TIEA311 Tietokonegrafiikan perusteet
kevät 2019

("Principles of Computer Graphics" – Spring 2019)

TIEA311 Tietokonegrafiikan perusteet – kevät 2019 ("Principles of Computer Graphics" – Spring 2019)

Adapted from: *Wojciech Matusik*, and *Frédo Durand*: 6.837 Computer Graphics. Fall 2012. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu/.

Frontpage of the local course version, held during Spring 2019 at the Faculty of Information technology, University of Jyväskylä:
`http://users.jyu.fi/~nieminen/tgp19/`

TIEA311 - Today in Jyväskylä

Plan for today:

- (Possible announcements or food-for-thought)
- Usual warm-up and group discussion
- Try to address the most urgent issues
- Break – reset the brain.
- Then continue with the theory.

TIEA311 - Today in Jyväskylä

We start by discussion, reflection and **questions**!

Work in groups of 3 students if possible:

- Fast warm-up: 90 seconds evenly split between group members (30s each in groups of 3), no interruptions from others: Foremost feelings right now?
- Reflection: Silent work, solo, 1 minute, **list words on paper**: What have you learned during the last week? Or since the course started?
- Interaction: 1.5 minutes group discussion: Compare if you learned the same or different things? Do those things feel useful? Why or why not?
  → Sum it up classwide.
- Interaction: Group work, 1.5 minutes or less if talk ends: At the moment, what would be the most helpful thing to help you (or others!)?
  → Sum it up classwide, and try to address the findings.

TIEA311 - Today in Jyväskylä

What were the findings in group discussion?

What were found to be the most important issues to address right now?

$\rightarrow$ Classwide discussion is found on the lecture video.

NOTE: Even if you watch at home, please think about the same things and try to be in ”virtual dialogue” with those in classroom. Use pen and paper! I believe, more and more every day, that doing so will make your brain perform activities that help **your own learning**.

NOTE: Contemplate if you could watch the lecture videos with some friends who would also like to learn computer graphics? Get some pizza and coke if it helps you get to the mood(?).

TIEA311 - Today in Jyväskylä

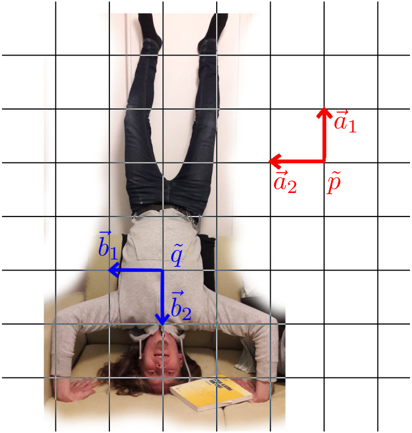
Plan for today:

- (Possible announcements or food-for-thought)
- Usual warm-up and group discussion
- Try to address the most urgent issues
- Break – reset the brain.
- Then continue with the theory.

Assignment 0 aftermath: How to do it "right"

- The "right way" to implement OBJ reading?
- In programming, "the **best way**" is somewhat **ill-defined**.
- All you can (should) do is **better than your earlier code**.
- **One measure** of a "**good enough** way" is that the code is successfully used in real products
- But never forget **safety** (as the most important measure), **readability**, **maintainability**, **performance** (which only matters in selected places! Almost never "comes first"!)...
- That said, two alternatives from real software to read OBJ meshes:
  - `https://github.com/openscenegraph/OpenSceneGraph/blob/master/src/osgPlugins/obj/obj.cpp`
  - `https://github.com/davll/ICG_SSAO/tree/master/source/nv`
- (The former is properly open source; about the latter I'm not certain - it seems to originate in some Nvidia SDK ...)

TIEA311



"Midterm" revisited

Live exercise time!

→ See lecture video.

This is important.

Do this!

(non-Finnish ones need to cope with English slides from MIT that will summarize this later; make sure you go through the frame switches using pen and paper, not only looking at them!)

