. SUI version two and three.

Generation	% Useful Shapes of Sofa using SUI Version 2	% Useful Shapes of Sofa using SUI Version 3
10	8.3%	16.6%
16	0%	8.3%
22	-	0%
28	-	-

Therefore, according to the participants, all three versions of the SUI approach have very similar variety of shapes but only the first version gives useful ideas of shapes and tends to create a shape that looks like the bounding volume, as can be seen in Figure 6.14.

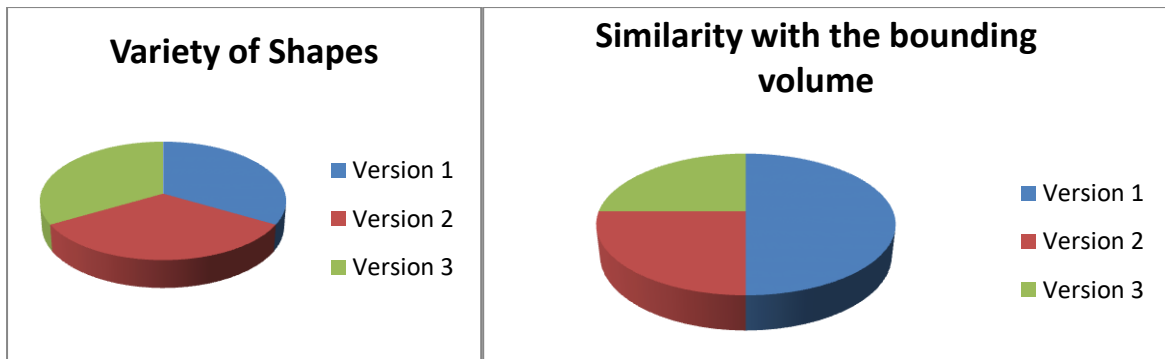


Figure 6.14. Left: All participants said that all versions of SUI approach have similar variety. Right: All participants said that only the first version can create shapes that are similar to the bounding volume.

However, all the participants stated that the SUI approach has neither better variety of shapes nor better useful ideas of shapes and similarity with the bounding box than the LBP approach does, as shown in Figure 6.15.

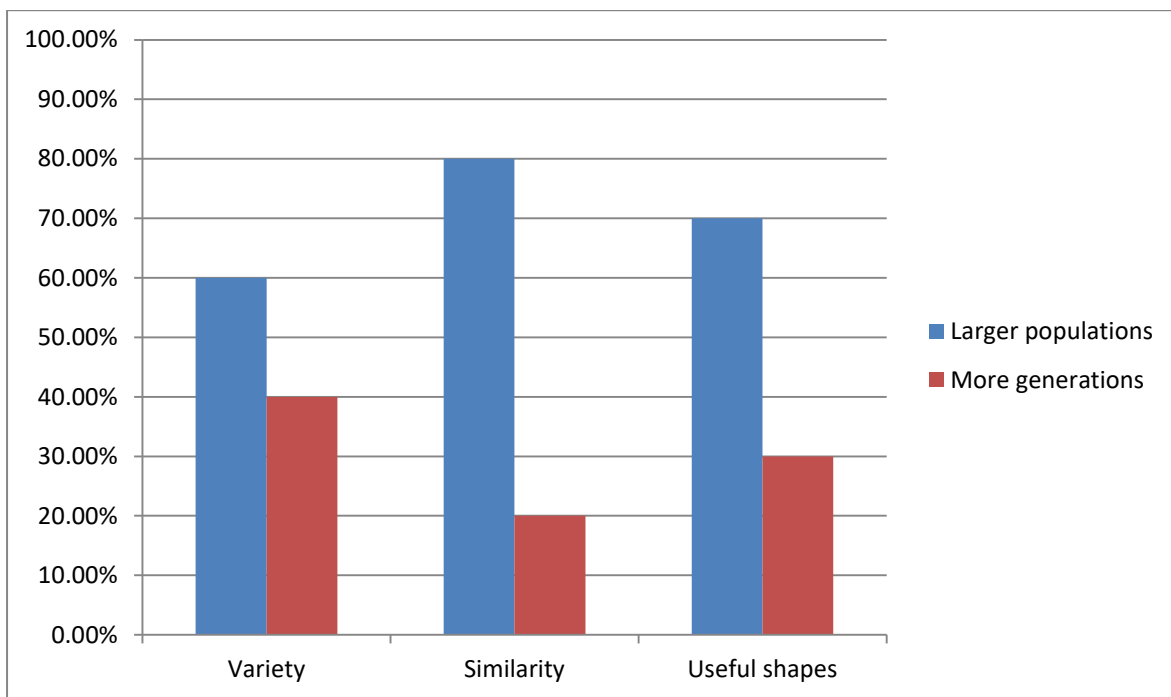


Figure 6.15. The variety in LBP approach is big enough to give useful ideas of shapes, and therefore to give shapes similar to the bounding volume.

The Figure 6.16 shows an example of three different sofas using the three system versions, how much the final products look like the bounding volume created by the participants at the beginning of the process when using the SUI approach, and how the SUI first version tends to create shapes that give better ideas of useful shapes than the other versions do.

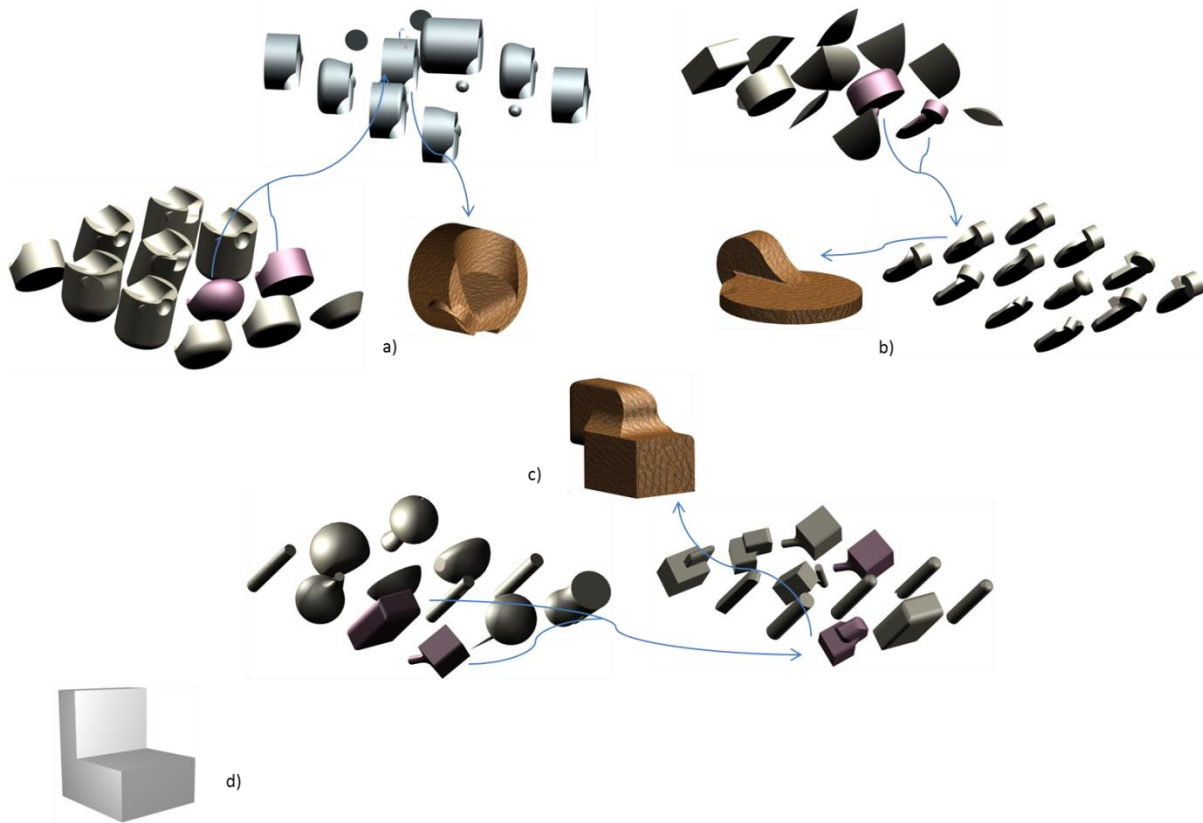
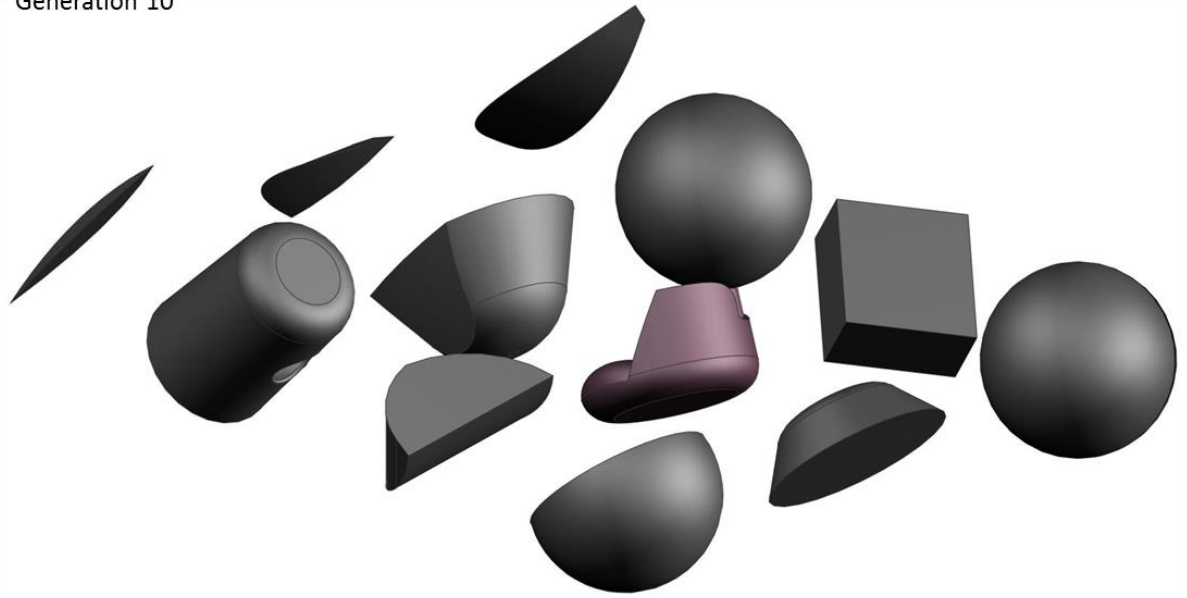


Figure 6.16. This figure shows three examples of sofas: a) created by using SUI first version, b) created by using SUI version two, and c) created by using SUI version three. d) Shows the bounding volume created at the beginning of the process by the participant when using SUI approach.

Figure 6.17, Figure 6.18, Figure 6.19, Figure 6.20, Figure 6.21 and Figure 6.22 show the final product of a sofa and a guitar as well as the ancestor generations of the products when using the SUI approach. The SUI Functions Map of the software is presented in Appendix K).

Generation 10



Generation 16

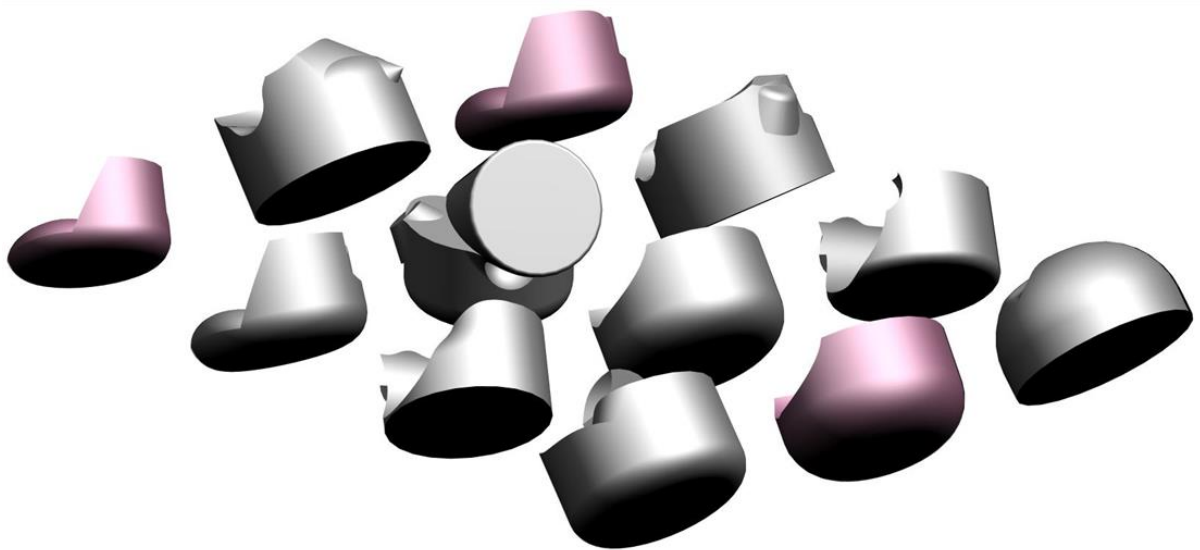
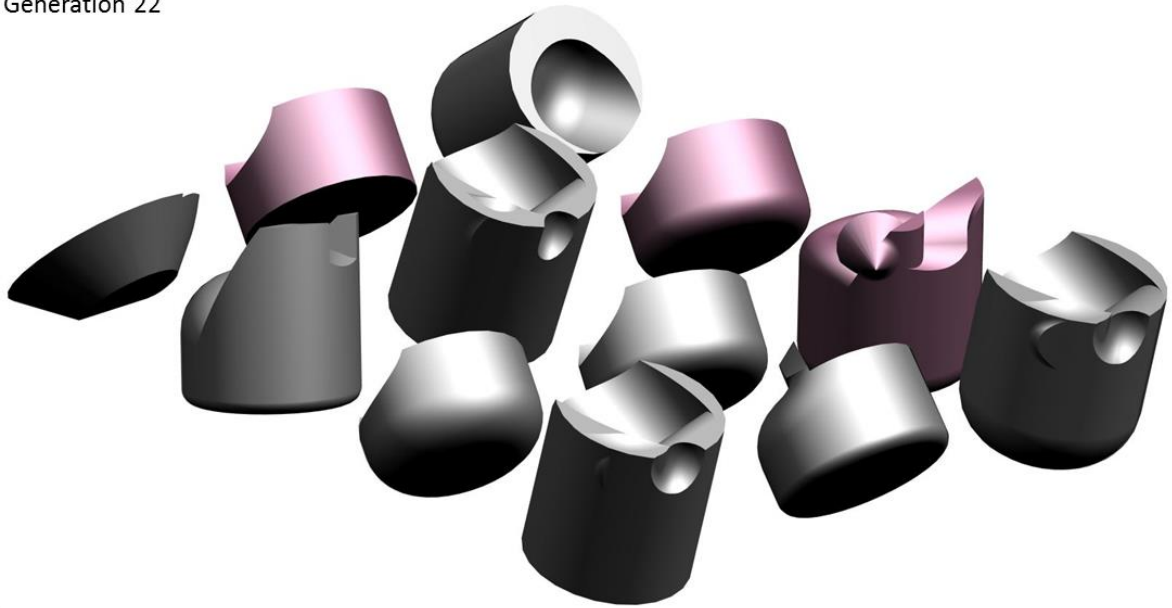


Figure 6.17. SUI: In generations 10 and 16, the participant chose one and three shapes respectively that they found useful and could inspire the form of a sofa. There is a variety of shapes but none of them can give good ideas for a sofa yet.

Generation 22



Generation 28

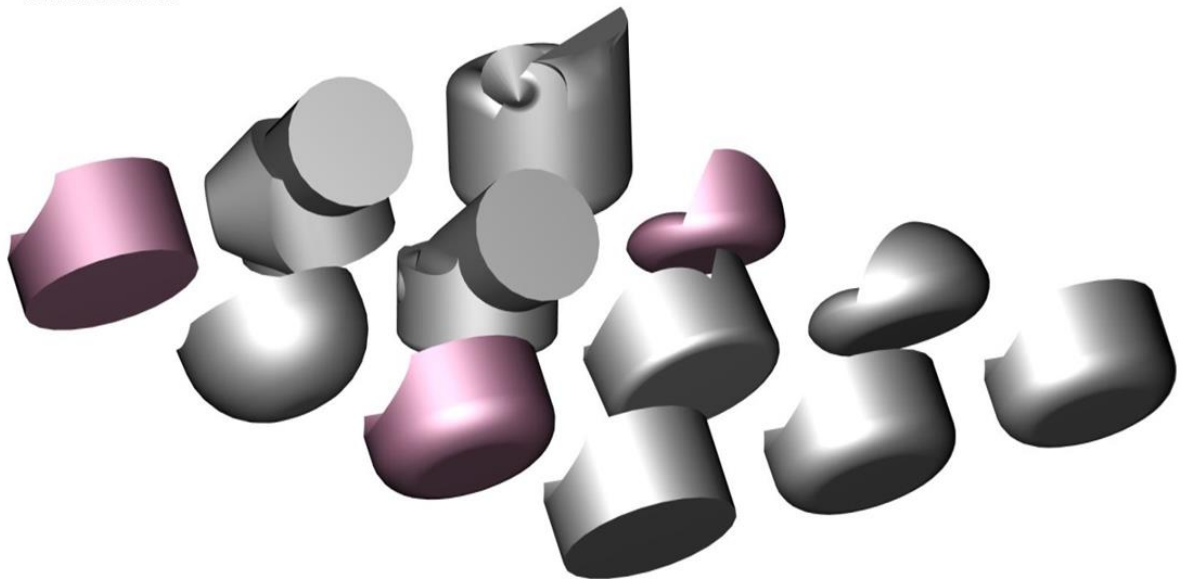
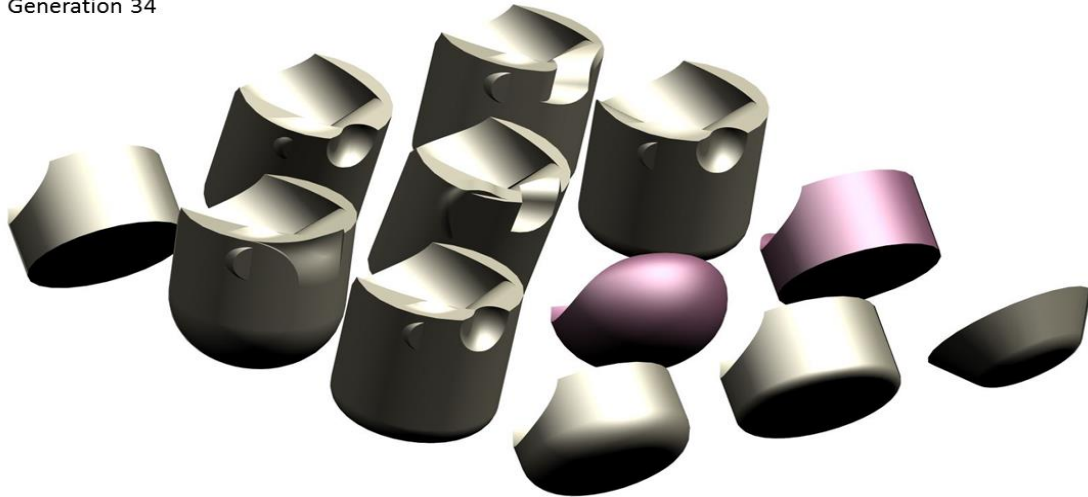


Figure 6.18. SUI: In both generations (22 and 28), the participant chose three shapes that they found useful and could inspire the form of a sofa. There is still a variety of shapes and some of them can already give ideas for a sofa.

