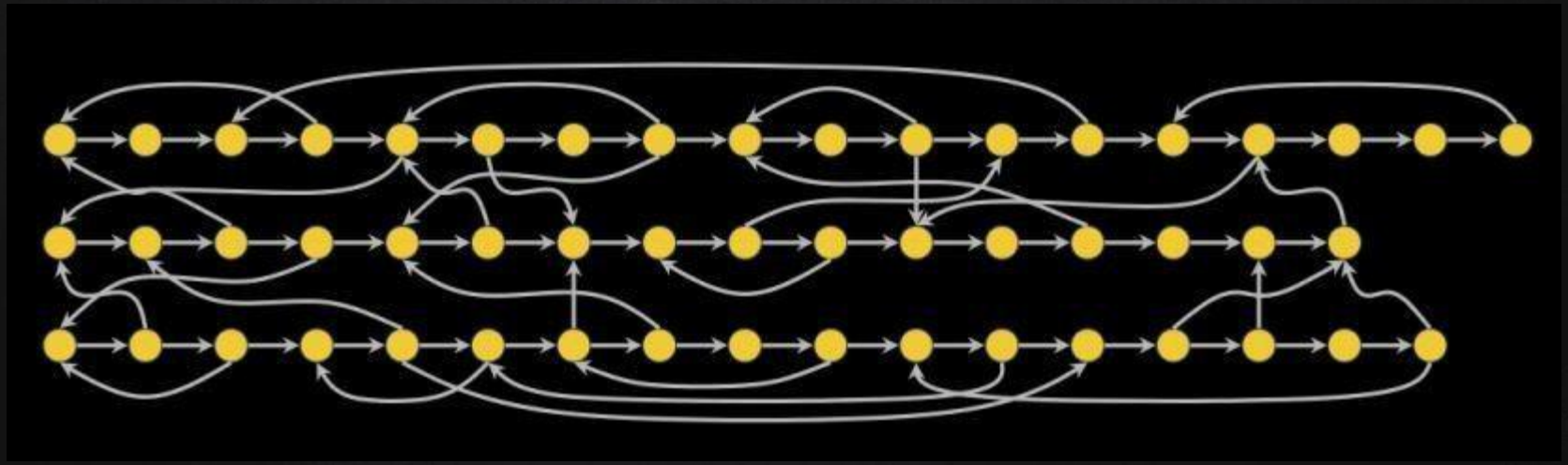
Motion Graph

Computer Animation Assignment 3

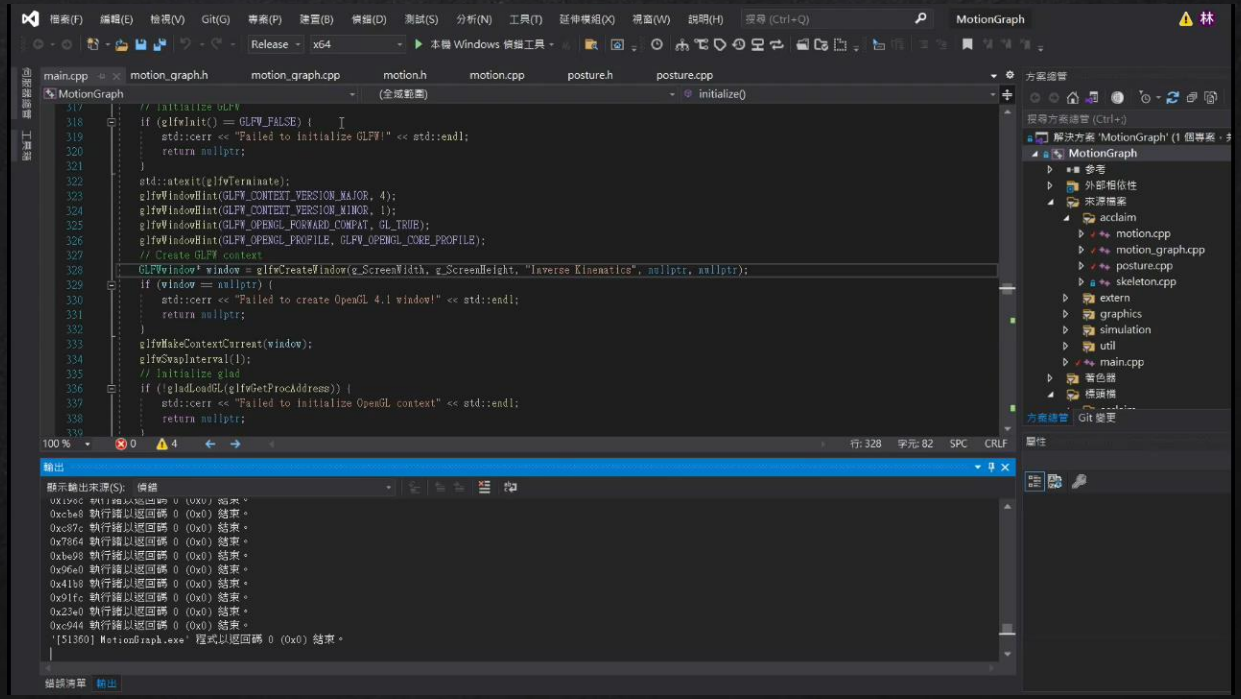
2024/05/06

Goal

- ✘ Build continuous & long sequence of motions based on unlabelled motion database



Demo


MotionGraph
Release
x64
main.cpp
motion_graph.h
motion_graph.cpp
motion.h
motion.cpp
posture.h
posture.cpp
initialize()
// Initialize GLFW
if (glfwInit() == GLFW_FALSE) {
std::cerr << "Failed to initialize GLFW!" << std::endl;
return nullptr;
}
std::atexit(glfwTerminate);
glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 4);
glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 1);
glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);
glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
