Abstract
For this two-minute 4K movie production, the continuous battle flow is simulated with a custom crowd animation system.

Background
"The War of Sekigahara" is a two-minute 4K movie production about the greatest battle in the history of Japan: the war of Sekigahara, where more than 100,000 samurais fought. The project uses live footage from a 4K digital camera composited with CGI. The final output of the project will also be in 4K format.

Problems
The battle sequences will have at least 10,000 CG simulated samurais. But one of the major issues in most of the productions in Japan is that the CGI team has no time to study 3rd party crowd simulation software, which usually requires full-time operators. So this is not a viable solution for most CGI teams as they cannot concentrate on researching and building the crowd animation system itself since they don't have programmers and researchers. This task is assigned to us, and our goal is to build smart and controllable tools for the artists.

The crowd system
By examining the scenario, the new crowd animation system should have the following features: moving towards the enemy along the height of the terrain, avoiding collisions with each other and fighting one-on-one with the enemy after finding him.

For the fighting animation (which is called "Event"), we use MOCAP data.

The event
While moving, a samurai locks on an enemy he wants to fight and proceeds to the event. The event itself has an IN-point and an OUT-point. The IN-point is the position and direction of the fighters when the event starts. The movement path of the samurais (during the battle) is then connected to the IN-point. The OUT-point of the event consists of the positions and directions of the fighters at the end of the battle. Because the event is basically MOCAP data, the IN and OUT points are fixed, so the agent movement path is adjusted to connect to the IN-point of the event in a smooth way.

Fix the path to connect
with the IN-point

IN
Samurai A

Samurai B

Original path

Event Area

Events may have different types of OUT-points: 0-OUT if the fighter is dead or injured (for one-man events, like "shot by an arrow" event), 1-OUT if one is dead and the other one proceeds further in the battle, or 2-OUT if both escape in some way.

Conclusion
So far, the system is in prototype-stage. It is made of 5 parts: layouting samurais, generating moving paths, generating blended motions, making proxies and rendering. The system components exchange data with the use of XML based files, with Maya being the "control room". Our efforts are directed towards Maya plug-ins and scripts development, because file based data exchange is inefficient. We are now working on gluing those parts together.

Acknowledgement
This project is supported by special coordination funds for promoting science and technology, MEXT Japan.

References

* {kaoru, kotamura, nobo, ark, mihai}@dhmsl.jp
** tomo@dhw.co.jp