Part-based Construction of digitized 3D objects

Daniela Borges
INESC-ID/IST/Technical
University of Lisbon
R. Alves Redol, 9, 1000-029
Lisboa, Portugal
daniela.borges@ist.utl.pt

Alfredo Ferreira
INESC-ID/IST/Technical
University of Lisbon
R. Alves Redol, 9, 1000-029
Lisboa, Portugal
alfredo.ferreira@inesc-id.pt

ABSTRACT

Nowadays, a few 3D acquisition devices are available at low-cost. While 3D capture is a commonplace, decompose the object into its components is not an easy task. Segmentation can help address this problem by suppling data which may be used to identify object components. However, it might not give complete and accurate information about components. In a context where a digital repository with every component that can belong to physical objects is available, retrieval algorithms can be used to construct a composed 3D model.

We propose a four phase solution to construct 3D digitized objects. We use Microsoft Kinect® to acquire 3D physical objects. A segmentation algorithm based on color information decomposes the object into a set of sub-parts. The component repository is queried using a shape-based retrieval algorithm, in order to identify which sub-part corresponds to each virtual component. Then, a 3D model of the physical object is constructed by assembling the retrieved components.

The work presented in this paper has a wide application domain, ranging from entertainment to health or mechanical industry. To validate our proposal, we implemented a toy-problem and evaluated its precision and efficiency. We used LEGO® blocks, which can provide challenges similar to real-world applications. The results were encouraging and we believe that our approach may even work better with greater object components, geometrically less similar to each other.

Keywords

Reconstruction, Acquisition, Segmentation, Retrieval.

1 INTRODUCTION

Technological advancements allowed the storage of objects such as audio, image and video through personal devices. These assets, previously considered tangible, can now be transported everywhere and are named digital media. Nowadays, 3D scanners are more accessible for anyone and several approaches have emerged to solve acquisition, analysis, classification, index and retrieval problems [VMC96]. Naturally, this also led to 3D model construction and new challenges have appeared.

Reconstruction techniques allow the acquisition of physical objects, in order to reach a digital model. However, if the object is composed by several components, it is impossible to identify every object component with a simple reconstruction.

Object construction is not a trivial challenge, because 3D object acquisition does not performs matching between known models and acquired objects. Moreover, the acquired objects could have several physical components which are also not identifiable. These considerations lead to an increase in the number of enthusiasts both from research and industry. Life of George[1] and Autodesk 123D[2] are examples of applications that had shown, respectively in 2D and 3D, the construction potential and why this is an interesting area. The establishment of new low-cost scanner devices increasingly accessible to anyone is also a motivation for our work.

Our research tries to ascertain if is possible to use a low-cost device to acquire a 3D object and perform a segmented construction that uses color and shape information. In other hand, we acquire, segment and retrieve a 3D physical object an all its components, in order to create a reliable 3D digital model.

To support our vision, we address two major challenges: (1) the segmentation of the acquired polygonal mesh, using color information; and (2) the identifica-

[1] http://george.lego.com/
[2] http://www.123dapp.com/

 ISBN 978-80-86943-75-6

tion of segmented physical components (sub-parts), that allows the construction. To be able to perform the construction, our work will assume two restrictions. The former is that adjacent components must have different colors and the latter is that every physical component allowed to be acquired must be known, i.e., must have a model (virtual component) in a repository. Therefore, our proposal aims to reconstruct 3D physical objects in real-time with an acceptable success rate. To accomplish these requirements, we selected algorithms that meet a tolerable time-quality relationship.

Next section discusses related work, presenting several approaches to solve construction problem through acquisition, segmentation and retrieval phases. Section 3 describes our solution and compares it with other approaches referred in the related work. Section 4 mentions how to evaluate our algorithm performance, shortcomings analysis and present relevant results. Finally, we present our conclusions and point out some future work.

2 RELATED WORK

3D object acquisition is allowing the scanning of a huge amount of objects mainly in the fields of computer-aided design (CAD), computer-aided manufacturing (CAM), cultural heritage, reverse engineering, among others. Range scanners, presented in [BR02], grant object acquisition and are divided in several systems such as triangulation systems, time-of-flight systems, among others. Although high range scanners (such as Comet L3d[3] or EXAscan[4]) have huge accuracy and resolution, some low-cost scanners that appeared recently are increasingly widespreading. As a result, some applications have emerged primarily in the area of video games.

Low-cost devices, such as Microsoft Kinect®[5] or Primesense sensor®[6], are becoming increasingly available to anyone, regarding the low price in comparison with other scanners. However, although these scanners acquire real-time RGBD data and produce relevant results through controlled scenarios, they do not have high accuracy. That is a high disparity between scanner prices is a reasonable reason for us to choose Microsoft Kinect® to acquire our 3D models. Moreover, we also consider the popularization around this scanner due to its low price. Because of this, lots of users are now using Kinect® for a wide variety of applications.

[3] http://www.steinbichler.com/products/surface-scanning/3d-digitizing/comet-l3d.html

[4] http://www.creaform3d.com/en/metrology-solutions/portable-3d-scanner-handyscan-3d

[5] http://www.microsoft.com/en-us/kinectforwindows/

[6] http://www.primesense.com/solutions/sensor/

Segmentation is useful for location, classification and feature extraction of 3D shapes. Although segmentation is easily performed by humans, computers need complex algorithms to achieve the same work. There are several segmentation algorithms that receive a point cloud, an object or an image as input. The goal is to decompose the object into patches or regions whatever is the input received. Some clues may help to reach segmentation such as normal calculation, curvatures or concavity around the boundaries. Up to now, in the field of segmentation a large number of algorithms have been proposed such as Region Growing [AB94], K-means [STK02], Fitting Primitives [AFS06], among others. We need to take into account that some algorithms must be used offline whereas others are faster and consequently better to interactive applications.