// Create GLFW context
GLFWwindow* window = glfwCreateWindow(g_ScreenWidth, g_ScreenHeight, "Inverse Kinematics", nullptr, nullptr);
if (window == nullptr) {
std::cerr << "Failed to create OpenGL 4.1 window!" << std::endl;
return nullptr;
}
glfwMakeContextCurrent(window);
glfwSwapInterval(1);
// Initialize glad
if (!gladLoadGL(glfwGetProcAddress)) {
std::cerr << "Failed to initialize OpenGL context" << std::endl;
return nullptr;
}
輸出
'[51360] MotionGraph.exe' 程式以返回碼 0 (0x0) 結束。
MotionGraph
acclaim
motion.cpp
motion_graph.cpp
posture.cpp
skeleton.cpp
extern
graphics
simulation
util
main.cpp

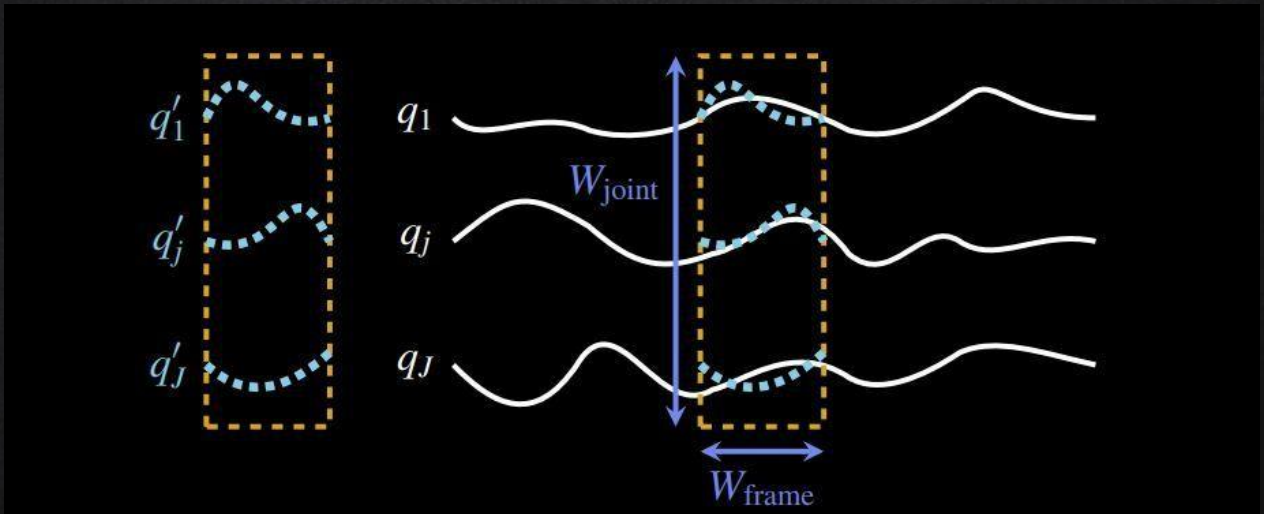
Components

- ✘ Motion matching
- ✘ Motion transformation and blending
- ✘ Motion graph
  - Reference Paper

