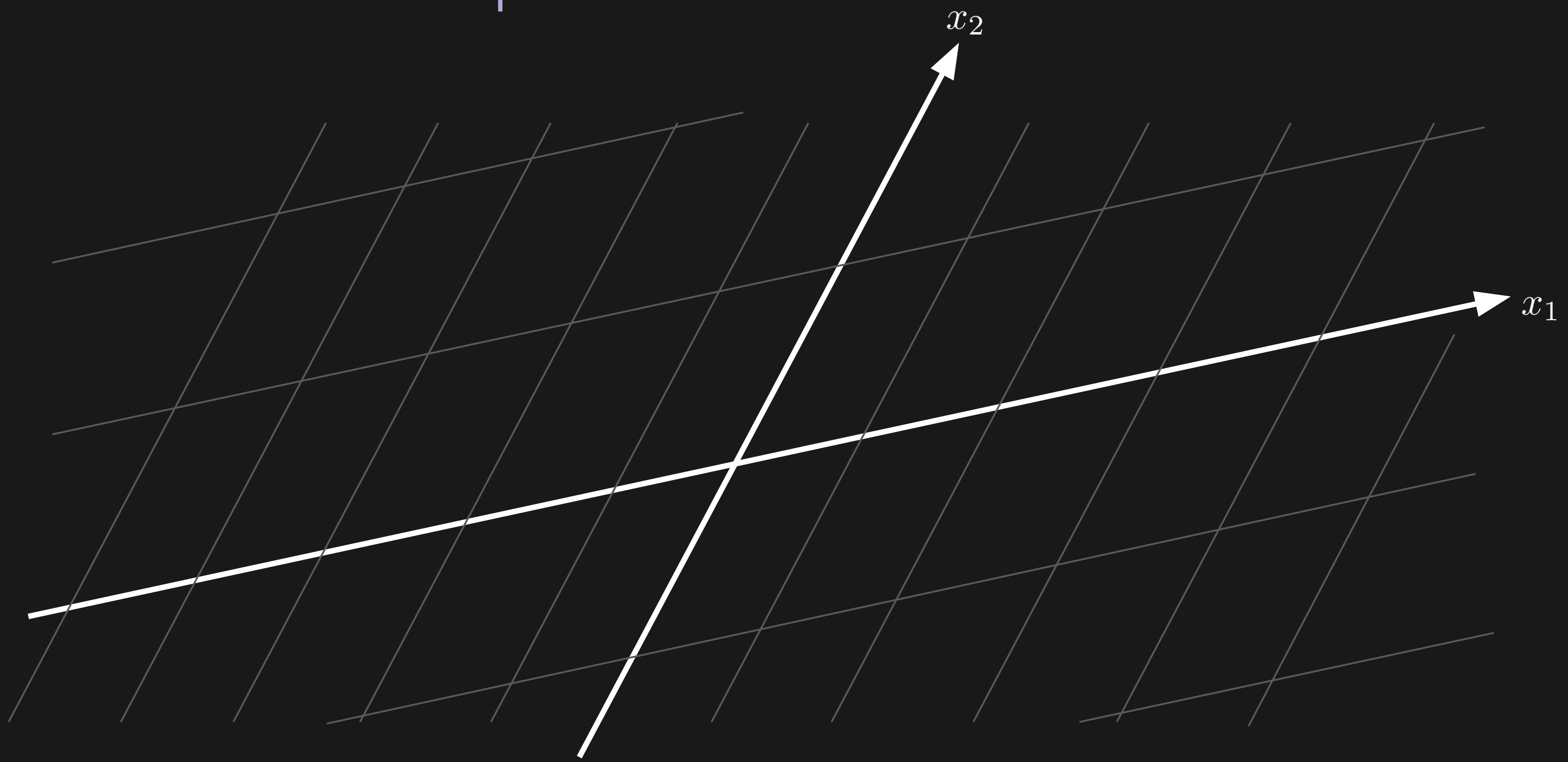
QR Factorization

Practical Linear Algebra | Lecture 10

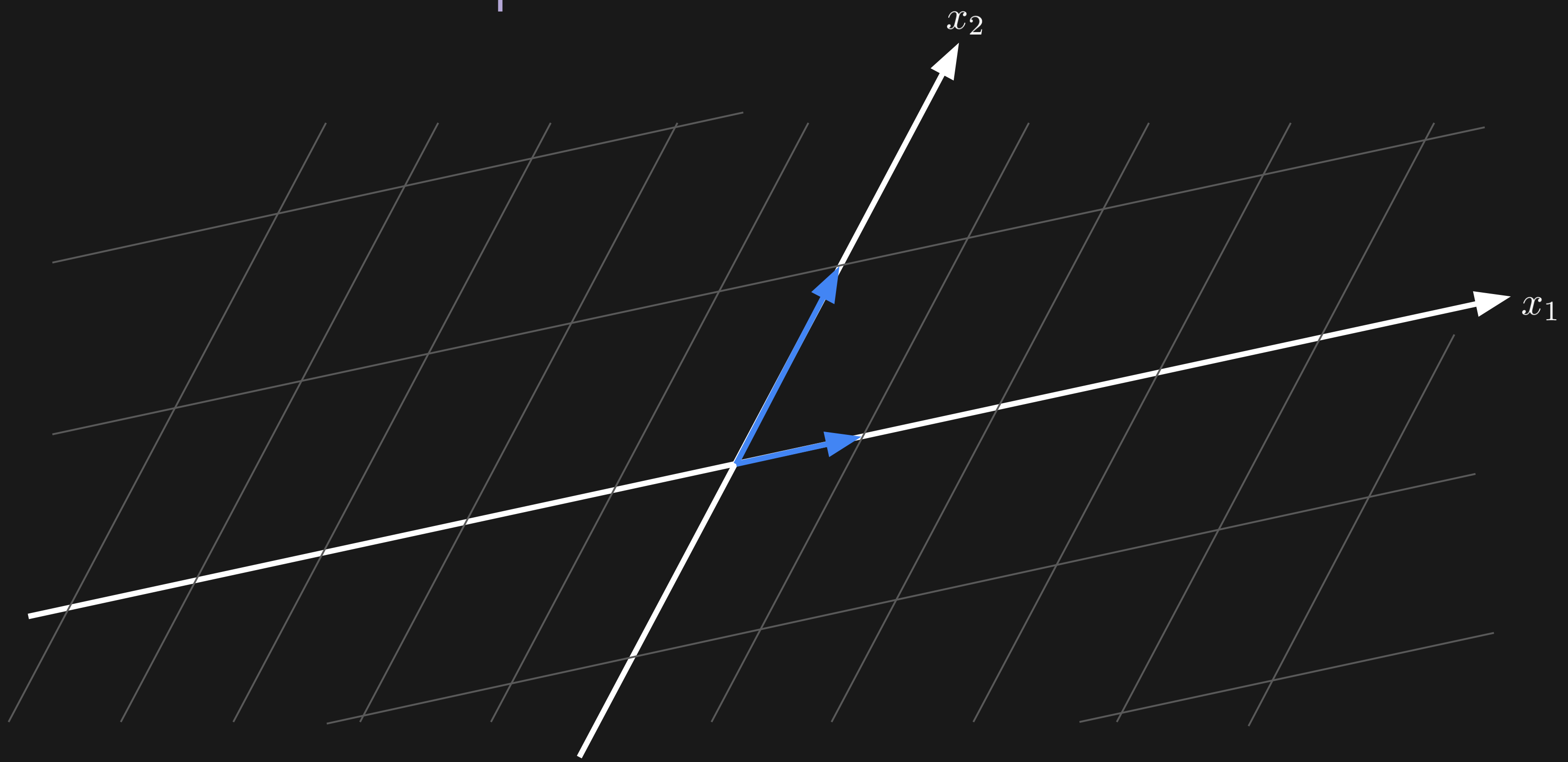
Problem Setup

- Given a set of vectors, we want to find an orthonormal basis for its span
- Why...?
