Rigid and Non-rigid Shape Matching for Mechanical Components Retrieval

Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello

Dip. di Scienze Ambientali, Informatica e Statistica, Universitá Ca' Foscari Venezia

Abstract. Reducing the setup time for a new production line is critical to the success of a manufacturer within the current competitive and cost-conscious market. To this end, being able to reuse already available machines, toolings and parts is paramount. However, matching a large warehouse of previously engineered parts to a new component to produce, is often more a matter of art and personal expertise rather than predictable science. In order to ease this process we developed a database retrieval approach for mechanical components that is able to deal with both rigid matching and deformable shapes. The intended use for the system is to match parts acquired with a 3D scanning system to a large database of components and to supply a list of results sorted according with a metric that expresses a structural distance.

1 Introduction

While the exact pipeline for bringing new goods into production varies significantly between markets, some common fundamental steps can be identified. Specifically, the most important milestones are the concept formulation, the design phase, the creation of a final prototype and the engineering of the production process [1]. Given the increasing competitiveness in the field of production, both in terms of time and cost, it is not surprising that a great effort is made to develop approaches that allow for a more efficient process for each of these four distinct phases.

The concept formulation alone could consume up to 50% of the time required for the whole cycle, thus a wide range of time saving approaches have been proposed in literature to reduce its impact [2, 3]. However, in this paper, we concentrate on the remaining three phases, since they are all dependant on technical judgements that can be eased by exploiting an automated tool.

The design phase is about the translation of the requirements emerging from the concept in a full product definition that can be implemented in a prototype. Depending on the market, the focus of the design step can be on the functionality of the product or on aspects related to fashion. Either way, the choices made at this stage can not be decoupled from the following phases. In Fact, the early verification of the relations between the design, the associated technical challenges and the actual capabilities of the available production system, could lead to significant reductions in the overall production cost and time to market [4, 5].

The prototyping is the bridge between the design and the assembly of the actual production process. At this stage, the requirements expressed during the design phase are casted into a physical object (usually handcrafted) and the limitation in the feasibility of some of the planned features may appear. At the same time, the requirements for the tooling (i.e. the set of machine configurations and custom tools needed for actual production) start to become apparent. Of course the quandary between changes in the design made with the goal of production semplification and implementation of new toolings must be managed. To this end, a partial overlapping and a regulated communication between phases has consistently been shown to be both useful and necessary [6–8].

The automatic tools available to help with the described process are many and diverse. In this paper we are focusing on a specific but very critical topic: the reuse of toolings.

Specifically, the most convenient way of reusing toolings is to adapt those made for the production of a component that is very similar (in shape and materials) to one already engineered in the past history of the factory. While this could seem a straightforward task, it must be taken in account that within, a medium to large factory, thousands of different new components can easily be introduced into the production each year. Currently, the most widely adopted method to solve this problem (at least in the factories we surveyed) is to resort to experts that have been working in the production department for a long time. Such experts are able to recall (by memory or by consulting a large archive of drawings) if a component similar to the one at issue have already been produced. Needless to say, this kind of approach can not be deemed as reliable or dependable for several reasons. To begin with, there is no guarantee that the experts are able to exhibit a good enough recall rate, moreover as the knowledge is not an asset of the company, but rather of individuals, the transfer of such assets as personnel turnover happens is difficult and very prone to errors.

The solution we are introducing automates this selection in a semi-supervised manner, making available to experts a structured tool to guide the crawling through a large product database. The proposed system (which will be described in depth in Sec. 2) is mainly based on shape-based recognition. Roughly speaking, shape matching is a technique widely developed in the field of Computer Vision and Pattern Recognition whose goal is to find the alignment or deformation transformation that relates a *model* object with one or more *data* scenes. There exist many different classifications of matching techniques, however, for our purposes, we break down them in two application domains and two transformation models. The application domains are respectively images (2D data) and surfaces (3D data). The transformation models which we are interested in are rigid, where model and data must align exactly and non-rigid, where a certain degree of elastic deformation is permitted.

