Recovering of body parts of articulated object

Anna M. Maureder

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Februar 2017

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, February 28, 2017

Anna M. Maureder

Contents

Abstract

This should be a 1-page (maximum) summary of your work in English.

Kurzfassung

An dieser Stelle steht eine Zusammenfassung der Arbeit, Umfang max. 1 Seite. Im Unterschied zu anderen Kapiteln ist die Kurzfassung (und das Abstract) üblicherweise nicht in Abschnitte und Unterabschnitte gegliedert. Auch Fußnoten sind hier falsch am Platz.

Kurzfassungen werden übrigens häufig – zusammen mit Autor und Titel der Arbeit – in Literaturdatenbanken aufgenommen. Es ist daher darauf zu achten, dass die Information in der Kurzfassung für sich *allein* (d. h. ohne weitere Teile der Arbeit) zusammenhängend und abgeschlossen ist. Insbesondere werden an dieser Stelle (wie u. a. auch im *Titel* der Arbeit und im *Abstract*) normalerweise *keine Literaturverweise* verwendet! Falls unbedingt solche benötigt werden – etwa weil die Arbeit eine Weiterentwicklung einer bestimmten, früheren Arbeit darstellt –, dann sind *vollständige* Quellenangaben in der Kurzfassung selbst notwendig, z. B. [Zobel J.: *Writing for Computer Science – The Art of Effective Communication.* Springer-Verlag, Singapur, 1997].

Auch sollte daran gedacht werden, dass bei der Aufnahme in Datenbanken Sonderzeichen oder etwa Aufzählungen mit „Knödellisten“ in der Regel verloren gehen. Dasselbe gilt natürlich auch für das *Abstract*.

Inhaltlich sollte die Kurzfassung *keine* Auflistung der einzelnen Kapitel sein (dafür ist das Einleitungskapitel vorgesehen), sondern dem Leser einen kompakten, inhaltlichen Überblick über die gesamte Arbeit verschaffen. Der hier verwendete Aufbau ist daher zwangsläufig anders als der in der Einleitung.

Chapter 1

Introduction

Chapter 2

Related Work

Here comes the State-of the Art. An overview of related methods to Non-rigid Registration for detecting rigid body parts of an articulated object are mentioned here [3] which mainly use the ICP (iterative closest point) and the PCA (Principal component analysis) to find corresponding body parts. This paper is based on the Correlated Correspondance algorithm [2] [1] and Symmetrization [5]. A following work to [3] is [4]. Other methods include temporal coherence, markers and user inputs.

2.1 Marker, User input

blabla

2.2 Non-rigid Registration

blabla

2.2.1 EM-algorithm

2.2.2 LRP

2.2.3 Symmetrization

Chapter 3

My contribution

This chapter optimizes existing methods for object segmentation and detection of part correspondances. As input the methods take two point clouds that represent two similar objects in two different poses.

3.1 EM algorithm for articulated objects with two rigid parts

3.1.1 Algorithm

The algorithm starts with two sets of point clouds S_0 and D_0 of the same object in different configurations. The objects are composed of two rigid parts. S_0 is used as a *template* to be registered with D_0. The goal is to find a part assignment $P = \{part_0,\ part_1\}$ and transformation $T = \{T_0,\ T_1\}$ for all points of the *template* that alligns them with all points of D_0. To iteratively find corresponding parts and transformations, S_0 as well as D_0 are divided into assumed rigid parts. The dividers d_S and d_D are initially defined with the secondary axis s_S and s_D. The resulting rigid parts are matched together with the ICP. Depending on the matching errors e_{left} and e_{right}, the dividers are slided alongside the principal axis p_S and p_D of the objects. The algorithm termintates if the total error $e_{total} = e_{right} + e_{left}$ doesn't get any smaller.

3.1.2 Steps

1. The centroids c_S and c_D of S_0 and D_0 are computed.
2. The principal axis p_S and p_D are computed through c_S and c_D in order to orient the point clouds horizonally around their centroids.
3. The secondary axis s_S and s_D perpendicular to p_S and p_D through c_S and c_D are computed.

