Implementation of an On-Line Multimedia Collaborative Linear Algebra Tutoring System

Carlos R. Morales, Nishant Kothary
Purdue University, Knoy Hall, Room 363, West Lafayette, IN, 47907

Abstract

Few courses undertaken by students in their first two years of engineering education are as abstract as linear algebra. The content is not only critical for future success, but considerably difficult at the same time, because most students are unfamiliar with abstract mathematical concepts at that point in their lives. Many courses around the nation augment their traditional instruction by exposing their students to computer algebra systems (CAS) such as MatLab, Mathematica, MathCad, Maple, etc. While these tools are great for analysis, they are not designed to be primary instructional tools. These tools favor those students that have a stronger theoretical understanding of the concepts, but do little to help them develop a better understanding of these. The author presents the development of an online multimedia collaboration tool that actively teaches the students basic linear algebra concepts. The system utilizes non-linear broadcast quality video and interactive 3D graphics in a multi-user environment. A pseudo intelligent mechanism analyzes the learner's choices and modifies instructional delivery based on a predetermined instructional-objectives map. The system in not meant to replace classroom instruction or utilization of computational CAS systems. Its primary purpose is to provide supplemental tutoring aid while fostering visualization skills in the learner.

Introduction and background

In order for students to succeed in technology or engineering they must master higher-level mathematical concepts. In Computer Graphics Technology mastery of Linear Algebra is critical.

A difficulty that many students encounter is the manner in which mathematics is taught at most of the larger universities. At Universities such as Purdue, students go to large lectures of 400 to 500 students. Lectures are usually supplemented by "recitation" sessions which have class sizes of 20-30 students.

The lack of immediate feedback and adequate resources for students is often a problem. In a large class, the availability of instructors is often limited. The materials that are available to the students for self-help are of utmost importance. Mastery of concepts often comes down to a rigid homework schedule that forces students to methodically work through important problem sets. At this point in time, most students are still unfamiliar with the language of Mathematics. Thus refering to a second book is often fruitless. For example, if a student is confused by an explanation for Taylor polynomials supplied in James Stewart's book, he has the option of looking into Thomas/Finney's book for a slightly different or more eludicating perspective on the same concept[1][2]. In reality, this does not work for engineering/technology students because they do not know the language of mathematics. Mathematics books are unlike books in the social sciences, where the same subject matter might be approached from slightly different perspective. Mathematicians write with a very terse and compact tone that must be mastered if one is to understand the book on one's own.

Other approaches tried by students are using rapid self-help manuals such as those offered by Schaum's and R.E.A. or to look on the Internet for video lectures, notes, power point presentations, or anything else that might help.

While both of these approaches are valid, they are not useful to all students. Rapid self-help manuals show the mechanics of how to work out problems, but do not explain the concepts needed for the student to succeed. Thus, if a student does grasp the concepts during regular lecture time, he/she is probably not going to pick them up here.

There are very few sources of well-developed instructional materials for higher-mathematics on the Internet. With the exception of Gilbert Strang's Linear Algebra lectures, the sites that I have found on-line were of poor quality both from an instructional and presentational perspective [3]. Most of the flaws stemmed from poor instructional design, poor graphics, and video clips that did not fully exploit the power that comes from combining top-multimedia [animation, video, interactive 3D graphics, etc.] to create a compelling instructional presentation. Even Strang's lecture could use some improvement. Instructionally they are effective, but do not fully exploit streaming video/visualization graphics.

Solution

From a bird's eye view, the solution is simple. We need to provide two things. The first thing is feedback. We need provide the students with immediate feedback or with the resources so the student can pursue that feedback at his own convenience. The student must have a way to check if the methods he/she is using are effective.

The second provision to be made is variety in how the subject matter is presented. If a student does not understand a concept from Stewart's book, he/she is unlikely to understand it from Thomas/Finney because the presentation of the subject matter is essentially the same. If we provide the students with a bank of lectures that present the same subject matter, but from different perspectives, then we would facilitate learning.

Technical solution

This paper details an intelligent educational system developed at the CGT department at Purdue University to serve as a supplemental tool used by the students. The system can deliver asynchronous DVD-quality video lectures while actively testing the student's knowledge against a set of pre-determined objectives and dynamically altering the delivery of instruction.

The first set of lessons built into the system helps to teach Linear Algebra concepts with a Computer Graphics Technology application bias. Let's now explore the development of the system.

Scenario for use

Students majoring in computer graphics technology at Purdue University do not have to formally take a class in linear algebra. Yet, senior level classes presume some rudimentary knowledge of the subject. Students are expected to know how to determine basis, linear dependence, perform matrix operations, etc.

