Natural-looking hair is a key component for presenting believable virtual humans, because the head and face form natural focal points of the human figure. In non-static scenes, hair behaviour is just as important as its looks. Principles of physics and dynamic simulation are often used for animating hair, because other traditional animation approaches—such as skeletal animation or motion capture—are difficult to apply to hair. Dynamic animation of hair is still an open problem without a known best solution, because hair has quite specific mechanical properties which, combined with the high number of hairs typically comprising a hairstyle, make realistic and efficient simulation challenging.

In this work, we focus on dynamic hair animation methods capable of providing real-time or interactive performance while staying physically plausible. Basing on research and analysis of hair properties from the cosmetic industry, we have devised a novel hair animation method which provides more realistic results than existing comparable methods while at the same time offering better performance and stability. We have applied this method to two different approaches to hair animation in order to prove its independence on any particular representation of hair. In one of these approaches, our method allows us to replace an iterative function minimisation with a direct computation, giving a speed-up of about an order of magnitude in this one stage of the simulation, while at the same time increasing its robustness.

Based on further observations natural behaviour of real hair, we have also proposed a novel organisation of simulated hair. This allows representing a larger number of hair strands by simulating fewer primitives without introducing artificial interpolation.

Additionally, we have analysed behaviour of real hair when in mutual contact and based on this analysis, proposed an efficient model of collision response for collisions between hair strands.

Our entire method is designed to be easily usable with current massively parallel computation architectures such as GPUs. To validate this design decision, we have also created a proof-of-concept implementation of the core parts of our simulation system on the GPU.