Deriving Complex Particle Motion Through Motion Vector Analysis

Johnathan R. Banta
AGRAPHA Productions

1. Introduction

Emitting a particle system from a 2D image sequence is limited to standard directional forces defined in most particle software, and is less likely to be driven by movement of objects depicted in the image. To inherit motion or avoid collision from movement in an image sequence, it is necessary to recreate that motion, most of the time through the use of matchmoved 3D geometry of some form, or 2D tracked points. Furthermore, forces which act on these particle systems are often nothing more than standard directional forces and forms of fractal-based turbulence.

This paper demonstrates motion vector analysis, and 2D alpha-derived shapes to cheaply recreate complex motion for particle systems. The technique approximates fluid advection, repulsing or attracting particles, while preserving a dimensional "feel" for the derived motion. It also proposes layered particle systems as a model for motion derived turbulent forces.

2. Exposition

A complex object emitting a trail of smoke, or fluid of some form is a common sight in motion picture entertainment. In the real world things get wet, dusty, or catch fire, and modern visual effects recreate that in many ways. If these objects have to match imagery already photographed, it is best to virtually recreate all the conditions of the event at the time of photography. Three dimensional geometry lined up, virtual cameras tracked, wind, gravity, timing, and artistry all mix together for a convincing result. It is also an expensive time investment.

Two-dimensional particle systems, often added to current compositing packages, are computationally cheaper than full 3D particle systems, or if they work in 3D space are accelerated or highly optimized. These systems, though limited in comparison to higher-end packages, are more affordable, and under direction of a competent artist produce spectacular results at a lower cost in time and money, but are the result of interpretation, rather than actual velocities from the imagery, or its geometric analog.

When a particle is emitted from a surface, due to Newton's laws of motion, it inherits the direction and velocity of that surface point at the time of emission. Most physical objects, like water or dust resist this motion, since they have a level of surface tension holding them to the object, until a large enough velocity is reached to the point that it will break free. Once free from the object it sails in a direct line in the direction of the motion, until another force acts upon it.

In a two dimensional image, the amount of force emitted by an object in the image plane, is most directly derived from its motion differential over time. This rate is most often measured in two planes with a motion-vector analysis, which records the disparity of each pixel on a frame-by-frame basis, and is stored as grayscale data in individual color channels in another image. The resulting motion-vector map is most often used to add or remove motion blur from an image, or to warp imagery to a novel position along the vector of motion for image retiming. However, this force and vector analysis can also directly modify the motion of particles as they are emitted from the imagery.

In three dimensional particle emission, it is common to use a low-subdivision fluid volume to store vector and magnitude information as it is transferred from the surface point into the surrounding environment. This fluid flow modifies the direction of any particle in the area, and allows particles not connected to the surface to be affected by its motion through the surrounding substance – be it air or liquid. In the two dimensional plane, this same effect is easily produced by using the motion vector [Ma, et al. 2009] to continually modify the particle system, rather than just use it for initial direction and magnitude.

The force within the motion vector map is rarely highly detailed, and will occupy space between image points, as it is designed to be a record of the entire journey of a feature point between time samples. This force is therefore affecting the space surrounding the image feature, and represents a "push" force to any particle in the area.

In effect, the motion vector map is a record of emissive vector and magnitude, as well as fluid vector advection. But fluid forces and directional forces like wind and gravity are not the only things that affect this motion. Since two objects do not generally occupy the same space in the real world, it is necessary to give some level of collision to surfaces within the image, and these are not recorded in the motion vector map.

In a 3D particle environment, a collision surface pushes back against a moving particle with its surface normal. A direct hit will reflect the particle away, which is labeled as a "bounce event" in most particle systems. In a 2D environment, the best way to represent this normal vector is with a normal map rendered for the object, but since we are not solving 3D geometry, and have no access to actual normal information for the image, approximating a screen-space normal is the next best thing — and this only requires that we know an object's silhouette.