In the past, the author dealt with this knowledge gap by integrating some of this instruction into his classes. For example, before addressing the RGB three-dimensional color model, the author would spend a lecture covering the concepts of linear dependence and basis.

In creating this system, the author offloaded the instruction of these mathematical concepts to the software. Thus, the students did receive live instruction in linear algebra prior to actually utilizing that knowledge in class. Instead, they were told to use the tool to fill their gaps in knowledge.

Instructional base

In order for the system to successfully deliver intelligent media, it must:

(1) have some knowledge of the content that it is delivering at the time of the presentation
(2) have a way to measure the students' performance
(3) prescribe changes in instructional delivery based on the student's performance

The AI portion of the program is based on Russell and Norvig's definition for an ideal rational agent. Russell and Novig tell us that "for each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has."[4] Thus, our system does not need to know the content matter to teach it intelligently, but it must be able to measure the student's performance against a preset of performance measures we specify with our objectives.

Three things are needed for this methodology to work: (1) a model of the learner, (2) a model of the lesson, and (3) a model of the interaction.

Model of Lesson

- Lesson Step
- Instructional Objective
- Instructional Objective Measure
- Measure
    - Filename
    - Threshold
    - Remediation

- Video Clips
- Filename
- Bias
- Sub-Objectives
- Common Mistake
- Mistake

- Identification
- Remediation

Each lesson is made up of steps. Typically one would think of steps only in a lesson that is procedural. Here a step refers to the current location within the educational presentation. For example, in teaching an attitudinal lesson, the steps wouldn't necessarily be dependent on each other. But one would have to select the topics he wishes to present and order them in a sequence. In this context, a step is a temporal measure of the topics presented.

Instructional objective behavior and the objective measure fields are used to both document the desired behavioral outcome and set an expected level of performance. We can have multiple measures of success/failure of the specified objective. Each measure is delivered as a standalone file --- this was done to allow for unlimited combination of interactivity. We also associate a threshold with each objective. Thus we can ask for varying levels of mastery before allowing the student to proceed. We also have a specific remediation recommendation at each step that can be used by the system in helping the student. In a nutshell, this part of the lesson model provides the data that the system needs in order to select the appropriate course of action during instructional delivery.

Next, our lesson model associates the video clips with this particular step. There are multiple video clips for each step that teach the instructional objective. We also associate a bias with each video clip. Note the bias utilizes user-defined parameters. This allows us to have as many or as few video clips and associations as necessary.

Sub-objectives allow the system to know which objective must be mastered by the student prior to being successful at the current objective. Note that some of the sub-objective might in fact be entry behaviors and not link to any remediation media assets.

The final part of the lesson model is used to endow the system with more than just the capability to redirect the user to alternate media clips based on the performance on a single measure. Here we provide the system with the common mistakes made by learners and some prescriptive changes as well. Let's consider the following example:

> The system is delivering instruction on the calculation of eigenvectors & eigenvalues. In the lesson model, we indicate that two common problems for students are the calculations of determinants and solving a homogenous system of linear equations. During the instructional delivery, the system can compare the weaknesses identified in the learner's model and with anything identified in these common mistakes. It can then make appropriate suggestions. The system does more than just compare these two. It creates a graph and identifies possible weaknesses based on previous performance. For example, in the case we just discussed, the system would implicitly know the student will have difficulty solving dynamical systems of differential equations even though there is no explicit data concerning the student's ability to find eigenvectors / eigenvalues in the learner model.

Learner model, interaction model and instructional Delivery

The learner model contains information about the learner's past performance as well as general demographical information. The most important aspect of the learner model is that it allows the system to any strengths and weaknesses identified to a specific user.

The interaction model is created during run-time and is used to make instructional presentation decisions. By combining the data from the learner and lesson models along with the behavior of the user, the system is able to create a model of how the user is interacting with the system. It is this interaction model that is used by the system to make intelligent decision.

The interaction model tracks the following: user's position within a lesson, common mistakes made by the user, number of times a student asks for extra help, and looks for relationships in the types of error made by the user.

The charts below detail the sequence of events that occur within the system during instructional delivery and the commands available to the learner.