Image-based shape matching is primarily performed by finding correspondences between point patterns which can be extracted from images using detectors [9–11] and descriptors [12, 13] that are locally invariant to illumination, scale and rotation. The matching itself happens through a number of different

techniques that ranges from Procrustes Alignment [14] to Graphical Models [15] for the rigid scenario, and from Relaxation Labeling [16] to Gaussian Mixture Models [17] for non-rigid matching.

In the first phase of our investigation we evaluated the adoption of image-based methods applied over shots of the components under a set of different angles. Unfortunetely this approach was not robust enough to grant a reasonable performance, additionally, the building of the database through image capturing was very time consuming and error prone. For this reason we resorted to the use of 3D matching techniques. Surface matching, albeit being addressed by literature for a long time, is recently boldly emerging due to the availability of cheaper and more accurate 3D digitzing hardware and to the increasing processing power of computer systems that allows for a feasible handling of the more complex volumetric information. In the last decade, more and more problems traditionally tackled with different techniques have been proven to be addressable by exploiting surface matching. For instance, in the field of biometric, the use of 3D surfaces as a substitute for images has shown to attain a far superior performance [18]. Exceptionally good results have been obtained in particular in the recognition of strongly characterizing traits such as ears [19] or fingers [20]. In classical biometric challenges, such as expression-independent face recognition, methods that are able to tell the difference even between the faces of two twins have been demonstrated [21]. In the industry, surface matching has been used extensively for defect analysis [22], reverse engineering [23] and motion capture [24].

Most 3D matching approaches work best when the transformation between model and data is rigid, as when a deformation is applied most Euclidean metrics can not be exploited. Recently, some effective non-rigid registration techniques that can be applied to 3D surfaces have begun to appear. Some of them are based on Graph Matching [25] or Game Theory [26]. Others perform the needed deformation by optimizing the parameters of Thin Plate Splines [27] or of a flexible surface regulated by a set of quadratic equations [28].

With respect to the pipeline described in the following sections, we resorted to the use of the very standard ICP rigid registration technique [29] since it is a fast algorithm that works well over the almost noiseless data that we are acquiring with a structured light scanner. By contrast, we designed a specially crafted method to tacke the non-rigid search. In fact, the knowledge of the restricted problem dominion, allows for an ad-hoc solution that specifically addresses the constraints we are dealing with.

2   A Shape-Based Pipeline to Maximize Tooling Reuse

The approach presented in this paper is the result of a study commissioned by Luxottica, a world leading designer and producer of eyewear. The goal of the study was the design and the implementation of a full system able to search for components similar to a given part into a database of nearly 150.000 components extracted from the about 30.000 eyeglasses model produced during the half a

