

Hybrid Motion Graphs for Character Animation

Kalouache Saida
LESIA Laboratory
Biskra University
Biskra, Algeria

Cherif Foudil
LESIA Laboratory
Biskra University
Biskra, Algeria

Abstract—Many works in the literature have improved the performance of motion graphs for synthesis the humanlike results in limited domains that necessity few constraints like dance, navigation in small game like environments or in games by the gesture of feedback on a snowboard tutorial. The humanlike cannot exist in an environment without interacting with the world surrounding them; the naturalness of the entire motion extremely depends on the animation of the walking character, the chosen path and the interaction motions. Addressing exact position of end-effectors is the main disadvantage of motion graphs which cause less importance expended to the search for motions with no collision in complex environments or manipulating motions. This fact motivates this approach which is the proposition of an hybrid motion graphs taking advantages of motion graphs to synthesis a natural locomotion and overcoming their limitations in synthesis manipulation motions by combined it with an inverse kinematic method for synthesis the upper-body motions.

Keywords—*motion graphs; inverse kinematic; virtual human; animation*

I. INTRODUCTION

Virtual humans have a wide range of applications with the developments of interactive entertainments such as games, learning and training applications or film animations. Responding to the needs of these applications, believable synthetic characters must populate the virtual environments. Character interactions and moves must be realistic to suit the visual richness of the virtual environments. To meet the interactivity demands, character must efficiently be animated and controllable by the simulation.

In literature model-based and data-driven are two semiautomatic techniques classes developed for creating human animations.

Simulation, search, and optimization are applied by model-based approaches to synthesis character motion by limiting the space of possible motion using biomechanical, kinematic, dynamic models, the representation of motion in these approaches is flexible, but may suffer from the difficulty of construction and control [11]. It was well used addressing human-like characters manipulating objects, but could fail to produce a motions with natural looking if the models do not appropriately constrain the motion particularly for motions with low energy, such as locomotion.

Data-driven approaches mainly refer to the reuse of existing 3D motion data to generate new motions. Which do not present a perfect solution either because when the motion is highly dynamic or involves constrained interactions with the

environment or other characters, the adaption of existing captured motions to new situations will mostly be difficult. To address these issues, several researchers have been proposed, the most popular approach is motion graphs where locomotion synthesis has been an active area of study. The search techniques based on motion graphs provide lowest distortion of the captured motions, and naturally allow the exploration of solutions in complex environments.

The main contribution of this paper is the proposition of a hybrid motion graphs based on the idea of using motion graphs to synthesis the locomotion motions to reach different locations in the environment and to achieve the reaching range of objects mainly needed for the forward and backward stages of the inverse kinematic method used to control the trajectories of the end-effectors in the generation of the upper body motions. Our hybrid approach proves to be very efficient for producing object manipulation in different locations in the environment with user control by allowing a coordination of hyper body actions with locomotion, and could be suitable for interactive application. Avoiding problems related to the explosion of the motion graph size in case of including both locomotion and upper-body actions which cause absolutely an increase of the search time and a complication of the search method which may fail to find the right motion in the right time, or problems related to only base on inverse kinematic methods that may fail in the generation of natural looking for low energy motions such as locomotion, the proposed hybrid motion graphs present a good solution for decrease search time also address a wide range of virtual character motions and could be useful to many applications in humanoid robotics, ergonomics, training, entertainment.

The paper is organized as follow:

In the first section, a review of works of the literature related to character locomotion, a discussion about data-driven methods, their advantages and drawbacks. Also works related to the inverse kinematic methods for the hyper body character animation. In the second section an overview on the proposed method that allows the synthesis of a natural locomotion and hyper body motions with a description of each step. In the third section a discussion of the obtained results of the proposed method.

II. RELATED WORK

A. Locomotion motions

Data-driven motion synthesis generates new motion from a library of motion clips. A lot of approaches have been proposed like blending approaches that create new motion by

blending multiple motions depending on blending weights, for synthesizing new motions that parameterize high level interesting features, it has been used to synthesize locomotion animation to lead a virtual character [9,24,26,27].

This approach have trouble to deal with the planning around obstacles, and problems in the generation of motions able to cover a sufficiently large solution space.

Motion capture data have been integrated with Planning methods in many way, Lau and Kuffner [14] use motion planning with pre-computed search trees to synthesis a long motion sequences even in dynamic environments containing many characters.