Generation 34



Generation 40

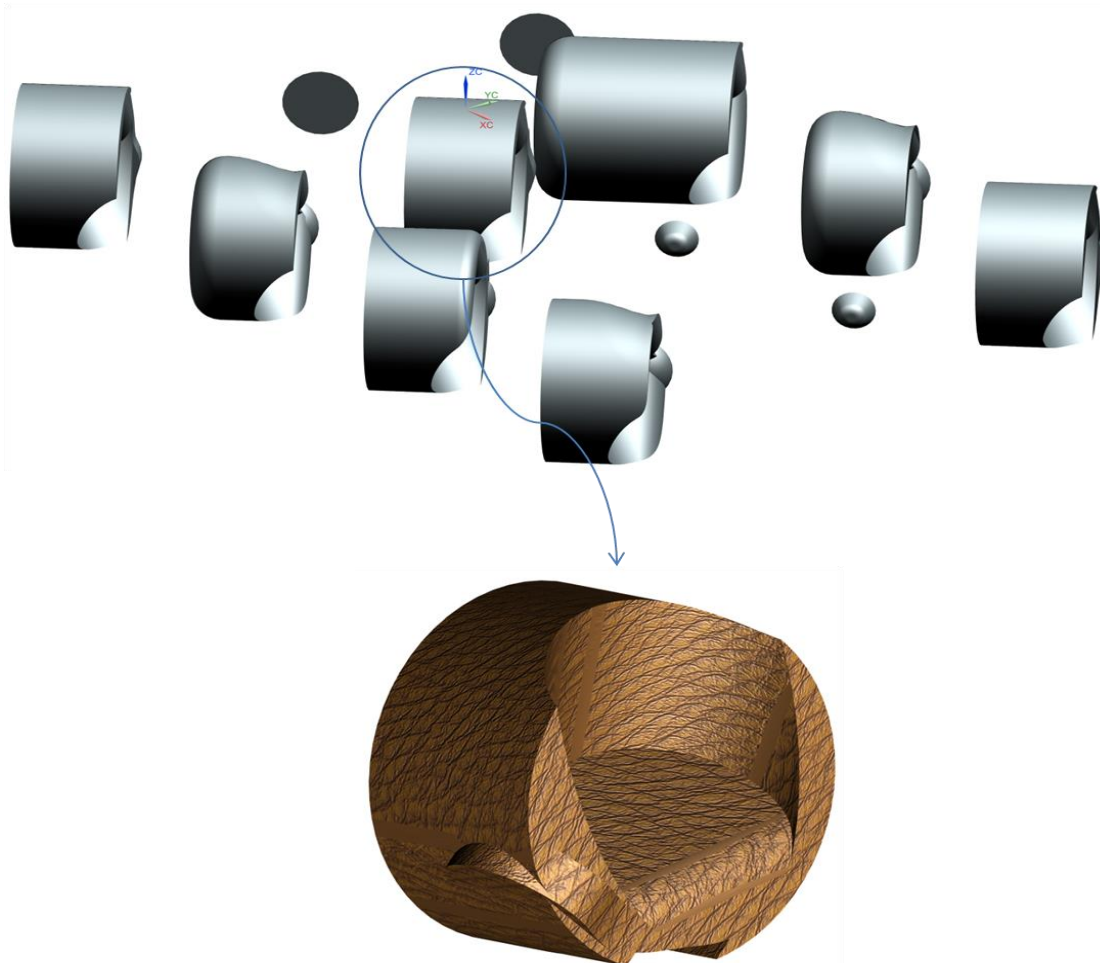
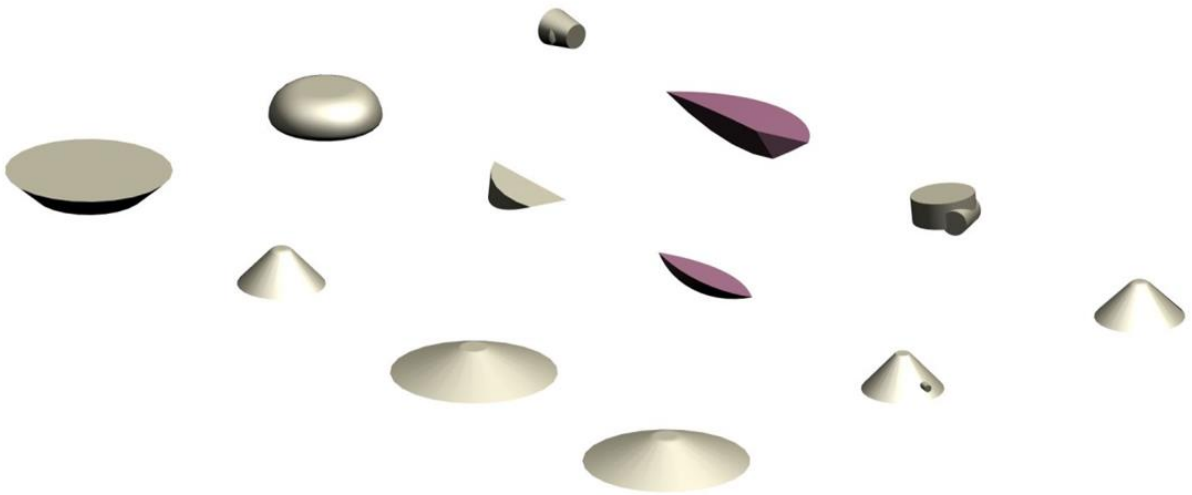


Figure 6.19. SUI: In generation 34, the participant chose two shapes that they found useful and could inspire the form of a sofa. Although the shapes have started to converge, there is still a variety of shapes and some of them can still give ideas for a sofa. In generation 40 there is at least one shape that the participant can choose to inspire the form of a sofa.

Generation 10

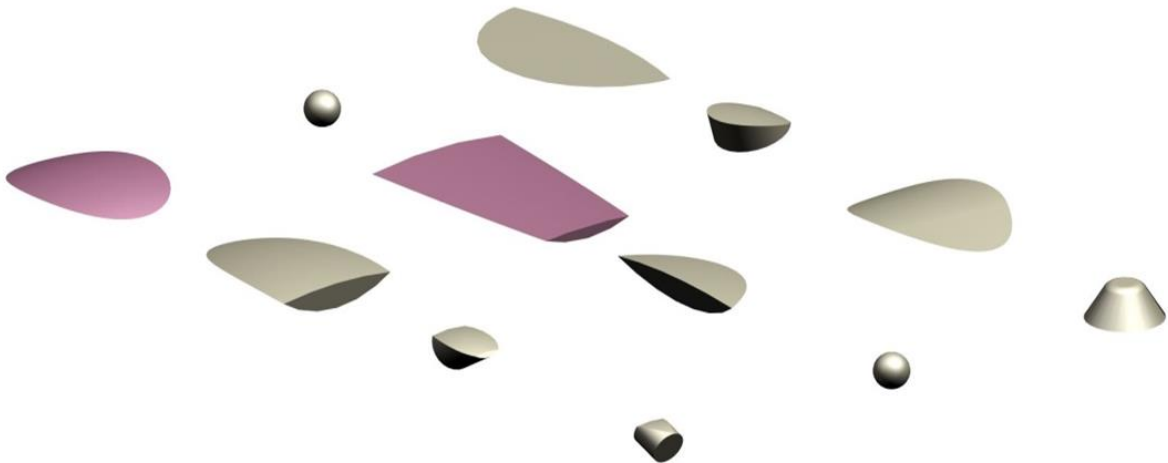


Generation 16



Figure 6.20. SUI: In generations 10 and 16, the participant chose two and three shapes respectively that they found useful and could inspire the form of a guitar. There is a small variety of shapes but none of them can give good ideas for a guitar yet.

Generation 22



Generation 28

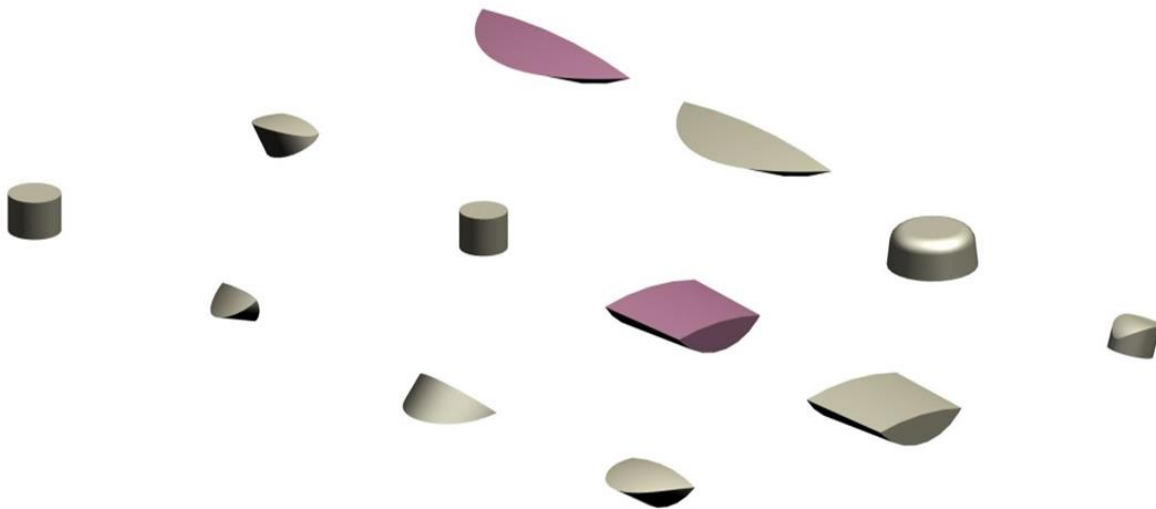


Figure 6.21. SUI: In both generations (22 and 28), the participant chose two shapes that they found useful and could inspire the form of a guitar. There is still a small variety of shapes but still none of them can give ideas for a guitar.

Generation 34

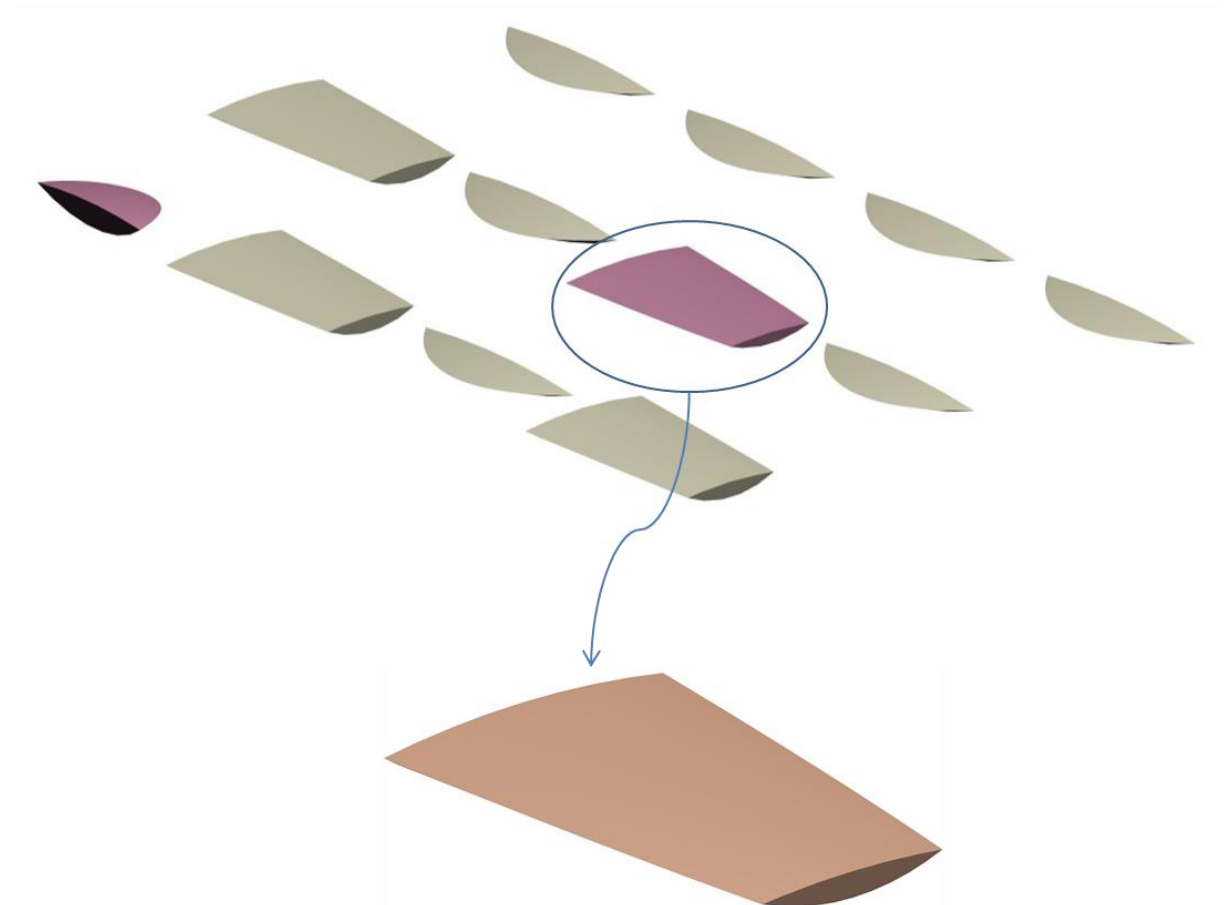


Figure 6.22. SUI: In generation 34, the participant chose two shapes that they found useful and could inspire the form of a guitar. The shapes have started to converge and there was no big variety of shapes to give ideas for a guitar. The participant stopped the process.

6.4 Discussion

User testing has been carried out with five users, ranging in experience from third-year engineering undergraduates to design professionals. In each test, which took around 1 hour, users evolved shapes that would inspire a range of consumer products (a sofa, a watch, a guitar, and a hand-held 3D mouse). Each of the five users tested EvoShape, as well as two versions of the software with GG (LBP of 80 shapes and LBP of 40 shapes). Two users tested three versions of GG with SUI approach. Guide volumes were created at the beginning of the process by the participants when testing the software with GG.

When interviewed, all participants stated a strong preference for versions that included GGs. Observations backed up these subjective findings in several ways, e.g. the time and number of iterations it took to arrive at shapes with which participants were satisfied, and the proportion of useful shapes generated in a given time; in the LBP of 80 shapes approach, after 400 shapes were evaluated by the system, the participants are satisfied with the results – at least one of the shapes gives a useful idea of a product (Figure 6.23).

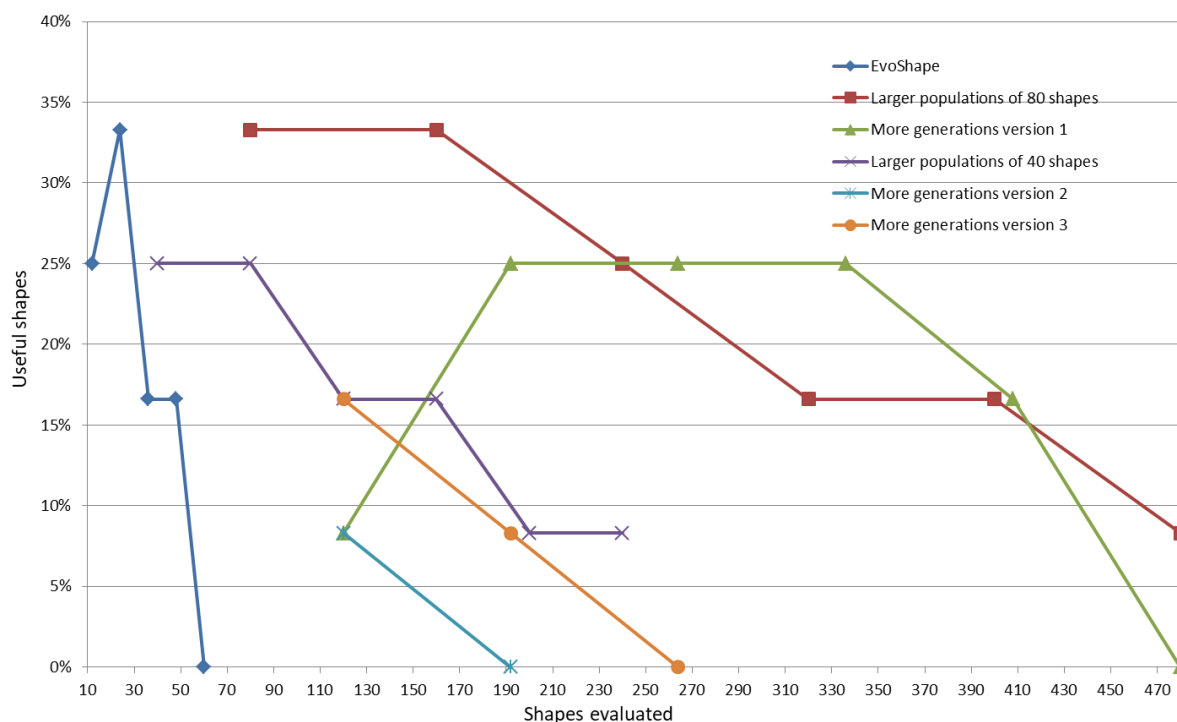


Figure 6.23. Comparison of %useful shapes that each of the three approaches provide.

Most participants showed a strong preference for the LBP version, with only one participant expressing the desire to ‘turn off the guides’ at several intervals during the sessions, to ‘speed things up’ and reduce convergence.

Some of the participants mentioned that the algorithm can be improved by increasing the probability of mutation through the generations (P 3.14). “This will increase the probability of having, in a population, a random shape that is different from the other 11 shapes and may save you by giving you a different idea of a useful shape”. Some others mentioned that

the algorithm could be improved by creating a database of different classes of shapes and using the right class with the right type of bounding volume. This idea could possibly increase the similarity with the bounding volume and remove unwanted shapes from the first generation.

6.5 Participant Testing Conclusions

Human intervention has been added to the system to subjectively test and assess the efficacy of the interactive evolutionary strategies and the efficacy of PN and shape comparison approaches being developed. This has been achieved by:

- ❖ Allowing the users to create soft geometric constraints (in the form of simple bounding geometry) at the start of the evolution process.
- ❖ Testing by the users of three different versions of the system:
 - EvoShape.
 - LBP.
 - SUI.

User testing during development has shown that the LBP interaction strategy best complements the evolutionary design process, and it is this implementation which will be carried forward into the next commercial release of the EvoShape software, with the addition of a toggle switch allowing the new guide functionality to be turned off and on by the user.

7 Conclusions and Future Work

7.1 Conclusions

Firstly, efficient interaction strategies for IEC design systems in the literature have been presented and compared. Some of these have directly influenced this research while others showed potential but were either too complex, computation-time demanding or would have required too much time to implement. The ‘pareto-optimal points’ interaction strategy gets close to the most preferred point and finds the best solution in only a few interactions with the users and without fatigue. Although this approach has been classified as a multi-objective problem, it has influenced the SUI approach of this research, in that the number of interactions with users is predefined.

Objective 1: Develop a representation for the simple GG

All the participants were asked to create a simple GG (soft geometric constraint in the form of simple bounding geometry), at the beginning of the evolutionary process, that will guide the evolving forms. The GG should be a rough representation of the *proportions* the participants would like to create. It should be a solid 3D bounding volume, which the users think will influence the evolving forms proportions, whose bounding lengths should be between 25mm and 100mm. Some participants had small experience in using NX CAD system; therefore they created their GG with the support of the developer.