century of company history. The main chelleges were the sheer number of objects to scan, which would imply several years of work with manual 3D scanners, and the need for a fast and accurate search system able to deal with rigid and non-rigid matching. In fact, eyeglasses frames are made up of both unarticulated parts, such as the front bar or the rims of the lenses, and of bending components such as the earpiece or the nose pads. In Fig. 1 an overall view of the proposed system is shown. As a preliminary step, the whole warehouse of available previously produced models is scanned using a structured light scanner customized to perform a fast and unattended digitzing of an eyeglass frame. The acquired models must be splitted into their basic components to be useful for the part search engine. Unfortunately, this kind of segmentation can not be done automatically, since the heterogeneity in component shapes and positions over the model is too high to grant an algorithm the ability to tell where actually a rim ends or a nosepad starts. To solve this problem in the most efficient manner we designed a semi-supervised segmentation tool that works by operating a greedy region growing regulated by the first derivative of the surface curvature. Using this tool a human operator is able to perform a complete model segmentation in a couple of minutes. Each part is subsequently labelled with its type. The component to be used as a query object is created in a similar way, by scanning the prototype frame instead of an archived eyepiece and by segmenting the part of choice. The matching is performed by first choosing the type of search (rigid or non-rigid) and by specifying the metric to be used (in the case of non-rigid matching). The query component is then compared with a suitable algorithm against all the item in the database. While this could seem to be a gargantuan task, both in terms of computing time and memory usage, it is indeed feasible without the need of a multi-level index structure. As a matter of fact, the size of a model component is within a few kilobytes and the typical amount of memory available on a modern server can easily handle hundred of thousands of components. Moreover, all the matching algorithms used can be executed in a few milliseconds, allowing for a full database scan less than a minute (as the
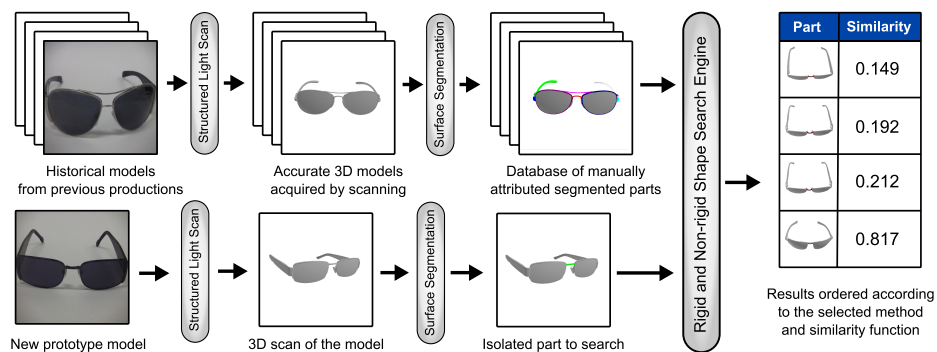

Fig. 1. Overview of the pipeline described in this paper for shape-based rigid and non-rigid component retrieval.

search is narrowed for type of component). In practice, however, we narrowed even further the search by filtering out components those total surface area deviates more than 30% with respect to the surface of the pursued part. This latter optimization provided, on a standard Intel based server, an average query time of about 10 seconds.

2.1 Capturing the 3D Shape of the Components

The building of the components database has been performed by means of a dedicated 3D scanner made up of a specialized support, an automated turntable, and a structured light scanning head (see Fig. 2). The support has been designed to be able to hold an eyewear frame with minimal occlusion and to present it to the scanning head with an angle that allows for a complete and watertight acquisition while turning on a single axis. The scanning head comprises two cameras and a DLP projector used to create light fringes according to the Phase Shift coding variant presented in [30]. A single range image requires about 3 seconds to be captured, and a total of 24 ranges are needed for the complete surface reconstruction. If the time required for the turntable rotation is also accounted, the whole process can be carried on in about two minutes.
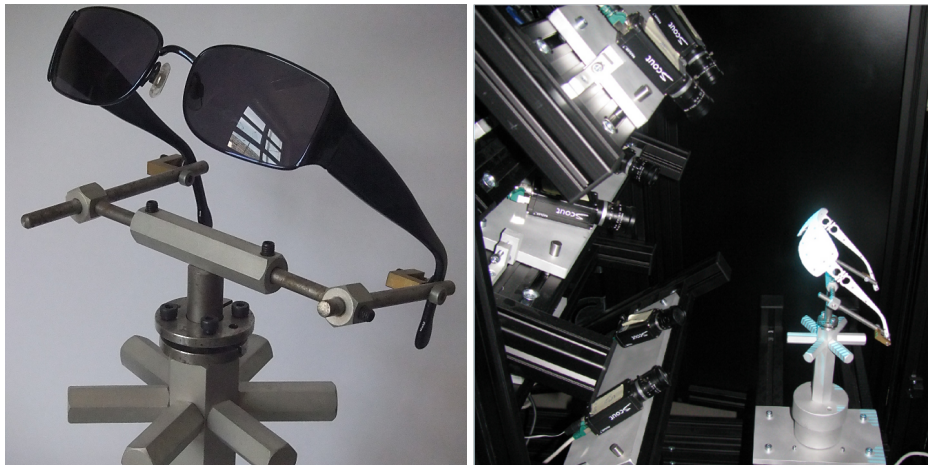

Fig. 2. The custom eyeglasses frame holder and the structured light scanner.

