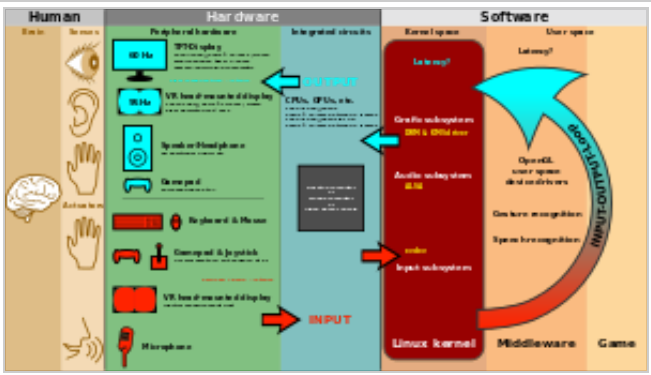
Gesture recognition

From Wikipedia, the free encyclopedia

Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. Current focuses in the field include emotion recognition from the face and hand gesture recognition. Many approaches have been made using cameras and computer vision algorithms to interpret sign language. However, the identification and recognition of posture, gait, proxemics, and human behaviors is also the subject of gesture recognition techniques.[1] Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs (graphical user interfaces), which still limit the majority of input to keyboard and mouse.

A child being sensed by a simple gesture recognition algorithm detecting hand location and movement

Gesture recognition enables humans to communicate with the machine (HMI) and interact naturally without any mechanical devices. Using the concept of gesture recognition, it is possible to point a finger at the computer screen so that the cursor will move accordingly. This could potentially make conventional input devices such as mouse, keyboards and even touch-screens redundant.



Gesture recognition is usually processed in middleware, the results are transmitted to the user applications.

Gesture recognition can be conducted with techniques from computer vision and image processing.

The literature includes ongoing work in the computer vision field on capturing gestures or more general human pose and movements by cameras connected to a computer.[2][3][4][5]

Gesture recognition and pen computing: This computing not only going to reduce the hardware impact of the system but also it increases the range of usage of physical world object instead of digital object like keyboards, mouses. Using this we can implement and can create a new thesis of creating of new hardware no requirement of monitors too. This idea may lead us to the creation of holographic display. The term gesture recognition has been used to refer more narrowly to non-text-input handwriting symbols, such as inking on a graphics tablet, multi-touch gestures, and mouse gesture recognition. This is computer interaction through the drawing of symbols with a pointing device cursor.[6][7][8] (see Pen computing)

Contents

Gesture types

In computer interfaces, two types of gestures are distinguished:[9] We consider online gestures, which can also be regarded as direct manipulations like scaling and rotating. In contrast, offline gestures are usually processed after the interaction is finished; e. g. a circle is drawn to activate a context menu.

- Offline gestures: Those gestures that are processed after the user interaction with the object. An example is the gesture to activate a menu.
- Online gestures: Direct manipulation gestures. They are used to scale or rotate a tangible object.

Input devices

The ability to track a person's movements and determine what gestures they may be performing can be achieved through various tools. Although there is a large amount of research done in image/video based gesture recognition, there is some variation within the tools and environments used between implementations.