Objective 2: Develop stand-alone programs to investigate the individual operations

During this research, through an Internal Open C API, different readily-available CAD functions were used to develop stand-alone programs and investigate the alignment (PN) functionality (rotation, scaling and translation) as well as the shape comparison functionality. Through these stand-alone programs a number of PN and shape comparison methods have been discovered.

Objective 3: Find efficient alignment methods

Since PN is essential to align the GG with the evolving forms, the nine most suitable PN technology candidates for IEC design system research have also been presented and compared. The PCA-based PN method is suitable for complex 3D models, and it was selected to be adapted for this research since the IEC design systems of interest generate quite abstract and complex 3D models.

The most efficient 3D alignment methods were found by developing a program using different readily-available CAD functions; PA for the orientation, Volume and BB for the scaling, and CoM and Centre of BB for the translation.

Objective 4: Find efficient shape comparison method

The most efficient 3D shape comparison algorithm was found by developing a program using different readily-available CAD functions; this algorithm is based on a simple method – the principle of non-intersecting volumes. This algorithm is implemented with standard geometric functions, is simple, accessible to other researchers and offers appropriate efficacy for the application – and objective tests to confirm this have been completed.

Objective 5: Integrate the most efficient operations with EvoShape Interactive Evolutionary Design (IED) system

The most efficient alignment and shape comparison methods, described in Objectives 3 and 4, were chosen to integrate with the GA-driven generative design program (EvoShape). This shortlist of successful methods could be tested mathematically for accuracy.

Objective 6: Turn the add-in program into an automated shapes matching program

To test and assess the efficacy of the GG as well as to find the most efficient alignment and shape comparison methods that will finally be used for the IED system, the add-in program was turned into an automated shape matching program running consistent experiments but without human intervention.

Objective 7: Test alignment and shape comparison methods efficiency

The automatic shape matching program run 16 iterations for one method of rotation (PA), 32 iterations for two methods of scaling (BB and volume) and 32 iterations for two methods of translation (Centre of BB and Centre of Mass), with four different starting populations and four different GG.

The PA method applied in rotation, while not perfect, and not as effective as more sophisticated techniques reviewed in the literature, is demonstrated to be good enough to be used for *guiding* applications, such as in IEC design systems. This has been confirmed by the results of a full system validation outlined in Chapter 5. The BB method applied in scaling, produces reasonable scaling estimations and achieves robustness with typical geometry of the Interactive Evolutionary CAD system described in Chapter 4. The centre of BB method produces reasonable translation estimations and achieves robustness with typical geometry of the Interactive Evolutionary CAD system described in this research.

Objective 8: Develop effective human-system interaction strategies

Investigation of several user interaction strategies were described, which examine how best to use the GG capabilities to help users in evolutionary shape searching. Objective and fitness values based on the shape comparison have been produced. The experimental results imply that the LBP of 80 shapes approach has the greatest benefit as it produces significantly better fitness than the benchmark approach. The variety of the forms is also important to investigate the interaction strategies that best complement the evolutionary

design process. To discard the benchmark approach, both benchmark and LBP approaches need further consideration with user interaction.

Objective 9: Add human intervention to the system

To test and assess the efficacy of the interactive evolutionary strategies, human interaction was added to the add-in program. This was achieved by either replacing the automated shape matching program with user interaction at intervals of between 4 and 11 generations (SUI approach) or by adding human interaction in every generation to evaluate a subset of the fittest shapes found using GG (LBP approach).

Objective 10: Test the human-system interaction strategies to the users

User testing during development has shown that the LBP interaction strategy best complements the evolutionary design process, and it is this implementation which will be carried forward into the next commercial release of the EvoShape software, with the addition of a toggle slider control allowing the new guide functionality to be turned off and on by the user, and to customise the 'strength' of the guides offset by speed. The 'strength' is dictated by the number of background shapes, and can be set at 0, 20, 40, 60 and 80.

Concluding statements

What make this research unique is both the concept and implementation of *guided* evolutionary shape searching, the concept and optimised implementation of background and foreground populations in interactive evolutionary design, and the integration of both of these elements. The implementation in the CAD domain and the specific application in form ideation during conceptual stages of product design is also unique.

Contributions

- ✓ This system better supports early design processes and ideation in conceptual design.
- ✓ This system better allows a designer to experiment with ideas freely and to interactively explore populations of evolving solutions.
- ✓ Users of the new Interactive Evolutionary CAD system have more control over the process beyond direct real-time interacting with the system through shape selection and scoring during the evolutionary process – by creating GG at the beginning of the process.

Limitations of the research

A limitation of the system is that it focuses on aesthetics rather than function, when in reality function is often the major influencer on form. Another limitation is that since PN algorithms have not been investigated in practice, and only simple, computationally quick methods that are easily adoptable by other researchers have been used, the system in this research does not produce good results when the GG are quite complex shapes – i.e. are not approximated with a single geometric primitive (e.g. roof). However, this problem was partially solved by adding user intervention to the system where a GG similar to 'roof' gave

good results; useful ideas of products that the users wanted to create, guided by the GG that they created at the beginning of the process.

7.2 Future Work

There are two broad recommendations which could be implemented in order to develop the research further; integration with topology optimisation and machine learning.

Another way that industrial designers can find inspiration for form is topology optimisation. When designers do not have an existing design or concept that could inspire them, they can simply start with a block of material, model the known forces on the part, and allow the optimiser to define the shape and size of the design. This is known as topology optimisation. Topology optimisation satisfies a set of constraints by minimising or maximising optimisation objectives such as mass, maximum stress or displacement (Hale, 2017). It is acknowledged that a weakness of the system developed during this research, is that it treats form independently from function, when in reality, function is often the major influencer. To reconcile this, topology optimisation could be integrated and run in parallel with the improved EvoShape system and assist the designer even more to find inspiration.

In Section 2.4.3 a Machine Learning approach for IED was presented and analysed. After the system shows the best forms to the user and the user defines a fitness value to each form according to their preference, Machine Learning should assimilate on-line the aesthetic preferences of the user. This should reduce the human interaction gradually from the first generation until the last, and increase the aesthetical solutions in terms of the user preferences. It is important that the system will sufficiently study the user's preferences. If enough information is given to the system with a good representation, then the algorithm will learn and assimilate the information easier and better and the machine learning system will be overloaded with the instances to be learned. This approach should be integrated with EvoShape to speed up the interactive evolutionary process and decrease the users' effort.

Takagi's (2003) method of Euclidean distances could be a useful addition to the GG algorithm that chooses which shapes to present to the user, as it improves variety of solutions. The method discards any similar solutions of a population and therefore increases the diversity of the solutions. It would have taken too long to integrate into the CAD system during this research but certainly would be worth investigating in the future, despite the concern that objects mathematically very similar subjectively could be quite different in appearance – a small difference to the geometry can make a big difference.

Additional funding from the EPSRC is enabling the outcomes of this doctoral research to be incorporated into the next release of EvoShape. This will then be supplied to some existing

customers for beta testing. As well as further validating the work described in this thesis, it is also likely to highlight opportunities for further related research.

References

- Adler, D. (1993) 'Genetic algorithms and simulated annealing: A marriage proposal', in *IEEE International Conference on Neural Networks - Conference Proceedings*. NY, pp. 1104–1109.
- Altair Company (2018) *SolidThinking*. Available at: <https://solidthinking.com/inspire2018.html>.
- Autodesk (2018). Available at: <https://www.autodesk.co.uk/>.
- Autodesk Research (2018) *Autodesk Project Dreamcatcher*. Available at: <https://autodeskresearch.com/projects/dreamcatcher>.
- Averkiou, M., Kim, V. G. and Mitra, N. J. (2015) 'Autocorrelation Descriptor for Efficient Co-alignment of 3D Shape Collections', *Computer Graphics forum*, 34, pp. 1–11.
- Axenopoulos, A., Litos, G. and Daras, P. (2011) '3D model retrieval using accurate pose estimation and view-based similarity', in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval - ICMR '11*. New York, USA: ACM Press, pp. 1–8.
- Baluja, S., Pomerleau, D. and Jochem, T. (1994) 'Towards Automated Artificial Evolution for Computer-generated Images', *Connection Science*, 6(2–3), pp. 325–354.
- Barequet, G. and Har-Peled, S. (2001) 'Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions', *Journal of Algorithms*, 38(1), pp. 91–109.
- Biles, J. (1994) 'GenJam: A genetic algorithm for generating jazz solos', in. Aarhus, Denmark: Int. Computer Music Conf. (ICMC'94), pp. 131–137.
- Biles, J., Anderson, P. and Loggi, L. (1996) 'Neural network fitness functions for a musical IGA', *Proceedings of the International ICSC Symposium on Intelligent Industrial Automation*. Reading, UK.
- Birkhoff, D. G. (2016) *Aesthetic Measure — George David Birkhoff | Harvard University Press*. E-DITION. Available at: <http://www.hup.harvard.edu/catalog.php?isbn=9780674734470> (Accessed: 4 February 2016).
- Branke, J. (1999) 'Evolutionary approaches to Dynamic Optimization Problems - {A} survey-', in *Proceedings of the Genetic and Evolutionary Computation Conference {GECCO}-1999, Workshop Program*. Karlsruhe, Germany, pp. 134–137.
- Caldwell, C. and Johnston, V. S. (1991) 'Tracking a criminal suspect through "face-space"

- with a genetic algorithm', in *4th Int. Conf. Genetic Algorithm (ICGA'91)*. San Mateo, CA: Morgan Kaufmann, pp. 416–421.
- Catchbook (2018). Available at:
<https://www.plm.automation.siemens.com/global/en/products/nx/catchbook.html>.
- Chan, C. K. and Tan, S. T. (2001) 'Determination of the minimum bounding box of an arbitrary solid: An iterative approach', *Computers and Structures*, 79(15), pp. 1433–1449.
- Chaouch, M. and Verroust-Blondet, A. (2009) 'Alignment of 3D models', *Graphical Models*, 71(2), pp. 63–76.
- Chen, D.Y., Tian, X.P., Shen, Y.T. and Ouhyoung, M. (2003) 'On visual similarity based 3D model retrieval', *Computer graphics forum*, 22(3), pp. 223–232.
- Cunha, M. and Sousa, J. (2001) 'Hydraulic Infrastructures Design Using Simulated Annealing', *Journal of Infrastructure Systems*, 7(1).
- Dang, Q.-V., Mouysset, S. and Morin, G. (2014) 'Symmetry-based alignment for 3D model retrieval', in *2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI)*. Klagenfurt, Austria: IEEE, pp. 1–6.
- Davidson, S. (2018) *Grasshopper*. Available at: <https://www.grasshopper3d.com/>.
- Dean, L., Atkinson, P. and Unver, E. (2005) 'Evolving individualised consumer products', in *6th International Conference of the European Academy of Design*. University of the Arts, Bremen., pp. 1–19.
- Deb, K., Sinha, A., Korhonen, P.J. and Wallenius, J. (2010) 'An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions', in *IEEE Transactions on Evolutionary Computation*, 14(5), pp. 723–739.
- DiPaola, S. and Gabora, L. M. (2009) 'Incorporating characteristics of human creativity into an evolutionary art algorithm', *Genetic Programming and Evolvable Machines*, 10(2), pp. 97–110.
- Dorigo, M., Maniezzo, V. and Coloni, A. (1996) 'The Ant System: Optimization by a colony of cooperating agents', *IEEE Transactions on Systems, Man, and Cybernetics—Part B IEEE Transactions on Systems, Man, and Cybernetics, Part-B*, 26(26), pp. 1–131.
- Dougherty D. E. and Marryott, R. A. (1991) 'Optimal Groundwater Management: 1. Simulated Annealing', *AN AGU Journal*, 27(10), pp. 2493–2508.
- Ekárt, A., Chalakov, S. and Sharma, D. (2011) 'Modelling human preference in evolutionary art', in *European Conference on the Applications of Evolutionary Computation*. Berlin, Heidelberg: Springer, pp. 303–312.
- Elbeltagi, E., Hegazy, T. and Grierson, D. (2005) 'Comparison among five evolutionary-based optimization algorithms', *Advanced Engineering Informatics*, 19(1), pp. 43–53.

- Eusuff, M. M. and Lansey, K. E. (2003) 'Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm', *Journal of Water Resources Planning and Management*, 129(3), pp. 210–225.
- EvoShape (2011) *EvoShape*. Available at: <http://www.evoshape.co.uk/>.
- Farrugia, T. and Barbarar, J. (2015) 'Pose normalisation for 3D vehicles', *Computer Analysis of Images and Patterns*. Valletta, Malta, pp. 235–245.
- Fogel, L.J., Owens, A.J. and Walsh, M. J. (1966) *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons Inc.
- Frustum (2018) *Frustum Generate*. Available at: https://www.frustum.com/?gclid=CjwKCAiArK_fBRABEiwA0gOOc2zbDwqDX-u_WbFSDM0n9OwSvGU5fRj5TFvzwpDfR27T14cqPFFs0BoCQe4QAvD_BwE#overview.
- Geltman, K. E. (2014) *The Simulated Annealing Algorithm*. Available at: <http://katrinaeg.com/>.
- GenoForm (2018). Available at: <https://www.solidworks.com/partner-product/genoform-genopal>.
- Gharib, I. (2013) *Integration of Sketch-based Ideation and 3D Modeling with CAD Systems*. Brunel University.
- Goldman, F. E. and Mays, L. W. (1999) 'The application of simulated annealing to the optimal operation of water systems', in *29th Annual Water Resources Planning and Management Conference, WRPMD*. Tempe, AZ, United States: American Society of Civil Engineers (ASCE).
- Graham, I.J. , Campbell, R.I. and Case, K. (2008) 'User-led development of an Interactive Evolutionary Design system', in *Proc. of the 8th International Conference in Adaptive Computing in Design and Manufacturing, Institute for People-Centred Computation*. Bristol, UK: ACDM.
- Graham, I. J. (2002) *Genetic algorithms for evolutionary product design*. Published Thesis (PhD), Loughborough University.
- Graham, I. J., Case, K. and Wood, R. L. (2001) 'Genetic algorithms in computer-aided design', *Journal of Materials Processing Technology*, 117(1–2), pp. 216–221.
- Hale, J. (2018) *How could generative design technologies fit into a traditional product design process?*, Placement Dissertation, Loughborough University.
- Hale, S. (2017) *What is Topology Optimization and Why Use It?*, CAE Associates. Available at: <https://caeai.com/blog/what-topology-optimization-and-why-use-it>.
- Hauck, A. (2018) *What is Generative Design?* Available at: <https://www.linkedin.com/pulse/what-generative-design-anthony-hauck/>.

- Hayashida, N. and Takagi, H. (1999) 'Introduction of an active selection to the interactive evolutionary computation by using multi-dimensional data visualization', in *SOFTKyushu Chapter Meeting*. Saga, Japan, pp. 17–20.
- Heczko, M., Keim, D. A., Saupe, D. and Vranic, D. V. (2002) 'A method for similarity search of 3D objects'.
- Hoenig, F. (2005) 'Defining computational aesthetics', *Proceedings of the First Eurographics conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Girona, Spain, pp. 13–18.
- Holland, J. H. (2016) *Adaptation in Natural and Artificial Systems* | The MIT Press. Available at: <https://mitpress.mit.edu/books/adaptation-natural-and-artificial-systems> (Accessed: 15 February 2016).
- Hu, Z.-H., Ding, Y.-S., Zhang, W.-B. and Yan, Q. (2008) 'An interactive co-evolutionary CAD system for garment pattern design', *Computer-Aided Design*, 40, pp. 1094–1104.
- Huebner, K., Ruthotto, S. and Kragic, D. (2008) 'Minimum volume bounding box decomposition for shape approximation in robot grasping', in *Proceedings - IEEE International Conference on Robotics and Automation*. Pasadena, CA, USA, pp. 1628–1633.
- InstantCAD (2017) *InstantCAD*. Available at: <https://www.neowin.net/news/instantcad-is-a-new-mit-developed-plugin-that-enables-real-time-cad-changes/>.
- Jin, Y. and Branke, J. (2005) 'Evolutionary Optimization in Uncertain Environments—A Survey', *IEEE Transactions on Evolutionary Computation*, 9(3).
- Johan, H., Li, B. and Wei, Y. (2011) '3D model alignment based on minimum projection area', *The Visual Computer*, 27(6–8), pp. 565–574.
- Jones, G. (1998) 'Genetic and evolutionary algorithms', *Encyclopedia of Computational Chemistry*. John Wiley. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.6043&rep=rep1&type=pdf>.
- Karim, A. (2004) *Computer-aided aesthetics in evolutionary computer aided design*. PhD Thesis, Loughborough University.
- Kazhdan, M. (2007) 'An Approximate and Efficient Method for Optimal Rotation Alignment of 3D Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7), pp. 1221–1229.
- Kazhdan, M. M., Funkhouser, T. A. and Rusinkiewicz, S. (2004) 'Symmetry descriptors and 3D shape matching', *Journal of Eurographics Symposium on Geometry Processing*, 71, pp. 115–123.
- Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', in *Proceedings of ICNN'95 - International Conference on Neural Networks*. Perth, WA, Australia: IEEE, pp. 1942–

1948.