Although the referred algorithms are traditional, new technologies bring depth and color information. Moreover, the combination of color and depth information to segment shapes is becoming common use.

One approach to construct a 3D model is to retrieve every component that belong to an object, identifying them. First of all, we need to index every possible component through the descriptor computation for every model. Descriptors define models through a signature and provide a way to retrieve those models efficiently. Therefore, have appeared several descriptors, namely Spherical Harmonics (SHA) [FMK+03], Light-Field Descriptor (LFD) [CTSO03], and more recently BOW-LSD [LSFG11] or PatchBOF [TDVC11]. Our work requires almost real-time and, consequently, we decided to use the most efficient descriptor. When all models are indexed, preferably using an indexing mechanism, we are allowed to retrieve them. Models are retrieved using queries that represent those models. This query is also the result of a descriptor computation. Afterwards, when the query is performed the most similar results may be retrieved.

A pioneer work in construction area is The Digital Michelangelo Project [LRG+00]. This project, which major requirement is the construction of high resolution digital models, uses specific software and hardware in order to scan cultural heritage. A triangulation system was used to acquire depth information, with the help of one motorized gantry. Due to the hardware used the final output contains billions of polygons (for instance, in David statue) which is not possible to handle with commercial applications. Our approach is different to this work because we use low resolution models, that have different requirements.

Recently, there are other projects [SW11, MMWG11] in the field of cultural heritage. Schwartz et al. [SW11] presented a work where 3D geometry is extremely required and optical properties of object surface are also desired (such as reflexion). Using a Bidirectional Tex-

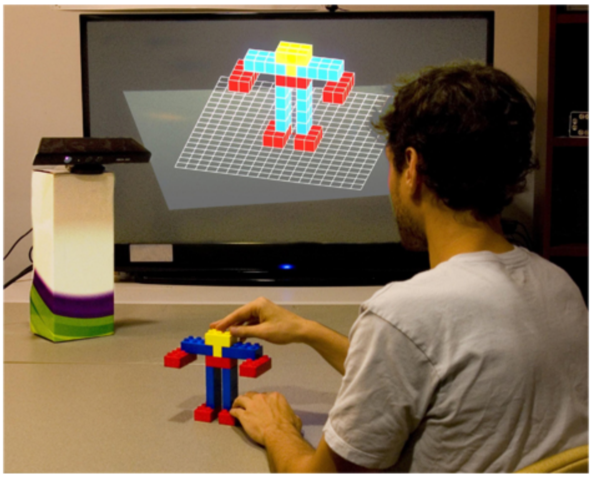

Figure 1: Setup of Lattice-First project.

ture Functions (BTF) they can achieve geometrical accuracy and provide a photo-realistic results. In order to obtain these results, they used 151 cameras with 12 megapixels that acquire High Dynamic Range (HDR) sequences. This configuration allows 151 simultaneous pictures with high geometrical accuracy. However, it is not possible to be used by everyone, hindering generalized use of the system.

The projects mentioned so far use acquisition hardware which reach high quality and accuracy, justified by dimension and quality required in cultural heritage projects. Other approaches, that require less accuracy in comparison with these projects, may take different acquisition hardware, such as Microsoft Kinect®.

Software such as KinectFusion[IKH+11, ND11], ReconstructMe[7] or Skanect[8] allow controlling of one Microsoft Kinect® (or other low-cost cameras) through a scenario and, consequently, the acquisition of this scenario through several viewpoints. Up to now, works such as 3D Puppetry [RTHA12] used the combination of software implementations and low-cost scanners to acquire objects. This is also an example of toy-problem project, which consider controlled scenarios and use toys to create an abstraction that can be generalized to other domains.

Nevertheless, other toy-problem works have appeared recently. Miller at al. [MWC+12] presented a solution (Lattice-First) that also uses Microsoft Kinect® in order to obtain depth information of LEGO® blocks (Figure 1). Although this approach just acquires 3 degrees-of-freedom (DOF) and assumes that the object could not be moved out of a certain area, real-time information is guaranteed (25 frames per second). As a result, user is able to manipulate and interact with the physical object though this area. In order to achieve these goals, pixels from user hands are segmented in acquisition phase. Color information is added afterwards, through rendering phase. This work has some shortcomings, namely the limitation of using orthogonal DUPLO® blocks due to Microsoft Kinect® low resolution and low dimensions of used blocks. Moreover, regarding the limitation of 3DOF, latitude movements are not allowed.

Other relevant work in this field is DuploTrack [GFCC12] which proposes an approach that uses instructions, similar to LEGO® Digital Designer [9] virtual tool. This system also presents information in real-time and is divided in two modes: Authoring and Guidance. Authoring mode allows user to construct a physical object which digital model is being constructed and presented on the screen. On the other hand, Guidance mode instruct user in order to construct an existing object, by giving him several instructions. The representation used is equal to Miller's work, a voxel grid, but this system is able to acquire 6-DOF. User hands, depth and color information are used to segment foreground from background and to apply color to digital model. Although DuploTrack solve problems that the previous system cannot deal with, it still requires orthogonal DUPLO® blocks. The main reason referred is also Microsoft Kinect® acquisition limitations, such as noise through point clouds and low accuracy. They explain that this system also requires minimum of five blocks in order not to lose track. Otherwise, blocks can be confounded as outliers.

