Quadratic Encoding for Hand Pose Reconstruction from Multi-Touch Input

S. Chung[1] J. Kim[1] S. Han[1] N. S. Pollard[1]

[1]Carnegie Mellon University, United States of America

Abstract

One of the most compelling challenges in virtual reality today is to allow users to carry out virtual manipulation tasks using their hands. Multi-touch devices are an interesting interface for this task, as they are widely available, they provide users with some haptic sensation of their motions, and they give very precise locations of the fingertips. We introduce a quadratic encoding technique to provide plausible and smooth hand reconstructions from multi-touch input at real-time rates suitable for virtual reality applications. Another nice feature of our data-driven approach is that it does not require explicit identification or registration of fingers. We show that quadratic encoding outperforms linear encoding, cubic encoding, and a PCA based inverse kinematics approach, and is well suited for performing real-time virtual manipulation using a multi-touch device.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Animation Computer Graphics [I.3.6]: Methodology and Techniques—Interaction Techniques

1. Introduction and Related Works

Real-time creation or reconstruction of hand motion is of increasing interest, as shown by advent of devices such as the LeapMotion [Lea] and the Nimble Sense [Nim] and the vast array of visual hand tracking research such as [QSW*14]. Multi-touch devices offer interesting opportunities in this space, as they are widely available and inexpensive, they deliver precise fingertip position information, and they provide some feeling of touch to the user and support for the hand. However, with a multi-touch device, we must be able to reconstruct complete hand motions from fingertip positions in real-time and without knowing which fingers are responsible for which contacts. Furthermore, the result should be simultaneously smooth, controlled, precise, and natural.

Other researchers have also worked on reconstruction of hand pose from reduced dimensional input data. El Koura and colleagues reconstruct hand motions specifically for guitar playing, using a motion capture database to capture sympathetic motions of the fingers [ES03]. Hamer and colleagues [HGUVG11] reconstruct hand motion from object motion by retrieving acceptable hand motions from a captured database. Ye and colleagues [YL12] reconstruct plausible hand motion from motion capture of the full body up to and including the wrist along with motion capture of the manipulated object. Mulatto et al. perform an inverse kinematics approach based on thumb and index finger positions measured by a haptic device, taking into account a set of linear dependencies between the joint angles that they call synergies [MFMP13]. Hoyet and colleagues [HRMO12] provide evidence from human subjects experiments for the perceptual validity of reconstructing hand motion from reduced marker sets. Chang and colleagues [CPMX07] explore minimal marker sets for grasp discrimination.

Our work differs from these works in that we use an encoding function as a compact representation of a motion capture database. Our quadratic encoding approach allows us to create smooth, natural hand motion in real-time from multi-touch fingertip inputs. Our approach gracefully handles changing dimensionality in input data as fingers are added to and removed from the multi-touch device. In addition, we automatically handle ambiguity in input signals, because we must identify which finger is responsible for which contact on the multi-touch device, even as the configuration and number of fingers are continuously changing.

Our results show that quadratic encoding is superior to linear encoding, cubic encoding, and PCA based inverse kine-

DOI: 10.2312/egsh.20151003

matics. We demonstrate the ability to simulate a hand manipulating objects in a virtual environment in real-time using a multi-touch device as input. The contributions of this paper are: first example of hand pose reconstruction from multi-touch input, quadratic encoding technique for smooth hand pose reconstruction, demonstration that changing contacts can be handled gracefully by using the same quadratic function for all contact conditions, an algorithm for estimating which fingers are responsible for which contact on a multi-touch input device, and demonstration that quadratic encoding can represent joint angle values successfully even for hand motion in manipulation tasks.

2. Quadratic Encoding

Quadratic encoding attempts to represent outputs as a quadratic function of inputs while retaining the full dimensionality (and ideally expressiveness) of both. Quadratic encoding has been used previously to encode energy values and center of mass trajectories for foot placements in humanoid robot walking [KPA13]. In this paper, we consider whether this approach can successfully encode the detail of hand pose consisting of joint angles and the wrist configuration in the challenging situation when inputs (fingertip positions) are not always observable and are frequently changing. The advantages of encoding hand pose in this way are exceptional speed from direct evaluation of a quadratic function and built-in smoothness from the quadratic function construction.

Let our input be vector $u = (p_1, \cdots, p_5, s_1, \cdots, s_5) \in \Re^{15}$, where $p_i = (x_i, y_i) \in \Re^2$ denotes the position of the i-th fingertip on the multi-touch device, s_i denotes contact status whose value is 1 if the fingertip is in contact and 0 otherwise, and the subscripts represent the indices of the fingertips.

