Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
225157	0	0	0															
270034	5	5	0	1	2	2	0	0										
293846	0	0	0															
295323	0	0	0															
345642	0	0	0															
348843	0	0	0															
349936	11	9	2	1	2	2	4	0		R5: normal transformed like point (w=1), also not normalized. GPU-SSD: xform_accum needs to be zero-initialized (-0p)	2							
350475	9.5	9.5	0	1	2	2	4	0.5		R3: two of your coord axes point to the same direction due to using the right-direction twice instead of right/up. (-0p) R5: normals not normalized.								
352091	0	0	0															
353980	0	0	0															
354439	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
355593	0	0	0															
356026	11.5	10	1.5	1	2	2	4	1		GPU SSD: normal not normalized in shader.	1.5							
361749	0	0	0															
369181	9	8	1	1	1	2	4	0		R5 + GPU SSD: normals transformed like points + not normalized.	1							
372660	0.5	0.5	0	0.5	0	0	0	0		R1: you are not using the to_parent matrices so nothing happens. use joint2.to_world = joint2.to_parent * parent_to_world instead.								
387370	0	0	0															
425575	0	0	0															
425614	11	9	2	1	2	2	3	1			2							
426419	2.5	2.5	0	1	1.5	0	0	0		R2: somewhat involved way of constructing the rotation - use the library functions. Rotation of a joint also moves the joint slightly, which is not expected.								
427489	9	9	0	1	2	2	4	0										
428022	0	0	0															
429487	0	0	0															
430829	3	3	0	1	2	0	0	0										
457598	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
460297	0	0	0															
464772	0	0	0															
46477D	0	0	0															
46596K	0	0	0															
474199	9	9	0	1	2	2	4	0		R5: normals treated like points and not normalized.								
474322	9	9	0	1	2	2	4	0										
474458	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized.								
474898	4.5	4.5	0	0.5	2	2	0	0		R1: wrong order of multiplication of matrices parent_to_world*to_parent. Fixing this fixes R2.								
475389	0	0	0															
475813	0	0	0															
475910	11	9.5	1.5	1	2	2	4	0.5		R5: normal not normalized. GPU SSD: you don't want to be using worldToClip with the normals when light direction is in world-space. Both the vectors need to be in the same space for the inner product to make sense!	1.5							

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
477329	9	9	0	1	2	1.5	4	0.5		R3: rotating a joint also affects the orientation of the children's coord systems. R5: normals not normalized.								
477811	10	10	0	1	2	2	4	1		R5: maybe a bit non-standard way of doing this, but checks out!								
478328	5	5	0	1	2	2												
478470	0	0	0															
478687	0	0	0															
479505	1	1	0	1	0	0	0	0		R1: You are not quite constructing the matrix correctly. You are not using the to_parent matrices (but rather only the translation component of those matrices). Hence it seems that R1 works since it is only translations, however since you are not actually using the to_parent matrices that get changed in R2, the rotations never get applied.								
479589	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized.								
479741	10	10	0	1	2	2	4	1										
480086	0	0	0															
480248	0	0	0															
480714	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
480798	0	0	0															
481577	9	9	0	1	2	2	4	0		R5: you are transforming directions like points + leaving result unnormalized.								
493840	11	9.5	1.5	1	2	2	4	0.5		R5 + GPU SSD: normals not normalized. Animation is completely broken - I don't really see any interpolation happening anywhere?	1.5	0						
506300	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
508285	10	9	1	1	2	2	4	0		R5 + GPU SSD: normals transformed like points, not normalized.	1							
51620U	0	0	0															
524926	5	5	0	1	2	2	0	0										
525417	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
525491	0	0	0															
525941	0	0	0															
526050	9	9	0	1	2	1.5	4	0.5		R3: you are using the rows of the orientation matrix instead of the cols, meaning your local coordinate system rotates to the opposite direction of the actual rotation. R5: normals not normalized.								
526319	5	5	0	1	2	2	0	0		R4: crashes on debug mode.								