Our proposal requires a controlled environment and aims to identify orthogonal and non-orthogonal 3D shapes, allowing the construction of simple and complex objects. We also use a low-cost camera to acquire physical objects, with the purpose of disseminate our solution. We choose DUPLO® blocks because they are easy to use, educational and accessible to everybody. Moreover, we can easily access to LEGO® database (LDraw[10]) in order to create a repository.

3 SYSTEM OVERVIEW

Our proposal aimed to construct 3D objects almost in real-time, with an acceptable success rate considering the number of components known. Our system is composed by four phases: (1) acquisition of a 3D polygonal mesh; (2) segmentation of acquired object; (3) retrieval of segmented object components (sub-parts); and (4) construction of a digital model, using retrieved LEGO® blocks. As a result, we considered both efficient segmentation and retrieval algorithms. In order to construct the model, we used descriptors to retrieve every physical component presented in the object.

[7] http://reconstructme.net/
[8] http://skanect.manctl.com/
[9] http://ldd.lego.com/
[10] http://www.ldraw.org/

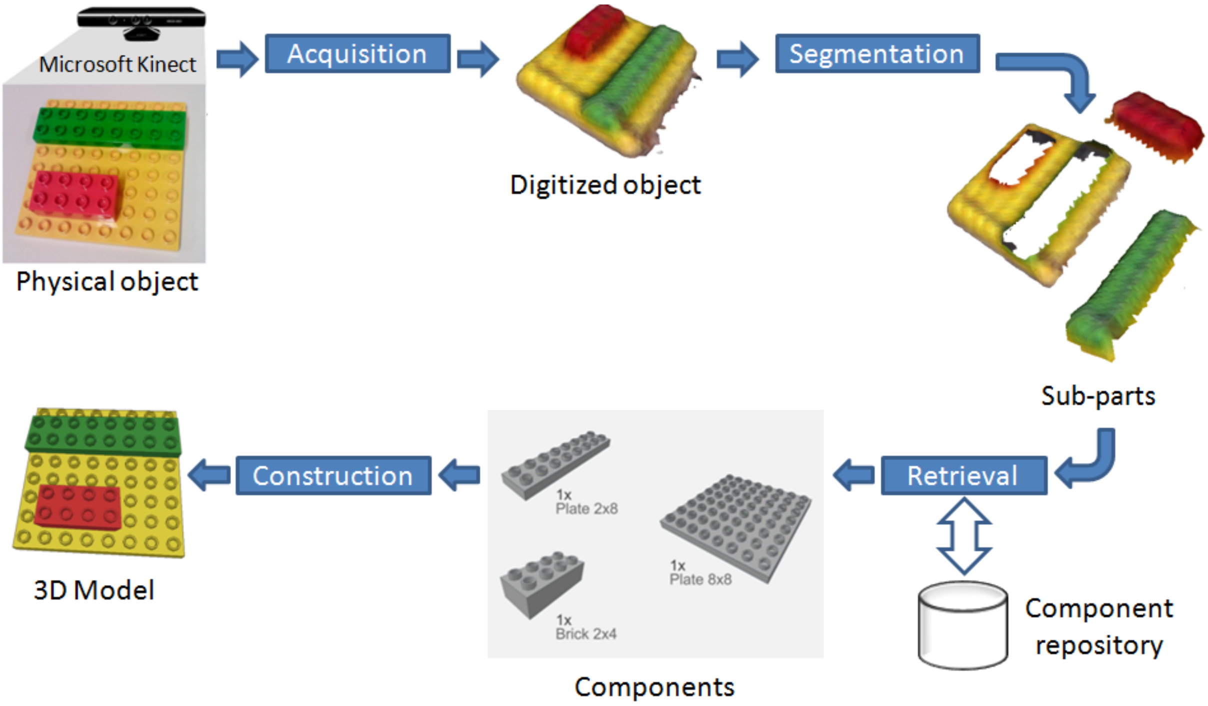

Figure 2: System overview, considering acquisition, segmentation, retrieval and construction phases.

Figure 2 describes an overview of our system, providing a construction algorithm capable to identify blocks that belong to an object. In particular, our approach aimed at going further than Lattice-First or DuploTrack, enabling the construction of blocks that are not considered in this projects (for example, curved blocks). Therefore, we pretended also a significant increase in the number of blocks that exist in the database (component repository).

Our system starts by acquiring a polygonal mesh, using Skanect. This is accomplished by moving one Microsoft Kinect® around the physical object. Using both depth and color information, we segment our object and get several sub-parts. As a result, each physical component is detached and we are ready to identify it. Subsequently, each sub-part is compared with components that exist in our repository and retrieval is performed. This process identifies physical components and makes the construction of LEGO® blocks possible.

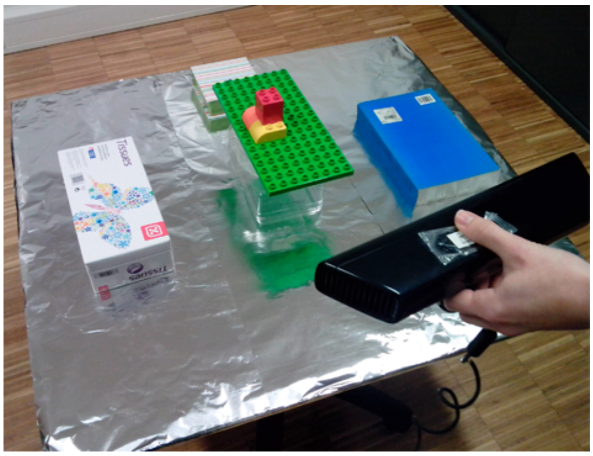

Figure 3: System setup. The object is created and subsequently positioned at the center of the table. Afterwards, it is acquired using one Microsoft Kinect® and able to be processed.