Then the n-th degree-of-freedom of the hand can be expressed as a quadratic function of u as follows:

$$q_n(u) = \sum_{i \leq j} a_{ijn} u_i u_j + \sum_i b_{in} u_i + d_n \tag{1}$$

where $a_{ijn} (i \leq j)$, b_{in} and d_n are the coefficients that must be determined. A total of N quadratic functions are needed to reconstruct the full configuration of an N-degree-of-freedom hand model. Because u vector is 15-dimensional, we solve for $120 + 15 + 1 = 136$ coefficients per degree-of-freedom.

The encoding function can alternatively be any desired degree, with correspondingly fewer or more coefficients. For example, equation 2 shows linear encoding and equation 3 gives a cubic encoding. We compare quadratic encoding against these two alternatives in our results section.

$$q_n(u) = \sum_i a_{in} u_i + d_n \tag{2}$$

$$q_n(u) = \sum_{i \leq j \leq k} a_{ijkn} u_i u_j u_k + \sum_{i \leq j} b_{ijn} u_i u_j + \sum_i c_{in} u_i + d_n \tag{3}$$

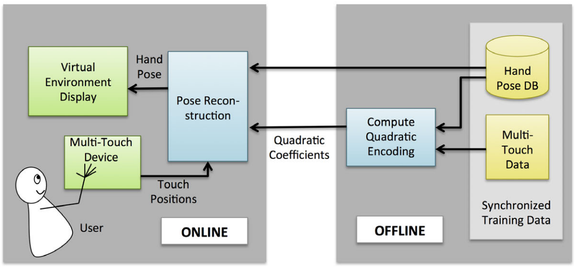

Figure 1: System flow. Quadratic encoding coefficients are computed offline from training data and used online for hand pose reconstruction from multi-touch input.

Our encoding technique is used in a complete system as shown in figure 1. Training is performed by computing quadratic encoding coefficients in an offline process. Given a training set of contact inputs u corresponding to hand postures $q_n(u)$, we solve equations 1,2, or 3 for coefficients a, b, c, and d using an off the shelf least squares solver (MATLAB's lsqcurvefit). It is worth noting that the same u vector may correspond to multiple hand poses in the training set due to ambiguity in lifted fingers' positions. The least squares fitting takes care of this problem by finding the best fit among these redundancies.

At runtime, we reconstruct hand pose in real time from a user's multi-touch inputs by evaluating equation 1 for each degree-of-freedom of the hand model using the coefficients a_{ijn}, b_{in}, and d_n that have been computed offline.

Reconstruction requires that the fingers be identified. For training, finger labelings are known. At runtime, our system automatically labels fingers at first contact with the device, and every time an unseen finger is introduced into contact. We use our hand pose reconstruction technique to choose the most likely finger labeling. Specifically, we reconstruct hand pose for all possible labelings, and choose the result that has the closest nearest neighbor in our training database.

In some situations (e.g., manipulation tasks), a physically based simulation of hand pose is desired. In these cases, we use the Bullet physics engine to track the reconstructed hand pose. In our implementation, we used out-of-the-box settings except for setting the maximum impulse values to $100N \cdot s$.

3. Principal Component Based Inverse Kinematics (PCA IK)

We compare our results to the PCA IK method of Mulatto et al. [MFMP13]. The number of principal components (PC's) that can be used to represent hand pose in this technique varies with the number of fingers in contact with the multi-touch device. For 5 fingers in contact, the 12 available PC's capture 97% of the variance in our training data, while 4 fingers (9 PC's) yield 94%, 3 fingers (6 PC's) yield 87%, and 2 fingers (3 PC's) yield 68%. This method relies on compliance values; we used a compliance of 0.1 for translation and

25.0 for rotation. We found 100 iterations to be sufficient for convergence and use 100 iterations in our experiments.

Unlike in the original work [MFMP13], fingertip labels are unknown for our application. Therefore, we perform an exhaustive search over fingertip labellings as in our quadratic encoding method. However, for PCA IK, we identify nearest neighbor based on fingertip contact positions, rather than reconstructing and comparing the complete hand pose. Comparing complete hand poses gave poorer results, perhaps because the PCA IK reconstruction was less faithful than that derived from quadratic encoding.

There are many options for improving the PCA-IK algorithm such as projection of the desired pose into null space. It would be interesting to compare against those in the future.

