OK. So, coming nearer the end of the course, this lecture will be a mixture of the linear algebra that comes with a change of basis.

And a change of basis from one basis to another basis is something you really do in applications.

And, I would like to talk about those applications.

I got a little bit involved with compression.

Compressing a signal, compressing an image.

And that's exactly change-of-basis. And then, the main theme in this chapter is th- the connection between a linear transformation, which doesn't have to have coordinates, and the matrix that tells us that transformation with respect to coordinates.

So the matrix is the coordinate-based description of the linear transformation.

Let me start out with the nice part, which is just to tell you something about image compression.

Those of you -- well, everybody's going to meet compression, because you know that the amount of data that we're getting -- well, these lectures are compressed.

So that, actually, probably you see my motion as

jerky? Shall I use that word?

Have you looked on the web?

I should like to find a better word.

Compressed, let's say.

So the complete signal is, of course, in those video cameras, and in the videotape, but that goes to the bottom of building nine, and out of that comes a jumpy motion because it uses a standard system for compressing images.

And, you'll notice that the stuff that sits on the board comes very clearly, but it's my motion that needs a whole lot of bits, right?

So, and if I were to run up and back up there and back, that would need too many bits, and I'd be compressed even more.

So, what does compression mean?

Let me just think of a still image.

And of course, satellites, and computations of the climate, computations of combustion, the computers and sensors of all kinds are just giving us overwhelming amounts of data.

The Web is, too.

Now, some compression can be done with no loss.

Lossless compression is possible just using, sort of, the fact that there are redundancies.

But I'm talking here about lossy compression.

So I'm talking about -- here's an image.

And what does an image consist of?

It consists of a lot of little pixels, right?

Maybe five hundred and twelve by five hundred and twelve.

Two to the ninth by two to the ninth pixels, and so this is pixel number one, one, so that's a pixel.

And if we're in black and white, the typical pixel would tell us a gray-scale, from zero to two fifty five.

So a pixel is usually a value of one of the xi, so this would be the i-th pixel, is -- it's usually a real number on a scale from zero to two fifty five.

In other words, two to the eighth possibilities.

So usually, that's the standard, so that's eight -- eight bits.

But then we have that for every pixel, so we have five hundred and twelve squared pixels, we're really operating x is a vector in R^n, but what is n?

n is five hundred and twelve squared.

That's our problem, right there.

A pixel is a vector that gives us the information about the image.

I'm sorry.

The image that comes through is a vector of that length that -- that's the information that we have about the image, if it's a color image, we would have three times that length, because we'd need three coordinates to get color.

So it would be three times five hundred and twelve squared.

It's an enormous amount of information, and we couldn't send out the image for these lectures without compressing it.

It would overload the system.

So it has to be compressed.

The standard compression, and still used with lectures is, called JPEG.

I think that stands for Joint Photographic Experts Group.

They established a system of compression.

And I just want to tell you what it's about.

It's a change-of-basis. What basis do we have?

The current basis we have is, you could say, the standard basis is, every pixel, give a value.

So that's like we have a vector x which is five hundred and twelve squared long and, in the i-th position, we get a number like one twenty one or something.

The pixel next to it might be one twenty four, maybe where my tie begins to enter, so if it was mostly blue shirt, this would be a slight difference in shading, but pretty close, then the tie would be a different color, so we might have quite a few pixels for the blue shirt, and a whole lot more for the blackboard, that are very close.

And that's what are very correlated.

And that's what gives us the possibility of compression.

For example, before the lecture starts, if we had a blank blackboard, then there's an image, but it would make no

sense to take that image and tell you what it is pixel by pixel.

I mean, there's a case in which all pixel values, all gray levels are the same -- or practically the same, depending on the erasing of the board, but extremely close -- and, so that's an image where the standard basis is lousy.

That's the basic fact, that the standard basis which gives the value of every pixel makes no use of the fact that we're getting a whole lot of pixels whose gray levels -- the neighboring pixels tend to have the same gray level as their neighbors.

So how do we take advantage of that fact?

Well, one basis vector that would be extremely nice to include in the basis would be a vector of all ones.

That's not in our standard basis, so let me just write again, the standard basis is our one, and all the rest zeroes, zero, one, and all the rest, zeroes, everybody knows what these standard basis is.