Motion clips are abstracted as high-level behaviours and then associated with a behaviour finite-state machine, motions are generated automatically by a planning algorithm that makes a global search of the state machine and computes a sequence of behaviours for the character that meet user requirements [15]. The motion database is annotated. Labels are used to describe the desired action, which is then built using dynamic programming [21].

Extracting motions from a motion graph structure is a popular approach for producing realistic full-body motions. Automatically processing motion capture data is the objective of motion graphs to facilitate the parameterization and interpolation of motions, and allow the motion planners to operate in the created parametric spaces. The motion graph structure is based on the idea of connecting similar frames in a database of motion capture clips. Once a motion graph is available, a graph search is performed to extract motions with desired properties [12,13,20,22].

The following work based motion graphs concentrated on achieving good connectivity in motion graphs by using interpolated motion clips with further similar poses than the original motion data to construct a well-connected motion graph [17], minimizing the size of the graphs by proposing an approach to select a good motion set by searching from a large motion graph that represent the motion capture data for a minimum size subgraph and using an Iterative Sub-graph Algorithm, to find a good approximation to the optimal solution [18], optimizing the search operations in motion graphs [28,29]. Mahmudi and Kallmann [20] propose feature-based motion graphs for the synthesis of locomotion motions among obstacles where transitions is chosen based on relevant features. They use a channel search method in motion graph search to avoid obstacles. Planning locomotion using motion graphs have received a lot of attention, in contrast to the manipulation of objects around obstacles because using motion graphs it is in general very hard to generate motion that needs tight coupling to the environment, like pointing to an object in the world, or walking up a stairs, unless exactly those motions are in the database. As more motion constraints are added, fewer paths will become available [7].

Yazhou [8] proposes a locomotion planner based motion graphs coupling with an upper-body actions planner that use a blending method to produce realistic motions of locomotion and interaction with the environment. The locomotion planner does a lot of iterations that required the execution of the upper

body motion each test to reach the target and in some cases it fails.

The interaction of the virtual human with the environment involves posing the character body to respect new constraints, motion graphs have the inconvenient to not precise the end effectors positions; methods of research sometimes can fail to generate the desired motion sequence because the motion corpus itself does not contain motion clips required.

B. Manipulation motions

The Animation of virtual character to realise many task and object interactions have been treated by many approaches, motion blending was used to generate reaching and grasping motions that satisfy spatial constraints [3,8].

Physics-based and procedural animation offers a great amount of control over motion by employing a mathematical model in the generation of the desired character motion depending on high-level parameters. Physically-based techniques use physic laws such as Newton's laws and conservation of energy. Determining the human motion using this techniques, require information more than joint angles and position, it need Information such as mass distribution for each body part and torques generated by each joint which is not easy specially for the synthesis of hyper body motions which involve object manipulation like reaching, grasp and displacing objects, often the Inverse Kinematics (IK) and collision avoidance are used to solve this type of problem, the sampling-based planners' methods are required when several obstacles are on the way.

Methods based on Inverse Kinematics (IK) mainly control the trajectories of the end-effectors, and can integrate some reactive collision avoidance behaviour in order to cope with obstacles. Inverse Kinematics is a method used to solve the problem of computing joint rotations in order to reach given targets with the end-effectors [19]. The analytical methods are the best choice to handle with simple kinematic chain that contains a few degrees of freedom [5]. These methods are fast but not support the complex articulated chain. Numerical methods based on Jacobian matrix are proposed to deal with complex articulated chain but suffer from the computational time, the heuristic method are proposed to solve this inconvenient [4,16,23]. Searching for a sequence of suitable hand locations in the workspace is one approach for reaching and arm planning. To reach each intermediate hand location, the IK method is used to lead the arm posture. Through interpolation of the postures, the final valid motion is obtained [11,25,30]. FABRIK (Forward And Backward Reaching Inverse Kinematics) [1,2] produces visually realistic poses with low computational cost, it bases on geometric algebra to resolve the inverse kinematic problem and support multiple chains with multiple end effectors because Constraints can easily be incorporated within FABRIK.

The virtual character is required to interact with different objects in different locations in the surrounding environment which involve posing the character body to respect new constraints, often in combination with environment navigation tasks. Motion graphs have the inconvenient to not precise the end effectors positions, this fact motivates this approach to

synthesis reactive motions in real-time, based on the use of the motion graphs in combination with an inverse kinematic method.

