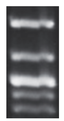
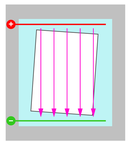
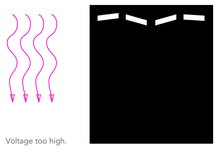
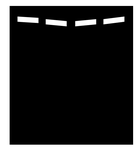
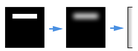
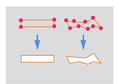
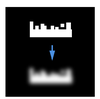
Lane flexion

	Diffusive blurring	Damaged well	Gradients	Stuck in well	Overloading	Smiling/frowning bands	Rotated gel	Wavy field	X-Drift	Thickness & Brightness	Smiling/frowning gel
Reference		a 	a 	a 	a  Goopy flames	b  Edges peel back					
Explanation		 Band shape mirrors that of well. If well is damaged, i.e. not-rectangular, then the bands will be too.	 Blur above is aggregating particles slowing things down. Blur below is degraded particles speeding up.	 Too much sample; it gets stuck trying to exit the well.	Too much sample. It is tripping over itself and can't easily find a way forward.	???	 Gel rotated in running buffer. So bands move at an angle relative to buffer edge.	 Voltage too high. Gel gets cooked.	 Sample and gel buffers salt differs.	 As concentration goes up, thickness and brightness go up.	 ??? Can be an arbitrary gel-wide deformation, smiling or frowning; not necessarily symmetrical.
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.	 Model each well as a polygon, and use it to generate bands.	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?	Stop Y-travel and possibly other features (blur, gradient, overloading deformation, etc...) if there is too much sample.	 Procedurally generate some "flames" at the back of the band. Blurring will complete the effect.  Deform (either pixels or mesh) with a slight smile effect at edges	Translate along a rotated axis. Other operations (e.g. flames, smiling, blurring) will probably also need to operate on this rotated axis.	• Same implementation as smiling/ frowning gel? (y += y_deform(gel_x))	• x += x_deform(gel_pos) • Generalize? p += deform(gel_pos)	Vary thickness and color with concentration.	y += y_deform(gel_x)
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.	• **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).	• 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?**	• Threshold for bp/mass/ aggregatoion. What is it a funciton of? What is the cutoff?	• How much to smile at edges. (w,h) = f(frag) • How much to flame? • deform: p += deform(band_pos)					• thickness(frag) • brightness(frag)	