Once the scans are completed, the resulting range images are coarsely placed in a common frame by exploiting the knowledge of the step angle of the electric motor that drives the turntable. Subsequently they are pairwise aligned with IPC [29] and globally aligned using Dual Quaternion Diffusion [31]. Finally, a standard reference frame must be imposed to the newly acquired object. This is necessary for obtaining a common orientation of each of the components that will be produced by the following segmentation step, which in turn is needed as an initialization for both rigid and non-rigid search. To this end, a simple Principal Components Analysis (PCA) is performed on the 3D data.

2.2 Semi-supervised Object Segmentation

To help the user in the component segmentation, we designed a semi-supervised tool that is able to separate the single components from a scanned model starting from a limited number of initial seeds supplied by a human operator. Those seeds are used as a hint that indicates areas that belong for certain to different parts. Each area is then grown in a greedy manner until it hits another area. The main idea is that the growing becomes slower when abrupt changes in surface normals are encountered and thus notches on the surface (that are typically associated to small gaps between parts) act as a containment border.

This tool does not work directly on the surface of the object, but rather on an apt dual graph representation [32]. As shown in Fig. 3 each node of this graph corresponds to the a triangle of the mesh. There are no geometrical relations between these nodes and the absolute position of the triangles in space. For this reason we do not need any attribute on the graph nodes. By converse, we are interested in the relations between adjacent faces, thus we are going to define a scalar attribute for the graph edges. Specifically, we want to assign to each edge a weight that is monotonical with the "effort" required to move between the two barycenters of the faces. This effort should be higher if the triangles exhibit a strong curvature with a short distance between their centers and it should be low if the opposite happens. To this extent, given two nodes of the graph associated to faces i and j, we define the weight between them as:

$$\omega(i,j) = \frac{1 - <n_i, n_j>}{|p_i - p_j|} \tag{1}$$

where $\bar{p} = (p_1, p_2...p_k)$ is the vector of the barycenters of the faces and $\bar{n} = (n_1, n_2...n_k)$ are the respective normals. $< \cdot, \cdot >$ denotes the scalar product and $|\cdot|$ the Euclidean norm.

In Fig. 3 (c) edge weight is represented by using a proportional width in the drawing of the line between two nodes. It can be seen how edges that connect faces with stronger curvatures exhibit larger weight.

Once the weighted graph has been created, the segmentation can happen. In our framework the surface is segmented starting from one or more hints
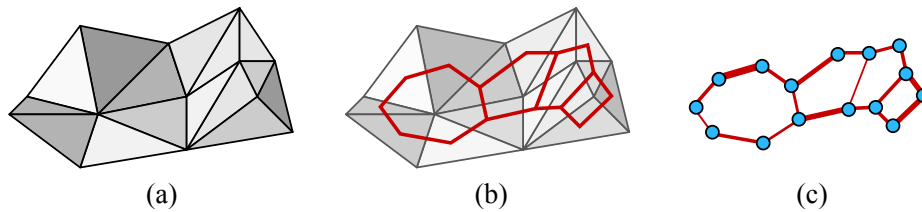

(a)   (b)   (c)

Fig. 3. Steps of the graph creation process. From the initial mesh (a) the dual graph is built creating a vertex for each face and connecting each pair of adjacent faces (b). Finally, each edge of this graph is then weighted according to the dot product between the normals of the connected faces.

provided by the user. This human hint expresses a binary condition on the mesh by assigning a small fraction of all the nodes to a set called *user selected green nodes* and another small portion to a set called *user selected red nodes*. We call *green nodes* the faces (nodes) belonging to the segment of interest and *red nodes* the ones that are not belonging to it, regardless of the fact that those nodes have been manually or automatically labeled. The proposed algorithm distributes all graph nodes in the *green nodes* and *red nodes* sets in a greedy way.