Now, any other basis for R -- so this is -- for this very high-dimensional space -- now I'm going to speak about a better basis.

Better basis -- and let me just emphasize, one vector that would be extremely nice to have in that basis is the vector of all ones.

Why is that?

Let me just say again, because that vector of all ones, by itself, one vector is able to completely give the information on a solid image.

Of course, our image won't be solid, it will have a mix of solid and signal.

So having that one vector in the basis is going to save us a whole lot.

Now, the question is, what other vectors should be in the basis?

The extreme vector in the basis might be a vector of one minus one, one minus one, one minus one.

That would be a vector that shows -- I mean, that's like a checkerboard vector, right?

That's a vector that would, if the image was like a huge checkerboard of plus, minus, plus, minus, plus, minus, that vector would carry the whole signal.

But much more common would be maybe to have half the image, darker and the other half lighter.

So another vector that might be quite useful in here would be half ones and half minus ones.

I'm just trying to get across the idea of that a basis could be where, that first of all, we've got the bases at our disposal.

Like, we're free to choose that.

And it's a billion-dollar decision what we choose.

So, and TV people would rather pre- would prefer one basis based on the way the signal is scanned, and movie people would prefer another, I mean, there's giant politics in this question that really reduces to a linear algebra problem, what basis to choose.

I'll just mention the best known basis, which JPEG uses, -- let me put that here -- is the Fourier basis.

So when you use the Fourier basis, that includes -- this is the constant vector, the D C vector if we're electrical engineers, the l- vector of all ones, so it would include one, one, one, one.

Often eight by eight is a good choice.

Eight by eight is a good choice.

So, what do I mean by this eight by eight?

I mean that the big signal, which is five twelve by five twelve, gets broken down, and JPEG does this, into eight by eight blocks.

And we -- sort of, this is too much to deal with at once.

So what JPEG does is take this eight by eight block, which is sixty four coefficients, sixty four, pixels, and changes the basis on that

piece. And then, now, let's see, I was going to write down Fourier, so you remember Fourier as this vector of all ones, and then, the vector -- oh, well, actually, I gave a lecture earlier about the Fourier matrix, this matrix whose columns are powers of a complex number w.

I won't repeat that, because I don't want to go into the details of the Fourier basis, just to tell you how compression works.

So what happens in JPEG?

What happens to the video, to each image, of these lectures?

It gets broken into eight by eight blocks.

OK. Within each block, we have sixty four coefficients, sixty four basis vectors, sixty four pixels, and we change basis in sixty four dimensional space using these Fourier vectors.

Just note, that was a lossless step.

Let me emphasize.

In comes the signal x.

We change basis.

This is the basis change.

Change basis.

Choose a better basis.

So it produces, the coefficients c.

So sixty four pixels come in, sixty four coefficients come out.

Now comes the compression.

Now come -- this was lossless.

It's just -- we know that R -- R sixty four has plenty of bases, and we've chosen one.

Now, in that basis, we write the signal in that basis, and that's what my lecture -- that's the math part of my lecture.

Now here's the application part.

The next part is going to be the compression step.

And that's lossy.

We're going to lose information.

And what will actually happen at that step?

Well, one thing we could do is just throw away the small coefficients.

So that's called thresholding, we set some threshold.

Every coefficient, every basis vector that's not in there more than the threshold value, and we set them threshold so that our eye can't see the difference, or can hardly see the difference, whether we throw away that little bit of that basis vector or keep it.

So this compression step produces a compressed set of

I'll just keep going here. coefficients.

So it keeps going, this compression step produces some coefficient c hat.

And with many zeroes. So that's where the compression came.

Probably, there is enough of this vector of all ones -- we very seldom throw that away.

Usually, its coefficient will be large.

But the coefficient of something like this, that quickly alternative vector, there's probably very little of that in any smooth signal.

That's high-frequency -- this is low-frequency, zero frequency.

This stuff is the highest frequency we could have, and if the noise, the jitter is producing that sort of output, but a smooth lecture like this one is, has very little of that highest frequency, very little noise in this lecture.

OK, so we throw away whatever there is, and we're left with just a few coefficients, and then we reconstruct a signal using those coefficients.