3.1 Acquisition

We used a low-cost scanner, Microsoft Kinect®, in order to acquire physical objects (composed by DUPLO® components). The justification behind our choice, is the fact that low-cost scanners are getting used by lots of enthusiasts, and it is a recent technology that is being more and more present at people homes.

Our application used Skanect software, that provides depth and color information. The main reason behind the use of this software is because it registers all views acquired through Microsoft Kinect®, allowing us to define a bounding box that excludes some outliers (such as walls or floor). Moreover, Skanect has the advantage of acquire color (and depth) information easily, in comparison with other approaches. However, it has some limitations: (1) it is only able to export a complete mesh (boundary representation); and (2) the resulting mesh has a huge amount of outliers, regarding the use of other objects which help Skanect not to lose track. To overcome these limitations, we used Point Cloud Library (PCL) [RC11] which is able to import the resulting mesh and convert it into a point cloud (it

 ISBN 978-80-86943-75-6

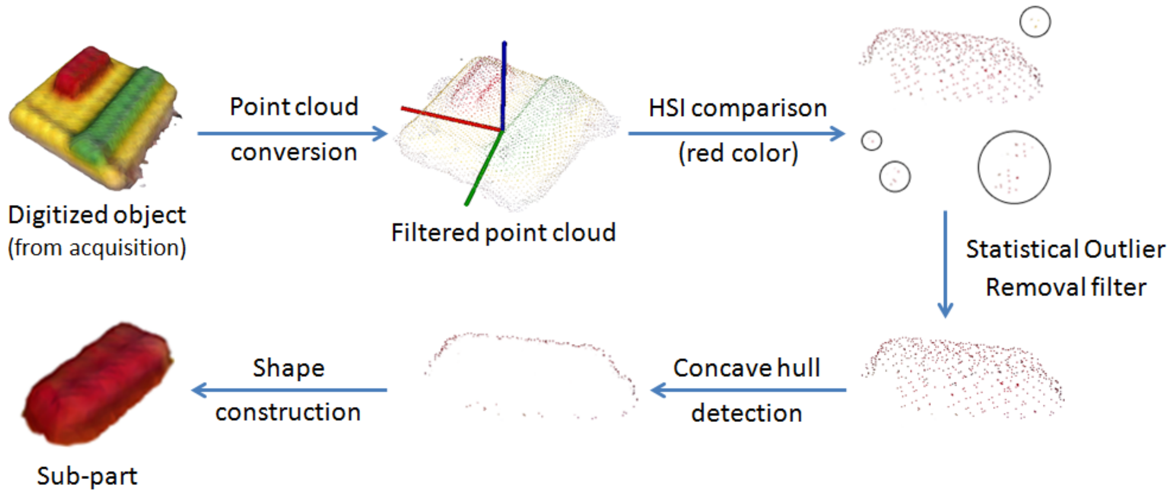

Figure 4: Example of segmentation for red color. The filtered point cloud from acquisition is used to identify the red color, through HSI comparison. We then filter outliers and detect concave hull. In the end, a shape construction is performed for every different color.

also speedup our filtering and segmentation processes). For this reason we can remove outliers and store depth and color information about our physical object.

Our setup is presented in Figure 3 where you can see a Microsoft Kinect®, several DUPLO® blocks and a turntable. Although a promising approach is to acquire objects though several table rotations, more valuable results were obtained by moving the scanner around a bounding box. Moreover, Skanect also helped us to configure our setup, providing real-time collaboration. We used an aluminium foil and transparent boxes in order to remove the maximum possible outliers and, consequently, accelerate the acquisition process.

The filtering is a process that is used by several algorithms to remove outliers from noise measures, lack of calibration, and imprecision problems due to registration. In our approach, we filter the acquired polygonal mesh in order to remove the objects added to help Skanect in registration. This process is performed by converting the acquired polygonal mesh into a point cloud. Using Principal Component analysis (PCA) [MN95], our point cloud is represented by a covariance matrix. As a result, we made a projection of the point cloud, ensuring that it is aligned with axes. The tabletop plane helped to perform this projection correctly. Therefore, we removed outliers that are behind length and width boundaries, an input of our algorithm which depend on the size of the acquired objects. Finally, we adjust the height boundary that depends on the height of the physical object, that is also an input of our algorithm. With this methodology, we reduce our point cloud from $\sim$12 Mb to $\sim$600 Kb, increasing the speed of our algorithm. Note that having the object aligned with the axes is determinant to provide a successful construction, because position and orientation of virtual components depend on this.

3.2 Segmentation

The acquired color and depth information from Skanect led us to create a boundary representation and, subsequently, a point cloud. However, none of this representations allow the identification of each component that belongs to the acquired object. In other words, it is not possible to recognize what components form the point cloud.

Therefore, our segmentation phase received the filtered point cloud and aimed to divide it in several sub-parts. We used color to segment each object component, assuming that adjacent components have different colors (Figure 4).

We used algorithms available on PCL that helped to filter each component by color. Hue, Saturation and Intensity (HSI) comparison [RC11] allows the creation of a set of filters that recognize each color. It also enables the creation of a filter that removes noise produced by registration failures, by removing points that have less than a certain number of neighbors.