- Kirkpatrick, S., Gelatt Jr. C. D. and Vecchi, P. M. (1983) 'Optimization by Simulated Annealing', *Science*, 220(4598), pp. 671–680.
- Koksalan, M. and Karahan, İ. (2010) 'An Interactive Territory Defining Evolutionary Algorithm: iTDEA', *IEEE Transactions on Evolutionary Computation*, 14(5), pp. 702–722.
- Komninou, A. (2018) *Emotion and Design, Interaction Design Foundation*. Available at: <https://www.interaction-design.org/literature/article/emotion-and-design-affect-and-design>
- Kouskouridas, R., Retzepi, T., Charalampoglou, E. and Gasteratos, A. (2012) 'Ontology-based 3D pose estimation for autonomous object manipulation', *IEEE International Conference on Imaging Systems and Techniques Proceedings*. Manchester, UK.
- Krippendorff, K. and Butter, R. (1984) 'Product Semantics: Exploring the Symbolic Qualities of Form', *Innovation*, 3(2), p. 4.
- Krish, S. (2018) *GenoForm*. Available at: https://prezi.com/vljywi_ndqkb/genoform/.
- Labadie, J. (2004) 'Optimal operation of multireservoir systems: State-of-the-art review', *J. Water Resour. Plann. Manage.*, 130(2), pp. 93–111.
- Li, Y. and Hu, C. (2015) 'Learning personalized exploration in evolutionary design using aesthetic descriptors', *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(3), pp. 489–501.
- Liu, Y., Wang, X. and Zha, H. (2008) 'Dimension Amnesic Pyramid Match Kernel', in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. Chicago, Illinois, pp. 652–658.
- Løvbjerg, M. (2002) *Improving Particle Swarm Optimization by hybridization of stochastic search heuristics and Self-Organized Criticality*. PhD Thesis, University of Aarhus.
- Machado, P., Romero, J., Cardoso, A. and Santos, A. (2005) 'Partially interactive evolutionary artists', *New Generation Computing*, 23(2), pp. 143–155.
- Machado, P. and Cardoso, A. (1998) 'LNAI 1515 - Computing Aesthetics', in *Advances in Artificial Intelligence*. Berlin, Heidelberg, pp. 219–228.
- Machwe, A. T. and Parmee, I. C. (2006) 'Introducing machine learning within an interactive evolutionary design environment', in *International Design Conference*. Dubrovnik - Croatia, pp. 283–290.
- Mallawaarachchi, V. (2017) *Introduction to Genetic Algorithms*. Available at: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>.
- Manandhar, I. (2017) *Affordance and Its Future In Designing Interfaces*. Available at: <https://blog.prototypr.io/affordance-and-its-future-in-designing-interfaces->

- Marryott, R. A. (1996) 'Optimal Ground-Water Remediation Design Using Multiple Control Technologies', *The groundwater association*, 34(3), pp. 425–433.
- Moscato, P. (1989) 'On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: towards memetic algorithms', *California Institute of Technology*. Pasadena, CA, USA.
- Nagao, M., Yamada, M., Suzuki, K. and Ohuchi, A. (1998) 'Evaluation of the image retrieval system using interactive evolutionary computation', in *Workshop on Interactive Evolutionary Computation*,. Fukuoka, Japan, pp. 25–30.
- Nguyen, T. T., Yang, S. and Branke, J. (2012) 'Evolutionary dynamic optimization: A survey of the state of the art', *Swarm and Evolutionary Computation*, 6, pp. 1–24.
- Nielsen, J. (2012) *How Many Test Users in a Usability Study?* Available at: <https://www.nngroup.com/articles/how-many-test-users/> (Accessed: 30 November 2017).
- Novotni, M. and Klein, R. (2001) 'A Geometric Approach to 3D Object Comparison', in *Proceedings International Conference on Shape Modeling and Applications*. Genova, Italy, pp. 167–175.
- Ohsaki, M., Takagi, H. and Ohya, K. (1998) 'An input method using discrete fitness values for interactive GA', *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*. IOS Press, 6(1), pp. 131–145.
- Papadakis, P., Pratikakis, I., Perantonis, S.J. and Theoharis, T. (2007) 'Efficient 3D Shape Matching and Retrieval using a Concrete Radialized Spherical Projection Representation', *Pattern Recognition*, 40(9), pp. 2437– 2452.
- Papadakis, P., Pratikakis, I., Theoharis, T., Passalis, G. and Perantonis, S. (2008) '3D object retrieval using an efficient and compact hybrid shape descriptor', in *Proceedings of the 1st Eurographics conference on 3D Object Retrieval*. Crete, Greece, pp. 9–16.
- Papadakis, P. (2014) 'Enhanced pose normalization and matching of non-rigid objects based on support vector machine modelling', *Pattern Recognition*, 47(1), pp. 216–227.
- Paquet, E., Rioux, M., Murching, A., Naveen, T. and Ali T. (2000) 'Description of shape information for 2-D and 3-D objects', *Signal Processing: Image Communication*, 16(1), pp. 103–122.
- Pears, N., Heseltine, T. and Romero, M. (2010) 'From 3D point clouds to pose-normalised depth maps', *International Journal of Computer Vision*, 89(2–3), pp. 152–176.
- Perfetti, B. Y. C. and Landesman, L. (2001) *Eight is Not Enough*. Available at: http://www.uie.com/articles/eight_is_not_enough/ (Accessed: 30 November 2017).
- Podolak, J. and Funkhouser, T. (2006) 'A Planar-Reflective Symmetry Transform for 3D Shapes', *Journal of ACM Transactions on Graphics*, 25(3), pp. 549–559.

- Ponnambalam, S. G. and Reddy, M. M. (2003) 'A GA-SA multiobjective hybrid search algorithm for integrating lot sizing and sequencing in flow-line scheduling', *International Journal of Advanced Manufacturing Technology*, 21(2), pp. 126–137.
- PTC Mathcad (2018). Available at: <https://www.ptc.com/en/products/mathcad>.
- Rechenberg, I. (2016) *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. (=Problemata Frommann-Holzboog; 15): Book*.
- Rigau, J., Feixas, M. and Sbert, M. (2016) 'Informational aesthetics measures.', *IEEE computer graphics and applications*, 28(2), pp. 24–34.
- Ross, B. J. and Ralph, W. (2006) 'Evolutionary Image Synthesis Using a Model of Aesthetics', in *2006 IEEE International Conference on Evolutionary Computation*. Vancouver, BC, Canada: IEEE, pp. 1087–1094.
- Rourke, J. O. (1985) 'Finding Minimal Enclosing Boxes', *International Journal of Computer and Information Sciences*, 14(3), pp. 183–199.
- Rustamov, R. M. (2007) 'Augmented symmetry transforms', in *IEEE International Conference on Shape Modeling and Applications*. Lyon, France, pp. 13–20.
- Said, B. L., Bechikh, S. and Ghedira, K. (2010) 'The r-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making', *IEEE Transactions on Evolutionary Computation*, 14(5), pp. 801–818.
- Sato, T. and Hagiwara, M. (2001) 'IDSET: Interactive Design System using Evolutionary Techniques', *Computer-Aided Design*, 33(5), pp. 367–377.
- Saupe, D. and Vranic, D. V. (2001) '3D model retrieval with spherical harmonics and moments', *Pattern Recognition*, 2191, pp. 392–397.
- Sfikas, K., Theoharis, T. and Pratikakis, I. (2011) 'ROSy+: 3D object pose normalization based on PCA and reflective object symmetry with application in 3D object retrieval', *International Journal of Computer Vision*, 91(3), pp. 262–279.
- Sfikas, K., Theoharis, T. and Pratikakis, I. (2014) 'Pose normalization of 3D models via reflective symmetry on panoramic views', *The Visual Computer*, 30(11), pp. 1261–1274.
- Shieh, H.-J. and Peralta, R. C. (2005) 'Optimal In Situ Bioremediation Design by Hybrid Genetic Algorithm-Simulated Annealing', *Journal of Water Resources Planning and Management*, 131(1), pp. 67–78.
- Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T. (2004) 'The princeton shape benchmark', in *Proceedings of the Shape Modeling International*. Washington, DC, USA, pp. 167–178.
- Sinclair, M. (2013) *The specification of a consumer design toolkit to support personalised production via additive manufacturing, Case studies in the use of design practice by*

PhD researchers. PhD Thesis, Loughborough University.

Sinha, A., Korhonen, P., Wallenius, J. and Deb, K. (2014) 'An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls', *European Journal of Operational Research*, 233(3), pp. 674–688.

SolidThinking (2018) *Solid thinking Inspire*. Available at:
<http://www.solidthinking.com/Inspire2016.html>.

Spool, J. and Schroeder, W. (2001) 'Testing web sites: Five users is nowhere near enough', in *CHI '01 extended abstracts on Human factors in computing systems - CHI '01*. Seattle, Washington, pp. 285–286.

Takagi, H. (2001) 'Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation', *Proceedings of the IEEE*, 89(9), pp. 1275–1296.

Takagi, H. (2003) 'Humanized computational intelligence with interactive evolutionary computation', *Proceedings of Computational Intelligence: the Experts Speak, Washington*, 89(9), pp. 2001–2001.

Takagi, H. and Kishi, K. (1999) 'On-line knowledge embedding for an interactive EC-based montage system', in *1999 Third International Conference on Knowledge-Based Intelligent Information Engineering Systems Proceedings*. Adelaide, SA, Australia: IEEE, pp. 280–283.

Takagi H. and Unemi T. (1998) 'Perspective on interactive evolutionary computing', *Journal-Japanese Society for Artificial Intelligence*, 13(5), pp. 692–703.

Tangelder, J. W. H. and Veltkamp, R. C. (2008) 'A survey of content based 3D shape retrieval methods', *Multimedia Tools and Applications*, 39(3), pp. 441–471.

The biomimicry Institute (2018). Available at: <https://biomimicry.org/janine-benyus/>.

Theodoridis, S. and Koutroumbas, K. (2006) *Pattern recognition*. Academic Press. Available at: <https://www.elsevier.com/books/pattern-recognition/theodoridis/978-1-59749-272-0>.

Tiainen, T., Ellman, A. and Syrilä, A. (2008) 'Interactive Evolutionary Design for Recognizing Customer Needs and Wishes', *Design*, (Bentley 1999), pp. 705–712.

Ventrella, J. (2008) 'Evolving the Mandelbrot Set to Imitate Figurative Art', *Design by Evolution: Advances in Evolutionary Design*, pp. 145–168.

Vranic, D. V. (2004) '3D model retrieval', *Ph. D. thesis, University of Leipzig*, pp. 1–13.

Vranic, D. V. (2005) 'Desire: A Composite 3D-Shape Descriptor', in *2005 IEEE International Conference on Multimedia and Expo*. Amsterdam, Netherlands: IEEE, pp. 962–965.

Vranic, D. V., Saupe, D. and Richter, J. (2001) 'Tools for 3D-object retrieval: Karhunen-loeve transform and spherical harmonics', in *IEEE Fourth Workshop on Multimedia Signal*

Processing. Cannes, France, pp. 293–298.

- Wu, H.-Y., Zha, H., Luo, T., Wang, X.-L. and Ma, S. (2010) 'Global and local isometry-invariant descriptor for 3D shape comparison and partial matching', in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, pp. 438–445.
- Yu, M., Atmosukarto, I., Leow, W. K., Huang, Z. and Xu, R. (2003) '3D Model Retrieval With Morphing-Based Geometric and Topological Feature Maps', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Madison, WI, USA, USA.
- Zaharia, T. B. and Prêteux, F. J. (2004) '3D versus 2D/3D shape descriptors: a comparative study', *Proceedings of the SPIE*, 5289, pp. 47–58.
- Zarpalas, D., Daras, P., Axenopoulos, A., Tzovaras, D. and Strintzis, M. G. (2007) '3D model search and retrieval using the spherical trace transform', *Applied Signal Processing*, 2007(1), p. 207.
- van Zyl, J., Savic, D. A., and Walters, G. A. (2004) 'Operational optimization of water distribution systems using a hybrid genetic algorithm method', *J. Water Resour. Plann. Manage.*, 130(2), pp. 160–170.

A)Appendix: Calculate fitness

```

/*****
/* calculate_fitness - process 3.0 */
*****/
#include <stdio.h>
#include <string.h>
#include <uf.h>
#include <uf_modl.h>
#include <uf_cfi.h>
#include <uf_ui.h>
#include <uf_obj.h>
#include "pop.tem"
#include "target.tem"
#include "analysis.tem"
#include "results.tem"
#include "parents.tem"
#include "root.tem"
#define null (int *)0
/*****
int calculate_fitness(struct root *root_ptr)
{
    int decode_chromosome(struct phenotype *,int);
    int insert_decoding_ranges(struct phenotype *,struct phenotype *);
    int assemble_analysis_input(struct phenotype *,struct efd_object_analysis *);
    int perform_analysis(struct efd_object_analysis *,struct calculated_values_root *);
    int compare_values(struct calculated_values_root *,struct target_values_root *,struct
phenotype *,struct root *);
    int geometry(struct root *);
    int convert_to_fitness(struct species *, struct root *);
    int elite_replacement(struct root *);
    int subset_to_the_user(struct root *);
    int prompt(char *);
    int run_mode;
    int b_response = 0;
    int return_code = 0;
    int return_code_b = 0;
    int return_code_d = 0;
    int status = 1;
    int no_species =0;
    int test = 1;

    char message[40] = "Ready to Decode Chromosomes";

    struct phenotype *phenotype_ptr;
    struct species *species_ptr;
    /*****
    printf("\n----- Calculate Fitness -----");
    printf("\n----- Generation %2d -----\\n",root_ptr->
generation_no);
    run_mode=root_ptr->ui_control_ptr->run_mode;
    if(run_mode==5) prompt(message);

    /*          DECODE CHROMOSOMES          */

    printf("\\nroot_ptr->id_number: %d\\n",root_ptr->id_number);

    species_ptr = root_ptr->first_species_ptr;
    while (species_ptr != null_species)
    {
        species_ptr->done=0;
        no_species++;
        species_ptr=species_ptr->next;
    }

    while (status>0)
    {
        species_ptr = root_ptr->first_species_ptr;
        while (species_ptr != null_species)
        {
            if ((species_ptr->done==0)&&((species_ptr->decoded_by>0&&species_ptr->
decoded_by_ptr->done==1)
|| (species_ptr->decoded_by == 0)))

```



```

        {
            phenotype_ptr = species_ptr->first_phenotype;
            while (phenotype_ptr != null_phenotype)
            {
                if (species_ptr->decoded_by>0&&species_ptr->decoded_by_ptr->done==1)
                    return_code = insert_decoding_ranges(phenotype_ptr, phenotype_ptr->
>decoded_by_ptr);
                return_code = decode_chromosome(phenotype_ptr, species_ptr->chrom_manip);
                phenotype_ptr = phenotype_ptr->next;
            }
            species_ptr->done=1;
            printf("Species %d decoded\n\n",species_ptr->species_id);
        }
        species_ptr = species_ptr->next;
    }
    status=0;
    species_ptr = root_ptr->first_species_ptr;
    while (species_ptr != null_species)
    {
        if (species_ptr->done==0)
            status++;
        species_ptr=species_ptr->next;
    }

    return_code = geometry(root_ptr);
    printf("-> %d\n",return_code);

    /*          FITNESS          */

    if (!( root_ptr->ui_control_ptr->quitswitch +
            root_ptr->ui_control_ptr->shoggle +
            root_ptr->ui_control_ptr->load_bitstream ))

        printf("\n Fitness processing...\n");

    species_ptr = root_ptr->first_species_ptr;
    while (species_ptr != null_species)
    {
        phenotype_ptr = species_ptr->first_phenotype;
        while (phenotype_ptr != null_phenotype)
        {
            return_code = assemble_analysis_input(phenotype_ptr, species_ptr->
>analysis_ptr);
            return_code = perform_analysis(species_ptr->analysis_ptr,species_ptr->
>calculated_values_root_ptr);
            return_code = compare_values(species_ptr->
>calculated_values_root_ptr,species_ptr->target_values_root_ptr,phenotype_ptr,root_ptr);
            phenotype_ptr = phenotype_ptr->next;
        }
        printf("\nFitness values\n");
        return_code_b = convert_to_fitness(species_ptr, root_ptr);
        if(run_mode==5) prompt("Check fitness values");

        if(root_ptr->generation_no > 1)
        {
            if (run_mode==5) prompt("OK");
            if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]!=1)

/*Only gg*/
                {
                    printf("\nCalling elite_replacement...\n\n");
                    return_code = elite_replacement(root_ptr);
                }

            if (run_mode==5) prompt("OK");
            if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]!=1) /*Only gg*/
        {
            printf("\nCalling subset_to_the_user...\n\n");
            return_code = subset_to_the_user(root_ptr);
        }
        if (run_mode==5) prompt("OK");
        if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]==1) /*Both gg and user
ratings*/

```

```

        {
            printf("\nCalling subset_to_the_user...\n\n");
            return_code = subset_to_the_user(root_ptr);
        }
        species_ptr = species_ptr->next;
    }
    printf("\n - Calculate Fitness -\n");
    printf(" ----- %d -> ",return_code);
    return(return_code);
}

```

B) Appendix: Geometry

```

/*****
/* geometry - process 4.0 */
*****/

#include <stdio.h>
#include <string.h>
#include <uf.h>
#include <uf_ui.h>
#include <uf_cfi.h>
#include <uf_part.h>
#include <uf_trns.h>
#include <uf_modl.h>
#include <uf_view.h>
#include "root.tem"
#include "pop.tem"
#include "results.tem"
#define null (int *)0

/*****

int geometry(struct root *root_ptr)
{
int create_form_data(struct root *);
int open_new_part(struct root *);
int create_objects(struct root *);
int analyse_created_geometry(struct root *);
int cheat_blend(struct root *);
int real_blend(struct root *);
int unite_all_solids(struct root *);
int geometric_fitness(struct root *);
int rotation_scalingBB(struct root *);
int rotation_scalingVOL(struct root *);
int translationBB_object_comparison(struct root *);
int translationCOFM_object_comparison(struct root *);
int interaction_loop(struct root *,int);
int prompt(char *);

int      return_code_a = 0,
        return_code_b = 0,
        return_code_c = 0,
        return_code_d = 0,
        return_code_e = 0,
        return_code    = 0,
        objects,
        run_mode;

char *prompt_message = "Look at the status window";

struct species *species_ptr;
struct phenotype *phenotype_ptr;

/*****

printf(" -----\n");
printf(" -   G e o m e t r y   -\n");

run_mode = root_ptr->ui_control_ptr->run_mode;

return_code_a    = open_new_part(root_ptr);
                printf(" %d\n",return_code_a);
UF_DISP_regenerate_display();
return_code_b    = create_form_data(root_ptr);
                printf(" %d\n",return_code_b);
objects          = create_objects(root_ptr);
return_code_d    = analyse_created_geometry(root_ptr);

if(root_ptr->create_mode!=6)
{
    return_code_d = unite_all_solids(root_ptr);
    return_code_d = analyse_created_geometry(root_ptr);

```

```

}
if(root_ptr->evaluate_mode[6]!=1)
    return_code = geometric_fitness(root_ptr);

if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]!=1) /*Geometric Guides*/
{
    return_code = rotation_scalingBB(root_ptr);
    return_code = translationBB_object_comparison(root_ptr);
}
species_ptr = root_ptr->first_species_ptr;
while (species_ptr != null_species)
{
    if(root_ptr->evaluate_mode[4]==1 && root_ptr->evaluate_mode[6]!=1) /*User Ratings*/
    {
        return_code_e = interaction_loop(root_ptr,objects); printf("%d\n", return_code_e);
    }
    else
    {
        phenotype_ptr=species_ptr->first_phenotype;
        while(phenotype_ptr!=null_phenotype)
        {
            phenotype_ptr->user_rating=0;
            phenotype_ptr=phenotype_ptr->next;
        }
    }
    species_ptr = species_ptr->next;
}

species_ptr = root_ptr->first_species_ptr;
while (species_ptr != null_species)
{
    if(root_ptr->evaluate_mode[4]==1 && root_ptr->evaluate_mode[6]==1) /*Both gg and user ratings*/
    {
        return_code = rotation_scalingBB(root_ptr);
        return_code = translationBB_object_comparison(root_ptr);
    }
    else
    {
        phenotype_ptr=species_ptr->first_phenotype;
        while(phenotype_ptr!=null_phenotype)
        {
            phenotype_ptr->user_rating=0;
            phenotype_ptr=phenotype_ptr->next;
        }
    }
    species_ptr = species_ptr->next;
}

printf("\n - G e o m e t r y  -\n ----- %d ",return_code_e);

return(return_code_e);
}