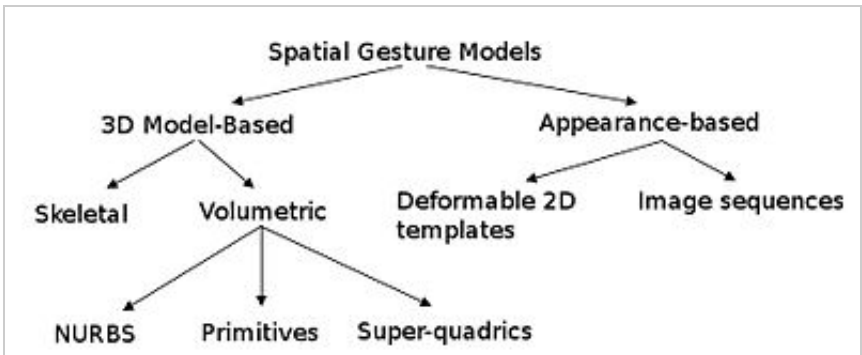
- **Wired gloves**. These can provide input to the computer about the position and rotation of the hands using magnetic or inertial tracking devices. Furthermore, some gloves can detect finger bending with a high degree of accuracy (5-10 degrees), or even provide haptic feedback to the user, which is a simulation of the sense of touch. The first commercially available hand-tracking glove-type device was the DataGlove,[10] a glove-type device which could detect hand position, movement and finger bending. This uses fiber optic cables running down the back of the hand. Light pulses are created and when the fingers are bent, light leaks through small cracks and the loss is registered, giving an approximation of the hand pose.
- **Depth-aware cameras**. Using specialized cameras such as structured light or time-of-flight cameras, one can generate a depth map of what is being seen through the camera at a short range, and use this data to approximate a 3d representation of what is being seen. These can be effective for detection of hand gestures due to their short range capabilities.[11]
- **Stereo cameras**. Using two cameras whose relations to one another are known, a 3d representation can be approximated by the output of the cameras. To get the cameras' relations,

one can use a positioning reference such as a lexian-stripe or infrared emitters.[12] In combination with direct motion measurement (6D-Vision) gestures can directly be detected.

- **Controller-based gestures**. These controllers act as an extension of the body so that when gestures are performed, some of their motion can be conveniently captured by software. Mouse gestures are one such example, where the motion of the mouse is correlated to a symbol being drawn by a person's hand, as is the Wii Remote or the Myo, which can study changes in acceleration over time to represent gestures.[13][14][15] Devices such as the LG Electronics Magic Wand, the Loop and the Scoop use Hillcrest Labs' Freespace technology, which uses MEMS accelerometers, gyroscopes and other sensors to translate gestures into cursor movement. The software also compensates for human tremor and inadvertent movement.[16][17][18] AudioCubes are another example. The sensors of these smart light emitting cubes can be used to sense hands and fingers as well as other objects nearby, and can be used to process data. Most applications are in music and sound synthesis,[19] but can be applied to other fields.
- **Single camera**. A standard 2D camera can be used for gesture recognition where the resources/environment would not be convenient for other forms of image-based recognition. Earlier it was thought that single camera may not be as effective as stereo or depth aware cameras, but some companies are challenging this theory. Software-based gesture recognition technology using a standard 2D camera that can detect robust hand gestures, hand signs, as well as track hands or fingertip at high accuracy has already been embedded in Lenovo's Yoga ultrabooks , Pantech's Vega LTE smartphones, Hisense's Smart TV models, among other devices.

Algorithms

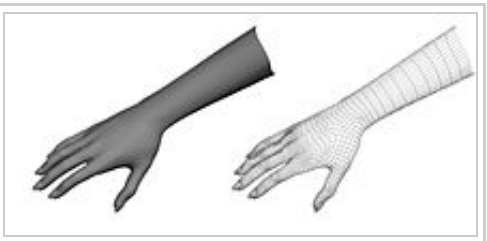
Depending on the type of the input data, the approach for interpreting a gesture could be done in different ways. However, most of the techniques rely on key pointers represented in a 3D coordinate system. Based on the relative motion of these, the gesture can be detected with a high accuracy, depending of the quality of the input and the algorithm's approach.
In order to interpret movements of the body, one has to classify them according to common properties and the message the movements may express. For example, in sign language each gesture represents a word or phrase. The taxonomy that seems very appropriate for Human-



Different ways of tracking and analyzing gestures exist, and some basic layout is given is in the diagram above. For example, volumetric models convey the necessary information required for an elaborate analysis, however they prove to be very intensive in terms of computational power and require further technological developments in order to be implemented for real-time analysis. On the other hand, appearance-based models are easier to process but usually lack the generality required for Human-Computer Interaction.

Computer Interaction has been proposed by Quek in "Toward a Vision-Based Hand Gesture Interface".[20] He presents several interactive gesture systems in order to capture the whole space of the gestures: 1. Manipulative; 2. Semaphoric; 3. Conversational.