Alpha inflation techniques commonly used to add lighting to flat artwork produce normals for the object by treating the alpha, or a blurred version of the alpha, as a height field. This height field is evaluated, and screen-space normals generated, to which lighting calculations are applied. In the case of particle collision, we only want the normals generated from the process. These normals represent a facing slope in XYZ space in an RGB image. Since we are currently only concerned about vectors in the two dimensional plane, this will work directly with the motion vector forces outlined above.

As a particle approaches the recovered normal, it looks forward in time to update its position, but reads an opposing force from the normal map, which will alter its direction, thus mimicking collision avoidance. For purposes of our discussion we will call this Collision Normals. However inverting this normal map produces the exact opposite event, and particles are attracted to the object as the distance closes. For purposes of our discussion we will call this Attraction Normals.

As a collision normal moves through a particle field, it pushes the particles around, while the Attraction Normal draws them in, at times producing particle motion that resembles eddies in a fluid stream. This phenomenon introduces a method to directly create a form of turbulent force based on layered particle emission.

The method to particle-based turbulent flow assumes that turbulence is a force thrown off of a moving object, which varies the flow of subsequent systems. By combining our motion vector emitter with spherical fields modeled on Attraction Normals, we can create a particle-based field that can be added back into the initial motion vector analysis, and directly affect the flow of the particles in a pseudo fluid volume directly influenced by the moving footage.

Each particle is replaced by a sprite containing a spherical attractor normal map. The existing emission system drives this low-count particle event. Generally these turbulence particles have a larger amount of drag, as their mass is less, and they are less affected by the other forces in the emission system. The resulting system is composited over the original motion vector extracted element. The turbulence particles need not be spherical. Any shape necessary to get an effect will suffice. Adding the particle forces together, and varying their opacity vary the outcome as well.

The compositing method for the turbulent system affects the final result. If the particles are composited in a standard OVER operation, each attractor has 100% contribution at that moment to the particle vector, and will quickly diverge its path. However if it is assumed these are additive forces, and the compositing setup is not floating point, then it is necessary to choose a transfer mode that splits the 0 to 1 range of the normal map, and converts it to positive and negative values in the range of -1 to 1, before adding it back to the mix, and re-normalizing. Adobe Photoshop's LINEAR LIGHT operator works in this manner.

Stacked motion vector analysis is also an effective way to introduce turbulence to the particle system. Whether it is an analysis of smoke, fire, or some other footage, or a secondary analysis of emitted particle systems, since it is visible imagery, it is possible to quickly ascertain the effect of the system, rather than merely wait for the result of the particle simulation to see what it does.

2.1 Elaboration

This technique was developed with off-the-shelf software, so specific code examples are not available. An attempt is made throughout this presentation not to directly reference software, but rather speak globally about the results.

The first step to test the methodology starts with motion vector analysis of the original footage. Two sources are used in the example, one of a gesticulating hand, and the other with more complex global motion. After analysis the motion vector is stored in 16-bit to avoid any mach banding, and to avoid constant recalculation.

Emitter Design.

The goal is derive all of the particle system forces or emitters from an existing image sequence, and not to rely on outside contribution, other than a few isolation mattes.

The examples emit particles from areas of greatest movement in the image. The emitter is a simple matte that differences (abs(A-B)) one preceding frame, and one following frame from the current frame. The pair of images is then combined with a minimum operation, so only the areas of movement in the current frame remain. This method does not account for movement between frames, and some aliasing of the emitter will occur, but it fits the need to contain the emitter to strict image analysis. This is called the "motion matte."

It is possible to expand the emission area to cover the missing temporal sample using the motion vectors to blur the emitter, and temporally anti-alias the motion isolation.

Normal map.

A rough extraction of the moving object via procedural matting and rotoscope give us the initial condition. This matte is treated with an algorithm that "puffs up" the layer by blurring it, and using the rusting gradient as a height map. Camera space tangent normals are derived from this faux inflation, for later use as a vector force in the particle systems.

