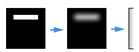
Gel simulation visual features

Diffusive blurring

Reference

Explanation

Simulation



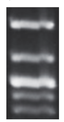
• Render each band to an texture, apply a gaussion blur shader, and then composite band back into main image.
• Blur amount is tied to diffusion factor.

Parameters/ Functions

• calcDiffusionForBP()
• getDiffusionForDye()
• Note that diffusion rate is a non-linear function of bp; it should tick up quickly below 1500bp.
• Maybe a custom ramp to map intensity values.

Damaged well

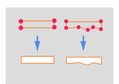
a



Explanation

Band shape mirrors that of well. If well is damaged, i.e. not-rectangular, then the bands will be too.

Simulation



Model each well as a polygon, and use it to generate bands.

Parameters/ Functions

• **Slider** for well damage
• Procedural well damage generator.
• Can we assume that damage is always deltay = f(x,damage)? If so, we could handle well shapes as a texture generated by a shader (see reference).

Gradients

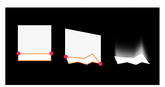
a

Explanation

Blur above is aggregating particles slowing things down. Blur below is degraded particles speeding up.

Simulation

1 2 3



1. Extrude band geometry, and fill with a gradient.

2. Will require some thought with non-rectangular gel geometry. Something with normals and extrusion; perhaps approximate with ray-casting; or get the convex hull, but this might have artifacts in some edge cases, too; maybe approximate custom gel geometry with a bounding box.

3. Combo of custom motion blur effect and geometry stretching/manipulation. Draw to texture, stretch texture (non-linearly?), fill it with a gradient?

Parameters/ Functions

• Existing aggregation/degradation positioning logic is to be used to figure out how far to blur.
• **Q: What about when our degrade param is >1?**

Stuck in well

a

Explanation

Too much sample; it gets stuck trying to exit the well.

Simulation

Stop Y-travel and possibly other features (blur, gradient, overloading deformation, etc...) if there is too much sample.

Parameters/ Functions

• Threshold for bp/mass/ aggregatoion. What is it a funciton of? What is the cutoff?

Overloading

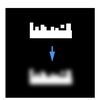
a

Goopy flames

Explanation

Too much sample. It is tripping over itself and can't easily find a way forward.

Simulation



Procedurally generate some "flames" at the back of the band. Blurring will complete the effect.

Parameters/ Functions

• How much to smile at edges. (w,h) = f(frag)
• How much to flame?
• deform: p += deform(band_pos)

Smiling/frowning bands

b

Edges peel back

Explanation

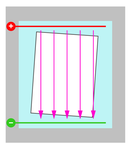
???

Simulation

Deform (either pixels or mesh) with a slight smile effect at edges

Rotated gel

Explanation



Gel rotated in running buffer. So bands move at an angle relative to buffer edge.

Simulation

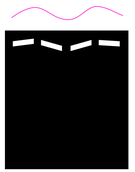
Translate along a rotated axis. Other operations (e.g. flames, smiling, blurring) will probably also need to operate on this rotated axis.

Parameters/ Functions

• **Needs slider/control**

Wavy field

Explanation



Voltage too high. Gel gets cooked.

Simulation

• Same implementation as smiling/ frowning gel? (y += y_deform(gel_x))

X-Drift (Lane flexion)

Explanation

Sample and gel buffers salt differs.

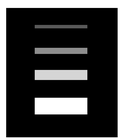
Simulation

• x += x_deform(gel_pos)

• Generalize?
p += deform(gel_pos)
• Or implement as a gel wide displacement map that indicates how far in x,y to move to get to result. For user interaction, we could invert the map by running an input in which each pixel is set to <src-x,src-y>, so the result indicates how to get back to input.

Thickness & Brightness

Explanation



As concentration goes up, thickness and brightness go up.

Simulation

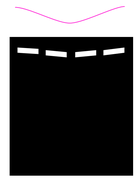
Vary thickness and color with concentration.

Parameters/ Functions

• thickness(frag)
• brightness(frag)

Smiling/frowning gel

Explanation



???

Can be an arbitrary gel-wide deformation, smiling or frowning; not necessarily symmetrical.

Simulation

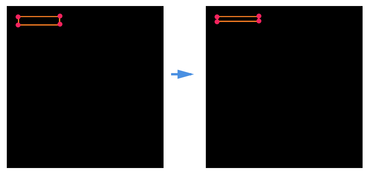
y += y_deform(gel_x)

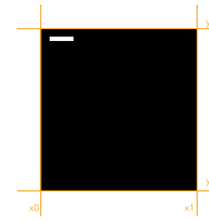
Gel band generation pipeline

Start

Get rectangular vertices for this well.

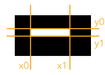
Set to desired thickness, position in well.





Draw it to a texture with desired.

Texture size is size of well bounds.



We know mapping of (0,0,1,1) => gel space (rectangle vertices we started with). Each stage has:
• Texture coords (0,0,1,1)
• Gel coords (rect vertices)
• T => Gel matrix, and inverse.
• So, we should be able to map positions wherever.

Would it be easier to just render each band at the resolution of the full gel sim? YES. No constant error-prone texture/coordinate space noodling/mapping. Almost like a full-gel sized CA simulation, per band, that we then composite all together.

Add any flame effects.