The output of this phase is one independent shape for each sub-part, providing important information to retrieval phase. These shapes are constructed through three steps: (1) outlier removal, through a Statistical Outlier Removal filter [RMB^{+}08]; (2) concave hull creation [RC11]; and (3) shape construction [RC11]. The first step uses the segmented sub-part and removes outliers, i. e. points that have the approximately the same color, but are distant neighbors. We performed Statistical Outlier Removal filter that uses statistical analysis techniques to remove these noisy measurements. This

 ISBN 978-80-86943-75-6

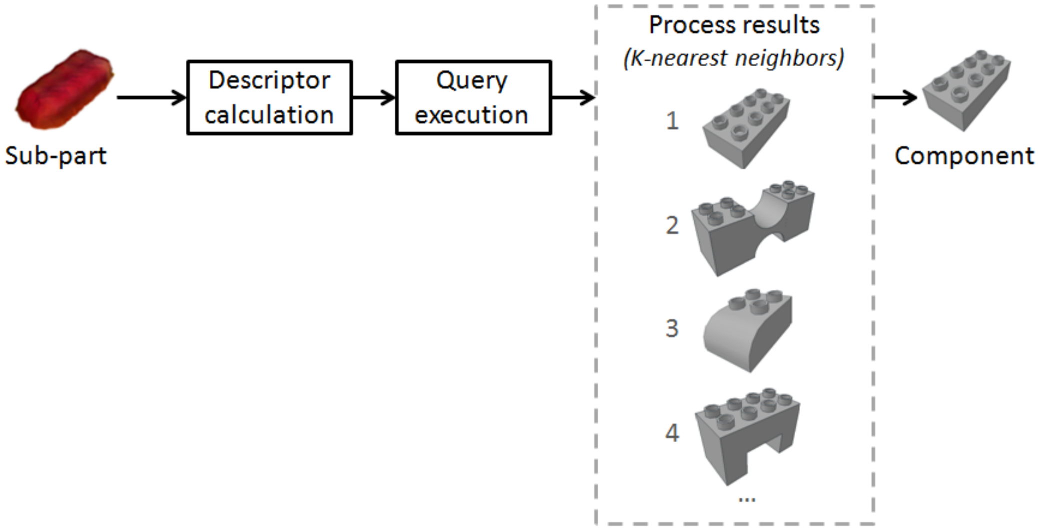

Figure 5: Retrieval phase. Our algorithm performs one query for each sub-part and returns the most similar parts.

filter is useful because acquisition outputs may have some points that are outliers due to tracking errors. We then create a concave hull in order to remove points that are not relevant for our solution. Take into account that concave hull creation tries to fill empty space created by non well acquired sub-parts, for example, due to occlusions. Finally, the shape is constructed, using the performed concave hull.

Adding to this output, there is also other information that needs to be stored for every sub-part: the color, the orientation and the centroid (one position in space). All this information is collected through filtered point cloud and required for construction phase.

3.3 Retrieval

When segmentation is concluded, we perform retrieval for each sub-part, with the purpose of constructing our model (Figure 5). Note that for every acquired physical component must exist a correspondent virtual component in our repository. First of all, one query is initiated for each sub-part. Afterwards, the descriptor is calculated for each sub-part, building one vector per part that represents that sub-part information in a more efficient way. Subsequently, the query is executed in our database, comparing the vector created with other descriptors. These descriptors were created through indexing phase and the process is similar: a descriptor was created for every virtual component that exists in our repository. The returned results of our algorithm are presented in two ways: either by best match selection or via k-nearest neighbors. The first best match selection for every sub-part is used for construction phase. On the other hand, k-nearest neighbors are used to disambiguate results, i.e, if the algorithm are not sure about one block, it can ask user using information on this list.

There are several algorithms which perform index and retrieval of 3D models. Our work required a time-efficient algorithm in order to fulfill our requirements. As a result, we chose Spherical Harmonics (SHA) [FMK+03] shape descriptor, that decomposes a 3D model in a collection of functions defined by concentric spheres. SHA is computed for every virtual component that exists in our repository, during offline indexing, and also for every sub-part that belongs to physical object, when retrieving.

We used a NB-Tree [FJ03], which is a powerful multi-dimensional structure, to index 70 models. This approach is efficient for high-dimensional data points, mapping those points to a 1D line through Euclidean norm. NB-Tree was relevant because it accelerates the indexing and retrieval times of our algorithm.

3.4 Construction

The sequential phases considered so far are used to construct our 3D model. In other words, we acquired a 3D physical object and segmented this components considering depth and color. Afterwards, we identified every sub-part through retrieval phase and construction is performed using this retrieved virtual components (Figure 6).

Construction is accomplished using best match selection for every block identified, regarding the information from retrieval. A 3D digital model is created taking into account centroid, orientation and color for every block (this information was given at segmentation phase). As a result, the relationships between physical components are kept. However, these relationships have some errors due to noise, taking into account that we do not deal with collisions.

3.5 Visualization and exploration

We aim to provide visual feedback of our constructed 3D model. Considering the domain of the toy-problem,

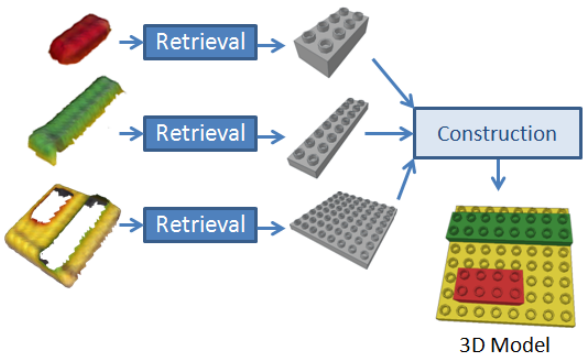

Figure 6: Construction phase.