```

C) Appendix: Rotation and scaling with BB

```

/*****
/* rotation_scalingBB - process 4.D */
*****/
#include <stdio.h>
#include <uf.h>
#include <uf_modl.h>
#include <uf_view.h>
#include <uf_trns.h>
#include <uf_ui.h>
#include <uf_csys.h>
#include <uf_so.h>
#include "root.tem"
#include "geometry.tem"
#include "pop.tem"
#include "results.tem"
#include "target.tem"

#define UF_CALL(X) (report( __FILE__, __LINE__, #X, (X)))
static int report( char *file, int line, char *call, int irc)
{
    if (irc)
    {
        char messg[133];
        printf("%s, line %d: %s\n", file, line, call);
        (UF_get_fail_message(irc, messg)) ?
            printf("    returned a %d\n", irc) :
            printf("    returned error %d: %s\n", irc, messg);
    }
    return(irc);
}
/*****
int rotation_scalingBB(struct root *root_ptr)
{
    int import_target(struct root *);
    int prompt(char *);
    int col=0,row=1,offset = 120;
    int status;
    int type = 1;
    int n_objects = 1,move_or_copy = 2,dest_layer = 0,trace_curves = 2;
    int i,p,m;

    double acc_value[11] = {0.9};
    double rot_mass_props[47],target_mass_props[47];
    double stats[13];
    double rot_move_data[2][12] = { {0,0,0,0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0,0,0} };
    double tran_move_data[2][12] = { {0,0,0,0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0,0,0} };
    double figit = 0.8;
    double rot_bounding_box[6],sc_bounding_box[6];
    double lengthx,lengthy,lengthz,sfx,sfy,sfz;
    double scale_factor[3] = {0,0,0}, point[3], matrix[16];
    double bounding_box[6];

    struct efd_object *efd_object_ptr;
    struct calculated_values_root *calculated_values_root_ptr;
    struct species *species_ptr;
    struct target_values_root *target_values_root_ptr;

    const double matrix_values[9] = { 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0 };
    tag_t matrix_id = NULL_TAG;
    tag_t csys_id = NULL_TAG;

    //Scale declarations
    UF_SCALE_TYPE_t scale_type_2 = UF_SCALE_TYPE_GENERAL;
    int num_body_tags_2 = 1;
    tag_t so_point_2 = NULL_TAG;
    tag_t so_dir_2 = NULL_TAG;
    char factors_2[3][UF_MAX_EXP_BUFSIZE] = { "scale_factor[0]", "scale_factor[1]", "scale_factor[2]" };
    tag_t *tags_2 = NULL;

    UF_SCALE_TYPE_t scale_type = UF_SCALE_TYPE_UNIFORM;

```

```

int num_body_tags = 1;
tag_t scalar = NULL_TAG;
tag_t so_dir = NULL_TAG;
tag_t so_csys = NULL_TAG;
char scale_exp[UF_MAX_EXP_LENGTH + 1];
tag_t *scaleFeat;

const UF_SO_update_option_t;
double new_scale = 1.5;

uf_list_p_t rot_feature_list;

tag_t trace_curve_group;

printf(" - GEOMETRIC GUIDES -\n");

/*****IMPORTS TARGET FILE*****/
UF_CALL(import_target(root_ptr));
UF_CALL(UF_VIEW_fit_view(NULL_TAG,figit));/*fits the geometry to the view. NULL_TAG = Work view*/

printf(" \n- POSE NORMALISATION -\n");
species_ptr = root_ptr->first_species_ptr;
calculated_values_root_ptr = species_ptr->calculated_values_root_ptr;
efd_object_ptr = root_ptr->first_efd_object_ptr;
target_values_root_ptr = species_ptr->target_values_root_ptr;

/*****ROTATION*****/
UF_CALL(UF_MODL_ask_mass_props_3d(&target_values_root_ptr-
>target,1,1,4,100,1,acc_value,target_mass_props,stats));/*finds principal axes of the target*/

while(efd_object_ptr != null_efd_object)/*Object Loop*/
{
    printf("-----\nObject %2d %d bodies %d\n",
    efd_object_ptr->efd_object_id,efd_object_ptr->first_efd_body->body,efd_object_ptr->bodies);

    UF_CALL(UF_MODL_ask_mass_props_3d(&efd_object_ptr->first_efd_body-
>body,1,1,4,100,1,acc_value,rot_mass_props,stats));/*finds principal axes of the objects*/
    UF_CALL(UF_MODL_ask_bounding_box(efd_object_ptr->first_efd_body->body,
rot_bounding_box));/*finds the center of bb of the objects*/

    for (i=0,p=0,m=3;i<=2,p<=2,m<=5;i++,p++,m++)
        rot_move_data[0][i] = (rot_bounding_box[p]+rot_bounding_box[m])/2;/*centre of rotation (x-y-z-
axis) = centre of bb, ORIGIN=ref POINT x=[0][0],y=[0][1],z=[0][2]*/

    for (i=3,p=22;i<=11,p<=30;i++,p++)
        rot_move_data[0][i] = rot_mass_props[p];/*FROM vector: [0][3]->[0][11]*/

    for (i=3,p=22;i<=11,p<=30;i++,p++)
        rot_move_data[1][i] = target_mass_props[p];/*TO vector: [1][3]->[1][11]---Move to csys of the
target*/

    printf("ROTATE OBJECT\n");
    UF_CALL(UF_MODL_ask_body_feats(efd_object_ptr->first_efd_body-
>body,&rot_feature_list));/*returns a list of features of each body*/
    UF_CALL(UF_MODL_move_feature(rot_feature_list,2,rot_move_data));/*moves a list of features from
CSYS to CSYS*/
    UF_CALL(UF_MODL_update());/*Update Undo Happened - Error*/
    UF_CALL(UF_MODL_set_update_fail_option(UF_MODL_UPDATE_ACCEPT));/*Accepts the warning of the
failed feature*/

/*****SCALING*****/
    printf("SCALE OBJECT\n");
    UF_CALL(UF_VIEW_fit_view(NULL_TAG,figit));

    UF_CALL(UF_MODL_ask_bounding_box(efd_object_ptr->first_efd_body->body, sc_bounding_box));

    lengthx = sc_bounding_box[3]-sc_bounding_box[0];
    lengthy = sc_bounding_box[4]-sc_bounding_box[1];
    lengthz = sc_bounding_box[5]-sc_bounding_box[2];

    sfx = (calculated_values_root_ptr->target_length_x/lengthx);
    sfy = (calculated_values_root_ptr->target_length_y/lengthy);
    sfz = (calculated_values_root_ptr->target_length_z/lengthz);

```

```

if (sfx <= sfy && sfx <= sfz)/*finds the scale factor*/
{
    sfx = (calculated_values_root_ptr->target_length_x/lengthx);
    scale_factor[0] = sfx;
    scale_factor[1] = sfx;
    scale_factor[2] = sfx;
}
else if (sfy <= sfx && sfy <= sfz)
{
    sfy = (calculated_values_root_ptr->target_length_y/lengthy);
    scale_factor[0] = sfy;
    scale_factor[1] = sfy;
    scale_factor[2] = sfy;
}
else if (sfz <= sfx && sfz <= sfy)
{
    sfz = (calculated_values_root_ptr->target_length_z/lengthz);
    scale_factor[0] = sfz;
    scale_factor[1] = sfz;
    scale_factor[2] = sfz;
}

/*****GENERAL SCALE*****/
printf("%f %f %f", scale_factor[0], scale_factor[1], scale_factor[2]);
for (i=0,p=0,m=3;i<=2,p<=2,m<=5;i++,p++,m++)
    point[i] = (sc_bounding_box[p]+sc_bounding_box[m])/2; /*Length x-y-z middle point*//*center of
bb - SMART POINT*/

    UF_CALL(UF_CSYS_create_matrix(matrix_values, &matrix_id));
    UF_CALL(UF_CSYS_create_temp_csys(point, matrix_id, &csys_id));
    UF_CALL(UF_MODL_create_scale(scale_type_2, &efd_object_ptr->first_efd_body->body,
num_body_tags_2, so_point_2, so_dir_2, csys_id, factors_2, &tags_2));

    UF_free(tags_2);

    UF_VIEW_fit_view(NULL_TAG,figit);
    UF_CALL(UF_MODL_update());

    efd_object_ptr = efd_object_ptr->next;
}
printf("- End of PN -\n -----\n");
printf(" - End of GEOMETRIC GUIDES -\n -----\n");
return(0);
}

```

D)Appendix: Import target

```

/*****
/* import_target - process 7 */
*****/
#include <stdio.h>
#include <uf.h>
#include <uf_modl.h>
#include <uf_disp.h>
#include <uf_view.h>
#include <uf_obj.h>
#include <uf_part.h>
#include <uf_trns.h>
#include "root.tem"
#include "pop.tem"
#include "results.tem"
#include "target.tem"

#define UF_CALL(X) (report( __FILE__, __LINE__, #X, (X)))
static int report( char *file, int line, char *call, int irc)
{
    if (irc)
    {
        char messg[133];
        printf("%s, line %d: %s\n", file, line, call);
        (UF_get_fail_message(irc, messg)) ?
            printf("    returned a %d\n", irc) :
            printf("    returned error %d: %s\n", irc, messg);
    }
    return(irc);
}
/*****
int import_target(struct root *root_ptr)
{
    int prompt(char *);
    int rcode;
    int count;
    int phenotypes;
    int big_number_of_phenotypes = 80;
    int small_number_of_phenotypes = 40;

    double acc_value[11] = {0.9};
    double rot_move_data_target[2][12] = { {0,0,0,0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0,0,0} };
    double dest_csys[6] = {1.,0.,0.,0.,1.,0.};
    double dest_point[3] = {50.,50.,50.};
    double scale = 1.0;

    struct calculated_values_root *calculated_values_root_ptr;
    struct species *species_ptr;
    struct target_values_root *target_values_root_ptr;

    char *filename = "C:\\\\EvoShape\\\\eS\\\\target_Laura.prt";

    uf_list_p_t list;
    UF_import_part_modes_t modes = { 1, 0, 0, 3, 0, TRUE, FALSE };

    tag_t group = NULL_TAG;
    tag_t inquiry = NULL_TAG;

    /*****IMPORT TARGET FILE*****/
    printf(" -----\\n - IMPORT TARGET -\\n");

    UF_CALL(UF_PART_import(filename,&modes,dest_csys,dest_point,scale,&group));

    UF_CALL(UF_MODL_create_list(&list));
    do
    {
        rcode = UF_MODL_ask_object(70,0,&inquiry);
        UF_CALL(UF_MODL_put_list_item(list,inquiry));
    }
    while (inquiry!=NULL_TAG);

```



```

UF_CALL(UF_MODL_ask_list_count(list,&count));
species_ptr = root_ptr->first_species_ptr;
target_values_root_ptr = species_ptr->target_values_root_ptr;

if (root_ptr->approach_mode[1] == 1) phenotypes = big_number_of_phenotypes;
if (root_ptr->approach_mode[2] == 1) phenotypes = small_number_of_phenotypes;

UF_CALL(UF_MODL_ask_list_item(list, phenotypes, &target_values_root_ptr->target));/*finds the tag of
the target*/
UF_CALL(UF_OBJ_set_color(target_values_root_ptr->target,183));/*changes the color of the target*/

/*****FINDING THE BOUNDING BOX OF THE TARGET OBJECT - TO FIND THE LENGTHS*****/
species_ptr = root_ptr->first_species_ptr;
calculated_values_root_ptr = species_ptr->calculated_values_root_ptr;
UF_CALL(UF_MODL_ask_bounding_box(target_values_root_ptr->target,calculated_values_root_ptr->
target_bounding_box));

/*****FINDING THE LENGTHS OF THE TARGET OBJECT - USE THEM IN SCALING*****/
calculated_values_root_ptr->target_length_x =
calculated_values_root_ptr->target_bounding_box[3]-
calculated_values_root_ptr->target_bounding_box[0];

calculated_values_root_ptr->target_length_y =
calculated_values_root_ptr->target_bounding_box[4]-
calculated_values_root_ptr->target_bounding_box[1];

calculated_values_root_ptr->target_length_z =
calculated_values_root_ptr->target_bounding_box[5]-
calculated_values_root_ptr->target_bounding_box[2];

return(0);
return(0);
}

```

E) Appendix: Compare values

```
/* **** */
/* compare_values - process 3.4 */
/* **** */
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <uf.h>
#include <uf_modl.h>
#include <uf_cfi.h>
#include <uf_ui.h>
#include <uf_obj.h>
#include "root.tem"
#include "pop.tem"
#include "target.tem"
#include "results.tem"
/* **** */
int compare_values(struct calculated_values_root *calculated, struct target_values_root *target, struct
phenotype *phenotype_ptr, struct root *root_ptr)
{
    int prompt(char *);

    int return_code = 0,
        run_mode;
    double sum = 0,
        sum1 = 0,
        sum2 = 0,
        p_diff_x = 0,
        p_diff_y = 0,
        p_diff_z = 0,
        temp_sum_one = 0,
        temp_sum_two = 0,
        difference = 0;
    /* **** */

    run_mode = root_ptr->ui_control_ptr->run_mode;

    /*Geometric Fitness*/
    if(root_ptr->evaluate_mode[0]>0 && root_ptr->evaluate_mode[6]!=1)
    {
        if(root_ptr->evaluate_mode[1]==1)
        {
            difference = calculated->h - target->bb_size;
            sum += sqrt(difference * difference);
        }
        if(root_ptr->evaluate_mode[2]==1)
        {
            difference = calculated->i - target->volume;
            sum += sqrt(difference * difference);
        }
        if(root_ptr->evaluate_mode[3]==1)
        {
            difference = calculated->j - target->area;
            sum += sqrt(difference * difference);
        }
        if(root_ptr->evaluate_mode[5]>0)
        {
            if(
                root_ptr->evaluate_mode[5]==100 ||
                root_ptr->evaluate_mode[5]==101 ||
                root_ptr->evaluate_mode[5]==110 ||
                root_ptr->evaluate_mode[5]==111
            )
            {
                printf("\ncalculated x : %lf",calculated->x);
                printf("\ntarget x : %lf",target->x_dim);
                difference = calculated->x - target->x_dim;
                printf("\ndifference : %lf",difference);
                temp_sum_one = difference * difference;
                p_diff_x = sqrt(temp_sum_one);
                printf("\nx-difference:%lf",p_diff_x);
                temp_sum_two += p_diff_x;
            }
        }
    }
}
```

```

        if(      root_ptr->evaluate_mode[5]==10 ||
                 root_ptr->evaluate_mode[5]==11 ||
                 root_ptr->evaluate_mode[5]==110 ||
                 root_ptr->evaluate_mode[5]==111      )
        {
            printf("\ncalculated y : %lf",calculated->y);
            printf("\ntarget y : %lf",target->y_dim);
            difference = calculated->y - target->y_dim;
            temp_sum_one = difference * difference;
            p_diff_y = sqrt(temp_sum_one);
            printf("\ny-difference:%lf",p_diff_y);
            temp_sum_two += p_diff_y;
        }
        if(      root_ptr->evaluate_mode[5]==1 ||
                 root_ptr->evaluate_mode[5]==11 ||
                 root_ptr->evaluate_mode[5]==101 ||
                 root_ptr->evaluate_mode[5]==111      )
        {
            printf("\ncalculated z : %lf",calculated->z);
            printf("\ntarget z : %lf",target->z_dim);
            difference = calculated->z - target->z_dim;
            temp_sum_one = difference * difference;
            p_diff_z = sqrt(temp_sum_one);
            printf("\nz-difference:%lf",p_diff_z);
            temp_sum_two += p_diff_z;
        }
        sum += temp_sum_two;
    }

}

/*Geometric Guides*/
printf("Phenotype %2d:\n", phenotype_ptr->phenotype_id);
if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]!=1) /*only gg*/
{
    phenotype_ptr->objective_function_value = sqrt(phenotype_ptr-
>non_intersected_volumes*phenotype_ptr->non_intersected_volumes);
    printf("non intersected volumes          = %f\n", phenotype_ptr-
>non_intersected_volumes);
    printf("Objective non intersected volumes = %f\n\n",phenotype_ptr-
>objective_function_value);
    if(run_mode==5) prompt("geometric guides fitness");
}

/*User Ratings*/
if(root_ptr->evaluate_mode[4]==1 && root_ptr->evaluate_mode[6]!=1) /*only user*/
{
    phenotype_ptr->objective_function_value = difference = 10 - phenotype_ptr->user_rating;
    printf("user rating difference          = %d\n", 10 - phenotype_ptr->user_rating);
    printf("Objective user rating            = %f\n\n",phenotype_ptr-
>objective_function_value);
}

/*Both*/
if(root_ptr->evaluate_mode[4]==1 && root_ptr->evaluate_mode[6]==1) /*Both gg and user ratings*/
{
    phenotype_ptr->objective_function_value = sqrt(phenotype_ptr-
>non_intersected_volumes*phenotype_ptr->non_intersected_volumes);
    printf("non intersected volumes          = %f\n", phenotype_ptr-
>non_intersected_volumes);
    printf("Objective non intersected volumes = %f\n\n",phenotype_ptr-
>objective_function_value);
    if(run_mode==5) prompt("geometric guides fitness");
}
if(run_mode==5) prompt("Next Phenotype");
return return_code;
}

```

F) Appendix: Convert to fitness

```
/* **** */
/* convert_to_fitness - process 3.5 */
/* **** */
#include <stdio.h>
#include "pop.tem"
#include <string.h>
#include <uf.h>
#include "root.tem"
#define null_member (struct member *)0
/* **** */
int convert_to_fitness( struct species *species_ptr, struct root *root_ptr)
{
    int merge_objective_functions(struct phenotype *, struct phenotype *);
    int return_code = 0;
    int prompt(char *);
    int run_mode;

    struct phenotype *phenotype_ptr;
    struct phenotype *target_phenotype_ptr;

    double scaling_factor;
    double maximum_value;
    /* **** */
    run_mode = root_ptr->ui_control_ptr->run_mode;

    maximum_value = 0.0;
    phenotype_ptr = species_ptr->first_phenotype;
    while (phenotype_ptr != null_phenotype)
    {
        if (phenotype_ptr->objective_function_value > maximum_value)
            maximum_value = phenotype_ptr->objective_function_value;
        phenotype_ptr = phenotype_ptr->next;
    }

    scaling_factor = species_ptr->fitness_scale_factor;
    phenotype_ptr = species_ptr->first_phenotype;
    if (species_ptr->decodes_ptr!=null_species)
        target_phenotype_ptr = species_ptr->decodes_ptr->first_phenotype;
    else
        target_phenotype_ptr = species_ptr->first_phenotype;

    while (phenotype_ptr != null_phenotype)
    {
        if (species_ptr->decodes_ptr!=null_species) /* M e r g e */
            return_code = merge_objective_functions(phenotype_ptr,target_phenotype_ptr);

        printf("Phenotype %2d :\n", phenotype_ptr->phenotype_id);
        if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]!=1) /*only gg*/
        {
            phenotype_ptr->fitness_function_value = scaling_factor / (scaling_factor +
            phenotype_ptr->objective_function_value);
            printf("gg Fitness values = %f\n\n", phenotype_ptr->fitness_function_value);
        }
        if(root_ptr->evaluate_mode[4]==1 && root_ptr->evaluate_mode[6]==1) /*Both gg and user
ratings*/
        {
            phenotype_ptr->fitness_function_value = scaling_factor / (scaling_factor +
            phenotype_ptr->objective_function_value);
            printf("gg Fitness values = %f\n\n", phenotype_ptr->fitness_function_value);
        }
        if(run_mode==5) prompt("Next Phenotype");
        target_phenotype_ptr = target_phenotype_ptr->next;
        phenotype_ptr = phenotype_ptr->next;
    }

    return(return_code);
}
```