Instructional Delivery

1. Learner logs onto system and selects a lesson
    a. Learner model is loaded
    b. Summary of available lessons are loaded
2. Student selects lesson and step to start from
    a. Specific lesson model is loaded
    b. Interaction model is set to initial values
3. Iterate though steps
    a. Interaction model is queried for appropriate video to use
    b. Video is displayed at resolution appropriate for receiving device
4. Randomly insert embedded question
5. Compare to performance to threshold
    a. Performance is classified into success or failure for current objective.
    b. Video for sub-objectives is ignored, offered, or forced upon the learner dependent on their performance.
6. Interaction Model is updated
    a. Interaction model is updated to reflect performance.
    b. Any common mistake conducted by user is highlighted in the interaction model.
7. Go to step 3 until done with lesson
8. User model is updated

Figure 1. Instructional Delivery Methodology

Commands Available to Learner

- Show Objectives / Objective map
- Pause

- Contact instructor
- Tell instructor I'm having problems
- Review sub-objective
- Show common test items
- Switch Language
- Interactive Activities
- Download to PDA

Figure 2. Software Commands

Setting up a lesson

Many users associate this system with a distance-learning delivery system because it has many of the same capabilities. It can deliver instructional materials including asynchronous video over the Internet. From the learner's perspective, this similarity holds.

Anyone setting up an instructional unit in this system will find that there additional instructional and media requirements that need to be supplied for the system to work properly. It is not enough to supply the system with video clips and questions to ask the students. The instructor must supply an extremely detailed instructional objectives map that serves as the backbone for the both instructional delivery and assessment phase of instruction. The user must supply assessment items that measure those objectives and supply the system with the measures and relationships between those objectives and the route for instructional delivery. Finally, the instructor must supply multiple discreet video clips for each instructional objective that address specific weaknesses within the learner. Without these pieces, the system cannot behave intelligently.

The system has been set up to facilitate the creation of these materials. When setting up a lesson, the instructor uses a wizard within the application to indicate the instructional objectives. He/she is then queried to enter the assessment items and measures. Finally, he/she is asked to import the video clips.

Preparation of the media

A television production model was used in creating the lessons. Lessons on linear dependence, basis, and vector spaces were prepared. A script was created for each lesson and the instructor was video-taped in front of a bluescreen. Then three-dimensional graphics were created and chroma-keyed onto the footage.

Figure shows one of the graphics created to illustrate the RGB color model with linearly dependent vectors.

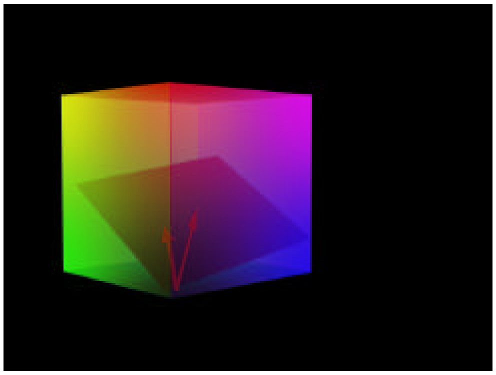

Figure 1. RGB Color Cube with Vector

Below is the script that links linear dependence to the RGB color model.

RGB Color Space Script

Video	Audio
	Color can be represented in many ways. One such ways is by considering each color to be made up of 3 independent colors ---- red, green and blue. Thus each color can be created from the primary colors red, green, and blue. We say that red, green, and blue are linearly independent because there is no way to create of these colors from the other two. That is, there is no way to create the color green from the colors blue and red. And likewise with the others. Thus each of these primary colors is independent from the others. Because each color is made up from combining these three independent colors, we can think of colors as being vectors in R^3 with each axis representing one of the primary colors. Most graphics systems allocate 8 bits to each color component, which gives us 256 levels of each color. We will therefore adopt a scale that ranges from 0 to 255 for each of our color components. The scale we use is not important as long as it is consistent for all of the axes. We adopt this scale to be consistent with most graphical systems. Now that we have a way to represent color, let's plot a

few colors.

The color black is <0,0,0> and is located at the orgin

The color white is created by setting each of the color components to its maximum. Thus white is located at <255,255,255>

Pure red is located at <255,0,0>
Pure green is found at <0,255,0>
Pure blue can be found at <0,0,255>

Lets also plot the subtractive primary colors.

Cyan is the absence of red. Thus its coordinates are <0,255,255>
Yellow is the absence of blue. Its coordinates are <255,255,0>
Magenta is the absence of green. Its coordinates are <255,0,255>

We refer to this as the RGB color cube.

An advantage to this representation is that all of our knowledge of linear algebra can be applied to answering questions.

Let's first consider the following question.

--
Question: Can the color <90,135,137> be created from the colors <25, 125, 132> and <130, 20, 10> ?