Some literature differentiates 2 different approaches in gesture recognition: a 3D model based and an appearance-based.[21] The foremost method makes use of 3D information of key elements of the body parts in order to obtain several important parameters, like palm position or joint angles. On the other hand, Appearance-based systems use images or videos for direct interpretation.



A real hand (left) is interpreted as a collection of vertices and lines in the 3D mesh version (right), and the software uses their relative position and interaction in order to infer the gesture.

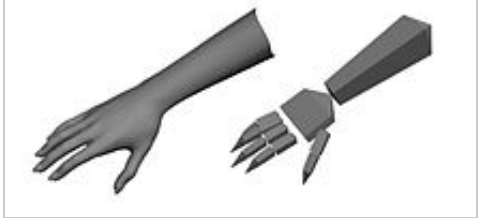
3D model-based algorithms

The 3D model approach can use volumetric or skeletal models, or even a combination of the two. Volumetric approaches have been heavily used in computer animation industry and for computer vision purposes. The models are generally created of complicated 3D surfaces, like NURBS or polygon meshes.

The drawback of this method is that is very computational intensive, and systems for live analysis are still to be developed. For the moment, a more interesting approach would be to map simple primitive objects to the person's most important body parts (for example cylinders for the arms and neck, sphere for the head) and analyse the way these interact with each other. Furthermore, some abstract structures like super-quadrics and generalised cylinders may be even more suitable for approximating the body parts. The exciting thing about this approach is that the parameters for these objects are quite simple. In order to better model the relation between these, we make use of constraints and hierarchies between our objects.

Skeletal-based algorithms

Instead of using intensive processing of the 3D models and dealing with a lot of parameters, one can just use a simplified version of joint angle parameters along with segment lengths. This is known as a skeletal representation of the body, where a virtual skeleton of the person is computed and parts of the body are mapped to certain segments. The analysis here is done using the position and orientation of these segments and the relation between each one of them(for example the angle between the joints and the relative position or orientation)



The skeletal version (right) is effectively modelling the hand (left). This has fewer parameters than the volumetric version and it's easier to compute, making it suitable for real-time gesture analysis systems.

Advantages of using skeletal models:

- Algorithms are faster because only key parameters are analyzed.
- Pattern matching against a template database is possible
- Using key points allows the detection program to focus on the significant parts of the body

Appearance-based models

These models don't use a spatial representation of the body anymore, because they derive the parameters directly from the images or videos using a template database. Some are based on the deformable 2D templates of the human parts of the body, particularly hands. Deformable templates are sets of points on the outline of an object, used as interpolation nodes for the object's outline approximation. One of the simplest interpolation function is linear, which performs an average shape from point sets, point variability parameters and external deformators. These template-based models are mostly used for hand-tracking, but could also be of use for simple gesture classification.

These binary silhouette(left) or contour(right) images represent typical input for appearance-based algorithms. They are compared with different hand templates and if they match, the correspondent gesture is inferred.

A second approach in gesture detecting using appearance-based models uses image sequences as gesture templates. Parameters for this method are either the images themselves, or certain features derived from these. Most of the time, only one (monoscopic) or two (stereoscopic) views are used.

Challenges

There are many challenges associated with the accuracy and usefulness of gesture recognition software. For image-based gesture recognition there are limitations on the equipment used and image noise. Images or video may not be under consistent lighting, or in the same location. Items in the background or distinct features of the users may make recognition more difficult.

The variety of implementations for image-based gesture recognition may also cause issue for viability of the technology to general usage. For example, an algorithm calibrated for one camera may not work for a different camera. The amount of background noise also causes tracking and recognition difficulties, especially when occlusions (partial and full) occur. Furthermore, the distance from the camera, and the camera's resolution and quality, also cause variations in recognition accuracy.

In order to capture human gestures by visual sensors, robust computer vision methods are also required, for example for hand tracking and hand posture recognition[22][23][24][25][26][27][28][29][30] or for capturing movements of the head, facial expressions or gaze direction.