Change of basis

- Critical in computer graphics
  - From world to car to arm to hand coordinate system
  - From Bezier splines to B splines and back

- problem with basis change:
  you never remember which is M or M^{-1}
  it's hard to keep track of where you are

Change of basis

- Assume we have two bases $\vec{\mathbf{a}}$ and $\vec{\mathbf{b}}$
- And we have the coordinates of $\vec{\mathbf{a}}$ in $\vec{\mathbf{b}}$
- e.g.

$$\vec{a_1} = \begin{bmatrix} \vec{b_1} & \vec{b_2} & \vec{b_3} \end{bmatrix} \begin{bmatrix} M_{11} \\ M_{21} \\ M_{31} \end{bmatrix}$$

- i.e.

$$\vec{\mathbf{a}}^t = \vec{\mathbf{b}}^t M$$

- which implies $\vec{\mathbf{a}}^t M^{-1} = \vec{\mathbf{b}}^t$

Change of basis

- We have $\vec{\mathbf{a}}^t = \vec{\mathbf{b}}^t M$ & $\vec{\mathbf{a}}^t M^{-1} = \vec{\mathbf{b}}^t$
- Given the coordinate of $\vec{v}$ in $\vec{\mathbf{b}}$: $\vec{v} = \vec{\mathbf{b}}^t \mathbf{c}$
- What are the coordinates in $\vec{\mathbf{a}}$?

Change of basis

- We have $\vec{\mathbf{a}}^t = \vec{\mathbf{b}}^t M$ & $\vec{\mathbf{a}}^t M^{-1} = \vec{\mathbf{b}}^t$
- Given the coordinate of $\vec{v}$ in $\vec{\mathbf{b}}$: $\vec{v} = \vec{\mathbf{b}}^t \mathbf{c}$
- Replace $\vec{\mathbf{b}}$ by its expression in $\vec{\mathbf{a}}$

$$\vec{v} = \vec{\mathbf{a}}^t M^{-1} \mathbf{c}$$

- $\vec{v}$ has coordinates $M^{-1}\mathbf{c}$ in $\vec{\mathbf{a}}$
- Note how we keep track of the coordinate system by having the basis on the left

Regarding the “left notation” and changing frames, the local lecturer of TIEA311 was dealing with something he hadn’t really used before.

Questions he had to **ask himself**: How is the math being used here, conceptually? How are the computations done? How does this relate to what he had learned before (on previous instantiations of the present local course, on Linear Algebra of the math dept., and a nasty one, from a substandard “young person next door”-tutorial back in highschool days!)?

Specifically, the order of A and A^{-1} and what it means to “move from a frame to another” were puzzling. How should we interpret “moving” from $\vec{\mathbf{f}}$ to $\vec{\mathbf{a}}$?

Were the slides correct (always possible to contain mistakes)? And assuming they were (which is more likely), which part of the concept did he not yet fully understand?

So… what did the lecturer have to do in order to understand?

(guesses, anyone?)

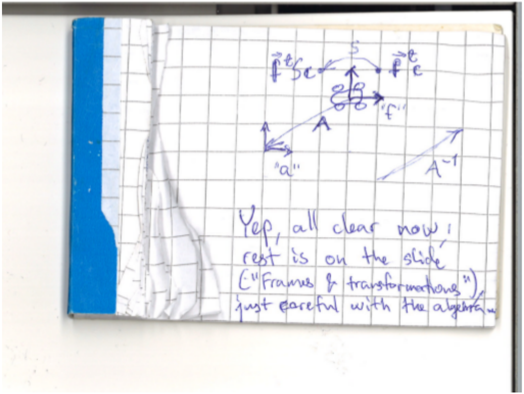
Take a **paper** and a **pen**, and use a **simple, concrete example** to **verify** that the equations **match the mental image**.

This time, it turns out that also the mental image needed to be adjusted (not much, but a little). This is called **learning**. It is painful, takes time, requires necessary tools (perhaps unique for everyone?), and then rewards.