III. HYBRID MOTION GRAPHS

The proposed approach presents a solution to avoid many problems related to the explosion of motion graph size in case of including both locomotion and upper-body actions and preserve produced realistic motions. We propose a hybrid motion graphs taking advantage of motion graphs to synthesis a natural locomotion and overcoming their limitations to synthesis manipulation motions by combined it with an inverse kinematic method for synthesis the upper-body motion.

A. Overview of the proposed method

An overview of the proposed method is illustrated in the Fig. 1

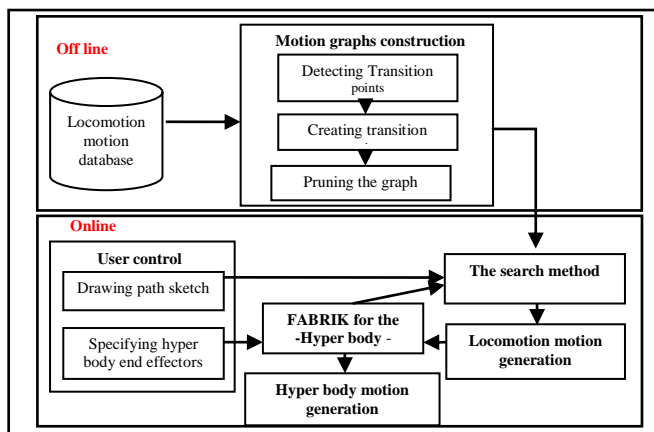


Fig. 1. System Overview

The proposed method has two phases:

1) Offline phase:

Allows the construction of the motion graphs according to Kovar and Gleicher [13] steps:

a) *Finding Candidate Transition Points:* Distance metrics are used to detect the similar frames for choosing the transition points, taking into account the relative importance of joints, changes in joint-velocity and acceleration and identifying compatible coordinate systems.

To calculate the distance $D(A_j, B_k)$ between two frames A_j and B_k , a point clouds is made over two windows of frames with length l defined by the user, every point cloud is composed of smaller point clouds that represent every frame position in the window.

The distance between A_j and B_k is calculated as follow:

$$\min_{\theta, x_0, z_0} \sum_i w_i \| p_i - T_{\theta, x_0, z_0} p_j \|^2 \quad (1)$$

Where p_i and p_j are the corresponding points in the two point clouds, a linear transformation T_{θ, x_0, z_0} formed of a rotation by θ degrees about the vertical axis and followed by a translation of (x_0, z_0) in the floor plane is applied to the second point cloud. Fig. 2 illustrates a point clouds made over two windows of frames.

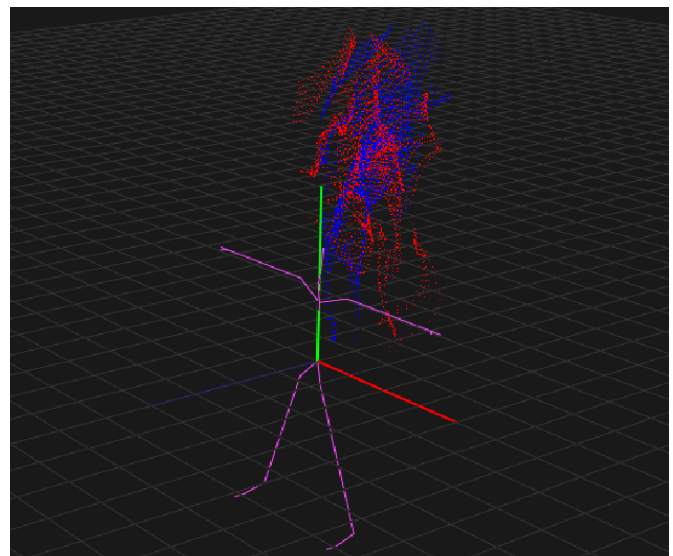


Fig. 2. Point clouds made over two windows of frames

The weights w_i may be taken to assign different importance to each joint in each frame and can be chosen by the user. Fig. 3 illustrates list of weights for each bones of an ASF/AMC skeleton [6].