We define a seed as triple $< n, t, w >$ where n is the graph node referred by this seed, t is a boolean flag that indicates if n has to be added to green or red nodes, w is a positive value in $\mathbb{R}^+$. At the initialization step, for each initial green and red node selected by the user, a seed is created and inserted into a priority queue with an initial weight value $w = 0$. All nodes are also added to the *unassigned nodes* set. At each step, the seed $< n, t, w >$ with lowest value of w is extracted from the priority queue and its referred node n is added to *green nodes* or *red nodes* according to the seed's t flag. The node is also removed from *unassigned nodes* to ensure that each node is evaluated exactly once during the execution of the algorithm. For each node $n' \in$ *unassigned nodes* connected to n in the graph, a new seed $< n', t' = t, w' = \omega(n, n') >$ is created and added into the queue. It has to be noted that it is not a direct consequence of such insertion that the final type of n' (either green or red) is determined by the type t' of this seed. At any time multiple seeds referring the same node can exist in the queue, with the only condition that a node type can be set only once. During the execution of algorithm either the region of green nodes and the region of red ones expands towards the nodes that would require less weight to be reached. Once all nodes in the same connected component are visited, the result of this assignment is shown to the user who can either refine his initial hint or accept the proposed segmentation. Of course the procedure can be iterated to obtain a hierarchical segmentation. In any condition, the algorithm will run in $O(N)$ time since, with the described greedy approach, each node is visited once.

In Fig. 4 and example of the segmentation produced by the tool is shown.
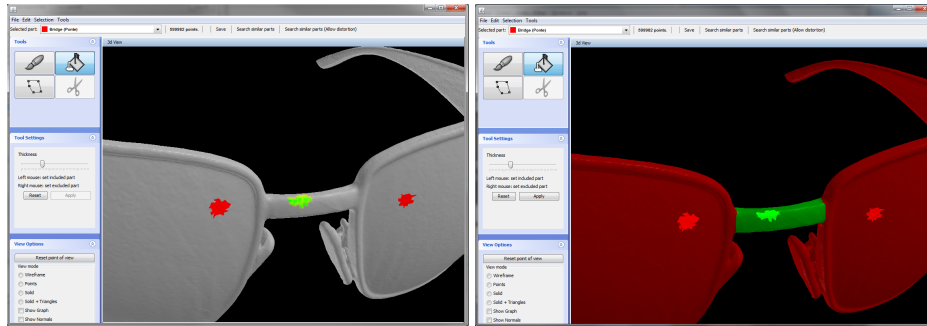

Fig. 4. An example showing an initial manual seeding and the associated semi-supervised segmentation.

2.3   Rigid Components Matching

The rigid matching consist in an alignment of each part in the database with the query part using an efficient variant [33] of the ICP (Iterative Closest Point) algorithm [29]. In detail, each component in the database is sampled and exactly 1000 3D points are extracted. The surface of the query object is coarsely aligned with those samples through the PCA previously computed over the whole model. Thus, for each sample point, the intersection between the vertex normal and the surface of the query object is found (if it exists). This intersection generates a new point, that is associated to the original vertex as the ideal mate on the query surface. After all the intersections have been computed a closed-form optimization is used to align mating points to model vertices [34]. The process is iterated many times, until the relative motion of the query with respect to the model becomes negligible. The idea of this algorithm is that the positional error committed when adopting the points generated by normal shooting as true correspondences becomes smaller at each iteration. In practice, the ability of ICP to converge to a correct global minimum strongly depends on the initial coarse alignment. In fact, a less than good initial estimation can easily lead to completely wrong final alignment, even with perfectly correspondent 3D objects. In our validation, however, the coarse registration supplied by the global PCA consistently allowed to obtain a local minimum when the object to be compared was the same or a component similar enough to be effectively used as a tooling source. After an optimal aligned is obtained, a last normal shooting is performed and the RMS of the distance between each vertex and its virtual mate is computed. This RMS is used as the metric distance for the ordering of the results. In Fig. 5 we show some results obtained by searching respectively for a bridge and a front hinge (highlighted in red in the figure). On the side of each result the corresponding RMS associated to the best alignment is reported.