To solve this problem we need to check if v:<90, 135, 137> is linearly dependent on v1: <25, 125, 132> and v2:<130, 20, 10>.

We seek to find two constants a and b such that av1 + bv2 = v. If such a combination exists then v can be created from v1 and v2.

Geometrically we want to check if the v lies in the plane spanned by v1 and v2.

The easiest way to check this is to construct a matrix as follows. We put in our vectors in our columns.

	First we put in v1 and v2. Finally v. We reduce our matrix. If we get a consistent system, then our vector is linearly dependent and it is in fact our color can be created from the other two colors. In this case, our color can be created by adding color 1 to ½ of color 2. We can readily ready this from our matrix. Let's now consider another question: **Question 2:** What are all of the colors that can be created from the colors <10,25,35>, <50,21,16>, and <130, 117, 137>? We are being asked to find a basis for the given subspace. It is tempting to think that because we have three colors that we can generate all colors in R3, but this is not necessarily the truth. Linearly dependent To solve this problem we place the vectors into the columns of our matrix. We then reduce the matrix. Any leading 1's in our columns correspond to the original vectors. In this case we have ones in columns 1 and 2. Thus the vectors <10,25,35> and <50,21,16> form a basis for our subspace. All the colors that can be generated will be a combination of these two colors and will be of the form a*<10,25,25> + b*<50,21,16> Because or scale goes from 0 to 255 any values of a that produce values above 255 will be out of our range.

Conclusion and future work

Overall the system worked as expected. It successfully delivered instructional materials in an intelligent manner. As long as the system was supplied with the appropriate instructional objective maps and accompanying instructional materials, the system performed quite well, and served as an aid to students who needed to augment their understanding of mathematics.

From an instructional perspective, the system was also able to fulfill its role an offline mathematical instructional tool. After deploying the system, the author was able to deliver instruction in his courses that required the instructional content delivered by the system without having to take time out from class to remediate the students' math skills. During informal conversations, the students felt the system allowed them to quickly look up the skills they needed in order to engage in formal class material in a manner that was directly applicable to their focus of study.

As designed the system is meant to supplement traditional classroom mathematical education. It would be interesting to expand the purpose of the system to include primary instructional delivery and validate its effectiveness against traditional classroom instruction. As of this point, this is outside of the scope of this project.

From a technological perspective, the system can be improved by incorporating video production functionality within the system. As it is currently configured, everything except for the actual production of the video assets is included within the system. The system is built on top of Microsoft technology. Using the video editing services within DirectX this should be possible to integrate into the system [5].

Another area to explore is to enhance the intelligent capabilities of the system. As is, the system accomplishes all of the tasks by using simple searching. If we restrict the domain of the system to a specific subject matter, then it might be possible to build a knowledge base of that subject matter within the system and then base decision off that knowledge base.

In closing, the applicability of this system as an aid to mathematical education in a technical program has worked out well.

Bibliography:

[1] Stewart, J. (1999). Calculus: Early Transcendental (4th Edition). Brooks/Cole Publishing. Pacific Grove, CA. ISBN: 0-534-36298-2

[2] Finney, R., Thomas, G. (1996). Calculus and Analytic Geometry (9th Edition). Addison Wesley Publishing. Boston, MA. ISBN: 0-201-53174-7

[3] Strang, G. On-line Video lectures. Massachussettes Institute of Technology. http://ocw.mit.edu/18/18.06/f02/video-lectures/index.html

[4]Russell, S., Norvig., P. (2003). Artificial Intelligence: A Modern Approach, Prentice Hall, Upper Saddle River, NJ. ISBN: 0-13-790395-2

[5] Active-X specifications. (2002) Microsoft Corporation. [On-line] . Available: http://www.microsoft.com/com/tech/activex.asp

CARLOS R. MORALES
Carlos R. Morales is an assistant professor of Computer Graphics Technology at Purdue University. He holds a BA in Telecommunications and an MS Ed. in Curriculum and Instruction. Prior to working at Purdue University, Carlos worked as a Technical Director. His research interest includes distance learning, animation, and multimedia development. He can be reached at crmorales@tech.purdue.edu

Nishant Kothary

Nishant is a freelance multimedia developer specializing in web design. He teaches courses at Purdue University dealing with the fundamentals of scripting and tagging, and currently serves as web developer for the Office of Technology Commercialization at Purdue University. Due to his background in computer science, mathematics and computer graphics, Nishant has been able to develop a progressive approach to the implementation of kinematics and mathematics in ActionScript. Nishant can be contacted at kothary@purdue.edu.