526775	0	0	0															
527143	0	0	0															
527389	0	0	0															
528867	0	0	0															
528883	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
529196	0	0	0															
529303	0	0	0															
530185	0	0	0															
530363	0	0	0															
530619	0	0	0															
530648	0	0	0															
530868	14	9.5	4.5	1	2	2	4	0.5		R5 + GPU SSD: normal not normalized.	1.5	3						

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
530981	7	7	0	1	2	2	2	0		R4: when computing toBindTransforms you are using "auto joint : joints_" which does not update the actual joints since it's not pass by reference but by value. Also, when computing the position, you use (sv.position, 0) instad of (sv.position, 1).								
540094	0	0	0															
540654	0	0	0															
544566	4	4	0	1	1	2	0	0		R2: rotations wrt angle controlled by left-right -keys causes weird warping. R3: you seem to use the up direction twice instead of up/ahead (-0p).								
549040	5	5	0	1	2	2	0	0										
549163	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized.								
55055P	14.5	9.5	5	1	2	2	4	0.5		R5: Normal transformed like position (w=1)	2	3						
552794	5	5	0	1	2	2	0	0										
552969	0	0	0															
554598	0	0	0															
563068	5	5	0	1	2	2	0	0		R3: you are using the .y -coord of each dir when adding up/ahead/right vectors to the joint positions (-0p)								
576149	0	0	0															
585716	0	0	0															
586333	11	9.5	1.5	1	2	2	4	0.5		R5: normals not normalized. GPU SSD: normals not normalized, some elements of wTB remain uninitialized which on my machine resulted in erratic ripples.	1.5							
586702	9	9	0	1	2	2	4	0										
586980	0	0	0															
587316	0	0	0															
587471	11.5	10	1.5	1	2	2	4	1		GPU SSD: normal transformed like point in shader.	1.5							
588289	9	9	0	1	2	2	4	0		R5: normals treated like points and not normalized. Also inverse-transpose of a sum of matrices may not be the sum of the inverse transposes (does not even work with scalars!) Inverting the matrix at all steps of the loop also kills any performance.								
589291	0	0	0															
589343	0	0	0															
589848	8	8	0	1	2	0.5	4	0.5		R3: bones only. R5: normals transformed like points.								
590921	0	0	0															
591904	5	5	0	1	2	2	0	0										
591946	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
592929	0	0	0															
593274	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
593847	0	0	0															
594435	0	0	0															
595926	10	10	0	1	2	2	4	1										
595997	0	0	0															
596747	0	0	0															
597429	9.5	9.5	0	1	2	2	4	0.5		R5: unnormalized normals. You don't need to call getSSDTransforms separately for each vertex. Removing this gives around 20x boost to FPS.								

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
597623	9	9	0	1	2	2	4											
597937	15	9.5	5.5	1	2	2	4	0.5		R5 + GPU SSD: normals not normalized after transformations.	1.5		4					
602851	0	0	0															
602893	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
603096	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
603245	8.5	8.5	0	1	2	2	3.5	0		R4: problem was caused by the uninitialized matrix sigma. Uncomment the sigma.setZero() to fix.								
604095	32	10	22	1	2	2	4	1		Pose capture/transfer just seems to be really broken :(Good effort, though!	2	3	4	5	8			
604273	7.5	7.5	0	1	2	1.5	3	0		R3: you are using the rows of the orientation instead of the columns. Causes rotation of the local system to be inverted. R4: the loop of getSSDTransforms is broken and trasforms-vector is always empty. R5: treating directions like points, unnormalized afterwards.								
606064	9.5	9.5	0	1	2	2	4	0.5		R5: normals not normalized.								
608949	0	0	0															
609155	0	0	0															
609249	0	0	0															
612472	10	10	0	1	2	2	4	1										
612498	0	0	0															
612870	11	9	2	1	2	2	4	0		R5, SSD on GPU: normals transformed like points (w=1), also unnormalized	2							
614577	0	0	0															
614580	0	0	0															
621308	6.5	6.5	0	1	2	0	3.5	0		R2: you are not using the coordinate system given by the transform matrices at all. R4: you are using "auto joint : joints_" instead of "auto& joint : joints_" in computeToBindTransforms, so the actual joints never get non-identity to_bind_joints - only their (discarded) copies do!								
628835	5	5	0	1	2	2												
63036R	0	0	0															
641922	0	0	0															
646804	8.5	8.5	0	1	2	1.5	4	0		R3: you are using the rows of the orientation matrix rather than the cols. Results in the joint coord system rotating in the opposite direction than expected. R5: treating directions as points + unnormalized result.								