"Gorilla arm"

"Gorilla arm" was a side-effect of vertically oriented touch-screen or light-pen use. In periods of prolonged use, users' arms began to feel fatigue and/or discomfort. This effect contributed to the decline of touch-screen input despite initial popularity in the 1980s.[31][32]

In order to measure arm fatigue and the gorilla arm side effect, researchers developed a technique called Consumed Endurance.[33]

References

1. ^ Matthias Rehm, Nikolaus Bee, Elisabeth André, Wave Like an Egyptian - Accelerometer Based Gesture Recognition for Culture Specific Interactions (http://mm-werkstatt.informatik.uni-augsburg.de/files/publications/199/wave_like_an_egyptian_final.pdf), British Computer Society, 2007
2. ^ Pavlovic, V., Sharma, R. & Huang, T. (1997), "Visual interpretation of hand gestures for human-computer

interaction: A review" (http://www.cs.rutgers.edu/~vladimir/pub/pavlovic97pami.pdf), IEEE Trans. Pattern Analysis and Machine Intelligence., July, 1997. Vol. 19(7), pp. 677 -695.

3. ^ R. Cipolla and A. Pentland, Computer Vision for Human-Machine Interaction (http://books.google.com/books?id=Pe7gG0LxEUIC&dq=pentland+cipolla+computer+vision+human+interaction&printsec=frontcover&source=bl&ots=O2q5ExL8PU&sig=FMhom_f4h9dqeib-6pSSpjbsB38&hl=en&ei=uzvsSbruBdqIsAaq5PCKBw&sa=X&oi=book_result&ct=result&resnum=1), Cambridge University Press, 1998, ISBN 978-0-521-62253-0
4. ^ Ying Wu and Thomas S. Huang, "Vision-Based Gesture Recognition: A Review" (http://reference.kfupm.edu.sa/content/v/i/vision_based_gesture_recognition__a_revi_291732.pdf), In: Gesture-Based Communication in Human-Computer Interaction, Volume 1739 of Springer Lecture Notes in Computer Science, pages 103-115, 1999, ISBN 978-3-540-66935-7, doi:10.1007/3-540-46616-9 (http://dx.doi.org/10.1007%2F3-540-46616-9)
5. ^ Alejandro Jaimesa and Nicu Sebe, Multimodal human–computer interaction: A survey (http://staff.science.uva.nl/~nicu/PUBS/PDF/2005/sebeHCI05.pdf), Computer Vision and Image Understanding Volume 108, Issues 1-2, October–November 2007, Pages 116-134 Special Issue on Vision for Human-Computer Interaction, doi:10.1016/j.cviu.2006.10.019 (http://dx.doi.org/10.1016%2Fj.cviu.2006.10.019)
6. ^ Dopertchouk, Oleg; "Recognition of Handwriting Gestures" (http://www.gamedev.net/page/resources/_/technical/game-programming/recognition-of-handwritten-gestures-r2039), *gamedev.net*, January 9, 2004
7. ^ Chen, Shijie; "Gesture Recognition Techniques in Handwriting Recognition Application" (http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5693514&tag=1), *Frontiers in Handwriting Recognition* p 142-147 November 2010
8. ^ Balaji, R; Deepu, V; Madhvanath, Sriganesh; Prabhakaran, Jayasree "Handwritten Gesture Recognition for Gesture Keyboard" (http://www.hpl.hp.com/india/documents/papers/GKB_IWFHR10_Final.pdf), *Hewlett-Packard Laboratories*
9. ^ Dietrich Kammer, Mandy Keck, Georg Freitag, Markus Wacker, Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features (http://vi-c.de/vic/sites/default/files/Taxonomy_and_Overview_of_Multi-touch_Frameworks_Revised.pdf)
10. ^ Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson and Young Harvill. http://portal.acm.org. "A HAND GESTURE INTERFACE DEVICE (http://netzspannung.org/cat/servlet/CatServlet/$files/228648/DataGlove+CHI+1987.pdf)." http://portal.acm.org.
11. ^ Yang Liu, Yunde Jia, A Robust Hand Tracking and Gesture Recognition Method for Wearable Visual Interfaces and Its Applications, Proceedings of the Third International Conference on Image and Graphics (ICIG'04), 2004
12. ^ Kue-Bum Lee, Jung-Hyun Kim, Kwang-Seok Hong, An Implementation of Multi-Modal Game Interface Based on PDAs, Fifth International Conference on Software Engineering Research, Management and Applications, 2007
13. ^ Per Malmestig, Sofie Sundberg, SignWiiver - implementation of sign language technology (http://www.tricomsolutions.com/academic_reports.html)
14. ^ Thomas Schlomer, Benjamin Poppinga, Niels Henze, Susanne Boll, Gesture Recognition with a Wii Controller (http://www.wiigee.com/download_files/gesture_recognition_with_a_wii_controller-

