Merging and Unmerging in Multi-Body Simulations

Otman Benchekroun, Eulalie Coevoet, Paul Kry

Abstract

Sleeping and waking bodies in rigid body simulations is a common technique in physics engines which reduces computation time by exploiting the property that only a subset of bodies are typically in motion at any given simulation scenario. Where these bodies sleep and wake in the inertial frame, it is also possible to have bodies sleep and wake in a body frame. We can simplify simulations by merging multiple bodies that appear to move together into a single entity. This allows us to bypass time-consuming collision detection/resolution step otherwise required between the bodies in a merged collection. We explore the construction of appropriate criteria that would allow seemless merging and unmerging operations to be performed while maintaining physically grounded motion.

Introduction

To solve multi-body systems with contact, we can solve Linear Complimentarity Problem (L.C.P.) [1][2] described by the discretized system of equations:

$$v^{t+1} = v^t + hM^{-1}J^T\lambda + hM^{-1}f_{ext}$$
$$w = Jv$$

Where λ is the magnitude of the contact forces, and v is the velocity of each body at the next timestep. After a substition:

$$w = hJM^{-1}J^T\lambda + J\left(q^t + hM^{-1}f_{ext}\right)$$

Where we set w to zero and select one of many numerical solvers. One commonly used numerical method is the Projected Gauss Seidel (P.G.S.) [1].

PGS:

- Iterates through each contacts many times, getting more accurate with each iteration
- Handles friction by projecting found force required for immobility to maximum of system $F_F = \mu F_N$

Solutions like these scale in computation with the size of the system's contacts and number of bodies. However, only a subset of these bodies move in a noticeably dramatic way.

Many subsets of the bodies in our system may move similarly due to static friction, and can be grouped together. We aim to determine explicit rules for both the identification of these grouped bodies and a criterion for which these bodies stop behaving as one.

Figure 1:
Left: Large Multi-Body system filled with bodies that can be left out of the LCP solution. [2]
Right: Contacting objects, when in motion, may sometimes predictably have near-zero relative velocity.

Related Work

Many physics engines (PhysX, Vortex) may employ heuristics that puts bodies to "sleep" when bodies have a low velocity. Speedup is obtained by not checking for collisions between sleeping bodies and reducing the size of the LCP system. These bodies "wake up" with broad rules involving new contact detection or changes in applied force[3]. Erleben suggests[4] adding a time dependancy on this criterion: if the velocity is low for a **N** timesteps, then the body can sleep. This mitigates the chances of a body falling asleep at the peak of a parabolic trajectory. Shu[5] deals with "deactivating" objects in piles by using pile models to determine which bodies contribute negligible motion. The Vortex physics engine also allows for groups of bodies with consistently low relative velocities to be "merged" into a single body. These bodies "unmerge" with very broad rules involving new contacts with external bodies and changes in applied force/velocity.

Method: Merging

Our method merges contacting bodies if they meet the following three conditions:

Rule 1:
- Have a relative velocity below a user set-threshold

Rule 2:
- Maintain the above condition for **N** timesteps

Rule 3:
- Have the relative velocity be monotonically decreasing with time

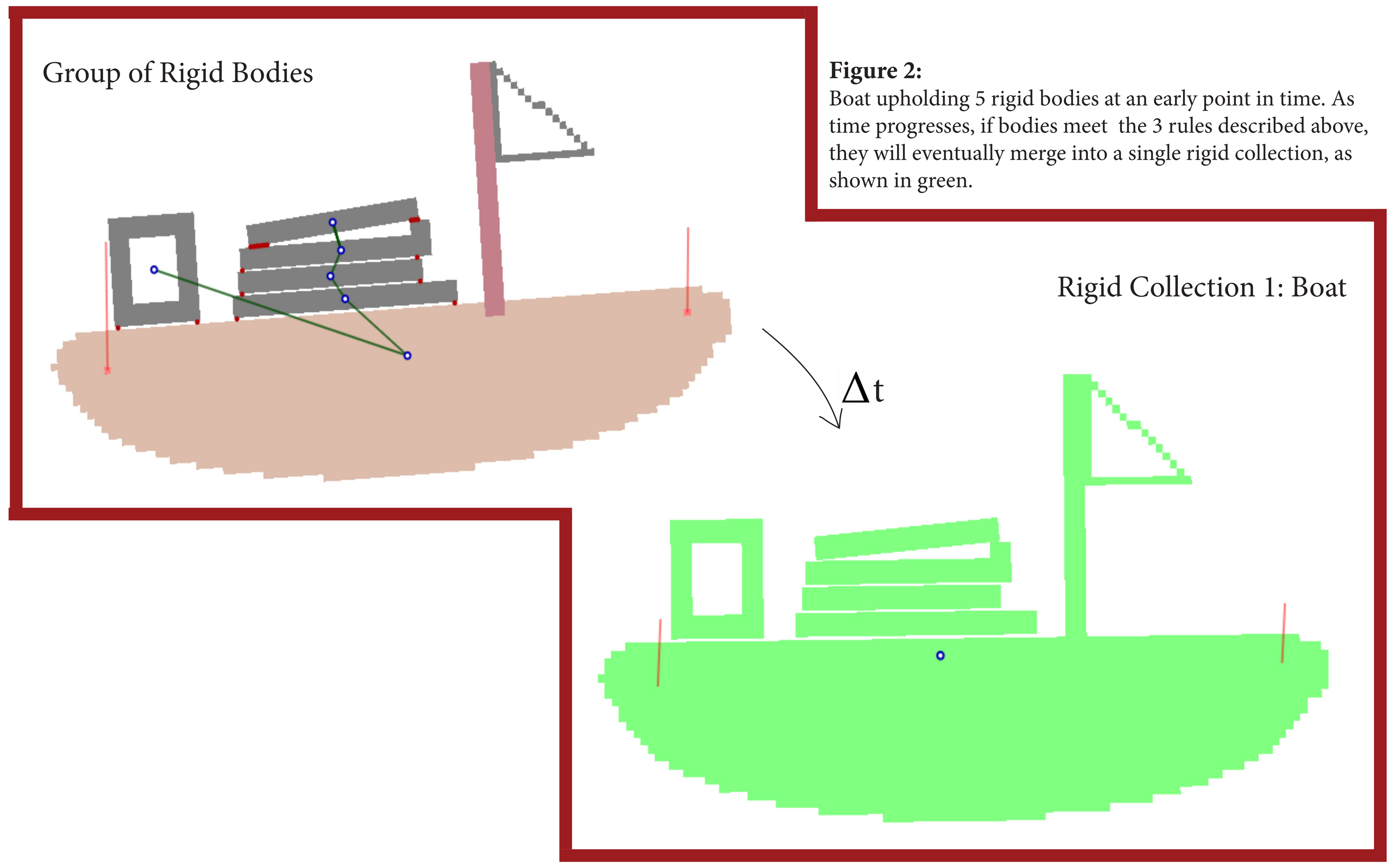

Figure 2:
Boat upholding 5 rigid bodies at an early point in time. As time progresses, if bodies meet the 3 rules described above, they will eventually merge into a single rigid collection, as shown in green.

Merged bodies are jointly referred to as **Rigid Collections**

Rigid Collections:
- Regulate each sub-body's motion
- Assume all external applied forces exerted on sub-bodies
- Remember internal contact information

Method: Unmerging

Unmerging must occur if the internal forces of a Rigid Collection experience a change such that the static friction gluing two sub-bodies is not enough to resist motion.

Rule 1:
- If the static friction force cannot resist motion, unmerge is required (Fig 3.)

Rule 2:
- If body is unmerged, need to recheck if any of it's neighbors unmerge.(Fig 4.)