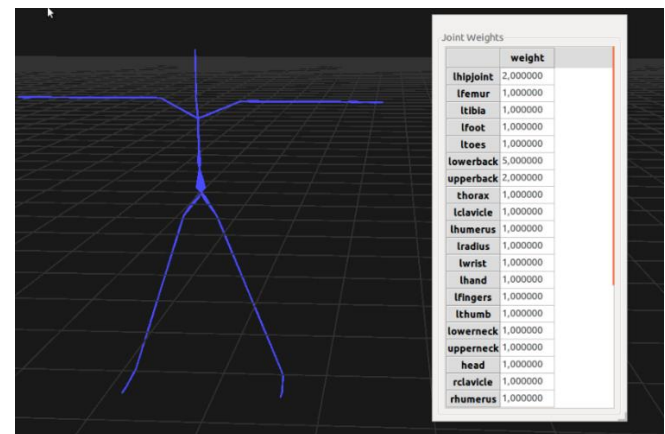


Fig. 3. ASF skeleton with list joints and weights

The distance metrics discussed above is computed for each pair of frames in the database to form a sampled 2D error function.

b) *Selecting transition points:* The distance that is found in this manner can be thought of to be a sampled 2D error or distance function. This function is normalized, so that, all the distances lie in the range $[0,1]$. We take this function and select only the local minima of this function, which gives us the candidate transition points. To guarantee that those local minima of this error function are actually good enough for transitions, a threshold must be defined by the user and the $D(A_j, B_k)$ must be smaller than it. The value of the threshold effect on the quality of the generated transitions, and the connectivity of the graph.

c) *Creating transition clips:* We have said that transitions are clips designed to connect two pieces of original

data, it is created by identifying the sufficiently similar portions of the initial clips that straight forward blending will produce a valid transitions. If the distance between frames A_j and B_k is less than the threshold and it is selected as a transition point, then we need to create the transition frames and add them as edges in the motion graph, by blending between the frames of the two windows of frames for A_j and B_k starting by applying the appropriate aligning 2D transformation to motion B. The frames A_j to A_{j+l-1} and B_{k-l+1} to B_k are interpolated as follows:

$$R_p = \alpha(p) R_{A_{j+p}} + [1 - \alpha(p)] R_{B_{k-l+1+p}} \quad (2)$$

Where R_p means the position of the root joint in frame p of the transition. The interpolation between the rotations is performed by:

$$q_p^i = \text{slerp}(q_{A_{j+p}}, q_{B_{k-l+1+p}}, \alpha(p)) \quad (3)$$

The following conditions must be fulfilled by the weight function of the interpolation

$$\alpha(p) = 1 \text{ for } p \leq -1, \alpha(p) = 0 \text{ for } p \geq k \quad (4)$$

To ensure the continuity:

$$\alpha(p) = 2 \left(\frac{p+1}{k}\right)^3 - 3\left(\frac{p+1}{k}\right)^2 + 1, -1 < p < k \quad (5)$$

To maintain the constraints of the original motion clips and ensure that the interpolation will not hurt it, the constraint will be activated in the first half of the new transition frames and passive in the second half in case it was active in the first clip but not in the second.

d) *Pruning the Graph*: In the resulted graph there may be portions with very high connectivity, and there may be portions with very low connectivity that may contain dead end nodes, which are nodes that are not part of any cycle in the graph, also contain other nodes that could be a part of one or more cycles which called sinks. These nodes may halt the motion search. To eliminate these problematic nodes we prune the graph by calculating the largest strongly connected component of the graph.

2) Online phase

The online phase allows the interactive generation of motions. This approach is similar to previous techniques that use sketching; the user draw the path sketched which will be followed by the character in the locomotion motions.

User chooses the end effectors position for the hyper body limbs before, after or during the locomotion motion which help the character to avoid obstacle (like moving head to avoid a bar) and reach objects.

The search method use a depth first search to obtain graph walks. Branch-and-bound strategy and incremental search are used to improve the efficiency [13].

The user supplies a scalar function $g(w;e)$ to evaluates the additional error accrued by appending an edge e to the existing path w The total error $f(w)$ of a graph walk is:

$$f(w) = f([e_1, \dots, e_n]) = \sum_{i=1}^n g([e_1, \dots, e_{i-1}], e_i) \quad (6)$$

Where w is comprised of the edges e_1, \dots, e_n . We assumed that $g(w,e)$ is never negative Therefore, the total error cannot decrease by adding an edge to a graph walk.

f is evaluated for all graph walks using a depth-search, then the best graph walks is selected.