G)Appendix: Subset of shapes to the user

```
/* **** */
/* p2.8b subset_to_the_user */
/* **** */
/*Select a subset of n objects by using g-guides */
/*present these n to the user. */
/* **** */
#include <stdio.h>
#include <uf.h>
#include <uf_ui.h>
#include <uf_obj.h>
#include <uf_modl.h>
#include <uf_udobj.h>
#include <uf_view.h>
#include <uf_disp.h>
#include "root.tem"
#include "pop.tem"
#include "geometry.tem"
#include "target.tem"
#define null_member (struct member *)0
/* **** */

int subset_to_the_user(struct root *root_ptr)
{
    int prompt(char *);
    int sort_phenotypes(struct species *);
    int real_blend(struct root *);
    int write_fit_values_gg_fittest(struct root *, struct species *);
    int interaction_loop(struct root *,int);

    int col=0,row=1,offset = 120;
    int return_code = 0;
    int run_mode;
    int sorted_counter;
    int fittest_phenotypes_no = 12;
    double figit = 1.0;
    double g_response[5] = {6,0.3,0,0,0};
    double scaling_factor;
    double tran_bounding_box[6];
    double tran_move_data[2][12] = { {0,0,0,0,0,0,0,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0,0,0} };
    tag_t inquiry = NULL_TAG;
    struct phenotype *phenotype_ptr;
    struct species *species_ptr;
    struct target_values_root *target_values_root_ptr;

    uf_list_p_t tran_feature_list;
    uf_list_p_t body_list;
    uf_list_p_t fit_phenotype_feature_list;
    uf_list_p_t target_feature_list;

    /* **** */
    printf("In subset_to_the_user now...\n");
    run_mode = root_ptr->ui_control_ptr->run_mode;

    printf("\nCalling sort_phenotypes...\n\n");
    return_code = sort_phenotypes(root_ptr->first_species_ptr);

    return_code = UF_MODL_create_list(&body_list);

    sorted_counter=1;
    phenotype_ptr=root_ptr->first_species_ptr->first_sorted;
    while (sorted_counter<=fittest_phenotypes_no)
    {
        return_code = UF_MODL_ask_body_feats(phenotype_ptr->efd_object_ptr->first_efd_body->body,&fit_phenotype_feature_list);
        return_code = UF_MODL_unsuppress_feature(fit_phenotype_feature_list);//12 fittest on
        printf("P%d Fit-val %F unsuppressed\n",phenotype_ptr->phenotype_id, phenotype_ptr->
        >fitness_function_value);//on
        sorted_counter++;
        phenotype_ptr=phenotype_ptr->next_sorted;
    }
}
```

```

/*****Blend the fittest*****/
return_code = real_blend(root_ptr);
/*****/

sorted_counter=1;
phenotype_ptr=root_ptr->first_species_ptr->first_sorted;
while (sorted_counter<=fittest_phenotypes_no)
{
    UF_DISP_set_display(UF_DISP_UNSUPPRESS_DISPLAY); //on

    /*****MOVE FITTEST OBJECTS*****/
    col++;
    if(col==5) /*sqrt(objects)*/
    {
        row++;
        col=1;
    }
    /*****FIND THE CENTER OF THE BASE OF EACH OBJECT*****/
    return_code = UF_MODL_ask_bounding_box(phenotype_ptr->efd_object_ptr->first_efd_body-
>body,tran_bounding_box);

    tran_move_data[0][0] = (tran_bounding_box[0]+tran_bounding_box[3])/2; /*FROM vector x*/
    tran_move_data[0][1] = (tran_bounding_box[1]+tran_bounding_box[4])/2; /*FROM vector y*/
    tran_move_data[0][2] = tran_bounding_box[2]; /*FROM vector z*/

    tran_move_data[1][0] = -((col*offset)+(offset/2)); /*TO vector x*/
    tran_move_data[1][1] = (row*offset)+(5*offset); /*TO vector y*/
    tran_move_data[1][2] = 0; /*TO vector z*/

    return_code = UF_MODL_ask_body_feats(phenotype_ptr->efd_object_ptr->first_efd_body-
>body,&tran_feature_list);
    return_code = UF_MODL_move_feature(tran_feature_list,0,tran_move_data);
    printf("----->Translate object\n\n");

    sorted_counter++;
    phenotype_ptr=phenotype_ptr->next_sorted;
}

species_ptr = root_ptr->first_species_ptr;
target_values_root_ptr = species_ptr->target_values_root_ptr;
return_code = UF_MODL_ask_body_feats(target_values_root_ptr->copy_target,&target_feature_list);
return_code = UF_MODL_suppress_feature(target_feature_list);
return_code = UF_VIEW_fit_view(NULL_TAG,figit);

write_fit_values_gg_fittest(root_ptr, root_ptr->first_species_ptr);

/*****FITTEST OBJECTS - USER RATINGS*****/
if(root_ptr->evaluate_mode[6]==1 && root_ptr->evaluate_mode[4]==1) /*Both gg and user ratings*/
{
    if(root_ptr->generation_no != g_response[0])
    {
        return_code = interaction_loop(root_ptr,phenotype_ptr->efd_object_ptr->first_efd_body-
>body); printf("%d\n", return_code);
        species_ptr = root_ptr->first_species_ptr;
        scaling_factor = species_ptr->fitness_scale_factor;
        phenotype_ptr=root_ptr->first_species_ptr->first_sorted;
        while (phenotype_ptr != null_phenotype)
        {
            phenotype_ptr->objective_function_value = 10 - phenotype_ptr->user_rating;
            printf("user rating difference = %d\n", 10 - phenotype_ptr-
>user_rating);
            printf("Objective user rating = %f\n\n",phenotype_ptr-
>objective_function_value);
            phenotype_ptr->fitness_function_value = scaling_factor / (scaling_factor +
phenotype_ptr->objective_function_value);
            if(phenotype_ptr->objective_function_value == 0.00)
            {
                phenotype_ptr->fitness_function_value = 2;
            }
            if(phenotype_ptr->objective_function_value == 11.00)
            {
                phenotype_ptr->fitness_function_value = 0;
            }
        }
    }
}

```

```

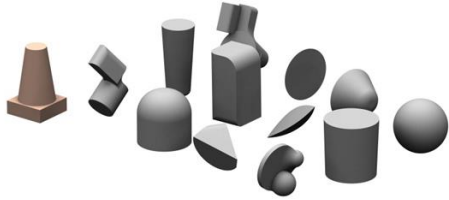
        if(root_ptr->ui_control_ptr->hidden_objects == 0 && phenotype_ptr->
>objective_function_value >= 10.00)
        {
            phenotype_ptr->fitness_function_value = 0;
        }
        printf("Fitness user rating = %f\n\n", phenotype_ptr->fitness_function_value);
        phenotype_ptr=phenotype_ptr->next_sorted;
    }
}
/*****/
return return_code;
}

```

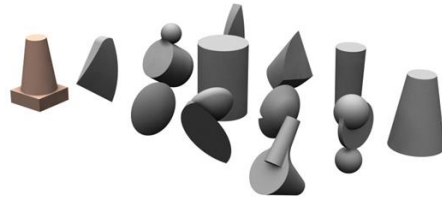
H)Appendix: Detailed benchmark Example

Shape matching – Generations 1-10

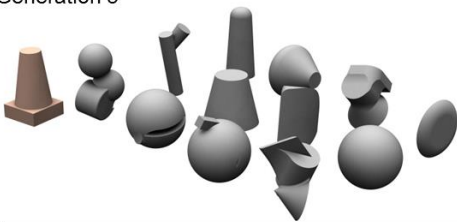
Generation 1



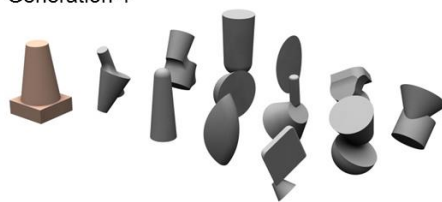
Generation 2



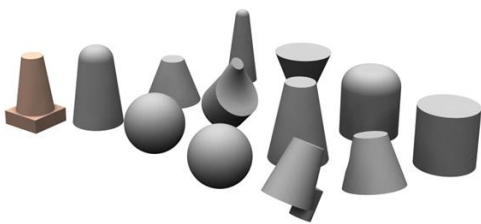
Generation 3



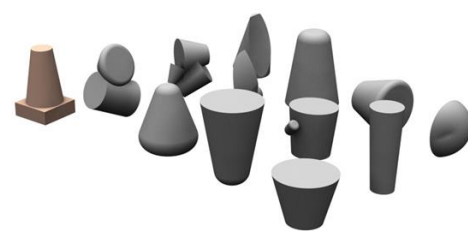
Generation 4



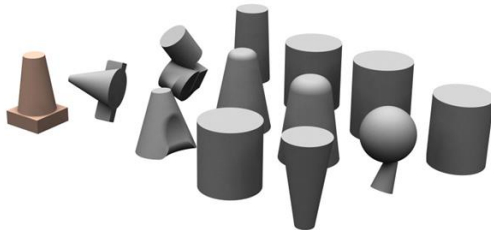
Generation 5



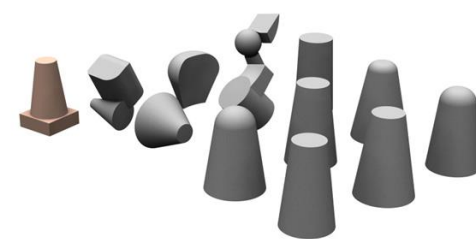
Generation 6



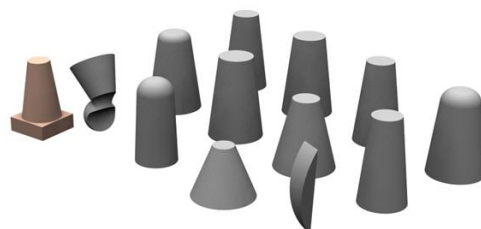
Generation 7



Generation 8



Generation 9

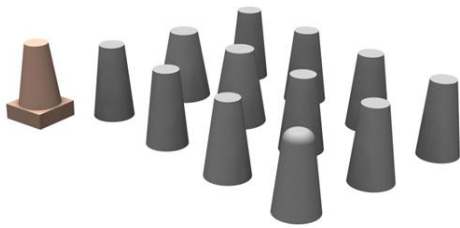


Generation 10

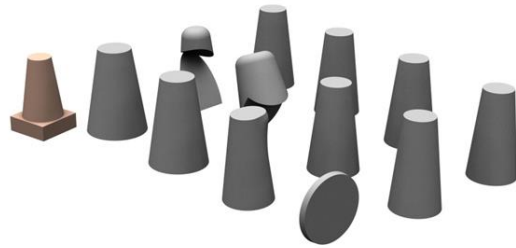


Shape matching – Generations 11-20

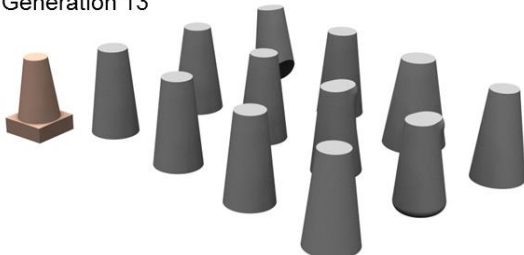
Generation 11



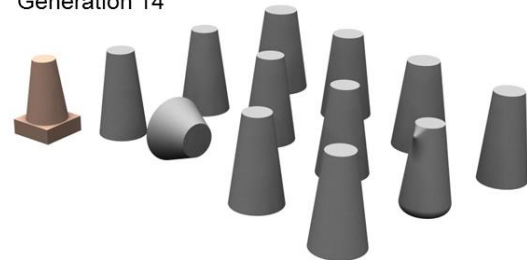
Generation 12



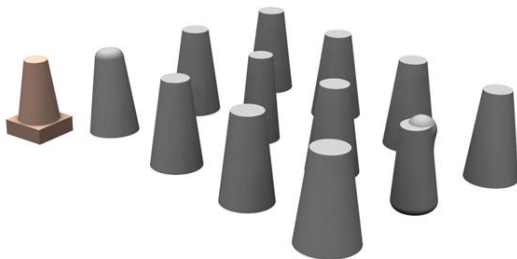
Generation 13



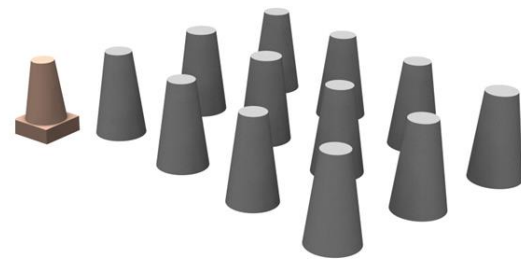
Generation 14



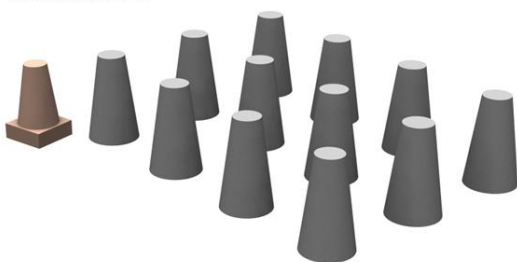
Generation 15



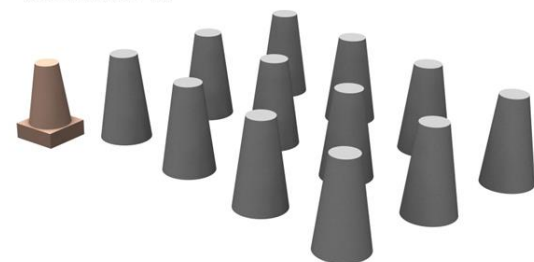
Generation 16



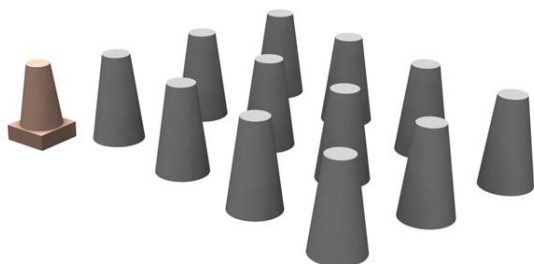
Generation 17



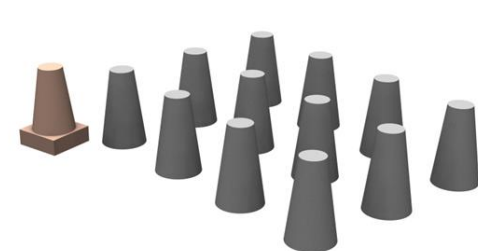
Generation 18



Generation 19



Generation 20



I) Appendix: Participants' Feedback

EvoShape - Participant 1 - Product Sofa

- 1.1 *"In the first population there is not much variety of shapes and none of the shapes looks like a sofa".*
- 1.2 *"In the second generation there is slightly more variety and there is a small similarity with the product I want to design".*
- 1.3 *"In the third generation there is again lack of variety, maybe because of the choices I have made. The similarity with the product is still small".*

LBP of 80 shapes - Participant 1 - Product Sofa

- 2.1 *"In the first generation, there is more variety than before (EvoShape without g-guides), more choices - but there is no similarity, with the product I would like to design, at all".*
- 2.2 *"In the second generation there is definitely variety of shapes and some similarity with the product as well".*
- 2.3 *"In the third generation there is good variety of shapes and more similarity with the product than before".*
- 2.4 *"In the fourth generation there is less variety but the shapes are getting lot closer to the product, same as in the fifth generation".*
- 2.5 *"In the last generation surprisingly there is again quite a lot of variety and the similarity with the product is high".*
- 2.6 *"This approach of EvoShape definitely gives useful ideas of shapes that I would not have thought".*

LBP of 40 shapes - Participant 1 - Product Sofa

- 3.1 *"In the first generation the variety is good and the shapes do not look like a sofa a lot but they are alright".*
- 3.2 *"In the second generation the variety is more than in the first generation and the similarity is pretty good – we have got some good shapes".*
- 3.3 *"In the third generation the variety is pretty good but there is less similarity with the product than the one before, same as in the fourth generation".*
- 3.4 *"In the fifth generation there is still good variety and the similarity is slightly better".*
- 3.5 *"In the last generation there is less variety and some similarity with the product".*
- 3.6 *"This approach of EvoShape gives some useful ideas of shapes as well".*
- 3.7 *"Both EvoShape with g-guides approaches have good variety of shapes, but the first approach (80 shapes) gives shapes that look like the product more than the second approach does".*
- 3.8 *"The first approach gives better ideas of useful shapes than the second one does".*
- 3.9 *"I prefer the first EvoShape with g-guides approach but I have also made better choices there".*
- 3.10 *"In general EvoShape with g-guides does not give bigger variety than EvoShape without g-guides but it gives more ideas of useful shapes that look like the product you would like to design".*
- 3.11 *"EvoShape with g-guides (80 shapes) is more useful if you have a specific shape in mind".*
- 3.12 *"EvoShape without g-guides is better when you have a general idea of what you want to create".*
- 3.13 *"Using g-guides is finally useful – it guides the process and gets rid of out-there ideas".*
- 3.14 *"What I would change in EvoShape with g-guides to make it better is: having a 'wild card', picking the 11 fittest shapes and creating one random shape in order to have something to 'save' you when you do not have very useful shapes. Maybe, I would do this by increasing the probability of mutation".*

3.15 *"EvoShape with g-guides is very easy to use".*

EvoShape - Participant 2 - Product Watch

4.1 *"In the first population there is variety of shapes but none of the shapes looks like a watch where in the second generation they started looking slightly like a watch".*

4.2 *"In the third generation there is still variety and the shapes started to look like more to the ideas I like".*

4.3 *"In the fourth generation there is less variety and in the fifth generation I have lost the shape I was carrying through".*

4.4 *"This version of EvoShape gives new ideas of useful shapes but, by the end of it, it wasn't exactly what I wanted".*

LBP of 80 shapes - Participant 2 - Product Watch

5.1 *"In the first generation there is less variety than the old EvoShape version, but is good. And, also, there is already some similarity with the product".*

5.2 *"In the second generation there is nice variation and the shapes are getting closer to the product, same as the third generation".*

5.3 *"In the fourth and fifth generation there is a variation in the details of the shapes and still a good similarity with the product".*

5.4 *"In the sixth generation the variety is not good and there are less interesting shapes".*

5.5 *"This EvoShape version definitely gives new ideas of useful shapes".*

LBP of 40 shapes - Participant 2 - Product Watch

6.1 *"In the first generation there is some variety and some shapes are quite similar to the product and in the second generation is getting even closer".*

6.2 *"In the third and fourth generations there is variation on the details and a lot of the shapes are quite similar to the product".*

- 6.3 *"In the fifth generation there is not a lot variation to inspire you but there are some good shapes that are similar to the product".*
- 6.4 *"In the sixth generation there is more variety (mutation) and a big similarity with the product".*
- 6.5 *"This approach (40 shapes) is better than the 80 shapes approach because the ideas are closer to a watch. However, the 80 shapes approach has bigger variety of shapes and gives better ideas of useful shapes".*
- 6.6 *"EvoShape with g-guides has less variety than the EvoShape without g-guides but finds the shape easier; you evolve your concept easier –nicer. EvoShape without g-guides is too random".*
- 6.7 *"EvoShape with g-guides is finally useful; it is a shape exploration that gives new ideas of useful shapes".*
- 6.8 *"To make EvoShape better I would keep certain features that I like and bring them together better".*
- 6.9 *"EvoShape is easy to use".*

EvoShape - Participant 3 - Product Building

- 7.1 *"In the first generation there is a lot of variety but nothing to use directly as a building structure".*
- 7.2 *"In the second generation there is still variety and quite interesting shapes".*
- 7.3 *"In the third generation there are no interesting or useful shapes".*

LBP of 80 shapes - Participant 3 - Product Building

- 8.1 *"In the first generation there are interesting and complex shapes and the variety is high".*
- 8.2 *"In the second generation there are nice ideas of shapes, less complex, the variety is smaller and it is quite settled".*
- 8.3 *"In the third generation the shapes are closer to the target already".*
- 8.4 *"In the fourth generation the shapes are cool but the variety small".*

LBP of 40 shapes - Participant 3 - Product Building

9.1 *"In the first generation there is good variety and nice shapes".*

9.2 *"In the second generation there are new ideas of shapes but not much variety".*