647764	9.5	9.5	0	1	2	2	4	0.5		R5: unnormalized normals. Running the program in VS reduces the framerate somewhat due to profiling or whatnot even in release mode. Running the executable directly gives more-or-less the same framerate as in the example.								
648530	0	0	0															
648860	0	0	0															
650191	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
650227	0	0	0															
650405	0	0	0															
650560	3	3	0	1	2	0	0	0										
650942	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
651527	0	0	0															

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
651585	14.5	9.5	5	1	2	2	4	0.5		R5: unnormalized normals (you did normalize in the extra at the shader though!!)	2	3						
651637	9	9	0	1	2	2	4	0		R5: normal transformed like points (w=1), not normalized								
651789	0	0	0															
652102	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
652131	0	0	0															
652335	16.5	9.5	7	1	2	2	4	0.5		R5: normal not normalized	2			5				
652649	11	9.5	1.5	1	2	2	4	0.5		R5 + GPU SSD: unnormalized normals.	1.5							
652898	0	0	0															
652937	9.5	9.5	0	1	2	2	4	0.5		R5: unnormalized normals. Might also be a good idea to explicitly initialize newpos and newnormal to zero.								
653127	12	10	2	1	2	2	4	1			2							
653596	0	0	0															
653693	10	10	0	1	2	2	4	1										
653871	0	0	0															
653907	9.5	9.5	0	1	2	2	4	0.5		R5: unnormalized normals.								
653910	10	10	0	1	2	2	4	1										
654595	19	10	9	1	2	2	4	1		Nice wrists!	2			5	2			
655057	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
655086	14	10	4	1	2	2	4	1		Dual quaternions: could not immediately see the problem, but code looks quite reasonable.	2		2					
655109	9.5	9.5	0	1	2	2	4	0.5		R5: unnormalized normal.								
655251	0	0	0															
655264	0	0	0															
655471	10.5	8.5	2	1	2	0.5	4	1		R3: right, up, and ahead are the cols of the transformation matrix, using world space offsets doesn't work. Q: Inv-transp is more correct, but differnces quite small here. SSD: accum. matrix needs to be zero-initialized (not identity)	2							
655691	0	0	0															
655853	0	0	0	0	0	0	0	0		Src folder empty								
656250	3	3	0	1	2	0	0	0										
656454	10	10	0	1	2	2	4	1										
656616	10.5	9.5	1	1	2	2	4	0.5		R5 + GPU SSD: unnormalized normals. GPU SSD: weighted_sum is uninitialized and on my machine results in completely erratic jittering	1							
657291	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
657314	0	0	0															
657327	9	9	0	1	2	2	4	0		R5: normals transformed like points + unnormalized.								
657482	0	0	0															
657796	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
657893	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
659914	25	10	15	1	2	2	4	1		GPU_SSD: aWeights2, aJoints2 indexing starts at 0, not 4 (-0p)	2			5	8			
660246	14.5	9.5	5	1	2	2	4	0.5		R5 + GPU SSD: unnormalized normals. Dual quaternions: there seemed to be a sign error at getDoubleQuaternions with x,y and z.	1.5		3.5					
660877	0	0	0															
660893	0	0	0															
663191	0	0	0															
663272	0	0	0															

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
665380	10	10	0	1	2	2	4	1		No readme, let us know if we missed any extras								
665898	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
666172	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
666350	12	10	2	1	2	2	4	1		Yup, normalizing is the right thing to do!	2							
666680	21.5	9.5	12	1	2	2	4	0.5		R5, GPU-SSD: normal not normalized. IK: a more rigorous approach would be to use cross products (http://www.cs.cmu.edu/~15464-s13/lectures/lecture6/IK.pdf) instead of finite differences	2			5	5			
667249	7.5	7.5	0	1	2	2	2	0.5		R4: getSSDTransforms unimplemented? R5: normal not normalized								
67137M	0	0	0															
67627H	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
677734	0	0	0															
678089	0	0	0															
68933B	0	0	0															
69247N	0	0	0															
700436	0	0	0															
705570	59	10	49	1	2	2	4	1			2	3	4	5	14		21	Weight editor (6p), wireframe visualization (1p), several IK modes (+6), DAE loader and converter (8p), animation blending and character control (6p)
706566	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized. Funnily enough, I had to undo your std::max fix to get the code to compile								