We use branch and bound algorithm in which we prune any graph incapable of producing an optimal graph walk.

We apply path synthesis that is generating a motion stream that follows a user specified path P . The idea is to estimate the actual path P' travelled by the character through a graph walk and measure how different it is from P [13].

To compare the original paths P and the generated path P' , at each frame, the root joint of the character is projected on the floor giving a linear curve. Then calculate

$$g(w,e) = \sum_{i=1}^n \|P'(s(e_i)) - P(s(e_i))\|^2 \quad (7)$$

$P(s)$ is the point on P with arc-length distance from the start of P is s . The completeness condition of P' is if P' is sufficient long or surpasses the length of P .

Fig. 4. Illustrate how to extract the motion that travel with the path specified by the user

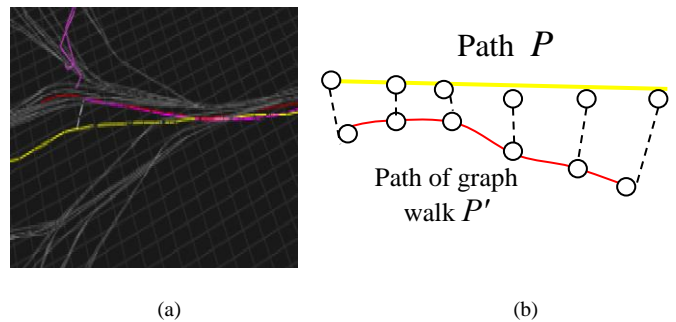


Fig. 4. (a), (b) Extract motion that travels along a path

For the hyper body IK method, the virtual character is modeled as a chain composed of rigid links attached at their ends by rotating joints, the chains can be formalized as follows: All joints with no children are labeled as end-effectors. FABRIK uses points and lines to solve the IK, it divide the problem into two relevant phases, a forward step and a backward steps, and support all the rotational joints limits and joints orientations by repositioning and re-orienting the target at each step.

The structure of the human skeleton is divided into smaller kinetic chains which are then sequentially adapted into the body posture in a hierarchical order [1,2]. Fig. 5 shows the hierarchical structure of the hyper body used in the approach.

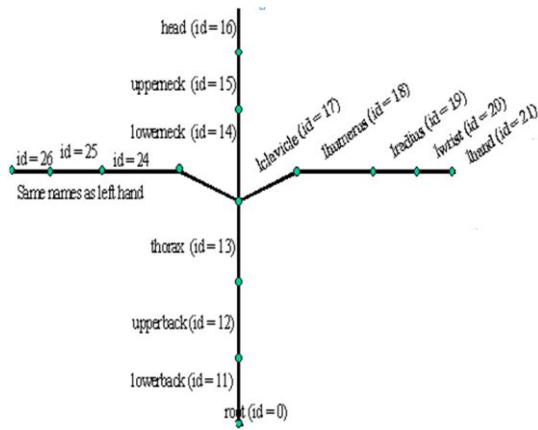


Fig. 5. Bones of the skeleton applied within the IK method[6]

The FABRIK method may be applied starting from the right hand (end effectors) until the root (pelvis), from the left hand (end effector) until the root, from the root until the head

When synthesizing reaching motion using the standard FABRIK, the focus tends to be on arms control, if the target is not reachable because the distance between the target and the first joint (root) is higher than the summation of the length of each links, the method do not provide a solution. However stepping motion can be a solution because humans often step in the direction of the target in order to help their reaching action. In doing so, the target falls within the convenient reaching range, in this approach this solution is provided by allowing the search method to place the character near the target location and ensure that the target is reachable by identifying a reaching range H where

$$H = D - \sum_{i=0}^{i=n} d_i \quad (8)$$

D is the distance between the target and the first joint (root), d_i is the joints distances of the chain.

To follows the user specified sketch presented by a 2D path planner on the floor plan, the branching factor of the search must be limited to only allow expansions nearby this path, else when the users choose to manipulate an object which is far from reaching, an extensions is integrated to customize the locomotion toward the object location.

The extension is a new root position that insures that the target is in the reaching range. Whenever the locomotion search expansion generates a character position that is close enough to the target the forward and backward stages are applied. An example of the unrolling of the motion graph with the expansions obtained is illustrated in Fig. 6.