9.3 *"In the third generation there are more versions of the same idea".*

9.4 *"The 80 shapes approach has better first generation than the 40 shapes approach – better variety and better ideas of shapes".*

9.5 *"I prefer the 80 shapes approach because it gives you more ideas for the whole object – considers the whole product, gives more applicable ideas to the whole product".*

9.6 *"LBP approach is useful but not better in every way, but it really has value".*

EvoShape - Participant 4 - Product Guitar

10.1 *"In the first generation there is not a lot of variety and nothing to inspire the form of a guitar".*

10.2 *"In the second generation there is still small variety and not very interesting shapes".*

10.3 *"In the third generation the variety is slightly better and there are some interesting shapes".*

10.4 *"In the fourth generation there is still some variety but there are no useful shapes any more".*

11.5 *"In the fifth generation there is no variety and no interesting shapes".*

LBP of 80 shapes - Participant 4 - Product Guitar

11.1 *"In the first generation the variety is high but there are no interesting shapes".*

11.2 *"In the second generation the variety is still good and there are some good ideas to inspire".*

11.3 *"In the third generation there is still variety and the shapes are getting more interesting".*

11.4 *"In the fourth generation the variety is small but there are some useful shapes".*

11.5 *"In the fifth generation there is not much variety of shapes but they are very close the target".*

LBP of 40 shapes - Participant 4 - Product Guitar

12.1 *"In the first generation there is good variety but there is no similarity with the target".*

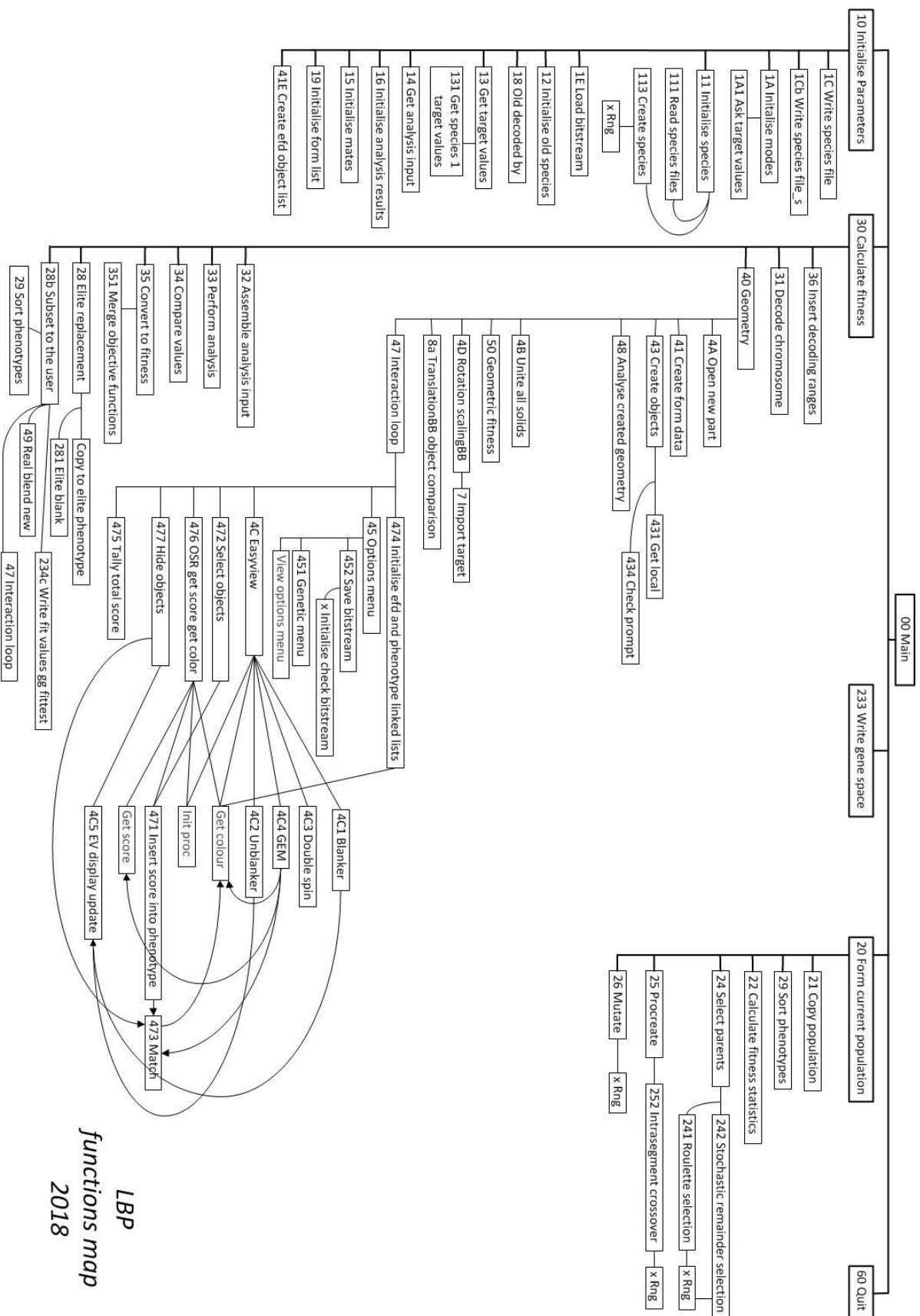
12.2 *"In the second, third and fourth generations there is good variety and the similarity with the target is increasing".*

12.3 *"In the fifth generation the variety is very small but the shapes are very close to the target".*

12.4 *"GG are finally useful and have improved EvS by guiding the process".*

12.5 *"GG could be improved by creating a database of different classes of shapes and using the right class with the right type of target".*

J) Appendix: LBP Functions Map



SUI
functions map
2018



L) Appendix: Publications

Published:

T.Retzepe, Y. M. Goh, I. J. Graham. "3D Object Comparison with Geometric Guides for Interactive Evolutionary CAD". ICMR International Conference on Manufacturing Research, Greenwich, United Kingdom, 5 Sept 2017, Vol. 6, pages 421-426.

T.Retzepe, Y. M. Goh, I. J. Graham. "3D Alignment for Interactive Evolutionary Design". ICMR International Conference on Manufacturing Research, Loughborough, United Kingdom, 7 Sept 2016, pages 281-286.

3D Object Comparison with Geometric Guides for Interactive Evolutionary CAD

Theodora RETZEPI, Yee Mey GOH, Ian J GRAHAM

*Wolfson School of Mechanical, Electrical and Manufacturing Engineering,
Loughborough University, Loughborough, LE11 3TU, UK.*

Abstract. 3D object comparison is presented as part of research into guided evolutionary Computer-Aided Design (CAD). CAD technology in development will combine human interaction and geometric optimization, within an existing evolutionary design system (EvoShape). Geometric Guides consist of simple 3D target objects (bounding volumes), to which evolving forms are compared. Before comparison, objects must be aligned and scaled, a process known as Pose Normalization (PN) in the literature. Both PN and object comparison have been implemented using standard geometric functions, enabling populations of evolving forms to be directed by the Geometric Guides. The algorithms and their implementation are presented alongside early results and analysis, discussion on limitations and robustness, and their suitability for Interactive Evolutionary CAD.

Keywords. CAD, Pose Normalization, Object Comparison, Geometric Guides, Interactive Evolutionary Computation, Interactive Optimization.

1. Introduction

For over two decades, Evolutionary Computation has formed the basis of evolutionary design systems which are traditionally either automated or interactive, with the latter well-suited for messy problems that are hard to evaluate using simulation (e.g. our primary application of form ideation within product design). Recently, relatively easy-to-use CAD applications combining elements of user interaction and engineering optimization have appeared in major proprietary CAD systems, particularly in the area of topology optimization. Our research interests are related, but are distinctive in focusing on highly interactive methods (Interactive Evolutionary Computation) supporting more subjective design activities (aesthetic concept generation).

EvoShape is a CAD application based on Interactive Evolutionary Computation (IEC), developed from the original Genetic Algorithm (GA) research described in Case et al [1]. It runs within a CAD system, utilizing its geometric modeller and user interface. The system relies solely on users to guide the evolution, from a random starting population of 3D shapes, by selecting or rating these shapes. The aim is to introduce more control by allowing users to create simple guiding geometry at the start of the evolution/exploration process. This has required research into Pose Normalization and object comparison, both of which have been implemented using standard geometric functions available within a CAD system, via its Application Programming Interface (API).

2. Pose Normalization

The CAD application being developed by the authors requires users to create a simple 3D geometric guide (hereafter, ‘target’), before they start to generate and explore 3D shapes (hereafter, ‘forms’). A target, which acts as a soft-constraint, may represent the overall proportions sought. To compare the evolving forms with the target, 3D object comparison algorithms are being developed (Section 3). For these to work effectively, forms need to be aligned with the target; this is commonly referred to in the specialist literature as “Pose Normalization” (PN) [2][3][4][5][6].

One driver for PN research is the need to search databases of CAD geometry (e.g. parts libraries). It is also highly applicable to the object comparison application in the IEC design system being developed. Before any processing, and due to arbitrary 3D orientation, scale, and position, a 3D alignment that finds the fittest position of each form compared to the target is required. PN is thus essential to align the target with the evolving forms being generated by the combination of GA and CAD modeller.

Basic PN has been investigated using readily-available deterministic geometric properties such as Center-of-Mass (CoM), Principle Axis (PA) and Bounding Box (BB). The results look to offer appropriate efficacy for the application – and objective tests to confirm this are being devised. However, sophisticated algorithms have been identified in the literature [2] should they be required. Rotation, scaling and translation are carried out in that order, and presented in the following sections.

2.1. Rotation

Aligning PA of target and form is computationally efficient. It is not perfect, and can produce counter-intuitive results, as shown in Figure 1 and detailed in our previous paper [7] – but these deviations are rarely significant within the context of the guiding application. The principal X Y Z axes of forms and target are identified, and aligned with the global axes. This has the effect of ‘standing all the objects up’ (as seen in Figure 3). Early user-testing has indicated that users would prefer the original orientation of the target to be maintained (some products naturally lie down, furniture for instance, and look odd aligned otherwise). The centre of the BB of each form is used as the centre of rotation – this works better than using the CoM, resulting in less of a jump. Functionally this is unimportant as the third translation step accounts for any undue movement of the object, but it produces a more pleasing result visually, if users are watching the analysis on screen.

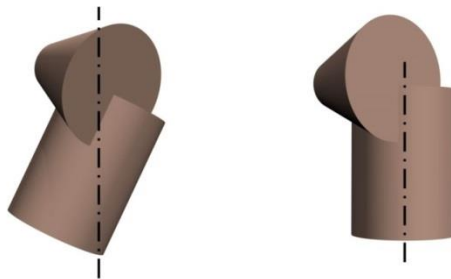


Figure 1. An object’s Principle Axis (PA) aligned with the Z-axis (left) compared to a more intuitive alignment of the object (right), where the PA of the dominant geometry, the cylinder, is used.

2.2. Scaling and translation

Firstly, the X, Y and Z lengths of each target and form's BB are established. Then the ratios (X Y Z lengths of the target BB divided by X Y Z lengths of the form BB) are calculated, and the form is scaled uniformly according to the smallest ratio of the three. This has the advantage of keeping the BB of the form within the BB of the target. It produces reasonable scaling estimations, preferred to the results of matching volumes (seen in Figure 2), and achieves robustness with typical EvoShape geometry.

The centre of the base of each form's BB is used as its local origin within each cell of a 4 x 3 grid, and to co-locate the target before object comparison. This works better than using the CoM, producing a more even spacing between forms (Figure 3).

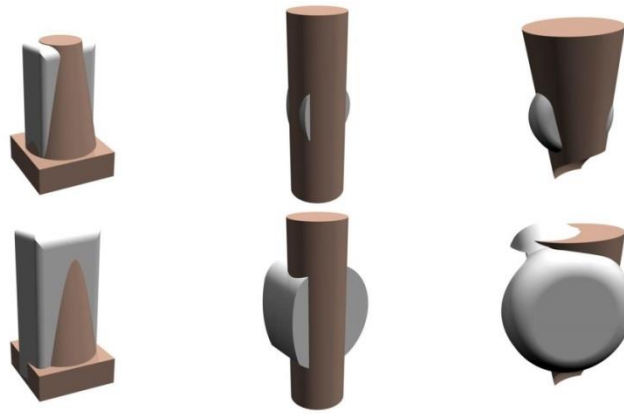


Figure 2. Scaling using BBs (upper): the BBs of the white forms do not exceed the BB of the brown targets. Scaling by volume (lower) is less satisfactory, and the form BBs generally exceed that of the target.

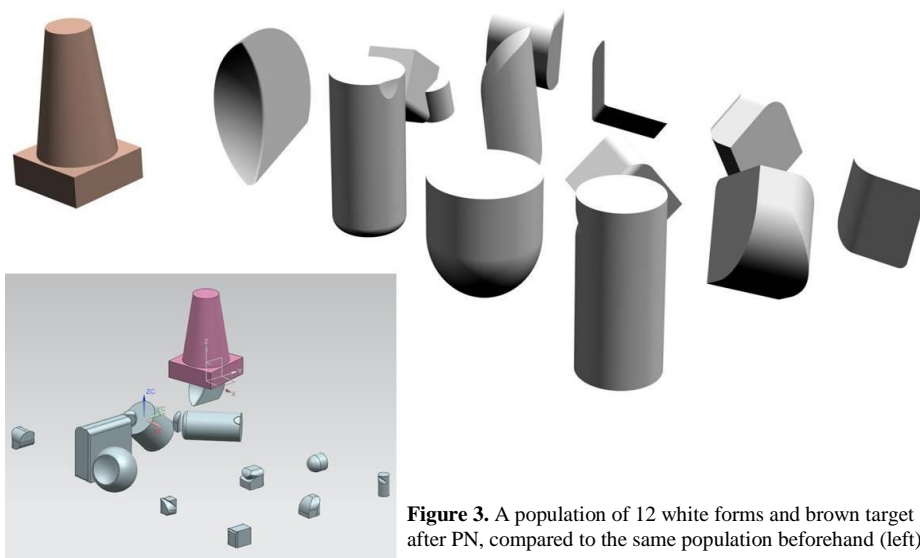


Figure 3. A population of 12 white forms and brown target after PN, compared to the same population beforehand (left).

3. Object Comparison

Once form and target are aligned, Boolean operators are used to calculate how closely they match. Our approach is based on the principle of non-intersecting volumes, i.e. any material volume outside the intersection of the two objects counts as a penalty. This can arise from target material not within the objects' intersection, and material from the form not within the objects' intersection. The sum of these two volumes acts as an Objective Function representing how closely the two objects match, shown graphically in Figure 4 and mathematically in equation (1). The sum non-intersecting volumes tend to zero as the evolving forms more closely match the target.

$$S_f = (V_{target} - (V_{target} \cap form)) + (V_{form} - (V_{target} \cap form)) \quad (1)$$

This can be simplified to: $S_f = V_{target} + V_{form} - 2(V_{target} \cap form)$.

- S_f is the Objective Function of an evolving form.
- V_{target} and V_{form} are the volumes of target and form.
- $V_{target} \cap form$ is the intersecting volume of target and form.

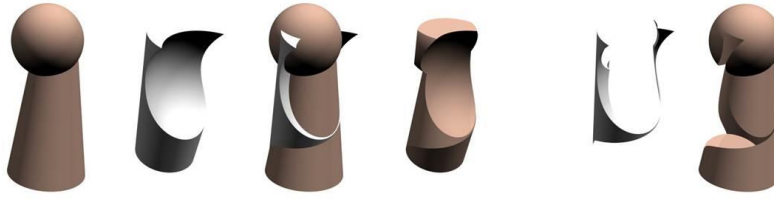


Figure 4. From left to right: target; form; after PN; intersection; non-intersecting volumes

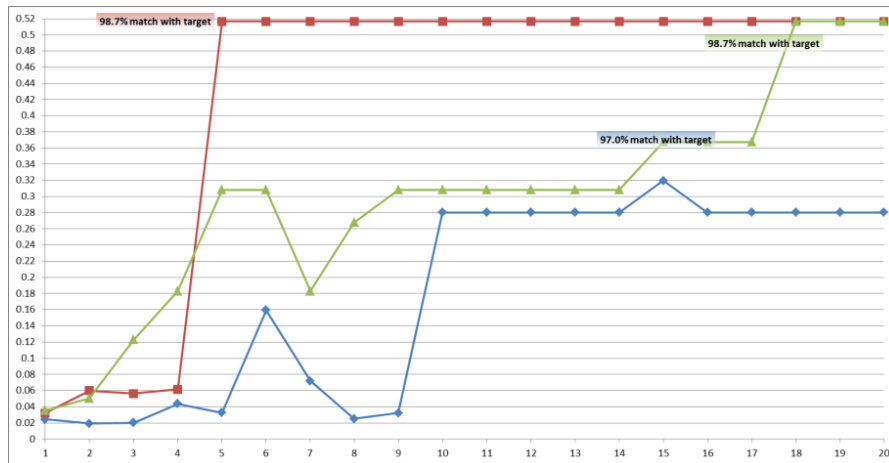


Figure 5. Max fitness, Y, vs generation, X, shows evolution of three different (random) starting populations.

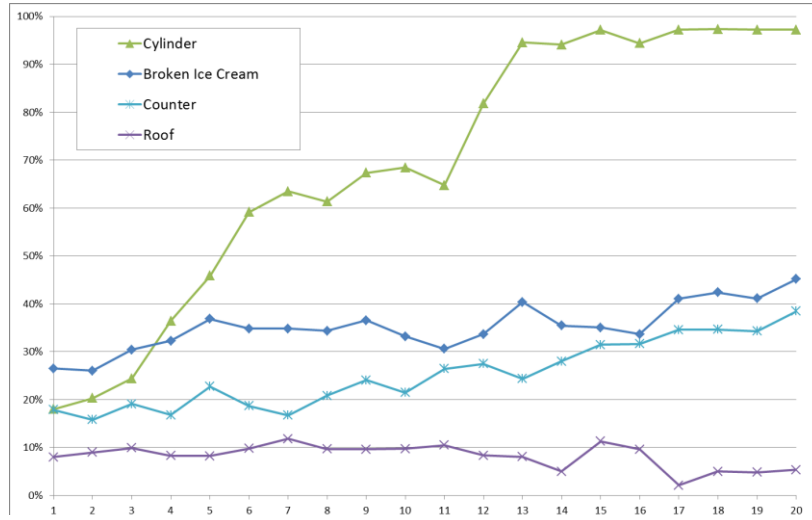


Figure 6. % match, Y, vs. generation, X, shows the evolution guided by four different targets.

4. Results

The IEC design system's aim is not to 'find the target', but this task is used here to assess the efficacy of the PN and object comparison. A form's % match with the target is sometimes used as a more intuitive expression than fitness functions.