708784	24.5	9.5	15	1	2	2	4	0.5		R5: normal not normalized. About workload: it is quite hard to know how much time any given extra will take for a given individual. Doing IK with dual numbers is more time consuming than computing analytic derivatives (with cross products) in this specific example, and is closer to hard than medium.	2			5	8			
708904	0	0	0															
708920	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
708933	0	0	0															
709291	9.5	9.5	0	1	2	2	4	0.5		R5: not normalized								
709628	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
710086	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
710497	14.5	9.5	5	1	2	2	4	0.5		R5: normal not normalized. Wrist joint behavior strange	2			3				
710743	10	10	0	1	2	2	4	1		R2: typo: rotz not used (-0p)								
710976	4.5	4.5	0	1	2	1.5				R3: right, up, and ahead are the cols of t, not the rows								
711182	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
711467	20.5	9.5	11	1	2	2	4	0.5		R5: normal not normalized	2		4	5				
711551	10	10	0	1	2	2	4	1		For loops are allowed in GLSL shaders								
711810	0	0	0															
711904	0	0	0															
712550	10	10	0	1	2	2	4	1		SSD probably slow due to push_back which causes allocations (you can combine the loops and get rid of the vectors)								
712686	0	0	0															

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
712819	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
712958	0	0	0															
713672	0	0	0															
714985	7	7	0	1	2	2	2			R4: You need w=1 when computing newpos. You also need to use references (auto &j : joints_) or index access in computeToBindTransforms() (currently changes are discarded)								
716080	9	9	0	1	2	2	4											
716462	8.5	8.5	0	1	2	1	4	0.5		R4: using rows instead of cols (frames rotate the wrong way). R5: not normalized								
716718	0	0	0															
716860	12	10	2	1	2	2	4	1		GPU SSD: first line of first loop has typo. Also, result and result2 need to be zero-initialized (both CPU and GPU)	2							
717377	7	7	0	1	2	2	2			R4: to_bind_joint should not be inverted in getSSDTransforms(). Also, the position vector in computeSSD() needs w coordinate 1								
717474	0	0	0															
717513	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
717539	0	0	0															
718020	0	0	0															
718871	0	0	0															
722427	0	0	0															
723691	9	9	0	1	2	1.5	4	0.5		R3: right, up, and ahead are cols of T, not rows. R5: normal not normalized								
723905	0	0	0															
728667	15.5	10	5.5	1	2	2	4	1		Wrist joint shrinks when rotated, behaves a bit strangely	2			3.5				
728900	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized	2							
729132	9	9	0	1	2	2	4	0		R5: normal transformed like point (w=1), not normalized								
729967	4.5	4.5	0	1	2	1.5				R3: Right, up, and out are the cols of T, not the rows								
730309	0	0	0															
732080	0	0	0															
732255	0	0	0															
732323	0	0	0															
732336	0	0	0															
732352	0	0	0															
732459	18.5	10	8.5	1	2	2	4	1		Remember to normalize normal in GPU-SSD too. Severe artifacts in wrist joints	2	3		3.5				
76509T	15.5	10	5.5	1	2	2	4	1		GPU SSD: normal not normalized.	1.5		4					
765510	19	10	9	1	2	2	4	1			2	3	4					
766331	0	0	0															
767042	0	0	0															
767136	8.5	8.5	0	1	2	1	4	0.5		R3: right, up, and ahead computed correctly but not used. R5: normal not normalized								
768504	0	0	0															
769396	12	10	2	1	2	2	4	1			2							
77388B	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized.								
779124	0	0	0															

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
780058	7	7	0	1	2	2	2			R4: almost there: you're missing an inverse in computeToBindTransforms(), you also have a bug in the SSD loop where the position accumulates forever without resetting								
780346	9.5	9.5	0	1	2	2	4	0.5		R5:normal not normalized								
782917	0	0	0															
783563	0	0	0															
783709	0	0	0															
786667	0	0	0															
78708M	0	0	0															
787543	0	0	0															
787640	0	0	0															
788380	0	0	0															
788678	0	0	0															
791982	0	0	0															
795700	0	0	0															
795755	0	0	0															
796039	0	0	0															
804183	11	9.5	1.5	1	2	2	4	0.5		R5 + GPU SSD: normal not normalized.	1.5							
829155	0	0	0															
838191	0	0	0															
83873J	0	0	0															
84308F	8.5	8.5	0	1	1	2	4	0.5		R5: normals not normalized.								