We take those coefficients, times their basis vectors, but this sum doesn't have sixty four terms any more.

Probably, it has about two or three terms.

So that would -- say it has three terms.

From sixty four down to three, that's compression of twenty one to one.

That's the kind of compression you're looking for.

And everybody is looking for that sort of compression.

Let's see, I guess I met the problem with the FBI and fingerprints.

So there's a whole lot of still images.

You know, with your thumb, you make these inky marks which go somewhere. it used to go to Washington and get stored in a big file.

So Washington had a file of thirty million murderers, cheaters on quizzes, other stuff, and actually, there was no way to retrieve them in time.

So suppose you're at the police station, they say, OK, this person may have done this, check with Washington, have they got -- are his or her fingerprints on file?

Well, Washington won't know the answer within a week if it's got filing cabinets full of fingerprints.

So of course, the natural step is digitizing.

So all fingerprints are now digitized, so now it's at least electronic, but still there's too much information in each one.

I mean, you can't search through that many, fingerprints if the digital image is five twelve squared by five twelve squared, if it's that many pixels.

So you get compressed.

So the FBI had to decide what basis to choose for compression of fingerprints.

And then they built a big new facility in West Virginia, and that's where fingerprints now are sent.

So I think, if you get your fingerprints done now at the police station, if it's an up-to-date police station, it happens digitally, and the signal is sent digitally, and then in West Virginia, it's compressed and indexed.

And then, if they want to find you, they can do it within minutes instead of within a week.

OK.

So this compression comes up for signals, for images, for video -- which is, like these lectures -- there's another aspect.

You could treat the video as one still image after another one, and compress each one, and then run them and

make a video.

But that misses -- well, you can see why that's not optimal.

In a video thing, you have a sequence of images, so video is really a sequence of images but what about one image to the next image?

They're extremely correlated.

I mean that I'm getting an image every split-second, and also, I'm moving slightly.

That's what's producing the, jumpy motion on the video.

But I'm not, like, you know -- each image in the sequence is pretty close to the one before.

So you have to use, like, prediction and correction.

I mean, the image of me one instant -- one time-step later, you would assume would be the same, and then plus a small correction.

And you would only code and digitize the correction, and compress the correction.

So a sequence of images that's highly correlated and the problem in compression is always to use this correlation, this fact that, in time, or in space, things don't change instantly, they're very often smooth changes, and, you can predict one value from the previous value.

OK.

So those are applications which are pure linear algebra.

I could, well, maybe you'll allow me to tell you, and the book describes, the new basis that's the competition for Fourier.

So the competition for Fourier is called wavelets, and I can describe what that basis is like, say, in the eight by eight case.

So the eight by eight wavelet basis is the vector of all ones, eight ones, then the vector of four ones and four minus ones, then the vector of two ones, and two minus ones, and four zeroes. And also the vector of four zeroes and two ones and two minus ones.

So now I'm up to four, and I need four more, right?

For R^8? The next basis vector will be one minus one and six zeroes, and then three more like that, with the one minus one there, and there, and there.

So those are eight vectors in eight-dimensional space, those are called wavelets, and it's a very simple wavelet choice, it's a more sophisticated

choice. This is a little jumpy, to jump between one and minus one.

And, actually, you can see, now, suppose you compare the wavelet basis with the Fourier basis above.

How could I write this guy, which is in the Fourier basis, it's an eight -- it's a vector in R^8. How would I write that as a combination of the wavelet basis?

Have I told you enough about the wavelet basis that you can see, how does this very fast guy -- what combination of the wavelet basis is that very fast guy?

It would be this one -- it would be the sum of these four, right?

That very fast guy will be that one minus one, and the next one, and the next one, and the next one.

So this is the sum of those last four wavelets.

This one, we've kept, and so on.

So, each -- well, every -- well, that's what a basis does.

Every vector in R^8 is some combination of those, and for the linear algebra -- so the linear algebra is this step, find the coefficient.

That's the step we want to take.

What if I give you the basis, like this wavelet basis, and I give you the pixel -- so here are the pixel values, P1, P2, down to P8 -- what's the job?

What's the linear algebra here?

So these are the values, this is in the standard basis, right?