Basic stuff. On the following slides, some scribbles from along the way.

Example: Back and forth between frames?

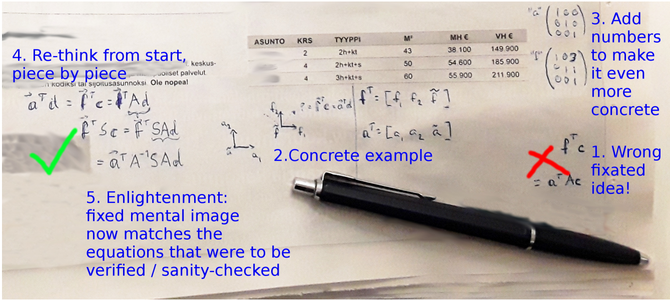
Pen and paper to help the brain (world + car + local origins and basis vectors + a point rotating in car frame):



Well, not yet enough.. provided only a momentary enlightenment that faded away overnight. . . followed an old fixation in thinking.

Example: Back and forth between frames!

More paper with whitespace, possibly same pen (very nice Ballograf one), re-start after thinking carefully about the slides from MIT:

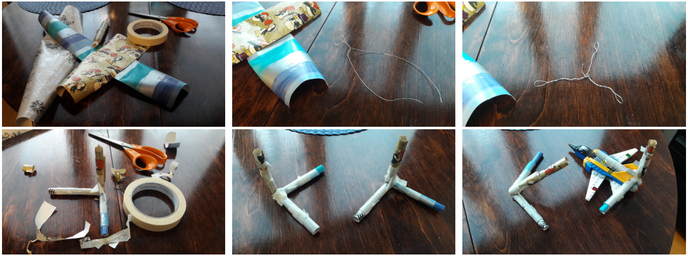


Finally, a corrected mental model of what is “a frame”, and “keeping track of the frame” as defined by the OCW slides. You **must** do this kind of stuff by **yourself** – in your **own way**!

(If you learn without, I think you have **superpowers** and should **go fight hostile aliens**, not waste those powers on IT studies)

TIEA311 Hobby Crafts Corner presents:

DIY Frame of Wire v0.1



The point being: Do whatever you need to do if you think it might help you understand things. . . Perhaps concrete and palpable artifacts (in ”true 3D”) work for you?

In 2019, I can tell you that it works for physicists: They seem to 3D-print scaled models of nanostructures to understand the chemistry invisible to the eye (or even microscope)!

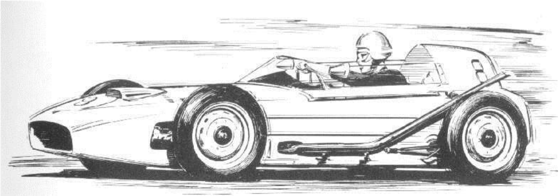
Recap

- Vectors can be expressed in a basis
  - Keep track of basis with left notation $\vec{v} = \vec{\mathbf{b}}^t \mathbf{c}$
  - Change basis $\vec{v} = \vec{\mathbf{a}}^t M^{-1} \mathbf{c}$
- Points can be expressed in a frame (origin+basis)
  - Keep track of frame with left notation
  - adds a dummy 4th coordinate always 1

$$\tilde{p} = \tilde{o} + \sum_i c_i \vec{b}_i = \begin{bmatrix} \vec{b}_1 & \vec{b}_2 & \vec{b}_3 & \tilde{o} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 1 \end{bmatrix} = \vec{\mathbf{f}}^t \mathbf{c}$$

Frames & hierarchical modeling

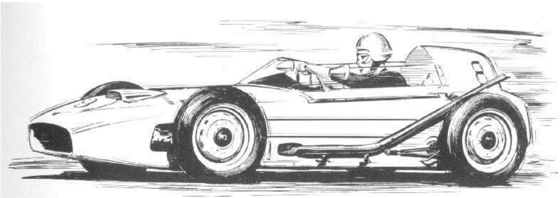
- Many coordinate systems (frames):
  - Camera
  - Static scene
  - car
  - driver
  - arm
  - hand
  - ...