84858E	0	0	0															
848754	24	9.5	14.5	1	2	2	4	0.5		General: having problems with uninitialized values in release mode. R5: normal not normalized. IK: almost impossible to control. Wrist joint: severa artifacts	2	3		3.5	6			
875170	10	8	2	1	2	0	4	1		R3: not visualizing local frame, but rather world-space coordinate axes (you need right, up, ahead from transformation matrix' columns)	2							
875251	9.5	9.5	0	1	2	2	4	0.5		R5: Normal not normalized								
875303	0	0	0															
875617	0	0	0															
876399	0	0	0															
877107	0	0	0															
877152	16.5	9.5	7	1	2	2	4	0.5		R5: Normal not normalized	2			5				
878591	16.5	9.5	7	1	2	2	4	0.5		R5: normal transformed like point (w=1)	2			5				
878627	11	9	2	1	2	2	4	0		R5: normal transformed incorrectly (using just inv instead of inv-transpose, using w=1 instead of w=0, not normalized)	2							
878889	7.5	7.5	0	1	2	0	4	0.5		R3: nothing drawn. R5: Normal not normalized								
879105	11.5	9.5	2	1	2	1.5	4	1		R3: right, up, and ahead are all found in the columns (typo?)	2							
882134	0	0	0															
885128	9.5	9.5	0	1	2	2	4	0.5		R5: Normal not normalized								
886648	10.5	9	1.5	1	2	2	4	0		R5: normal transformed like point (w=1), not normalized. Wrist: modifying weights (main challenge) not done				1.5				
889645	0	0	0															
892292	0	0	0															
898351	8.5	8.5	0	1	2	2	3	0.5		R4: position needs w=1. R5: Normal not normalized								

Student number	point total	req total	extra total	R1 Joint pos (1p)	R2 Joint rot (2p)	R3 Joint coord frame (2p)	R4 SSD (4p)	R5 normal skinning (1p)	mod	notes	SSD on GPU (2p)	animation (3p)	dual quaternion skinning (4p)	wrist joints (5p)	IK (8p)	other skinned models (5p)	other (put points here)	what other extras?
899130	13.5	9.5	4	1	2	2	4	0.5		R5: Normal not normalized. Wrist: quite a lot of unnatural warping in forearm				4				
900016	0	0	0															
901170	0	0	0															
901196	0	0	0															
913249	0	0	0															
913333	0	0	0															
913346	0	0	0															
913566	12	10	2	1	2	2	4	1		SSD-GPU: remember to zero-initialize accumulation variables	2							
915221	0	0	0															
915250	0	0	0															
917863	12	10	2	1	2	2	4	1			2							
918150	0	0	0															
918228	0	0	0															
918257	0	0	0															
918309	11.5	9.5	2	1	2	2	4	0.5		R5: Normal not normalized	2							
918396	9.5	9.5	0	1	2	2	4	0.5		R5: Normal not normalized								
918464	11.5	9.5	2	1	2	2	4	0.5		R5: normal not normalized. Yes, sorry about the late lecture videos. Looking through materials beforehand to get started early is always a good idea regardless of video schedule	2							
918671	9.5	9.5	0	1	2	2	4	0.5		R5: normal not normalized								
918875	9.5	9.5	0	1	2	2	4	0.5		R5: Normal not normalized								
930484	6.5	6.5	0	1	1	2	2.5			R2: you want canonical unit rotation vectors, pos not needed for those (just use coord. value 1). R4: computeToBindTransforms() missing reference in loop, inversion never applied. Also, SSD loop discards translation (fourth column) of transformation matrix								
932440	11	9	2	1	2	2	4	0		R5: normal transformed like point (w=1), not normalized	2							
935625	5	5	0	1	2	2												
939375	0	0	0															
942618	15	10	5	1	2	2	4	1			2	3						
k28342	0	0	0															