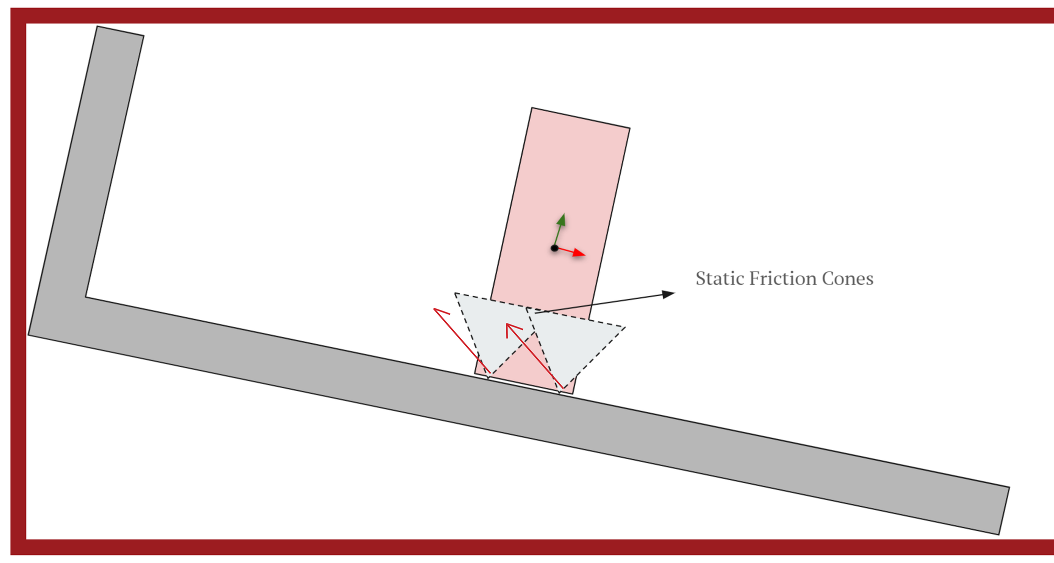

Figure 3:
2D representation of a dumpster. Assuming the two bodies are merged: because the contact force required to resist motion lies outside of the friction cone, the red body should unmerge.

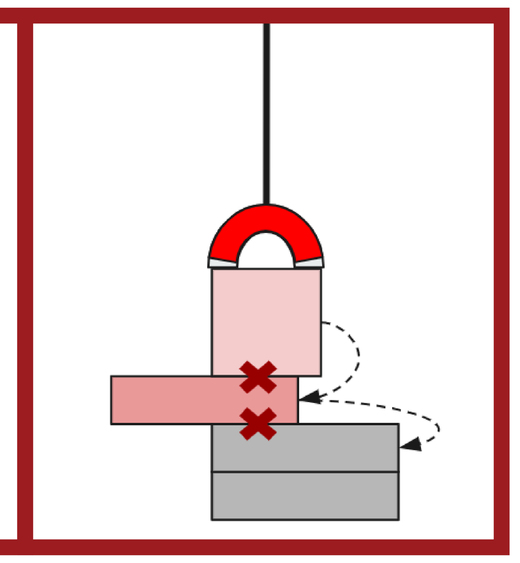

Figure 4:
The magnet activates on the square body and causes it to unmerge. We then need to check all neighbors of the unmerged square to find the rectangular body who's contact criteria is no longer met and should therefore also unmerge.

With no update, internal contact forces may quickly become inaccurate due to new merges/unmerges.

For **Rule 1** to work, internal contact forces **must**:
- Be kept up-to-date
- Do the above without a full LCP PGS solve

We suggest, for each timestep and for each collection:
- Treat collection as seperate RigidBody system
- Solve system with 1-iteration PGS solver, with warm start from previous time step

Now, if a new contact event occurs, our internal forces will adjust themselves with time to the correct value.

Conclusion

To simplify multi-body systems, we merge contacting bodies into Rigid Collections if their relative velocity is small and decreasing with time. This allows us to bypass the collision detection and reduced the number of bodies for in our LCP PGS solve. To make sure the internal contacts information does not become obsolete,we suggest a warm-started 1-iteration PGS loop within the system described by our Collection. With now up-to-date internal contact information, we can conservatively determine whether or not to unmerge a body if the contact force lies outside of the allowable friction cone.

References

[1] Kenny Erleben, Velocity-based shock propagation for multibody dynamics animation, ACM Transactions on Graphics (TOG), v.26 n.2, p.12-es, June 07

[2] Danny M. Kaufman , Timothy Edmunds , Dinesh K. Pai, Fast frictional dynamics for rigid bodies, ACM Transactions on Graphics (TOG), v.24 n.3, July 2005

[3] Ullman , TD, Spelke, ES, Battaglia, P, Tenenbaum, JB, Mind Games: Game Engines as an Architecture for In tuitive Physics, Trends in Cognitive Science, v.21, iss.9, p.649 - 665, June 17

[4] Kenny Erleban, Stable, Robust, and Versatile Multi-Body Dynamics Animation, PhD. Thesis, University of Coopenhaagen, Denmark, November 04

[5] Shu-Wei Hsu , John Keyser, Piles of objects, ACM Transactions on Graphics (TOG), v.29 n.6, December 2010 Multibody Dynamics Animation