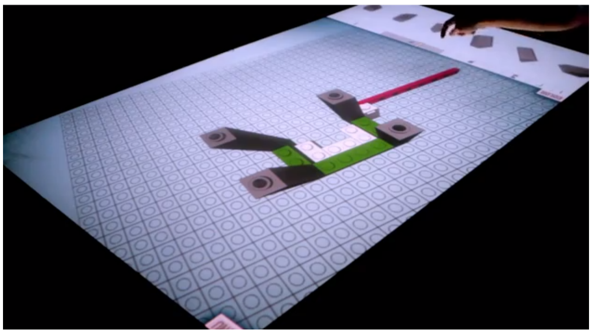

Figure 7: LTouchIT prototype. Through this application visualization and exploration of constructed models is possible.

LTouchIT [MF11] (Figure 7) application is going to be used. This application allows the construction of a 3D digital model using LEGO® blocks, through a multi-touch table. The user interacts with the table and is able to create several models, using blocks that exist in the database.

LTouchIT will represent the output of our construction algorithm, providing visual feedback and user input. It will be used mainly for user testing, allowing visualization and exploration of constructed models through several views. As a result, it helps to analyze the quality of the performed construction. However, LTouchIT application has some shortcomings regarding our approach. First, it uses a grid with lower dimensions due to the use of LEGO® blocks in spite of DUPLO® blocks. Second, the database used is incorrect, for the reasons mentioned above. Thus, some adaptation is needed in order to fulfill the requirements of our solution.

4 RESULTS

This section corresponds to the evaluation phase and obtained results of our algorithm. We divided this section in different subsections in order to explain what is evaluated and how this evaluation was performed.

4.1 Acquired models

As said before, we considered a toy-problem in order to validate our algorithm. As a result, for this particular problem we classified DUPLO® blocks through four different categories: standard blocks; additional blocks; curved blocks; and complex blocks. The standard blocks are orthogonal blocks that are considered in state-of-art projects ([MWC$^+$12, GFCC12]). The additional blocks are orthogonal blocks that are not considered in standard blocks category. The curved blocks are blocks that have at least one curve but are relatively simple. Finally, the complex blocks are blocks that are not considered in other categories.

The dimensions of the acquired blocks varied between about 1.5 to 14.5 centimeters with respect to height, and about 3 to 27 centimeters with respect to the length and width. We took into consideration that we mainly have small blocks to acquire. Moreover, Microsoft Kinect® accuracy should also be considered in order to produce relevant acquisition results.

In relation to our component repository, some of the indexed models came from LDraw library, which is actualized by LEGO® enthusiasts. Because of this, we needed to be careful regarding the use of those blocks. As a result, we performed a correction of errors (such as normals) in the used models. Nevertheless, in order to have the 70 models, we also created some virtual components according to LDraw format standards.

4.2 Evaluation methodology

Our approach was evaluated considering objective appreciations. Objective measures aim to evaluate the algorithm through precision and time. Precision is the percentage of retrieved components that are relevant, i.e, number of correct components regarding the total number of components. Time is the sum of the time of all phases and intend to evaluate algorithm efficiency. Objective metrics gave us percentages and efficiency measures to analyze our algorithm.

Using the categories mentioned in last subsection, ten physical constructions were performed (Figure 8), where DUPLO® blocks vary between two to five blocks. There are two aspects that we took into account: (1) the color of each block, ensuring that two adjacent blocks could not have the same color; (2) regarding Kinect® shortcomings, we excluded transparent and bright blocks.

4.3 Discussion

For a faithful construction, it is necessary to have retrieved virtual components that match the acquired components of the physical object. However, the results achieved do not precisely construct the physical object.

The objective evaluation was performed through our ten physical constructions, and the results are summarized in Figure 9. We conclude that 22% of the time our algorithm gave the correct answer at the first time (best

 ISBN 978-80-86943-75-6

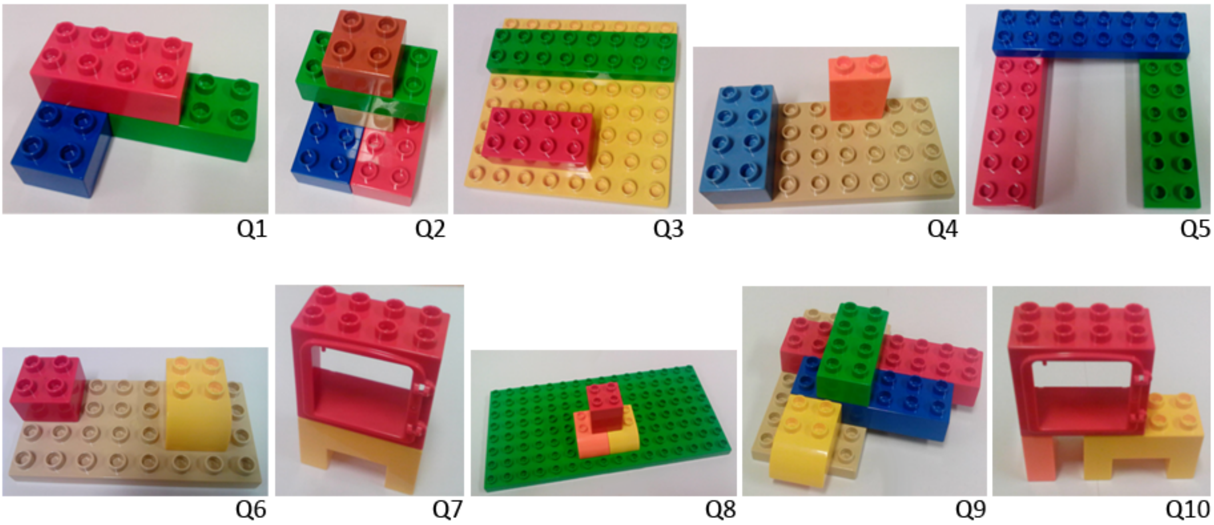

Figure 8: Acquired objects with multiple components (ten physical constructions were performed). Objects are composed by two to five blocks.