4. Results and Discussion

For our results, we trained all models using a set of hand motions captured by using a conventional marker-based motion capture system and a multi-touch device simultaneously. We used a 10 camera Vicon system to capture 3D hand motions, using 23 markers per hand. An iPad® 2 and TUIO [TUI] software was used to obtain the fingertip contact positions. The multi-touch device orientation was fixed and the subject was sitting directly in front of the device. The hand model used in our experiments has 14 ball joints and 6 degrees of freedom in the root joint for a total of 48 degrees of freedom, and was automatically generated by our Vicon system.

Our training set consists of 7,083 total frames of one subject making moving, grasping, pinching, and rotating motions with various combinations of fingers. 1-finger motions were not captured due to their inherent ambiguity on a multi-touch surface. This dataset was used to solve for the encoding functions as discussed in Section 2, with total time for offline processing of approximately 5 minutes. A separate database was captured on a different day from the same subject to be used in ground truth evaluation. The subject was allowed to perform free-form gestures on the multi-touch device as if he were interacting with a virtual clay blob. We removed 1 finger poses and brief glitches due to noise in the iPad® capture system. The resulting ground truth dataset contains a broad variety of motions and finger combinations in 2,538 total frames. For both training and testing data, we chose the fingertip position (x_i, y_i) to be given in inches from the top left corner of the screen. We found the exact value of a no-contact ($s_i = 0$) fingertip position to make little difference. It is set as $(-1,-1)$ in our experiments. Translation invariance was enforced by measuring all fingertip locations relative to the leftmost finger contact point.

Table 1 gives results comparing all techniques on the ground truth dataset. Each row in the table is a different reconstruction technique. The "Distance" columns give the mean and standard error of mean for Euclidean distance between the wrist position as measured from the motion capture data and the wrist position as reconstructed by each algorithm. The "Angle" columns give the mean and standard error of mean of the joint angle differences between ground truth and reconstruction, averaged over all joint angles. We see that quadratic encoding performs better than all other approaches. In particular, we suspected that cubic encoding may be suffering from overfitting, and performed the Copas test over 10-fold cross validation of training data to measure overfitness. We found that cubic encoding has an average of 18.6 joints which are overfitted to the training data as opposed to 7.3 joints for linear and 9.8 joints for quadratic.

	Distance (cm)		Angle (degrees)	
	Mean	SEM	Mean	SEM
Linear	3.9837	0.0510	9.0774	0.0362
Quadratic	2.5520	0.0479	8.5656	0.0368
Cubic	3.1652	0.0415	9.9411	0.0424
PCA IK	4.1488	0.0440	11.7594	0.0455

Table 1: Wrist position errors and overall joint angle errors in different methods. SEM stands for standard error of mean.

	Accuracy (%)			
	2 Fingers	3 Fingers	4 Fingers	5 Fingers
Linear	52.66	34.04	64.35	66.87
Quadratic	55.66	70.72	79.36	99.88
Cubic	56.24	52.38	86.49	99.33
PCA IK	51.85	51.50	48.15	62.88

Table 2: Finger labeling accuracy in different methods per number of fingers.

Table 2 shows the ability of the various techniques to compute a correct finger labeling. Again, quadratic encoding has the best overall performance. We believe cubic encoding was able to outperform in 4-finger labeling accuracy, because there was less overfitting due to less training data (874 frames for 4-fingers vs. 2020 frames for 3-fingers).

We also examined smoothness of the motion, finding that quadratic encoding is substantially smoother than the other encoding methods. Cubic encoding is also typically smooth but occasionally shows large divergences, possibly due to overfitting. Figure 2 shows some side by side comparisons of the various approaches, indicating some failure modes observed. PCA IK (figure 2(e)) tries to fit the target position excessively and produces an unnatural pose. Across the encoding methods (figure 2(b), 2(c), and 2(d)), we see that quadratic encoding (figure 2(c)) generalizes best over the training data and handles unseen data well.

We tested our real-time system for ability to allow users to manipulate objects in a virtual environment. Tasks included moving a virtual box to a goal position, rotating and sliding a virtual wrench, and moving and stacking boxes (figures 3). All of these tasks were successfully performed without user training. We informally tested with users of different hand

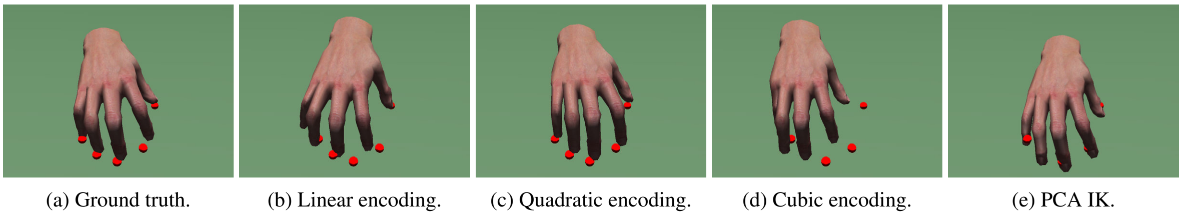
(a) Ground truth. (b) Linear encoding. (c) Quadratic encoding. (d) Cubic encoding. (e) PCA IK.