Particle systems

A basic two-dimensional particle system will suffice, that has standard directional and gravity forces, but it must have the ability to emit from a layer of imagery, and accept other color input as a force. Many off the shelf particle systems have this capability, whether they are 2D or 3D. Often the color information is used to control the life or size of a particle, but in this instance it must have the ability to apply the individual color channels as a vector force in XY(z).

3. Results

Example 1. The first example is the control sample. A particle system emitting from our emitter, dropping under a directional gravity force. It is interesting to look at, as the emitter moves around it creates the illusion of movement in the system, but without any secondary force to change direction.

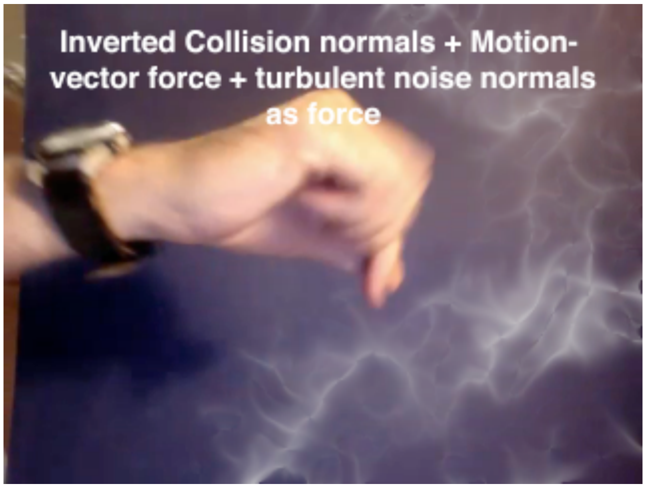
Example 2. The second example shows the same parameters, but with a radial force spreading the particles out from the point of emission before they are captured by gravity. The motion is more complex, but still inherits no more motion from the image sequence, than its point of origin.

Example 3. The third example uses the motion vector disparity map as an emission force applied to the particles just affected by gravity, at the birth of each particle. Particles are thrown from the emitter, and inherit the velocity of the moving — a far more complex result that has the appearance of existing in three-dimensional space.

Example 4. This is another control situation. A particle stream is emitted into the space around the moving object. Gravity is slightly reversed to give the particles the feeling of rising smoke, and treated with a post-process to improve the illusion of smoke. As the hand moves through it, the "smoke" is unaffected.

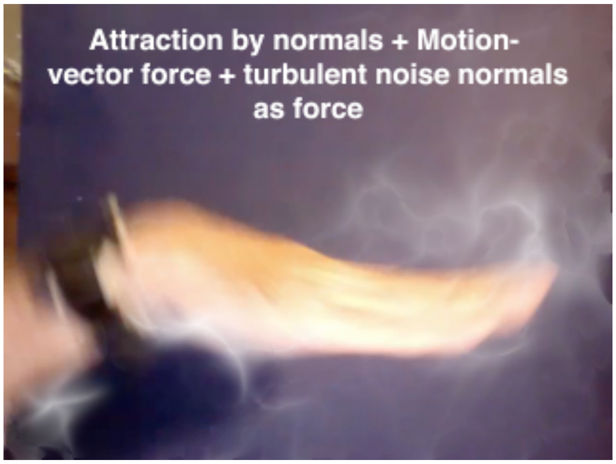
Example 5. The extracted normal map is used as a constant vector force to affect the particles after birth. As the object moves through the frame, the force from the normal map pushes the "smoke" away.

Example 6. The extracted normal map is inverted and used as a constant vector force to affect the particles after birth. As the object moves through the frame, the force from the inverted normal map attracts the "smoke" to its center, introducing a turbulent wake as particles are temporarily sucked into a vacuum.



Example 7. Motion vectors are used as a constant vector force to affect the particles after birth. As the object moves through the frame, the force from the vectors pushes the particles out of the way along the vector of motion, producing results similar to pressure forces in a fluid container.