4. The dividers d_S and d_T to segment S_0 and D_0 into its assumed two rigid parts are initialized with the secondary axis s_S and s_D.
5. The points $P_{0...N}$ of S_0 are either allocated to S_{left} or S_{right} depending on its position to d_S. The same procedure is done with all points of D_0.
6. ICP is computed between the rigid parts S_{left} and D_{left} as well as S_{right} and D_{right}.
7. An error distance e_{left} and e_{right} is obtained. The part with the most error per point is assumed to be not rigid which gives back an indicator where to divide S_0 and D_0.
8. The dividers d_S and d_D are shifted to the direction of the highest error. To be continued from step 5 until the total error e_{total} doesn't get smaller.

3.1.3 Segmenting objects with more parts

In case of having an unknown number of rigid parts n, the algorithm above has to be applied recursively in order to find all part assignments $P = \{part_0 \; ... \; part_n\}$. S_0 and D_0 are thereby initially divided into two assumed rigid parts by the dividers d_S and d_D initialized with s_S and s_D. The goal is now to find single parts by sliding another divider over S_{left} and D_{left} as well as S_{right} and D_{right} until the error e for one part doesn't get any smaller. The total error e_{total} is not used any more as dividing one part into two doesn't ensure that they are both rigid. After assigning points to a Part P the geodesic distance between points of rigid parts can be used to find further connecting parts. By taking the dividers as joints and taking into account that rigid parts are located between the same joints, rigid parts in the middle of the object can be easier detected.

3.2 Points-to-Ellipse fitting

3.2.1 Algorithm

This algorithm only requires one point cloud. The basic idea is to segment the non-rigid object S_0 into its rigid parts $part_1$ and part$_2$ by fitting ellipses to its rigid parts. S_0 is divided perpendicular to its principal axis p_0 into two assumed rigid parts S_{left} and S_{right}, initially defining the divider d with the secondary axis s_0. The points of S_{left} and S_{right} are verified to form an ellipse by using its formular

$$\frac{x^2}{r_1^2} + \frac{y^2}{r_2^2} = 1$$

Assuming to verify S_{left} forming an ellipse, r_1 is half the length of the principal axis p_{left} of S_{left} through its centroid c_{left}. Furthermore, r_2 is half the length of the secondary axis s_{left} of S_{left}. Thereby,the centroid c_{left} needs to be located in the origin (0,0). Now, to check whether a point pt_i of S_{left} is located on the ellipse, the formular is remodeled and its x values is applied.

$$(1 - \frac{x^2}{r_1^2}) \cdot r_2^2 = y^2$$

The resulting y-value of the ellipse is compared to the points actual y-value. Given a certain threshold τ a point either accounts to the number of total points lying on the ellipse n, or not.

$$n = \sum_{i=0}^{m} \{pt_i| \quad \|pt_i.y^2 - y^2\| < \tau\}$$

The algorithm is repeated by sliding d in the direction of the highest error e. To be continued until the total error $e_{total} = e_{left} + {}_{right}$ reaches its minimum.

3.2.2 Steps

1. The centroid c_0 of S_0 is computed.
2. The principal axis p_0 is computed through c_0 and S_0 horizontally oriented.
3. The secondary axis s_0 perpendicular to p_0 through c_0 is computed.
4. The divider d is initialized with the secondary axis s_0 to segment S_0 into two assumed rigid parts .
5. The points of S_0 are either allocated to S_{left} or S_{right} depending on its position to d_0.
6. The ellipse formular is applied on S_{left} and S_{right}.
7. An error e_{left} and e_{right} is obtained implying how many points of S_{left} and S_{right} form an ellipse.
8. The divider d is shifted to the direction of the highest error. To be continued from step 5 until the total error e_{total} doesn't get smaller.

3.2.3 Reusing detected shapes

After termination of the algorithm, one point cloud can be segmented into its rigid parts P $\{\mathrm{part}_1, ..., \mathrm{part}_n\}$. Their variables like the ellipses' centroid c_i and radii r_1, r_2 can be used to segment similar point clouds in different configurations. As the shapes to be matched are already known, e.g. how they are linked, finding the position to be segmented is a lot easier.