Query	Result 1	Score	Result 5	Score	Result 10	Score
		0.155		0.209		0.266
		0.508		0.715		1.0

Fig. 5. Results obtained by the rigid search engine for some query objects (best viewed in colors).

2.4 Non-Rigid Components Matching

We decided to address the problem of non-rigid components matching as the process of finding the best global or local alignment between two linear sequences. Every component is first sliced into a vector of equally-spaced slices along its median axis.

The slicing procedure starts by roughly aligning the first two principal vectors of the component along x and z axis using the PCA. After that, starting from the farthest vertex along the negative side of x axis, a set of n equally-spaced planes parallel to yz plane are used to intersect the component defining n different closed planar contours. For each of those contours, the size w_i and h_i respectively along z and y axes are used to characterize each slice with the quadruple $s_i = \langle w_{i-1}^s, h_{i-1}^s, w_i^s, h_i^s \rangle$. For each component, the vector containing all slices defined above is stored into the database, together with the component id and its size. After that, we defined the similarity between two slices s_i and t_i as

$$s(s_i, t_i) = \begin{cases} m(s_i) - m(t_i) \Leftrightarrow C_i^{s,t} \\ 0 \qquad otherwise \end{cases} \tag{2}$$

where $C_i^{s,t} = (w_{i-1}^s \leqslant w_{i-1}^t \leqslant w_i^s) \wedge (h_{i-1}^s \leqslant h_{i-1}^t \leqslant h_i^s) \bigvee (w_{i-1}^t \leqslant w_{i-1}^s \leqslant w_i^t) \wedge (h_{i-1}^t \leqslant h_{i-1}^s \leqslant h_i^t)$ and $m(s_i) = \frac{w_{i-1}^s + w_i^s}{2} + \frac{h_{i-1}^s + h_i^s}{2}$

Once the similarity measure is defined, we search for a best global alignment by exploiting the well known Needleman-Wunsch algorithm first presented in [35] and the local alignment using Smith-Waterman [36]. Each alignment is performed iteratively between all stored components and the results are sorted by similarity. Depending of specific application requirements, local alignment may offer better results than the global approach, vice-versa. In this extent, we decided to show both ordering to the user, along with the similarity measure. In Fig. 6 an example of the ordering obtained respectively with Needleman-Wunsch and Smith-Waterman algorithm is shown.

Query	Result 1	Score	Result 5	Score	Result 10	Score	Result 20	Score
		6.781		7.499		8.380		8.540
		0.182		0.254		2.737		5.799

Fig. 6. Results obtained by the non-rigid search engine using Needleman-Wunsch (top) and Smith-Waterman (bottom) algorithms (best viewed in colors).

