



Home People Projects Products References

SEE++ (2005) Critique

Remote Sites

SEE-KID at UAR

These criticisms of SEE++ and suggestions for its improvement were offered to the authors in connection with the version of SEE++ that was current in 2005, when my colleagues and I last reviewed it. The current version of SEE++ may or may not be significantly different.

There is no dispute that SEE++ is a modern, well-structured software application, and that in many ways it is quite beautiful. Unfortunately, its implementation of "Orbit™1.8" is flawed. Further, some advanced aspects of the model are simple lookup tables, rather than proper biomechanical simulations.

There is no escaping the fact that, in a eye with elastic pulleys or fat pads that allow globe translation, muscle forces affect muscle paths and, for the case in which one is solving for them, affect eye positions as well. These effects alter muscle path lengths, and so, muscle forces. Iteration (or something equivalent) is inescapable. Porrell et-al (2000) made many approximations in their model, which was only intended to simulate normal eyes, and was only intended for study of a particular theoretical issue (separability). Porrell et-al made their approximations with clear understanding of their limited intents. The authors of SEE++, in contrast, have apparently found it computationally convenient to make similar approximations, despite the fact that they are highly inappropriate in a program intended to simulate arbitrary clinical abnormalities. For this reason alone, SEE++ must be less accurate than Orbit.

The SEE++ implementation of "active pulleys" is simply an incorporation of Kono et-al's (2002) measurements as an internal lookup table. SEE++ does not implement a biomechanical model with movable pulleys. A proper model must, of course, represent the underlying mechanics, and then *produce* measurements such as Kono's, as an emergent property.

In May of 2005, we prepared a list of suggestions to the UAR group, and provide it here for your critical evaluation:

Short-Term Suggestions: Position SEE++ to Replace Orbit 1.8

Fix Obvious SEE++ Problems

1. 3D Eye: "Coordinate Axes" are unspecified as to reference, though they are apparently head-fixed. They are unlabeled & unsigned.
2. 3D Eye & Hess Chart: Fixing & Following positions of 3D Eye are shown on the Hess Chart (fixing eye on one chart and corresponding following eye on the other) inconsistently with how Fixing & Following positions are shown within a Hess Chart (corresponding fixing and following eyes on same chart).
3. Muscle Data: Muscle Strength: I know what you mean here from Orbit, but I'll bet it's confusing to almost everyone who is not an Orbit user.
4. Muscle Data: Total Strength: This is a bad idea. It indulges users who can't distinguish elastic from contractile muscle force. This parameter also breaks the graphical connection between the 3 force surfaces.
5. Muscle force surfaces have undefined & illegible axes & units. Providing 3D rotation of these graphs is just eye candy, offering opportunities for confusion.
6. Muscle Force Distribution: Why the bias in favor of showing variation with horizontal gaze? Shouldn't this be some sort of 3D plot that treats equally, at least horizontal and vertical gaze components?
7. Hess Diagram: Why don't you show torsion graphically? Even old-fashioned Orbit does!
8. "Functional Topography" is undefined. I find no mention of it in the User's Manual. If it is actually something interesting & useful, it must be clearly explained.
9. Eliminate the "Pulley Model", which as I understand, is some sort of rudimentary Orbit-like model. It is nowhere described or validated, and causes only worry and confusion.
10. Where is the equivalent of Orbit's "Mechanical State Viewer", which showed simulation details? How are we to learn about the underlying mechanics without it? Its absence makes it impossible to really compare the SEE++ implementation of "Orbit core" with that of Orbit 1.8. Its absence causes worry and doubt! This level of insight into the model's functioning is essential!
11. The "Muscle Force Vector" diagram should probably be eliminated. If I correctly guess what it is, it is pretty meaningless if more than 1 muscle is selected. Where is this diagram explained?

Consider making the following improvements

1. User Levels in Orbit was a good idea! Why did you abandon this helpful interface concept? It makes it easy to include full details for advanced users (and for anyone who doubts the program's credibility!), while providing a simple interface for less knowledgeable users.
2. Where is the "Parameter Fitter"? That was a very cool feature of Orbit! Smart guys like you can certainly implement a fitting method better than Orbit's "brute force" search.
3. Orbit had extensive help, including "balloon help" (sort of like Window's little yellow "tool tips" tags), which tried to assure that the meaning of every item in the interface was clear. Is the meaning of everything in the SEE++ interface clear? Links to sections of the User Manual are not enough!
4. Implement Orbit's "steepest descent" solution method as an alternative to SEE++'s Marquart-Levenberg method, to facilitate comparisons with Orbit 1.8.
5. You need to clarify the relationship of SEE++ to Orbit 1.8. We have spoken about this in detail.
6. The SEE++ software architecture utilizes the Orbit 1.8 core code in a somewhat different way than does Orbit 1.8 itself. There are reasons to worry that these differences limit SEE++ validity to relatively "normal" eyes! Explain differences clearly and completely. Characterize resulting differences in simulations, including the magnitudes & situations in which these differences occur. It's not fair to simply feature simulations in which these differences do not show up!

Mid-Term Suggestions: Target Specific Problems in Strabismus and EOM Coordination, such as

1. So-called muscle "overaction" syndromes. For each, you will need to get several sets of patient data with good alignment measurements, preferably before & after well-described surgery. You can probably show that such syndromes are mainly or totally due to abnormalities of elastic force, not contractile force.

Long-Term Suggestions: Leverage SEE++'s Modern Software Architecture

1. Produce a new, conceptually unified, User Interface, not bound to Windows graphics (probably in Java).
2. Use the SEE++ container architecture to provide access to truly new biomechanical models:
3. Active pulley models.
4. Calculate, rather than assume, Sherrington's Law of Reciprocal Innervation.
5. Include structures, such as the "neurofibrovascular bundle" (Stager, 1996), the lateral levator aponeurosis and other prominent intermuscular tissues.
6. Explicitly include target distance, vergence input. Is L2 an emergent property?
7. Include otolith input.
8. Explicit implementation of common coordinate systems.
9. Allow non-spherical (particularly deeply myopic) eyes.
10. Implement common muscle-splitting operations.
11. Make muscle surfaces part of the biomechanical model (not just for show).