Motion Matching

- A crucial part before Motion blending
  - Find pairs of motion segments (m1, m2) such that the tail of m1 matches with the head of m2
  - Define the distance between two motion segments by the sum of posture distances between the tail of m1 and the head of m2

Motion Matching (cont'd)



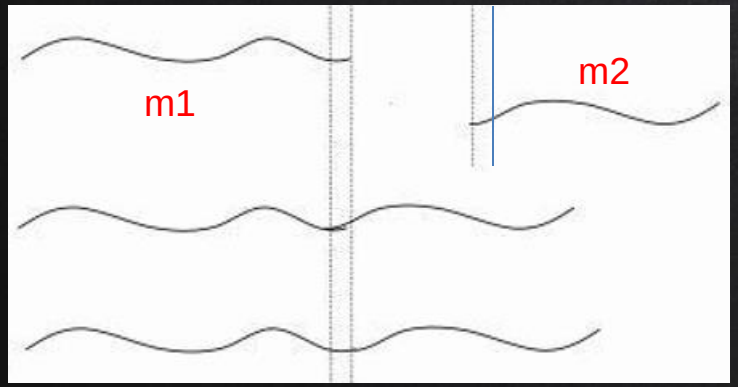
$$\text{dist} = \sum W_{\text{frame}}[f] \sum W_{\text{joint}}[f] \cdot \text{joint\_dist}\left(q_j[f] - q'_j[f]\right)$$

Motion Matching (cont'd)

- ✘ Posture difference can be defined differently
  - Use angle/velocity/position in different spaces
  - We provide sample code that uses weighted joint angle differences
  - You are free to try other methods

Motion Blending

- Connect two motions by blending the connecting motion clip



(a) Use motion matching to find suitable pairs of motion segments (m1, m2)

(b)Perform motion transformation on m2 (facing angle and root position)

(c) Perform ease in/ease out blending on overlapped frames (blending window)

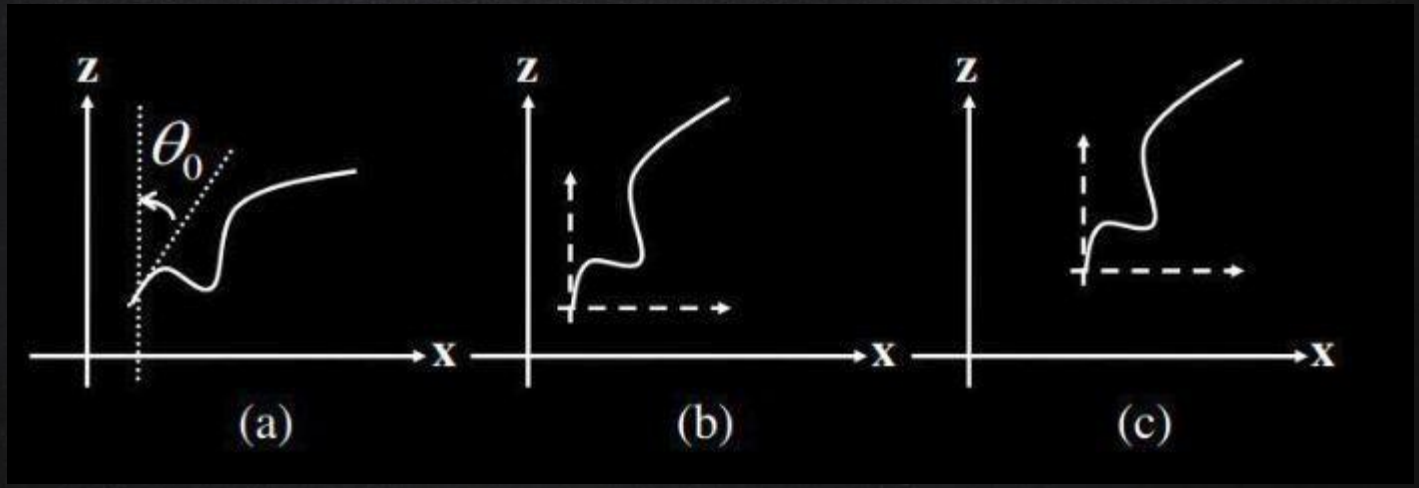
(d-2) Contatenate the blended segment to m1, then m2 to m1