Figure 5 shows that starting populations affect outcome. The graph shows one starting population achieves a maximum fitness corresponding to a 99% match (with the Cylinder target) after 4 generations. With 12 objects in each population this amounts to 48 total forms evaluated. A different starting population reaches the same fitness but not until the 18th generation (G18). A third reaches only a 97% peak in G15 but then settles at 96%. For this reason, the data presented in Figure 6 for four different targets are the mean of three different random starting populations. The graphs also show that maximum fitness drops at certain points –

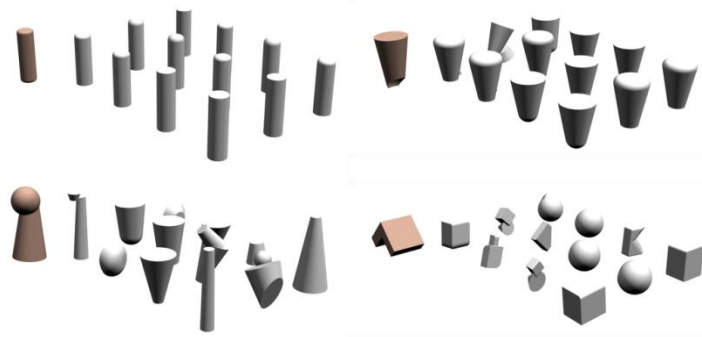


Figure 7. Evolved populations alongside their target shapes, in order of effectiveness: Cylinder, G15 (upper-left); Broken Ice Cream, G20 (upper-right); Counter, G20 (lower-left); Roof, G15 (lower-right).

this will be addressed by introducing a GA technique called ‘elite replacement’, where the fittest member is automatically carried over into the next population.

5. Conclusions

Our PN and object comparison algorithms, implemented with standard geometric functions, successfully enable meaningful objective function allocation to evolving forms. Simple targets (Cylinder, Broken Ice Cream in Figure 7) produce excellent results – accurate and efficient. The plain cylindrical target populations converge at G15 with a 97% average match and a 99% best. The inverted cone shape target populations average 45% by G20 (80% best). The system is less effective as target complexity increases, particularly where targets cannot be represented with single geometric primitives (e.g. Counter, 38% average by G20, 70% max), especially where these kinds of targets occupy a low % of their BB (e.g. Roof, 11% mean match in G15, but no improvement over time, or convergence). Although the fitness of some of these populations is still rising at G20, our belief is that object comparison methods should be investigated further in order to improve efficacy.

5.1. Related and future work

The next step in the research is combining g-guides with user-selection; to investigate and develop the human-machine interaction strategies which best complement the evolutionary design process. There are two broad approaches: larger populations, or more generations. The former involves processing larger populations ‘behind-the-scenes’ and using the g-guides to select a subset of 12 forms to present to the user. The latter option involves keeping the population sizes at 12, using g-guides to evolve the majority of populations but presenting every n^{th} generation to the user. The results presented in this paper suggest that larger populations will have the greatest benefit. Progress on this aspect of the research will be presented at the conference.

6. References

- [1] K. Case I. Graham R. Wood, “Shape modification using genetic algorithms,” *Eng. Manuf.*, vol. 218, pp. 827–832, 2004.
- [2] P. Papadakis, “Enhanced pose normalization and matching of non-rigid objects based on support vector machine modelling,” *Pattern Recognit.*, vol. 47, no. 1, pp. 216–227, Jan. 2014.
- [3] T. Farrugia and J. Barbarar, “Pose normalisation for 3D vehicles,” *Comput. Anal. Images Patterns*, pp. 235–245, 2015.
- [4] A. Axenopoulos, G. Litos, and P. Daras, “3D model retrieval using accurate pose estimation and view-based similarity,” in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval - ICMR '11*, 2011, pp. 1–8.
- [5] E. Paquet, M. Rioux, A. M. Murching, T. Naveen, and A. J. Tabatabai, “Description of shape information for 2D and 3D objects,” *Signal Proc. Image Commun.*, vol. 16, pp. 103–122, 2000.
- [6] S. P. P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, “3D object retrieval using an efficient and compact hybrid shape descriptor,” pp. 9–16, 2008.
- [7] T. Retzepi I. J. Graham Y. M. Goh, “3D Alignment for Interactive Evolutionary Design,” in *Procs. International Conference on Manufacturing Research*, 2016.

3D Alignment for Interactive Evolutionary Design

Theodora RETZEPI¹, Ian J GRAHAM, Yee Mey GOH
*Wolfson School of Mechanical, Electrical and Manufacturing Engineering,
Loughborough University, Loughborough, LE11 3TU, UK.*

Abstract. 3D model alignment ('Pose Normalization' in the literature) is investigated as part of wider research into guided evolutionary Computer-Aided Design. CAD technology in development will combine human interaction and geometric optimization, within an evolutionary design system. Evolving shapes will be influenced by simple pre-set geometric fuzzy-constraints – internal voids and external bounding geometry created by users. To compare evolving candidate shapes with these pre-set constraints they must first be aligned (rotated, scaled, and co-located). A shortlist of five promising alignment techniques is described. Benchmark data generated using standard CAD functions (centre of gravity, principle axes etc.) will be presented at the conference.

Keywords. Computer Aided Design, Pose Normalization, Interactive Evolutionary Computation, Geometric Evolutionary Optimization.

1. Introduction

Evolutionary Computation (EC) is often used for engineering and design problems that are too complex to tackle deterministically, and is generally focused on optimization and the later stages of design. EC is usually either automatic or interactive, with the latter well-suited for messy problems that are hard to model (e.g. the authors' primary area of interest of ideation within product design). Usable Interactive Evolutionary Computation (IEC) systems for design are emerging, but our research interests are distinct in focusing on Computer-Aided Design (CAD) methodologies to support early-stage concept generation by combining engineering optimization and IEC.

EvoShape (Figure 1) is a CAD application based on IEC, developed from original research into a Genetic Algorithm (GA) based system described in Graham et al [1]. It runs within a CAD environment, utilizing its geometric modeller and User Interface. Users guide the evolutionary process, from a random starting population of 3D shapes, purely through shape selection or rating.

The intention is to introduce more control to the users by allowing them to create simple guiding geometry at the start of the shape evolution process. This intention has resulted in the need to investigate 3D alignment techniques, the five most promising of which, shortlisted from a study of around 10, are analysed and compared in this paper.

¹ Corresponding Author. T.Retzeppi@lboro.ac.uk

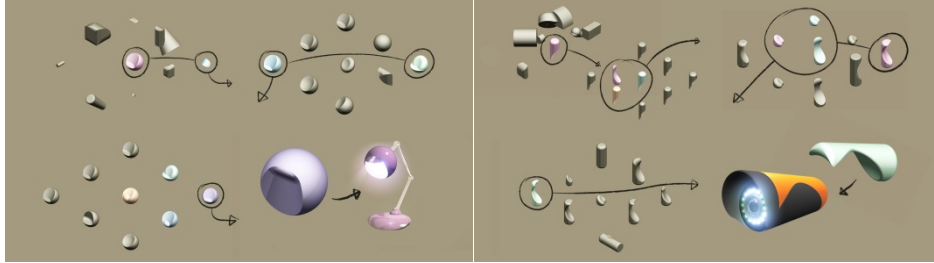


Figure 1. EvoShape.

2. Pose Normalization

The CAD application being developed by the authors requires users to create simple bounding geometry and internal voids, before they start to generate and explore 3D shapes. Bounding geometry may represent the overall proportions sought, and the voids could represent components within the products. To compare the evolving shapes with these soft-constraints, 3D form-comparison algorithms are being developed. But for these to work effectively, candidate and target shapes need to be aligned; this is commonly referred to in the literature as “Pose Normalization” (PN) [2,3,4,5,6].

2.1. Principal Component Analysis Methods

Farrugia et al [3] investigate various PN techniques for 3D vehicle models, concluding that there are two main techniques for rotational normalization:

- Principal Component Analysis (PCA) algorithm
- Computation of the symmetrical planes of a 3D model.

The symmetrical nature of vehicles should support the use of the symmetrical plane method but there is only one symmetrical plane on each vehicle. Farrugia et al [3] tested four different PCA methods on a downloaded database. These methods are:

- Principal Component Analysis (PCA)
- Centre of Gravity-PCA (CoG-PCA) [5]
- Normal-PCA (NPCA) [6] and
- Continuous-PCA (CPCA) [7]

The CoG-PCA method calculates the CoG of each mesh face rather than the actual vertices. In NPCA technique, the principal axes are identified by the covariance of the mesh face and not by the vertex points. The CPCA is the PCA method applied to an infinite continuous point set rather than a traditional discrete point set.

All these developments of the PCA differ from the original PCA only in the computation of the covariance matrix and they have been studied because many researchers have shown that the PCA technique may lead to inconsistent results.

Using the experimental results given by the authors, the CoG-PCA method was more effective regarding the computation time. The CPCA method had slightly better performance [8] than the CoG-PCA and noticeably better performance than the PCA and NPCA. Regarding reflection normalization, PCA was inadequate [9] while NPCA had inaccurate results in the three axes’ identification.

The CPCA-based PN method, in some cases can be un-successful in detecting some specific characteristics of 3D models such as symmetries, but they should be sufficiently effective when applied to IEC design systems.

2.2. Combined Pose Estimation (CPE)

Axenopoulos et al [4] combine plane reflection symmetry and recti linearity² to attain a 3D model alignment using the PCA method. Firstly, the 3D object is translated by locating the centre of the mass to the centre of the coordinate system, and then is scaled to fit within a bounding sphere. The object generates 2D views, and 2D rotation-invariant functionals are applied for each view. Rotation estimation takes place using a novel CPE method which combines the CPCA with plane symmetry and recti linearity.

If there are two or three planes of symmetry the transformation is kept as it is, otherwise the algorithm needs to be corrected using recti linearity. Recti linearity is invariant to scaling, translating and rotating. PCA and recti linearity are taking place at the same time and the one that produces the best rotation estimation is chosen.

After alignment, a set of 2D black/white views is extracted, and from each 2D view a descriptor vector is also extracted. These vectors are converted to a one-descriptor vector which describes the shape of the object. View-based similarity is computed by matching each 2D view between 3D models, so that the 3D models to be properly aligned in a coordinate frame.

The experimental results given by Axenopoulos et al [4] prove that CPE produce accurate rotation estimation results compared with the comparative techniques, and should achieve robustness in IED systems that use simple models. However, using the centre of the mass for translation normalization will not produce consistent alignment in more complex objects, as analysed in a later section.

2.3. Reflective Symmetry

A reflective symmetry computed on panoramic³ views is used for a novel pose normalization method, as described by Sfikas et al. [10]. The symmetry plane of a 3D model is detected and the first axis of the model is computed. The other principal axes are computed by calculating the variance of the panoramic views. First, on a chosen cylinder the surface of the model is projected and aligned with a principal axis in space. A panoramic view representation is created by unfolding the 3D model over a 2D image plane. The 3D model's plane of symmetry is defined by the axis of the projection cylinder and the axis of maximum reflective symmetry. The 3D model's centroid is used to achieve translation normalization, and scale normalization is followed so that the 3D model exactly fits inside the unit sphere. A rotation of the symmetry plane takes place by orientating the surface so that it includes the Z axis, and the plane of symmetry is detected in the panoramic image where the symmetry score graph is extracted, as shown in Figure 2.

² The maximum ratio of the surface area to the sum of three orthogonal projected areas of the mesh.

³ A panoramic view is a wide-angle view in a three-dimensional model.

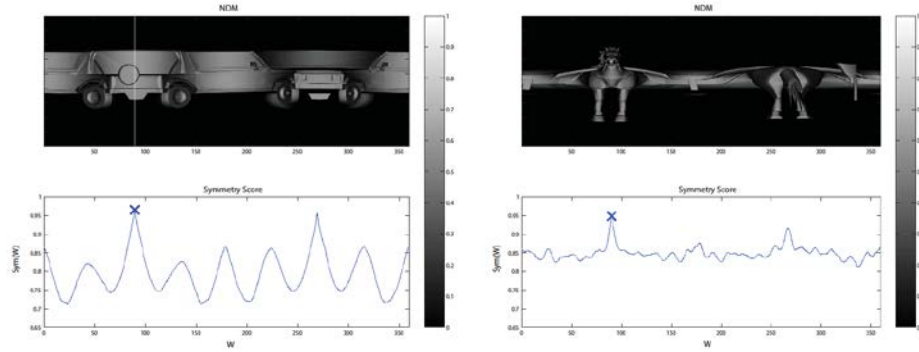


Figure 2. Panoramic images representing symmetric planes and associated symmetry score graphs [10].

Experimental results show that this method is accurate and with good performance but cannot handle complex 3D objects because symmetry detection either focuses on small fragments or bigger abstract areas of the 3D objects. In IEC design systems that use simple 3D models the reflective symmetry algorithm should achieve robustness. But systems with more complex models could produce inefficient alignments when using this method.

2.4. Non-rigid Shapes

Papadakis [2] describes the use of One-Class Support Vector Machines (OCSVM) to increase the consistency of translation and scale normalization under non-rigid shapes.

In translation normalization, the centre of mass of a 3D model is usually computed and located to the coordinates' origin. However, this technique is not effective in 3D objects that are articulated or have extruding parts and outliers (Figure 3).

To alleviate this problem, Papadakis [2] considers the surface of a 3D object as a collection of 3D points and use OCSVM to compute the decision surface and find the volume constrained within the boundaries of the decision surface. The centre of the object is the centroid of the distribution of the volume. The algorithm is identifying parts that are extremely small regarding the whole shape and that could derive negative results during the computation of the translation and scale.

In scale normalization, fitting a 3D model with these characteristics inside a unit cube is not effective due to the possible presence of outlying parts (Figure 3).

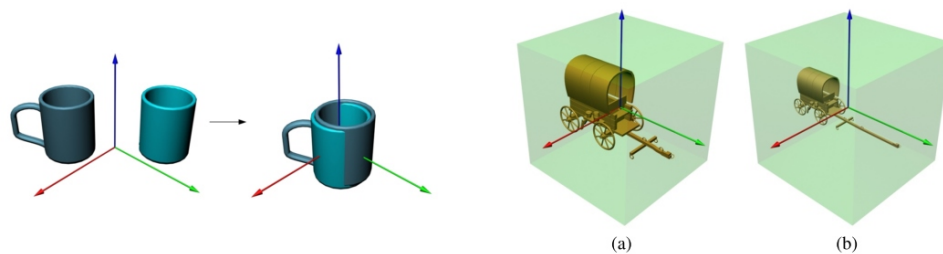


Figure 3. Left: Translation normalization using the centroid of the surface of the object, example in an object with and without extruding parts. Right: Scale normalization to the unit cube [2].

To normalize the scale, they calculate the average distance of the distribution of the object from its centre and scale the object in order this distance to be unit.

In the presence of non-rigid transformations, a method similar to the one described by Sfikas et al. [10], that depends on the symmetrical properties of the objects, becomes less descriptive. Many state-of-the-art approaches cannot accommodate 3D objects with articulations or extrusions, because the assumption that the centre of a 3D object is the centre of mass of the surface is not robust. The OCSVM methodology alleviates this problem and can probably be applied to IEC design systems that handle complex 3D objects with extruding and outlying parts, performing more consistent translation and scale normalization.

2.5. Planar-Reflective Symmetry Transform

Podolak et al. [11] describe a planar reflective symmetry transform (PRST) for 3D models that captures a continuous measure of the reflectional symmetry of a shape respectfully to all potential planes. This transform is computed by a Monte Carlo sampling algorithm which is constant under transformations, and determines the centre of symmetry as well as the principal symmetry axes which are essential for aligning models in a canonical system.

As shown in Figure 4, the darkest point (which indicates the greatest symmetry) is the centre of the object, and the main lines are the main axes. Alignments are usually computed with PCA using the centre of mass as the origin and the principal axes as the orientation. However, the results are not always correct (Figure 4).

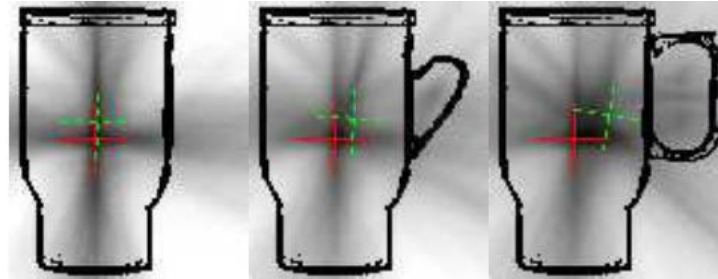


Figure 4. The PRST captures the degree of symmetry of arbitrary shapes with respect to reflection through all planes in space. The centre of mass and PCA axes are drawn in dotted green (they move depending on the presence of handles). The centre of symmetry and principal symmetry axes using PRST are shown in solid red (they remain constant under transformation of the shape) [11].

Podolak et al. [11] produce better alignments using PRST and introducing the centre of symmetry (COS) as well as the principal symmetry axes (PSA). The plane with maximal symmetry is the first PSA, the maximal symmetry perpendicular to the first is the second PSA, the maximal symmetry perpendicular to both first and second axis is the third PSA. Finally, the intersection of those three planes is the centre of symmetry. This algorithm should achieve efficient normalization when applied in IEC design systems as the centre of symmetry and principal symmetry axes remain constant under transformation of shapes.

3. Conclusions

The main driver for PN research is the need to search databases of CAD geometry (e.g. parts libraries). Conveniently, it is also applicable to shape-comparison applications in the IEC design system being developed. Here, PN is essential to align pre-set guiding geometry with evolving forms generated by the combination of GA and CAD modeller. Five suitable PN candidates for IEC design system research were presented and compared. Since the IEC design system generates quite abstract and complex 3D models, the PCA-based PN method is recommended. This should be combined with a robust method similar to OCSVM to find the best ‘centre’ of objects for translation.

3.1. Related and Future Work

Basic PN functionality has been achieved using the CAD functions available through an Application Programming Interface (centre of mass, minimized bounding box, and principle axes). It is likely that this approach is not sufficiently accurate for the application, hence the parallel research into more sophisticated PN techniques. The next step is to develop shape-comparison algorithms, which will be used to allocate objective functions to evolving shapes – these will be combined with user scores to direct the evolutionary process. The results of this work will be presented at the conference for interest, and for future benchmarking purposes.

References

- [1] K. Case, I. Graham, and R. Wood, Shape modification using genetic algorithms, *Procs. Inst. Mech. Eng. Part B: J. Eng. Manufacture*, **218**(7) (2004), 827-832.
- [2] P. Papadakis, Enhanced pose normalization and matching of non-rigid objects, *Pattern Recognit.* **47** (2014), 216–227.
- [3] T. Farrugia and J. Barbarar, Pose normalisation for 3D vehicles, *Comput. Anal. Images Patterns* (2015), 235–245.
- [4] A. Axenopoulos, G. Litos, and P. Daras, 3D model retrieval using accurate pose estimation and view-based similarity, *Procs. 1st ACM Int. Conf. Multimedia Retrieval - ICMR* (2011), 1–8.
- [5] E. Paquet, M. Rioux, A. M. Murching, et al, Description of shape information for 2D and 3D objects, *Signal Proc. Image Commun.* **16** (2000), 103–122.
- [6] S. P. P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, 3D object retrieval using an efficient and compact hybrid shape descriptor, *Eurographics Workshop on 3D Object Retrieval* (2008), 9–16.
- [7] D. V. Vranic, Desire: A Composite 3D-Shape Descriptor, *IEEE Int. Conf. Multimedia and Expo.* (2005), 962–965.
- [8] J. W. H. Tangelder and R. C. Veltkamp, A survey of content based 3D shape retrieval methods, *Multimed. Tools Appl.* **39**(3) (2008), 441–471.
- [9] N. Pears, T. Heseltine, and M. Romero, From 3D point clouds to pose-normalised depth maps, *Int. J. Comput. Vis.* **89**(2–3) (2010), 152–176.
- [10] K. Sfikas, I. Pratikakis, and T. Theoharis, Pose normalization of 3D models via reflective symmetry on panoramic views, *Vis. Comput.* (2014), 1261–1274.
- [11] J. Podolak, P. Shilane, A. Golovinskiy et al., A planar-reflective symmetry transform for 3D shapes, *ACM Transactions on Graphics*, **25** (2006), 549–559.