Those are just the values at eight successive points.

I guess I'm dropping down to one dimension, instead of eight by eight, I'm just going to take eight pixel values

along that first top row.

So what do I want to do?

In standard basis, here are the pixel values.

I want to write that as a combination of c1 times this guy, plus c2 times this guy, plus c3, these are the coefficients, plus c4 times this one -- do you see what I'm doing?

I want to write this vector P as a combination of c1 times the first wavelet plus c8 times the eighth wavelet.

That's the transform step.

That's the lossless step.

That's the step from P -- oh, I'm calling it P here, and I called it x there, so let me -- at the risk of moving, and therefore making this jumpy -- suppose the signal I'm now calling P, that a pixel values, and I'm looking for the coefficients.

OK, tell me how to do it.

If I give you eight basis vectors, and I give you the input signal, and I ask for the coefficients, what do I do?

What's the step?

I'm trying to solve this, I want to know the eight coefficients, so I'm changing from the standard basis, which is just the eight gray-scale values to the wavelet basis, where the same vector is represented by eight numbers.

It's got to take eight numbers to tell you a vector in R^8, and those eight numbers are the coefficients of the basis.

Look, we've done this thing before.

There is the equation in vector notation, we want to see it as a matrix.

This is a combination of columns of the wavelet matrix,

right? This is P equals c1, c2, down to c8, and these guys are the columns.

I mean, this is the step that we're constantly taking in this course, the first basis vector goes in the first column, the second basis vector goes in the second column, and so on, the eight columns of this wavelet matrix are the eight basis vectors.

This is a wavelet matrix W.

So, the step to change basis -- so now I'm finally coming to this change-of-basis, so the change of basis that, let me stay with this board, but -- well, let me just go above it, here.

So the standard basis, we know, the wavelet basis we have here, and the transform is simply, solve the equations,

P=W C. So the coefficients are W inverse P. Right.

This shows a critical point.

A good basis has a nice, fast, inverse.

So good basis means what?

So this is like the billion-dollar competition,

Eh? and it's not over yet.

People are going to come up with better bases than these.

So a good basis will be, first good thing would be fast.

I have to be able to multiply by W fast, and multiply by W -- by its inverse fast.

That's -- if a basis doesn't allow you to do that fast, then it's going to take so much time that you can't afford it.

So these bases -- the Fourier basis, everybody said, OK, I know how to deal quickly with the Fourier basis, because we have something called the Fast Fourier Transform.

So there's a FFT that came in my earlier lecture, and comes in the last chapter of the book, so change-of-basis is done -- if, for the Fourier basis, it's done fast by the FFT and there's a fast wavelet

transform. I can change, for this wavelet example, this matrix is easy to invert.

It's just somebody had a smart idea in choosing that wavelet basis and inverting it, it has a nice inverse.

Actually, you can see why it has a nice inverse.

Do you see any property of these eight basis vectors?

Well, I've only written five of them, but if you see that property for those five, you'll see it for the three

remaining. Well, if I give you those eight vectors and ask, what's a nice property?

Well, you would say, first, they're all ones and minus ones and zeroes. So every multiplication is very fast using -- just in binary.

But what's the other great property of those vectors?

Anybody see it?

So, of course, when I think about a basis, one nice property -- I don't have to have it, but I'm happy if it's there -- is that they're orthogonal.

If the basis vectors are orthogonal, then I'm in good shape.

And these are... do you see?

Take the dot product of that with that, you get four plus ones and four minus ones, you get zero.

Take the dot product of that with that.

You get two plus ones and two minus ones.

Or the dot product of that with that.

Two plus ones and two minus ones.

You can easily check that that's an orthogonal basis.

It's not orthonormal.

To fix it up, I should divide by the length, to make them unit vectors.

Let's suppose I do that.

So somewhere in here, I've got to account for the fact that this has length square root of eight, that has length square root of four, that has length square root of two.

But that's just a constant factor that's easy to -- so suppose we've done that.

Then, tell me what's W inverse?

That's what chapter four, section four point four was about.

If we have orthonormal columns then the inverse is the same as the transpose.

So if we have a fast way to multiply by W, which we do, the inverse is going to look just the same, and we'll have a fast way to do

W inverse. So that's the wavelet basis passes this requirement for fast.