Figure 2: Reconstruction results across different methods from an unseen input.

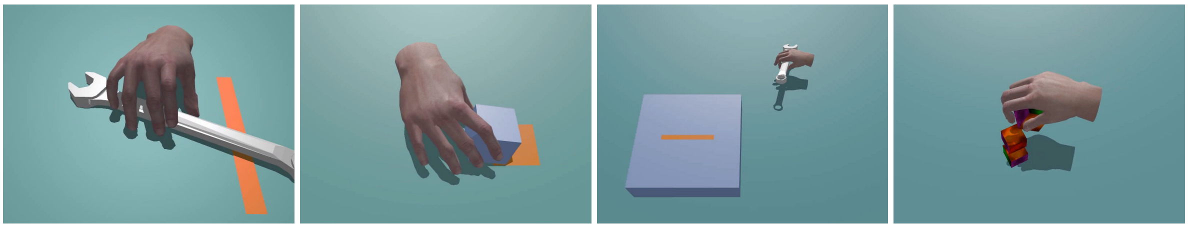
Figure 3: Manipulation tasks.

sizes and found that users with hand sizes similar to that of the training data can use the system effectively.

There are several ways our system can be improved. We could enforce that fingertips make contact at the measured contact points using simple corrective IK. We could also use historical information to make more informed estimates of finger labeling. Such estimates may enable us to extend our system to handle situations with a single finger in contact.

However, we find that current system performs extremely well in the real-time, interactive situations for which it was designed. Users can view smooth hand reconstructions and use their hands in a virtual environment to manipulate objects in real-time with the tactile support of the multi-touch interface. Even untrained users quickly compensate for errors in finger labeling or placement. In contrast to iterative IK approaches (66.1898 ms for 100 iterations), quadratic encoding (0.0021 ms per pose) is so fast that multiple possibilities for finger labels can be explored without causing a visible lag. In contrast to other encoding techniques, results represent ground truth well and are smooth over varying poses and varying finger contacts. In the future, we will explore using our system with 3D devices and immersive environments and explore bimanual and multi-user interactions. We will also explore applications of quadratic encoding beyond hand motion, e.g., to motions of the entire body.

Acknowledgements

We would like to thank Justin Macey for his help in motion capture, Moshe Mahler for his help in hand model, Yuzuko Nakamura for her help in drawing system overview diagram, and Daniel Troniak for his help in video editing. This research was supported by NSF awards IIS-1218182 and IIS-1344361.

References

[CPMX07] CHANG L. Y., POLLARD N. S., MITCHELL T. M., XING E. P.: Feature selection for grasp recognition from optical markers. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (2007), IEEE, pp. 2944–2950. 1

[ES03] ELKOURA G., SINGH K.: Handrix: animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 110–119. 1

[HGUVG11] HAMER H., GALL J., URTASUN R., VAN GOOL L.: Data-driven animation of hand-object interactions. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on* (2011), IEEE, pp. 360–367. 1

[HRMO12] HOYET L., RYALL K., MCDONNELL R., O'SULLIVAN C.: Sleight of hand: perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2012), ACM, pp. 79–86. 1

[KPA13] KIM J., POLLARD N. S., ATKESON C. G.: Quadratic encoding of optimized humanoid walking. In *IEEE/RAS International Conference on Humanoid Robot (ICHR)* (2013). 2

[Lea] Leap Motion. https://www.leapmotion.com. Accessed: 2014-01-21. 1

[MFMP13] MULATTO S., FORMAGLIO A., MALVEZZI M., PRATTICHIZZO D.: Using postural synergies to animate a low-dimensional hand avatar in haptic simulation. *IEEE Trans. Haptics 6*, 1 (Jan. 2013), 106–116. URL: http://dx.doi.org/10.1109/TOH.2012.13, doi:10.1109/TOH.2012.13. 1, 2, 3

[Nim] Nimble VR. http://www.nimblevr.com. Accessed: 2014-11-30. 1

[QSW*14] QIAN C., SUN X., WEI Y., TANG X., SUN J.: Real-time and robust hand tracking from depth. 1

[TUI] TUIO. http://tuio.org. Accessed: 2014-01-21. 3

[YL12] YE Y., LIU C. K.: Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 41. 1