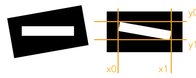
Grow rect in gel space.
Grow texture size.

And we can can compute a parallel pick shape that follows the same/similar logic to get a user interface handle. (Even more bonkers would be to do some image processing and use the generated texture data to handle picking.)

Deform texture with well damage. Apply a random y-displacement function (reusing smiling band logic).

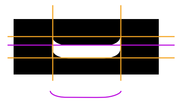
Same as gel smile band logic.

Rotate texture to negative gel rotation.

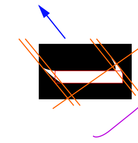


Rotate rect vertices in gel space, calc new bounding box.
Use bounding box delta to grow texture.
Use that rotation matrix to draw input texture rotated for output.

Deform texture with band smile effect. Apply an arbitrary y-displacement deformation to image.

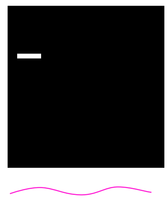


Expand rect vertices up (or down for drown) in gel space by max smile amount.
Apply deformation.

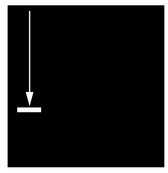


Gel wide texture will require that we can do this displacement on any arbitrary axis to compensate for rotation. Should just be as simple as writing y-displacement version, then adding a transform matrix in there somewhere to handle translate/rotate.

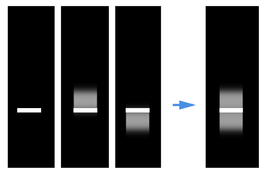
Wavy-field effect.
Smiling gel effect
(Do in one or two passes.)



Translate in y according to calcDeltaY simulation factors.

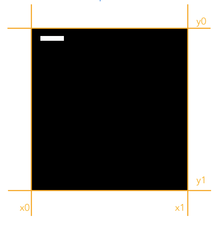


Aggregation / degradation blur. Composite back into itself with a kind of motion blur effect along the movement axis

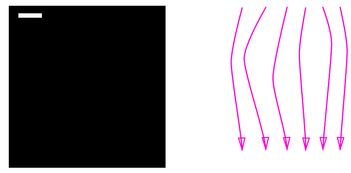


Diffusive blur.

Composite into the scene, and apply brightness and color.

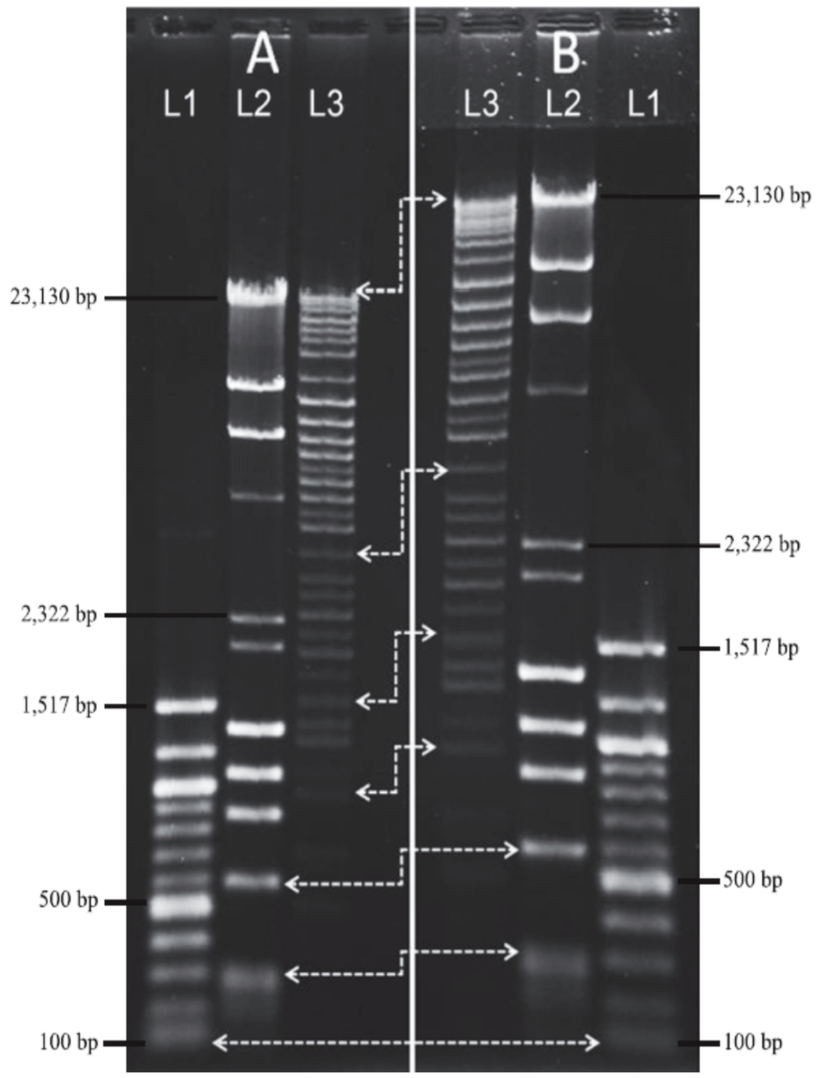


Apply flexion / x-drift effect to composited scene.



References

a. Lee and Bahaman (2012)



b. 1kb ladder