- Need to understand nested transformations

Frames & hierarchical modeling

- Example: what if I rotate the wheel of the moving car:
- frame 1: world
- frame 2: car
- transformation: rotation



Linear algebra is friendly and simple **after the initial pain** of learning it. (this slide is transcripted from MIT OCW originals; I think the matrices got inversed A vs A^{-1} w.r.t. our lecture example in Finnish. But that is the point: we learn how to re-learn this any time we need to!)

Some transformation is specified by a matrix S in "car" frame $\vec{\mathbf{f}}$ as $\vec{\mathbf{f}}^t\mathbf{c} \rightarrow \vec{\mathbf{f}}^t S\mathbf{c}$.
How is the world frame $\vec{\mathbf{a}}$ affected by this?

- Frame can be interchanged with matrix and inverse: $\vec{\mathbf{f}}^t = \vec{\mathbf{a}}^t A^{-1}$ and $\vec{\mathbf{a}}^t = \vec{\mathbf{f}}^t A$.
- Coordinates transform too: $\vec{\mathbf{a}}^t\mathbf{d} = (\vec{\mathbf{f}}^t A)\mathbf{d} = \vec{\mathbf{f}}^t(A\mathbf{d})$ and $\vec{\mathbf{f}}^t\mathbf{c} = (\vec{\mathbf{a}}^t A^{-1})\mathbf{c} = \vec{\mathbf{a}}^t(A^{-1}\mathbf{c})$.
- So, start from transformation given in $\vec{\mathbf{f}}$:
  $\vec{\mathbf{f}}^t\mathbf{c} \rightarrow \vec{\mathbf{f}}^t S\mathbf{c}$
- Plug in the above expressions. Transformation then reads: $(\vec{\mathbf{a}}^t A^{-1})(A\mathbf{d}) \rightarrow (\vec{\mathbf{a}}^t A^{-1})S(A\mathbf{d})$
- Rearrange parentheses: $\vec{\mathbf{a}}^t(A^{-1}A)\mathbf{d} \rightarrow \vec{\mathbf{a}}^t(A^{-1}SA)\mathbf{d}$
- Rid of identity matrix: $\vec{\mathbf{a}}^t\mathbf{d} \rightarrow \vec{\mathbf{a}}^t(A^{-1}SA)\mathbf{d}$. Done!

Linear algebra is friendly and simple **after the initial pain** of learning it. (this slide **uses the notations we created together** during the Finnish lecture example! And this, if anything, proves the main point: we **have learned how to re-learn** and **verify** this any time we need to!)

Some transformation is specified by a matrix R in ”dude” frame $\vec{\mathbf{b}}$ as $\vec{\mathbf{b}}^t\mathbf{c} \to \vec{\mathbf{b}}^t R\mathbf{c}$.
How is the world frame $\vec{\mathbf{a}}$ affected by this?

- Frame can be interchanged with matrix and inverse: $\vec{\mathbf{b}}^t = \vec{\mathbf{a}}^t A$ and $\vec{\mathbf{a}}^t = \vec{\mathbf{b}}^t A^{-1}$.
- Coordinates transform too: $\vec{\mathbf{a}}^t\mathbf{d} = (\vec{\mathbf{b}}^t A^{-1})\mathbf{d} = \vec{\mathbf{b}}^t(A^{-1}\mathbf{d})$ and $\vec{\mathbf{f}}^t\mathbf{c} = (\vec{\mathbf{a}}^t A)\mathbf{c} = \vec{\mathbf{a}}^t(A\mathbf{c})$.
- So, start from transformation given in $\vec{\mathbf{f}}$:
  $\vec{\mathbf{f}}^t\mathbf{c} \to \vec{\mathbf{f}}^t R\mathbf{c}$
- Plug in the above expressions. Transformation then reads: $(\vec{\mathbf{a}}^t A)(A^{-1}\mathbf{d}) \to (\vec{\mathbf{a}}^t A)R(A^{-1}\mathbf{d})$
- Rearrange parentheses: $\vec{\mathbf{a}}^t(AA^{-1})\mathbf{d} \to \vec{\mathbf{a}}^t(ARA^{-1})\mathbf{d}$
- Rid of identity matrix: $\vec{\mathbf{a}}^t\mathbf{d} \to \vec{\mathbf{a}}^t(ARA^{-1})\mathbf{d}$. Done!