Procedure - Motion Blending

1. Set the blending weight vector
   -Determine the temporal length of the blending part

2. Transform the facing angles & root positions of the 2nd segment so that its head overlaps with the tail of the 1st motion segment (the overlapping length is the blend window size)
   - (TODO) motion.cpp: Motion::transform()

Procedure - Motion Blending



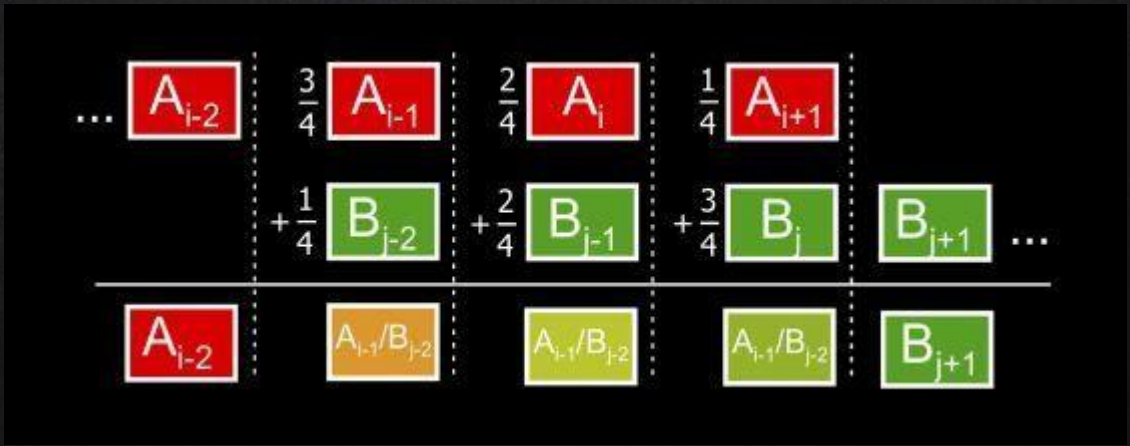
(a) Compute Facing Angle

(b) Rotate the postures about the root position of the 1st frame

(c) Translate to the new position

Rec Procedure - Motion Blending

3. Blend the overlapped frames by SLERP (spherical lerp)
   - (TODO) motion.cpp: Motion::blend()



Procedure - Motion Blending

For ease in/ease out blending, blending weight u can be set as…

$$W_b[f] = \frac{1}{2}\sin\left(\frac{f}{N_b - 1}\pi - \frac{\pi}{2}\right), f = 0, \ldots, N_b - 1$$