We can use it fast.

But there's a second requirement, is it any good?

Because the the very fastest thing we could do is not to change basis at all.

Right?

The fastest thing would be, OK, stay with the standard basis, stay with eight pixel values.

But that was poor from compression point of view,

right? Those eight pixel values, if I just took those eight numbers, I can't throw some of those away.

If I throw away ninety percent -- if I compress ten to one, and throw away ninety percent of my pixel values, well, my picture's just gone dark.

Whereas, the basis that was good, the wavelet basis or the Fourier basis, if I throw away c5, c6, c7, and c8, all I'm throwing away is little blips that are probably there in very small amounts.

So the second property that we need is good compression.

So first, it has to be fast, and secondly, a few basis vectors should come close to the signal.

So a few is enough.

Can I write it that way?

A few basis vectors are enough to reproduce the image just exactly as on a video of these 18.06 lectures. Uh, I don't know what the compression rate is, I'll ask, David, who does the compression -- and, by the way, I'll try to get the lectures, that are relevant for the quiz up onto the Web in time.

So I'll send them a message today.

So, he's using the Fourier basis because the JPEG -- so JPEG two thousand, which will be the next standard for image compression, will include wavelets.

So, I mean, you're actually getting a kind of up-to-date, picture of where this big world of signal and image processing

is. That Fourier is what everybody knew, and what people automatically used, and the new one is wavelets, where this is the simplest set of wavelets.

And this isn't the one that the FBI uses, by the way, the FBI uses a smoother wavelet, instead of jumping from one to minus one, it's a smooth, Cutoff. and, that's what we'll be in in JPEG two thousand.

OK, so that's that application.

Now, let me come to the math, the linear algebra part of the lecture.

Well, we've actually seen a change-of-basis. So let -- let me just review that eh-eh change-of-basis idea, and then the i- and then the transformation to a matrix.

OK. So this, I hope you see that these applications are really big.

Now, I have to talk a little about change-of-basis, and a little about that.

The matrix.

OK. OK.

OK. So change-of-basis. Basically, forgive that put, OK, I have, I have my vector in one basis, and I want to change to a different one.

Actually, you saw it for the wavelet case.

So I need the -- let the matrix W, and the columns of W be the new basis vectors.

Then the change-of-basis involves, just as it did there, W inverse.

So we have the vector, say, x, in the old basis, and that converts to a vector, let's say, c, in the new basis, and the relation is exactly what we had there, that x is W c.

That's the step we have to take.

There's a matrix W that gives us a change-of-basis. OK.

What I want to do is think about transformations on matrices.

So here's the question to complete this lecture.

Suppose I have a linear transformation T.

So we would think of it as an eight -- as a n by n matrix.

And it's computed with respect to a certain basis.

So T -- no, I'm sorry.

I've got the transformation T, period.

That's taking eight-dimensional space to eight-dimensional space.

Now, let's get matrices in there.

OK.

So, with respect to a first basis, say v1 up to v8, it has a matrix A.

I'm just setting up letters here.

With respect to a second basis, say, I'll make it u1 up to -- or w1, since I've used (w)s, w1 up to w8, it has a matrix B.

And my question is, what's the connection between A

How is the matrix -- the transformation T is settled. and B?

We could say, it's a rotation, for example.

So that would be one transformation of eight-dimensional space, just spin it a little.

Or project it.

Or whatever linear transformation we've got.

Now, we have to remember -- my first step is to remind you how you create that matrix A.

Then my second step is, we would use the same method to create B, but because it came from the same transformation, there's got to be a relation between A and B.

What's the relation between A and B?

And let me jump to the answer on that one.

That if I have the same transformation, and I'm compute on its matrix in one basis, and then I computer it in another basis, those two matrices are similar.

So these two matrices are similar.

Now, do you remember what similar matrices meant?

Similar.

A is similar to -- the two matrices are similar.

Similar.

And what do I mean by that?

I mean that I take the matrix B, and I can compute it from the matrix A using some similarity, some matrix M on one side, and M inverse on the other.

And this M will be the change-of-basis matrix.

This part of the lecture is, admittedly, compressed.

What I wanted you to -- it's really the conclusion that I want you to spot.