3.2.4 Results

3.3 LRP algorithm

3.3.1 Overview

As an initial step, the LRP algorithm tries to find the most reliable correspondences, the so-called largest rigid part (LRP), subsequently all other parts are detected that are linked to the LRP. The initial alignment stage tries to find sparse correspondences between two point clouds by applying a single rigid transformation to detect the largest subsets of points in two point clouds. Starting from the LRP all other parts are detected recursively.

3.3.2 Algorithm

Finding the LRP

The algorithm also takes two point clouds S_0 and T_0 of the same object in different configurations as input. The goal is to find a single rigid transformation T_{init} for all points of S_0 to get potential corresponding points $C_0 = \{(s_i, t_j)\}$ in T_0. For that, local descriptors of S_0 and T_0 are computed. The requirement for a sparse correspondance between two points s_i and t_j is that they are *reciprocal*, which means that the Euclidean distance $d(s_i, t_j)$ between them is the smallest in both directions. Some of the sparse correspondances are asumed to be wrong. Therefore, RANSAC is used on the sparse correspondances C_0 to estimate a rigid alignment that is supported by the largest number of points n from S_0 and T_0. To assign the LRP in S_0 and T_0, the biggest point clusters C_s and C_t of the overlapping area $G_s = \{C_1, \dots , C_n\}$ and $G_t = \{C_1, \dots , C_n\}$ are detected.

Part discovery

The remaining clusters from S_0 and T_0 that have not been registered yet are matched recursively by starting with clusters connected to already matched parts. First, all matched parts are excluded from the input point clouds $G_{s(l+1)} = S_0$ - C_{sl} and $G_{t(l+1)} = T_0$ - C_{tl} defining l as the number of already matched parts {1, ..., n}, C_{sl}. For that clusters are formed, using region taking into account that they are attached to already registered parts. The algorithm explained is applied until all body parts have been discovered.

3.3.3 Steps

1. The centroids c_s and c_t of S_0 and T_0 are computed.
2. The principal axis p_s and p_t are computed through c_s and c_t in order to horizontally orient the objects around their centroids.

3. The ICP is conducted as a first guess to find a transformation T_{init} for all points from S_0 that results in the highest number of corresponding points n in T_0, given the threshold T.
4. C_0 contains the corresponding points from S_0 and T_0, resulting from $T_{init}(S_0)$.
5. The RANSAC approach is applied on C_0 to find a T_f that results in the highest number of corresponding points n between $T_f(S_0)$ and T_0.
6. The LRP is assigned to C_s and C_t from the resulting point clusters G_s and G_t.
7. Starting from parts that are connected to the LRP, corresponding points C_i for unmatched points from S_0 and T_0 are seeked. The clusters are given as a input from Step 5.

3.3.4 Results

3.4 Symmetrization

Chapter 4

Conclusion

To conclude, I proposed... The results are ...

4.1 Future work

Future developments can be done by ....

References

Literature

[1] Dragomir Anguelov et al. “Recovering Articulated Object Models from 3D Range Data”. In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI ’04. Banff, Canada: AUAI Press, 2004, pp. 18–26. URL: http://dl.acm.org/citation.cfm?id=1036843.1036846 (cit. on p. 2).

[2] Dragomir Anguelov et al. “The Correlated Correspondence Algorithm for Unsupervised Registration of Nonrigid Surfaces”. In: *Advances in Neural Information Processing Systems 17*. Ed. by L. K. Saul, Y. Weiss, and L. Bottou. MIT Press, 2005, pp. 33–40. URL: http://papers.nips.cc/paper/2601-the-correlated-correspondence-algorithm-for-unsupervised-registration-of-nonrigid-surfaces.pdf (cit. on p. 2).

[3] Will Chang and Matthias Zwicker. “Automatic Registration for Articulated Shapes”. *Computer Graphics Forum (Proceedings of SGP 2008)* 27.5 (2008), pp. 1459–1468 (cit. on p. 2).

[4] Will Chang and Matthias Zwicker. “Range Scan Registration Using Reduced Deformable Models”. *Computer Graphics Forum (Proceedings of Eurographics 2009), to appear* () (cit. on p. 2).

[5] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly. “Symmetrization”. *ACM Trans. Graph.* 26.3 (July 2007). URL: http://doi.acm.org/10.1145/1276377.1276456 (cit. on p. 2).