To show the progress of the IK method an example of eight link manipulator is used Fig. 7, which is similar to the virtual human hands; the same principal is applied for the head but for fewer links (joints). The initial pose of end effectors and target position and each point is expressed by the coordinate of Cartesian space.



Fig. 6. Unrolling of the motion graph with the expansions obtained, white line expansions used to branch toward the reaching range (red circle)

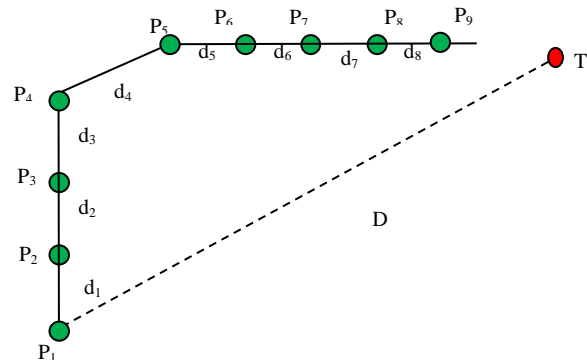


Fig. 7. example of eight link manipulator

The Fabrik [1] algorithm is extended as follow:

Input :

The target position T

The joints positions P_1, P_2, \dots, P_9

Distances between each joint d_1, d_2, \dots, d_8

Output : The generated joint positions for P_1, P_2, \dots, P_9

- 1- **Distance test** Check that the distance between the target T and the first joint P_1 is within the summation of length of each link.

$$D = P_1 - T$$

If $D \leq (d_1 + d_2 + \dots + d_8)$ target is reachable

2- run *Forward* stage

- (a) Move the hand P_9 end effectors to the target position, and the wrist point P_8 .

P_8 was moved on the line from P_9 to create P'_9 and P'_8 .

The distance between P'_9 and P'_8 is the same as d_8

- (b) Continue step (a) until new P'_1 is reached

3- Run *Backward* stage

- (c) Move P'_1 to its initial position P_1 .

- (d) Repeat procedure as (c), and P''_9 moves closer to the target than P_9 .

Run the *Forward* stage until the end-effectors is satisfactorily close to the target.

If $D \geq (d_1 + d_2 + \dots + d_8)$ target is not reachable

- 1- Calculate the new position P^*_1 of P_1

- (e) Calculate the threshold reaching range

$$H = D - (d_1 + d_2 + \dots + d_8)$$

- (f) Branch to H
- 2- run *Forward* stage
- 3- Run *Backward* stage

End

IV. RESULTS

In this section we investigate the results of the performed experiments relative to the automatic construction of motion graphs based on connecting between different motion clips in the database and then searching in the resulted graph for motions that satisfy user constraints using the procedures discusses above and the interactive generation of realistic controllable locomotion motions for character locomotion, other results demonstrate the hyper-body motions adjustment to reach different location using the inverse kinematic method.

For the locomotion motion the graph is automatically generated from an input set of motion clips like running clips, sharp turns used to animate a human-like character in a virtual environment. In experimentation the time to build the motion graph was 40 seconds for an example of a motion graph formed over 8 motion clips. Fig. 8 illustrates the motion clips list, the weight of each joints and the generated point cloud of the distance metric.

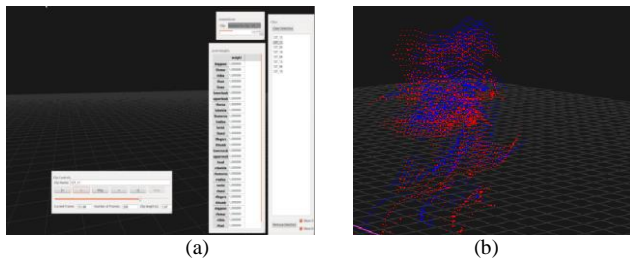


Fig. 8. (a) Motion clips list and joint weights (b)Point cloud distance metric

For the path synthesis the user first draw a sketch using the mouse, this sketch will be followed in the phase of generation of the new motions Fig. 9

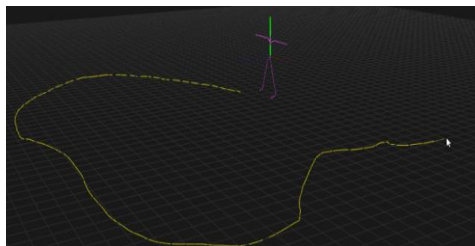


Fig. 9. Example of a user sketch

Fig. 10, illustrates the path synthesis and the generation of the new motion following the yellow line represents the user sketched path and the white lines represent the edges in the graph walk and red line represents the actual path followed by the character. The hyper-body pose are adjusted to new constraints using the inverse kinematic method. Fig. 11 and Fig. 12 illustrate right and left hand adjustment using the IK method.