schloemer_poppinga_henze_boll.pdf), Proceedings of the 2nd international Conference on Tangible and Embedded interaction, 2008

15. ^ AiLive Inc., LiveMove White Paper (http://www.ailive.net/papers/LiveMoveWhitePaper_en.pdf), 2006
16. ^ *Electronic Design* September 8, 2011. William Wong. Natural User Interface Employs Sensor Integration. (http://electronicdesign.com/article/embedded/Natural-User-Interface-Employs-Sensor-Integration.aspx)
17. ^ *Cable & Satellite International* September/October, 2011. Stephen Cousins. A view to a thrill. (http://www.csimagazine.com/csi/A-view-to-a-thrill.php)
18. ^ *TechJournal South* January 7, 2008. Hillcrest Labs rings up $25M D round. (http://www.techjournalsouth.com/2008/01/hillcrest-labs-rings-up-25m-d-round/)
19. ^ *Percussa AudioCubes Blog* October 4, 2012. Gestural Control in Sound Synthesis. (http://www.percussa.com/2012/10/04/gestural-control-of-sound-synthesis-featured-question/)
20. ^ Quek, F., "Toward a vision-based hand gesture interface (http://books.google.co.uk/books?hl=en&lr=&id=ywTGrWPf518C&oi=fnd&pg=PA17&dq=towards+a+vision-based+hand+gesture&ots=Vr9HpJO6IE&sig=PJHYTFu-MtRKn0Q2Ng86vEhZNhU#v=onepage&q=towards%20a%20vision-based%20hand%20gesture&f=false)" Proceedings of the Virtual Reality System Technology Conference, pp. 17-29, August 23–26, 1994, Singapore
21. ^ Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction (http://www.cs.rutgers.edu/~vladimir/pub/pavlovic97pami.pdf); A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997
22. ^ Ivan Laptev and Tony Lindeberg "Tracking of Multi-state Hand Models Using Particle Filtering and a Hierarchy of Multi-scale Image Features" (http://www.csc.kth.se/cvap/abstracts/cvap245.html), Proceedings Scale-Space and Morphology in Computer Vision, Volume 2106 of Springer Lecture Notes in Computer Science, pages 63-74, Vancouver, BC, 1999. ISBN 978-3-540-42317-1, doi:10.1007/3-540-47778-0 (http://dx.doi.org/10.1007%2F3-540-47778-0)
23. ^ von Hardenberg, Christian; Bérard, François (2001). "Bare-hand human-computer interaction". *Proceedings of the 2001 workshop on Perceptive user interfaces*. ACM International Conference Proceeding Series. 15 archive. Orlando, Florida. pp. 1–8. CiteSeerX: 10.1.1.23.4541 (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.4541).
24. ^ Lars Bretzner, Ivan Laptev, Tony Lindeberg "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering" (http://www.csc.kth.se/cvap/abstracts/BreLapLin-FG02.html), Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 21–21 May 2002, pages 423-428. ISBN 0-7695-1602-5, doi:10.1109/AFGR.2002.1004190 (http://dx.doi.org/10.1109%2FAFGR.2002.1004190)
25. ^ Domitilla Del Vecchio, Richard M. Murray Pietro Perona, "Decomposition of human motion into dynamics-based primitives with application to drawing tasks" (http://www.cds.caltech.edu/~ddomitilla/reports/AutomaticaReport.pdf), Automatica Volume 39, Issue 12, December 2003, Pages 2085-2098 , doi:10.1016/S0005-1098(03)00250-4 (http://dx.doi.org/10.1016%2FS0005-1098%2803%2900250-4).
26. ^ Thomas B. Moeslund and Lau Nørgaard, "A Brief Overview of Hand Gestures used in Wearable Human Computer Interfaces" (http://www.vision.auc.dk/~tbm/Publications/gesture-hci.pdf), Technical report: CVMT 03-02, ISSN: 1601-3646, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark.
27. ^ M. Kolsch and M. Turk "Fast 2D Hand Tracking with Flocks of Features and Multi-Cue Integration" (http://ilab.cs.ucsb.edu/projects/mathias/KolschTurk2004Fast2DHandTrackingWithFlocksOfFeatures.pdf),