Those who saw the lecture 13 of TIEA311 Spring 2019 either live or on video witnessed the following:

- Insecure teacher, in panic, trying to figure out if he got it right this time (after two consecutive years of failing the first attempt at explaining this bit) or not.
- The **effect** of panic and extreme **deadline pressure** on somebody who thinks he can do this thing any time, and (seemingly) can, too:). Circumstances matter.
- In the end, there was ultimately no mistake, but uncertainty was acknowledged. And **that** is **the main ingredient**, folks!!

Learnings to take home:

- This stuff is easy. . . but only **after getting it right** and being **sure** it was right.
- It is **necessary to doubt everything**, starting especially from yourself. The same in all math **and programming**!
- Finally: math (and software!) does not lie. It works or it doesn't, and there is a reason. It **can be verified/tested**.

Further necessary exercises:

Compute the thing with actual matrices, using the power tools of pen and paper, and verify it works for a simple transform, like the 90 degree rotation we did together on Finnish lectures.

Celebrate the “magic” of mathematics that **you can now perform**: the **algebraic equation** we sort of **found** is valid for **any** affine transform and **any** two frames!

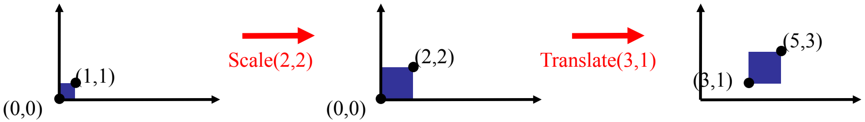
Think about how you can follow either the transformations of the frame (multiply frame from right), or transformations of the coordinates (multiply coordinates from left) one-by-one and end up with the **same result**. Real-world objects may help the brain.

Observe that after computing either way, **there is finally only one** matrix $M = ARA^{-1}$ that performs the whole transform.

This is the same for **any number of combined transforms**!

How are transforms combined?

Scale then Translate



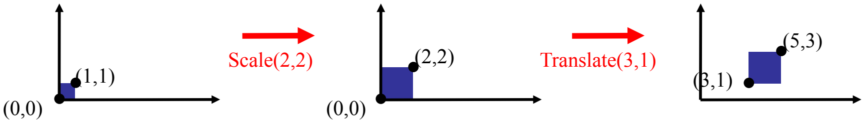
Use matrix multiplication: p' = T (S p) = TS p

$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

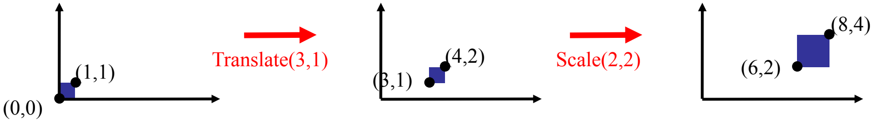
Caution: matrix multiplication is NOT commutative!

Non-commutative Composition

Scale then Translate: $p' = T(Sp) = TSp$



Translate then Scale: $p' = S(Tp) = STp$



Non-commutative Composition

Scale then Translate: p' = T (S p) = TS p

$$TS = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Translate then Scale: p' = S (T p) = ST p

$$ST = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

TIEA311 - Today in Jyväskylä

The time allotted for this lecture is now over.

Now: Break until tomorrow morning. Sleep if you have time.

But also **try to wake up and come to the lecture**!

We will pick up our thoughts soon enough!