Now, I have to go back and say, what does it mean for A to be the matrix of this transformation T.

So I have to remind you what that meant, that was in the last lecture.

Then this is the conclusion that if I change to a different basis, we now know -- see, if I change to a different basis, two things happen.

Every vector has new coordinates.

There, the rule is this one, between the old coordinates and the new ones.

Every matrix changes, every transformation has a new

matrix. And the new matrix is related this way, the M could be the same as the W.

The M there would be the W here.

OK. So, can I, in the remaining minutes, recapture my lecture -- the end of my lecture that was just before Thanksgiving, about the matrix?

OK.

What's the matrix?

And I'll just take one basis.

So now this part is going to go onto this board here.

What is the matrix?

What is A?

OK. Using a basis v1 up to v8. Mm. OK. What's the point?

The point is, if I know what the transformation does to those eight basis vectors, I know it completely.

I know T, I know everything about T, I know T completely from knowing T of V -- what T does to v1, what T does to v2, what T does to v8. Why is that?

It's because T is a linear transformation.

So that if I know what these outputs are -- so these are the inputs v1 up to v8, these are the outputs from the transformation, like everyone rotated, everyone projected, whatever transformation I've done, then why is it that I know everything?

How does linearity work?

Why?

This is because every x is some combination of these basis vectors, right? c1v1, c2v2, c8v8, they were a basis.

That's the whole point of a basis, that every vector is a combination of the basis vectors in exactly one way.

And then, what is T of x?

The point is, I claim that we know T of x completely for every x, because every x is a combination of those -- and now we use the linear transformation part to say that the output from x has to be c1 times the output from v1 plus v2 times the output from v2, and so on.

Up through c8 times the output from v8. So this is like just saying, OK.

We know everything when we know what T does to each basis vector.

OK. So those are the eight things we need.

Now -- but we need these answers in this basis.

So this first output is some combination of the eight basis vectors.

So write T acting on the first input -- in other words, write the first output as a combination of the basis vectors, say a11 v1 + a21 v2 and so on a81 v8. Write T of v2 as some combination a12 of v1, a22 of v2 and so on.

I'm creating the matrix A, column by column.

Those numbers go in the first column, these numbers go in the second column, the matrix A that thi- this -- this is our matrix that represents T in this basis is these numbers. a11 down to a18, a21 down to a28, and so on.

OK. That's the recipe.

In other words, if I give you a transformation, and a basis.

So that's what I have to give you.

The inputs are the basis and to tell you what the transformation

is. And then, you tell me -- you compute T for each basis, expand that result in the basis, and that gives you the sixty four numbers that go into the matrix A.

Let me suppose -- let's close with the best example of all.

Suppose v1 to v8, this basis, is the eigenvectors.

Suppose we have an eigenvector basis so that T(vi) is in the same direction of vi.

Now, my question is, what is A?

Can you carry through the steps?

Let's do them together, because we can do it in one minute.

So, we've chosen this perfect basis.

And, actually, with signal image processing, they might look for the eigenvectors.

But that would take more calculation time that just saying, OK, we'll use the wavelet basis.

Or, OK, we'll use the Fourier basis.

But the very best basis is the eigenvector basis.

OK, what's the matrix?

So, what's the first column of the matrix?

How do I get the first column?

I take the first basis vector v1. I opt -- I look to see, what does the transformation do to it?

The output is lambda one v1. I express that output as a combination so the first input

is v1. Its output is lambda one v1. Now write lambda one v1 as a combination of the basis vectors, well, it's already done.

It's just lambda one times the first basis vector and zero times the others.

So this first column will have lambda one and zeroes. OK. Second input is v2. Output is lambda two v2. OK, write that output as a combination of the (v)s. It's already done.

It's just lambda two times the second v.

So we need, in the second column, we have lambda two times the second v.

Well, you see what's coming, that in that basis, in the eigenvector basis, the matrix is diagonal.

So that's the perfect basis, that's the basis we'd love to have for image processing, but to find the eigenvectors of our pixel matrix would be too expensive.

So we do something cheaper and close, which is to choose a good basis like wavelets.

OK, thanks.

So I'll -- quiz review on Wednesday, all day.

Thanks.