3   Evaluation of the Implemented System

An initial evaluation of the system was performed by acquiring different instances of the same objects and, after manual segmentation, by automatically searching for each single part, which was replicated several times in the test database. To test the robustness of the matching process, the components were modified by adding random Gaussian noise to their vertices. The goal of this validation step was simply to assess the ability of the system to produce an ordering were consistent clusters of components appear before the remaining results. While this test was successful, obtaining a 100% recognition rate, it is only meaningful to check that the search algorithm is correctly working as a pure object retrieval tool, but it does not really indicates if similar, but not identical, parts can be retrieved with a reasonable ordering. In order to obtain a measure of how well the system works for the intended purpouse, we let the experts use it for some months. During this trial, the experts have been asked to annotate, after each component search, the position in the result list of the first occurrence that they could consider to be similar enough for tooling reuse purposes. If a similar enough component is not found at all (or the expert knows that it does not exists in the database) the annotation does not happen and the event is deemed to be a true negative. The results of this test are given in Fig. 7 as the percentage of cases where the sought component is found among the first N answers supplied by the engine. The first graph shows the performance of the rigid matcher. As expected, in most cases the correct component can be found as the first match and in no cases more than 50 results must be scanned by the user. The capability of the non-rigid matcher, shown in the second graph of Fig. 7, is a bit lower. In detail, the local sequence alignment performed by Smith-Waterman algorithm seems to behave a little better than the global alignment obtained by Needleman-Wunsch. However, for both algorithms, the correct result can be found most of the times immediately and within the first 50 extracted components in more than 90% of the cases. This level of performance is clearly good enough for an effective use in production environment and arguably better than what could be attained without the assistance of the system.
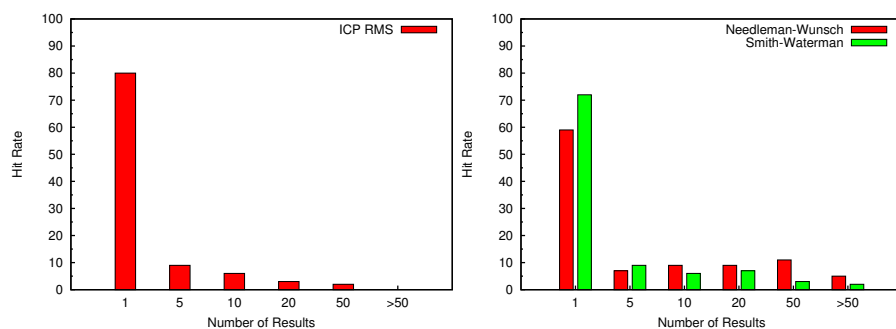

Fig. 7. Evaluation of the hit rate of the search system with respect to the number of results inspected by a human expert for validating purposes.

4 Conclusions

In this paper we described a complete pipeline to help field experts during the design process for new production lines. The presented approach is based on both well-known methods and novel domain-specific techniques. In details, the approach allows to easily find toolings and process definition to be reused in new productions by comparing the shape of the newly designed component with a large database of previously engineered parts. This comparison needs to be performed both in a rigid fashion and allowing for some deformation, depending on the type of part to be searched and on the good sense of the expert that is using the tool. Each search produces an ordering of the parts in the database, which can be computed with respect to a set of different metrics. Overall, a sizeable set of trials performed with a database of sunglasses components have been deemed successful by the users, as the most relevant parts were consistently found in the first few results produced by the search engine.

References

1. Ulrich, K.T., Eppinger, S.D.: Product Design and Development. McGraw-Hill (1995)
2. Kim, J., Wilemon, D.: Focusing the fuzzy front–end in new product development. R&D Management **32** (2002) 269–279
3. Millson, M.R., Raj, S., Wilemon, D.: A survey of major approaches for accelerating new product development. Journal of Prod. Innov. Manag. **9** (1992) 53 – 69
4. Dieter, G.E., Schmidt, L.C., AZARM, S.: Engineering design, 4[sup th] edition,. Journal of Mechanical Design **131** (2009) 056501
5. Nihtila, J.: R and d production integration in the early phases of new product development projects. Journal of Eng. and Tech. Manag. **16** (1999) 55 – 81
6. Takeuchi, H., Nonaka, I.: The new new product development game. Harvard Business Review **64** (1986) 137–146
7. Mabert, V.A., Muth, J.F., Schmenner, R.W.: Collapsing new product development times: Six case studies. Journal of Prod. Innov. Manag. **9** (1992) 200 – 212
8. Zha, X.F., Sriram, R.D.: Platform-based product design and development: A knowledge-intensive support approach. Know-Based Syst. **19** (2006) 524–543
9. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94). (1994) 593–600
10. Smith, S.M., Brady, J.M.: Susan—a new approach to low level image processing. Int. J. Comput. Vision **23** (1997) 45–78
11. Rosten, E., Porter, R., Drummond, T.: Faster and better: a machine learning approach to corner detection. CoRR **abs/0810.2434** (2008)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision. Volume 20. (2003) 91–110
13. Bay, H., Ess, A., Tuytelaars, T., Gool, L.J.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110** (2008) 346–359
14. Luo, B., Hancock, E.R.: Matching point-sets using procrustes alignment and the em algorithm. Computer (1999) 43–52
15. Caetano, T.S., Caelli, T., Schuurmans, D., Barone, D.A.C.: Graphical models and point pattern matching. IEEE Trans. PAMI **28** (2006) 2006