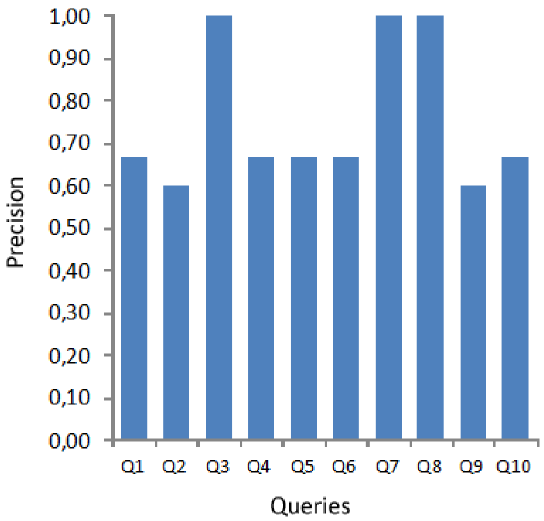

Figure 9: Objective tests: precision results. The plot represents precision obtained considering correct retrieved components within the first five results.

match selection). However, if we consider the first five results from retrieval, the correct answer was 75% of the time. This lead us to propose an application that asks if the construction is well performed and suggests the first five results if the construction is not correct at first glance. This result need to be considered because we also observed that most often plates were confused with bricks, which have slightly higher height.

In relation to time measures, we measured the total indexing time and the average time to construct an object part. Although the indexing time is done offline, the time spent to index all models in our database is quite insignificant (less than two minutes). This result is possible due to the use of SHA descriptor. The average time to construct a 3D object is defined by the sum of all phases. We realized that average time to retrieve sub-parts may become difficult to handle only when we have several sub-parts in the acquired scene. However, with the queries performed, it does not compromise our requirements (about 7 seconds to retrieve three sub-parts). In relation to average time to construct a 3D digital model, we evaluate this time by time spent to acquire the scene and compute the mesh (thereabout 25 seconds) plus $\sim$16 seconds (average) to perform the algorithm.

We conclude that the results achieved depend on the database used. First of all, the number of models in the component repository (our repository considers 70 models). Approaches referred in related work have much less models (1$\sim$3), which clearly help to retrieve object sub-parts in a more effective way. Second, there are a great amount of components in our repository that are similar among themselves. As a result, the identification of what virtual component correspond to an acquired physical component is a non-trivial challenge. In addiction to this problem, the use of blocks that fit one another increase the difficulty because blocks that have other blocks above that block are not well acquired. Although we surpassed this problem by creating a concave hull for every brick that fills the empty space, it is not totally reliable.

We also confirm that it is possible to have an application that represents models almost in real-time with this approach. However, the correctness of the construction performed depends strongly on the success of the acquisition, on the accuracy and also on the repository used.

 ISBN 978-80-86943-75-6

4.4 Limitations

For our problem setting, there are several limitations to overcome: (1) scanner resolution; (2) block size; and (3) Skanect acquisition results.

Low-cost scanners have some shortcomings, namely poor resolution, poor accuracy and also produce more noise. However, we hope that new advancements in this field are going to generate new technologies that decrease the problems mentioned.

The major limitation of our entire project is that the use of small blocks, allied with low resolution scanning, produces weak points clouds. Moreover, some DUPLO® blocks can cover one another preventing them from being totally captured. As a result, the generated shapes are partially incorrect and retrieved results are inaccurate.

The use of a software, such as Skanect or ReconstructMe, allows the registration of several views produced by scanner. Although the algorithms behind this kind of application are tested and are extremely efficient, they have some disadvantages, mainly for small size objects. Thus, we need to add several objects to our scene in order not to lose track. Afterwards, we got greater files to process and filtering are not trivial.

5 FUTURE WORK

Microsoft® is now working on Kinect 2.0, which have higher resolution when compared with the current version. Moreover, the RGB camera is upgraded from 34-bit RGB to 16-bit YUV, and a new infrared sensor is added. Therefore, we hope that using this configuration is going to help to acquire objects more efficiently.

In relation to segmentation, our algorithm needs and upgrade that allows to acquire at least two blocks with the same color. In order to achieve this, we will need to use a segmentation algorithm such as Region Growing. An upgrade that segments more different colors can also be added.

One part of our project consists on the integration of LTouchIT, providing an application that allows manipulation of acquired 3D objects. This is going to help users to evaluate our algorithm through a Likert scale. This evaluation is required because, although some results are not precisely correct, they may be close to what would be expected. Moreover, this will lead to ask if the presented result is correct and, if it is not correct the application will suggest the first five results (from retrieval phase). This process can be an automated task which is possible due to fine-tuning of our algorithm in combination with other methodologies.

We also consider to let users evaluate our algorithm, through subjective measures. On one hand, objective metrics gave us percentages and efficiency measures. On other hand, we want to know how acceptable could be one result in terms of quality. This evaluation is relevant because, although some results are not precisely correct, they may be close to what would be expected.