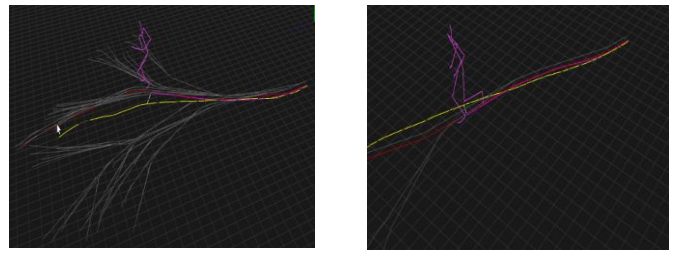


Fig. 10. Example of a motion extracted from a motion graphs , (a) from 8 motion clips (b)Another example with less motion

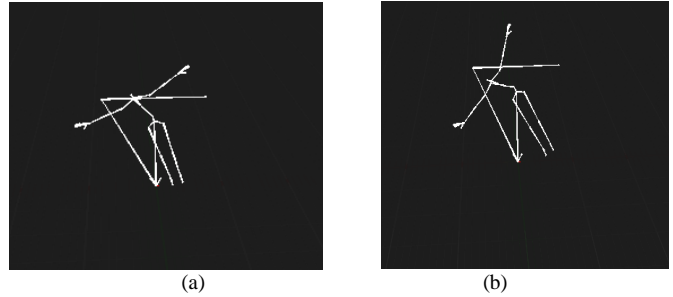


Fig. 11. (a),(b) Right hand adjustment using the IK method

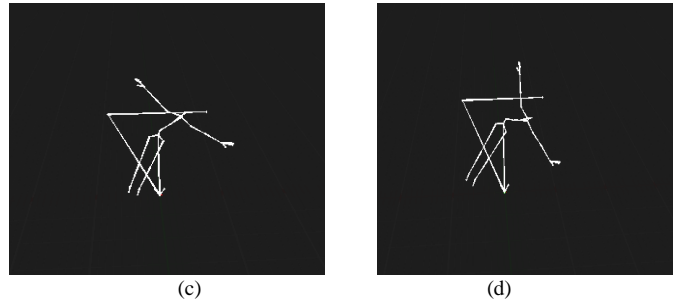


Fig. 12. (c),(d) left hand adjustment using the IK method

Hybrid motion graphs is a framework for generating realistic, controllable whole-body motion through the combination between two techniques, motion graphs and Inverse kinematic method to resolve the problems related to including both locomotion and upper-body actions in the same graph from the explosion of the size of the graph, the increase of the search time and the complication of the search method, also problems related to the fail in the generation of natural motions in case of using just inverse kinematic methods.

V. CONCLUSION

In this paper we have proposed a hybrid motion graphs for generating realistic whole body motion taking advantages of motion graphs to synthesis a natural locomotion and overcoming their limitations to synthesis manipulation motions by combined it with an inverse kinematic method for synthesis the upper-body motion.

ACKNOWLEDGMENT

A special thanks to lucas panian for his help with the motion graphs implementation.