CVPRW '04. Proceedings Computer Vision and Pattern Recognition Workshop, May 27-June 2, 2004, doi:10.1109/CVPR.2004.71 (http://dx.doi.org/10.1109%2FCVPR.2004.71)
28. ^ Xia Liu Fujimura, K., "Hand gesture recognition using depth data", Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, May 17–19, 2004 pages 529- 534, ISBN 0-7695-2122-3, doi:10.1109/AFGR.2004.1301587 (http://dx.doi.org/10.1109%2FAFGR.2004.1301587).
29. ^ Stenger B, Thayananthan A, Torr PH, Cipolla R: "Model-based hand tracking using a hierarchical Bayesian filter" (http://www.bmva.ac.uk/sullivan/prizethesis-2005.pdf), IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(9):1372-84, Sep 2006.
30. ^ A Erol, G Bebis, M Nicolescu, RD Boyle, X Twombly, "Vision-based hand pose estimation: A review" (http://www.cse.unr.edu/~bebis/handposerev.pdf), Computer Vision and Image Understanding Volume 108, Issues 1-2, October–November 2007, Pages 52-73 Special Issue on Vision for Human-Computer Interaction, doi:10.1016/j.cviu.2006.10.012 (http://dx.doi.org/10.1016%2Fj.cviu.2006.10.012).
31. ^ Windows 7? No arm in it - Mixed Signals - Rupert Goodwins's Blog at ZDNet.co.uk Community (http://www.zdnet.com/windows-7-no-arm-in-it-4010008314/)
32. ^ The Jargon File - Gorilla Arm (http://www.catb.org/jargon/html/G/gorilla-arm.html)
33. ^ Consumed endurance: a metric to quantify arm fatigue of mid-air interactions (http://dx.doi.org/10.1145/2556288.2557130)

External links

- Annotated bibliography of references to gesture and pen computing (http://ruetersward.com/biblio.html)
- Notes on the History of Pen-based Computing (YouTube) (http://www.youtube.com/watch?v=4xnqKdWMa_8&feature=youtu.be)
- The future, it is all a Gesture (http://www.bruceongames.com/2007/10/02/the-future-it-is-all-a-gesture/)—Gesture interfaces and video gaming
- Ford's Gesturally Interactive Advert (http://www.inition.co.uk/case-study/ford-c-max-campaign-ar-gestural-interface)—Gestures used to interact with digital signage
- Kinetic Space (http://kineticspace.colorfulbit.com/)—3D Gesture Recognition, Training & Analysis
- Look Ma No Trackpad (http://www.wired.com/epicenter/2012/03/look-ma-no-trackpad/)—Gestures used to control Spotify & iTunes
- The Gesture Recognition Toolkit (http://www.nickgillian.com/wiki/pmwiki.php?n=GRT.GestureRecognitionToolkit)—An open-source toolkit for gesture recognition

Retrieved from "http://en.wikipedia.org/w/index.php?title=Gesture_recognition&oldid=619126019"

Categories: Applications of computer vision | Virtual reality | Object recognition and categorization | User interface techniques | History of human–computer interaction

- This page was last modified on 30 July 2014 at 12:10.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.