N_b : length of the blending window

Procedure - Motion Blending

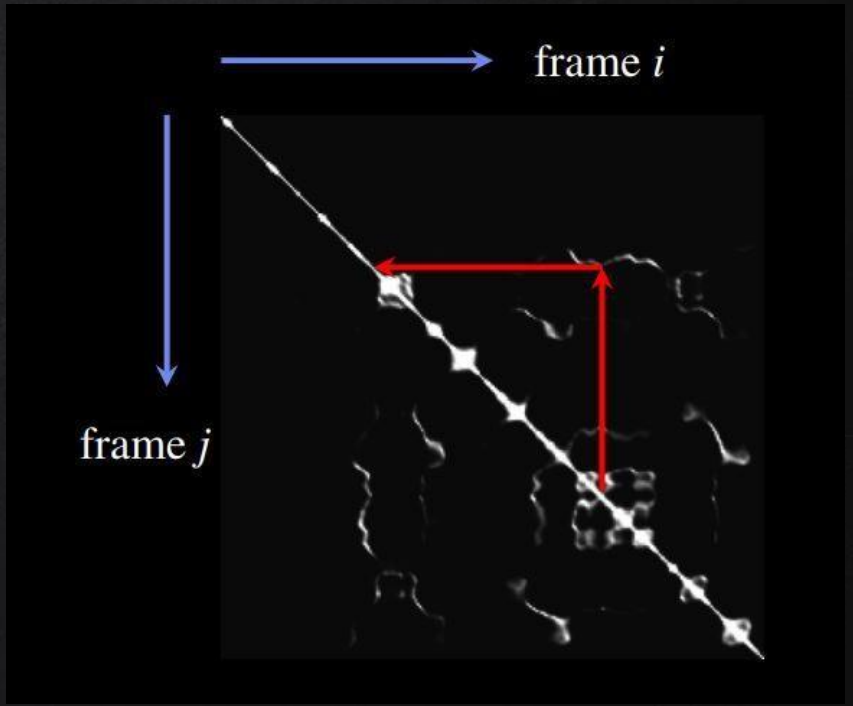
4. Concatenate these motion clips...

    1) 1st motion with ending clip trimmed
    2) Blended clip
    3) Transformed 2nd motion with starting clip trimmed

Procedure - Motion Graph

1. Cut large motion clips into small segments, then construct pose difference matrix between segments. Each segment will be seen as a node.

  - Finding good transitions (similar window of frames)
  - "Dense" map
  - This is before pruning



Procedure - Motion Graph

2. Draw edges between nodes according to difference matrix and define their weights (weight sum of all outgoing edges of a node should be 1). Each edge represent a suitable transition, and the weights represent the possibility of transition.
   - (TODO) motion_graph.cpp: MotionGraph::constructGraph()

3. Pruning bad edges(Remove bad candidates of transition)
   - Many techniques
   - “Sparse” map
   - (TODO) motion_graph.cpp: MotionGraph::constructGraph()

4. Blending at transitions

Handling .amc files and UI

- ✗ We will provide three default .amc files in the project and some others in a separate folder. You can add them to the assets/Acclaim folder to try them out.
- ✗ Longer motions are much preferred since segment length and blend window length has to be long enough for motion graph to work
- ✗ Fetching .amc files: lines 51~52, 136~141 in main.cpp
- ✗ When executing the project, you can adjust camera view with the “Camera panel” button in the UI.

Recommended Outline of Report

- ✘ Introduction
- ✘ Implementation
- ✘ Result and Discussion
- ✘ Conclusion

Required Content of Report

- ✘ Implementation of motion transformation and blending
- ✘ How you draw edges (transitions) and decide their weights for your motion graph
- ✘ Effects of parameters like blending window length and segment length (in number of frames)
- ✗ Names of the .amc files that you would like TAs to use while testing your code
- ✗ (Optional) Your modifications to the posture difference calculation or any part of the given code

Submission - Requirement

- ✘ Complete project files
  - The project should be able to build successfully
- ✘ .amc files that you want TAs to use while testing your code
- ✘ Report in pdf format, named HW3_report_<StudentID>.pdf

Submission

- ✘ Compress upmentioned materials in to a zip file
  - Named HW3_<StudentID>.zip

Grading

- ✗ (10%) In main.cpp, change the window name to <YOUR_STUDENT_NUMBER> (e.g. 110XXXXXX)

- ✘ (60%) Execution result, judged by:
  - Smoothness of transition (transform, blending)
  - Different sequence of segments between executions (Tas will run your code multiple times)

- ✗ (30%) Report, you will get at least 20 points if you meet the required contents mentioned above.

Policies

- Late policy
  - Penalty of 10 points per late day

- Cheating policy
  - 0 points for any cheating on assignments
  - Allowing another student to examine your code is also considered as cheating

Reminder

- ✘ Refer to the document “acclaim_FK_IKnote.pdf”, it has hints for implementation
- ✘ You can download new motion files and form your own motion capture database
  - Google “cmu motion capture database”

Reminder

- ✘ You can choose to report a paper instead of doing this project
  - Or... you can do both
  - Link to Paper presentation form can be found on e3, in the announcements section

Q & A