16. Lee, J.H., Won, C.H.: Topology preserving relaxation labeling for nonrigid point matching. IEEE Trans. Pattern Anal. Mach. Intell. (2011) 427–432
17. Jian, B., Vemuri, B.C.: Robust point set registration using gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 1633–1645
18. Woodard, D.L., Faltemier, T.C., Yan, P., Flynn, P.J., Bowyer, K.W.: A comparison of 3d biometric modalities. In: Proceedings of CVPR 2006. CVPRW '06, Washington, DC, USA, IEEE Computer Society (2006) 57–
19. Chen, H., Bhanu, B.: Contour matching for 3d ear recognition. In: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1 - Volume 01. WACV-MOTION '05, Washington, DC, USA, IEEE Computer Society (2005) 123–128
20. Woodard, D.L., Flynn, P.J.: Finger surface as a biometric identifier. Comput. Vis. Image Underst. **100** (2005) 357–384
21. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Expression-invariant representations of faces. IEEE TRANS. PAMI (2007) 1042–1053
22. Li, Q., Wang, M., Gu, W.: Computer vision based system for apple surface defect detection. Computers and Electronics in Agriculture **36** (2002) 215 – 223
23. Park, S.C., Chang, M.: Reverse engineering with a structured light system. Computers and Industrial Engineering **57** (2009) 1377 – 1384
24. Plänkers, R., Fua, P.: Articulated soft objects for multiview shape and motion capture. IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003) 1182–1187
25. Zeng, Y., Wang, C., 0001, Y.W., Gu, X., Samaras, D., Paragios, N.: Dense nonrigid surface registration using high-order graph matching. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2010, IEEE (2010) 382–389
26. Emanuele, R., Alex, B., Andrea, A., Filippo, B., Andrea, T.: A game-theoretic approach to deformable shape matching. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, IEEE (2012)
27. Zou, G., Hua, J., Muzik, O.: Non-rigid surface registration using spherical thin-plate splines. In: Proceedings of the 10th international conference on Medical image computing and computer-assisted intervention - Volume Part I. MICCAI'07, Berlin, Heidelberg, Springer-Verlag (2007) 367–374
28. Salzmann, M., Moreno-Noguer, F., Lepetit, V., Fua, P.: Closed-form solution to non-rigid 3d surface registration. In: ECCV. (2008) 581–594
29. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14** (1992) 239–256
30. Lilienblum, E., Michaelis, B.: Optical 3d surface reconstruction by a multi-period phase shift method. JCP **2** (2007) 73–83
31. Torsello, A., Rodolá, E., Albarelli, A.: Multiview registration via graph diffusion of dual quaternions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, IEEE (2011) 2441–2448
32. Bergamasco Filippo, Albarelli Andrea, T.A.: A graph-based technique for semi-supervised segmentation of 3d surfaces. Pattern Recognition Letters (2012)
33. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling. (2001) 145–152
34. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. J. of the Optical Society of America. A **4** (1987) 629–642
35. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48** (1970) 443 – 453
36. Smith, T., Waterman, M.: Identification of common molecular subsequences. Journal of Molecular Biology **147** (1981) 195 – 197