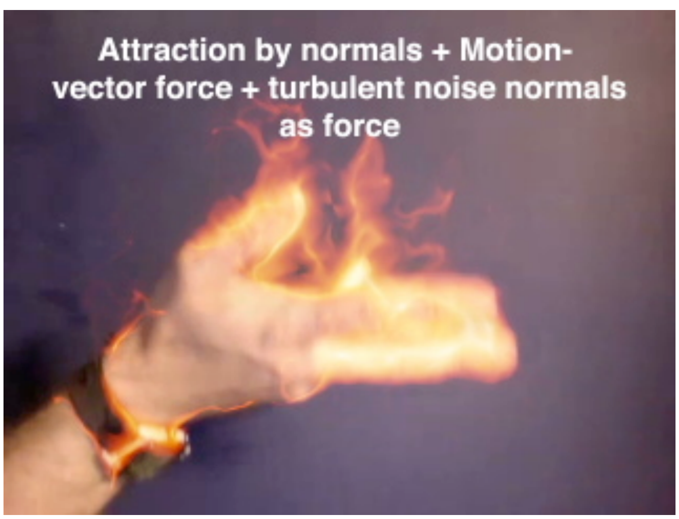
Example 8. Motion vectors and normal maps are combined and used as a constant vector force to affect the particles after birth. As the object moves through the frame, the force from the vectors pushes the particles out of the way along the vector of motion, and away from the surface.



Example 9. Motion vectors and inverted normal maps are combined and used as a constant vector force to affect the particles after birth. As the object moves through the frame, the force from the vectors pushes the particles out of the way along the vector of motion, and the inverted normal map attracts the them and introducing turbulent artifacts.

Example 10. Combining all the forces together. A particle stream is emitted form the object with motion vector forces modifying the initial vector. To look like smoke, the magnitude of the force is reduced, and gravity reversed to "loft the smoke" upward. A combined attraction normal and motion vector are used as a constant force after emission, to introduce the fluid-like and turbulent properties.

Example 11. Similar to example 9, but the particle life is reduced, and the post compositing changed to look like flame.



Example 12. Sprites designed to spherical attractors are emitted from the object using vector motion. These will be used in the next example to explicitly model turbulence. The particles are given a greater amount of drag than the systems they will emit, to suggest a variance in mass property from future emitted systems.

Example 13. Particles are emitted from the moving object using motion vectors as an initial force. The turbulent particles from Example 12 are combined with the object normal map, and motion vectors and applied as a constant force to the particle system. The emitted particles push away from the surface, and fly along their initial motion vector, after which they advect through the faux fluid system, and twist around based on the affect of the turbulent particles.

Example 14. 2D particle emission from a writhing CG tentacle, affected by gravity. Secondary layered particle systems used to create turbulent foam at the base of the tentacle — derived form the existing emitter, but cropped to the implied ocean height. This example created by an available product that was modified to use motion vector input because of this research.

4. Conclusions

The research in this presentation is entirely a two-dimensional evaluation of motion used to drive a two-dimensional particle system, but it is possible to use this in 3D as a planar emitter, and scale the motion on the Z-axis in any manner you determine. The motion vectors could also be extruded along the frustum of the viewing camera and used as a three dimensional volume to affect emission. Standard turbulent forces such as fluid noise could be mixed into this motion vector voxel, or explicit turbulent particles

Motion vectors and other image analysis processes are an effective way to drive particle motion that has the appearance of a three dimensional particle system, and induce a fluid dynamics feel to the motion cheaply. The method works with computer generated imagery or photography. It is proven in production, and achievable with off-the-shelf technology, as well as any custom software yet to be created.

References

Ma, Chongyang, Wei, Li-Yi, Guo, Baining, Zhou, Kun. Motion field texture synthesis. In *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2009),* vol. 28 Issue 5, December 2009, Article No. 110