REFERENCES

- [1] A. Aristidou, and J. Lasenby, "FABRIK: a fast, iterative solver for the inverse kinematics problem," *Graphical Models*, vol.73, no.5, pp. 243-260, 2011.
- [2] A. Aristidou, Y. Chrysanthou, and J. Lasenby, "Extending FABRIK with model constraints," *Computer Animation and Virtual Worlds*, 2015.
- [3] A. W. Feng, Y. Xu, A. Shapiro, "An example-based motion synthesis technique for locomotion and object manipulation," *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2012.
- [4] C. Welman, "Inverse kinematics and geometric constraints for articulated figure manipulation," *Diss. Simon Fraser University*, 1993.
- [5] D. Tolani, G. Ambarish, and I. B. Norman, "Real-time inverse kinematics techniques for anthropomorphic limbs," *Graphical models* vol. 62, no. 5, pp. 353-388, 2000.
- [6] Graphics Lab, Carnegie-Mellon University, \Carnegie-Mellon Mocap Database." <http://mocap.cs.cmu.edu>, 2015.
- [7] H. V. Welbergen, B. J. H. van Basten, A. Egges, Zs. Ruttkay, and M. H. Overmars, "Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control," *Computer Graphics Forum*. Vol. 29. No. 8. Blackwell Publishing Ltd, 2010.
- [8] H. Yazhou, M. Mahmudi, and M. Kallmann, "Planning humanlike actions in blending spaces," *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference*. IEEE, 2011.
- [9] J. Pette, and J. P. Laumond, "A motion capture - based control - space approach for walking mannequins," *Computer Animation and Virtual Worlds*, vol. 17, no. 2, pp. 109-126. 2006.
- [10] K. Yamane, J. J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," *ACM Transactions on Graphics (Proceedings of SIGGRAPH'04)*, vol. 23, no. 3, pp. 532-539, 2004.
- [11] K. Yamane, Katsu, J. J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," *ACM Transactions on Graphics (TOG)*. Vol. 23. No. 3. ACM, 2004.
- [12] L. Kovar, and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, ACM 2004.
- [13] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," *ACM transactions on graphics (TOG)*, vol. 21, no. 3, ACM, 2002.
- [14] L. Manfred, and J. J. Kuffner, "Behavior planning for character animation," *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2005.
- [15] L. Manfred, and J. J. Kuffner, "Precomputed search trees: planning for interactive goal-driven animation," *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006.
- [16] L. T. Wang, and C. C. Chen, "A combined optimization method for solving the inverse kinematics problems of mechanical manipulators," *Robotics and Automation, IEEE Transactions*, vol. 7, no. 4, pp. 489-499, 1991.
- [17] L. Zhao, and A. Safonova, "Achieving good connectivity in motion graphs," *Graphical Models*, vol. 71, no. 4, pp. 139-152, 2009.
- [18] L. Zhaoy, A. Normoyle, S. Khanna, and A. Safonova, "Automatic construction of a minimum size motion graph," *Proceedings of the 2009 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2009.
- [19] M. Kallmann, "Autonomous object manipulation for virtual humans," *ACM SIGGRAPH 2008 classes*. ACM, 2008
- [20] M. Mahmudi, and M. Kallmann, "Analyzing locomotion synthesis with feature-based motion graphs," *Visualization and Computer Graphics*, vol. 19, no. 5, pp. 774-786, 2013.
- [21] O. Arikan, D. A. Forsyth, and J. F. O'Brien, "Motion synthesis from annotations," *ACM Transactions on Graphics(TOG)*, vol. 22, no. 3, pp. 402-408, 2003.
- [22] R. Heck, and M. Gleicher, "Parametric motion graphs," *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, ACM 2007.
- [23] R. Kulpa, and F. Multon, "Fast inverse kinematics and kinetics solver for human-like figures," *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots*. 2005.
- [24] R. S. Johansen, "Automated semi-procedural animation for character locomotion," *Thèse de doctorat*. Aarhus Universitet, Institut for Informations-og Medievidenskab, 2009.
- [25] S. Bandi and D. Thalmann, "Path finding for humanmotion in virtual environments," *Computational Geometry: Theory and Applications*, vol. 15, no. 1, pp. 103-127, 2000.
- [26] S. I. Park, H. J. Shin, and S. Y. Shin, "On-line locomotion generation based on motion blending," *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002.
- [27] T. Kwon, and S. Y. Shin, "Motion modeling for on-line locomotion synthesis," *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2005.
- [28] W. Lo - Yen, and Z. Matthias, "Bidirectional search for interactive motion synthesis," *Computer Graphics Forum*, vol. 29, no. 2, 2010.
- [29] Y. Gao, Z. CHEN, M. CHEN, and Y. SHEN, "An improved approach to the efficient construction of and search operations in motion graphs." *Science China Information Sciences*, vol. 55, no. 5, pp. 1042-1051, 2012.
- [30] Y. Liu, and N. I. Badler, "Real-time reach planning for animated characters using hardware acceleration," *In Proceedings of Computer Animation and Social Agents (CASA'03)*, pp. 86-93, 2003.