Our solution uses a toy-problem, LEGO® blocks, to demonstrate the construction algorithm. We would like to propose several proof-of-concept applications that include a similar approach, namely for health, mechanical or electric industries. For example, in mechanical industry, the identification of several physical components that belong to the car, such as radiator and engine. For all mentioned applications, the concept of learning can be improved to 3D and many appliances can be done. Imagine if I could scan a set of bones of a leg and add them virtually to other structure, using a construction algorithm to provide help.

6 CONCLUSION

This research is focused on identification of object components in order to construct a digitized 3D model. Our approach considers newly low-cost technologies, such as Microsoft Kinect® and the increase in computing power that allows storage and faster data availability. As a result, some efficient retrieval algorithms have appeared. Having in mind the recent technological boost, our solution aims to identify all physical components that belong to an object. The main contribution of this work is an application that performs a segmented construction of a 3D physical object acquired through a low-cost device, using color and shape information.

In comparison with similar approaches, our solution has the advantage of having a large number of components allowed to be identifiable, from a repository. In particular, our repository guarantees that 70 different physical components can be acquired through a low-cost device and detectable with our algorithm. However, based on the fact that blocks are covering each other, the precision achieved is reduced.

The experiments clearly indicate that small size blocks are complex to analyze through Microsoft Kinect® acquisition, taking into account its low accuracy. Subsequently, the produced point cloud is insufficient to identify physical components correctly. Therefore, the obtained results are not yet enough to generalize our approach. However, we consider that our solution may be applicable to many other domains if those physical components are larger in comparison with the size of our acquired blocks. Moreover, if the components in the repository were more different among themselves, it would produce more accurate results.

7 ACKNOWLEDGMENTS

This work was partially supported by FCT through the PIDDAC Program funds (INESC-ID multiannual funding), reference PEst-OE/EEI/LA0021/2013, and through the project 3DORuS, reference PTDC/EIA-EIA/102930/2008.

8 REFERENCES

[AB94] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[AFS06] Marco Attene, Bianca Falcidieno, and Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 2006.

[BR02] Fausto Bernardini and Holly Rushmeier. The 3D Model Acquisition Pipeline. *Comput. Graph. Forum*, 21(2):149–172, 2002.

[CTSO03] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.

[FJ03] MJ Fonseca and JA Jorge. NB-Tree: An indexing structure for content-based retrieval in large databases. *Technical report*, pages 1–25, 2003.

[FMK+03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22(1):83–105, 2003.

[GFCC12] Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. DuploTrack: a real-time system for authoring and guiding duplo block assembly. *In Proceedings of 25th ACM Symposium on User Interface Software and Technology (UIST)*, pages 389–401, 2012.

[IKH+11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard A. Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew J. Davison, and Andrew W. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In Jeffrey S. Pierce, Maneesh Agrawala, and Scott R. Klemmer, editors, *UIST*, pages 559–568. ACM, 2011.

[LRG+00] Marc Levoy, Szymon Rusinkiewicz, Matt Ginzton, Jeremy Ginsberg, Kari Pulli, David Koller, Sean Anderson, Jonathan Shade, Brian Curless, Lucas Pereira, James Davis, and Duane Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. *SIGGRAPH*, pages 131–144, 2000.

[LSFG11] H Laga, T Schreck, A Ferreira, and A Godil. Bag of words and local spectral descriptor for 3d partial shape retrieval. *Proceedings of the Eurographics Workshop on 3D Object Retrieval (3DOR'11)*, pages 41–48, 2011.

[MF11] Daniel Mendes and Alfredo Ferreira. Virtual LEGO Modelling on Multi-Touch Surfaces. *Visualization and Computer Vision (WSCG)*, 2011.

[MMWG11] Markus Mathias, Andelo Martinovic, Julien Weissenberg, and Luc Van Gool. Procedural 3D Building Reconstruction Using Shape Grammars and Detectors. *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 304–311, 2011.

[MN95] Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[MWC+12] Andrew Miller, Brandyn White, Emiko Charbonneau, Zach Kanzler, and Joseph J LaViola. Interactive 3D model acquisition and tracking of building block structures. *IEEE transactions on visualization and computer graphics*, 18(4):651–9, 2012.

[ND11] RA Newcombe and AJ Davison. KinectFusion: Real-time dense surface mapping and tracking. *(ISMAR)*, pages 127–136, 2011.

[RC11] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation*, pages 1–4, 2011.

[RMB+08] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 2008.

[RTHA12] Brian Curless Robert T. Held, Ankit Gupta and Maneesh Agrawala. 3d puppetry: A kinect-based interface for 3d animation. *Proceedings of the 25th annual ACM symposium adjunct on User interface software and technology*, 2012.

[STK02] Shymon Shlafman, Ayellet Tal, and Sagi Katz. Metamorphosis of Polyhedral Surfaces using Decomposition. *Computer Graphics Forum*, 21(3):219–228, 2002.

[SW11] C Schwartz and M Weinmann. Integrated high-quality acquisition of geometry and appearance for cultural heritage. *In proceedings of The 12th International Symposium on Virtual Reality, Archeology and Cultural Heritage*, 2011.

[TDVC11] Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. Deformable shape retrieval using bag-of-feature techniques. *Proceedings of the 3D Image Processing (3DIP'11)*, 2011.

[VMC96] T Varady, RR Martin, and Jordan Cox. Reverse engineering of geometric models - an introduction. *Computer-Aided Design*, pages 1–28, 1996.

 ISBN 978-80-86943-75-6