  - It's more natural to work with orthonormal bases
  - It's easier to see which vectors are linearly dependent

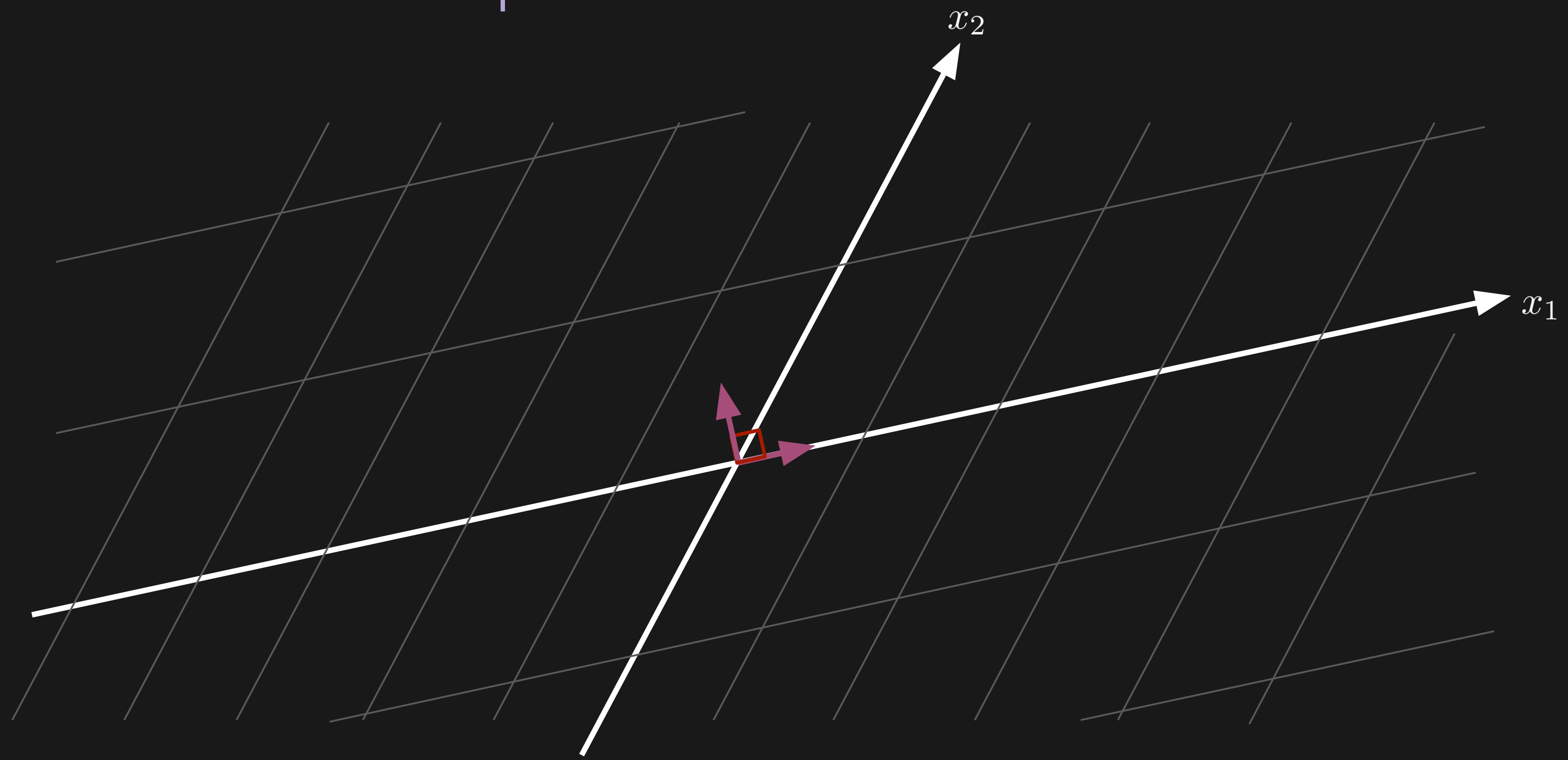
Geometric Interpretation



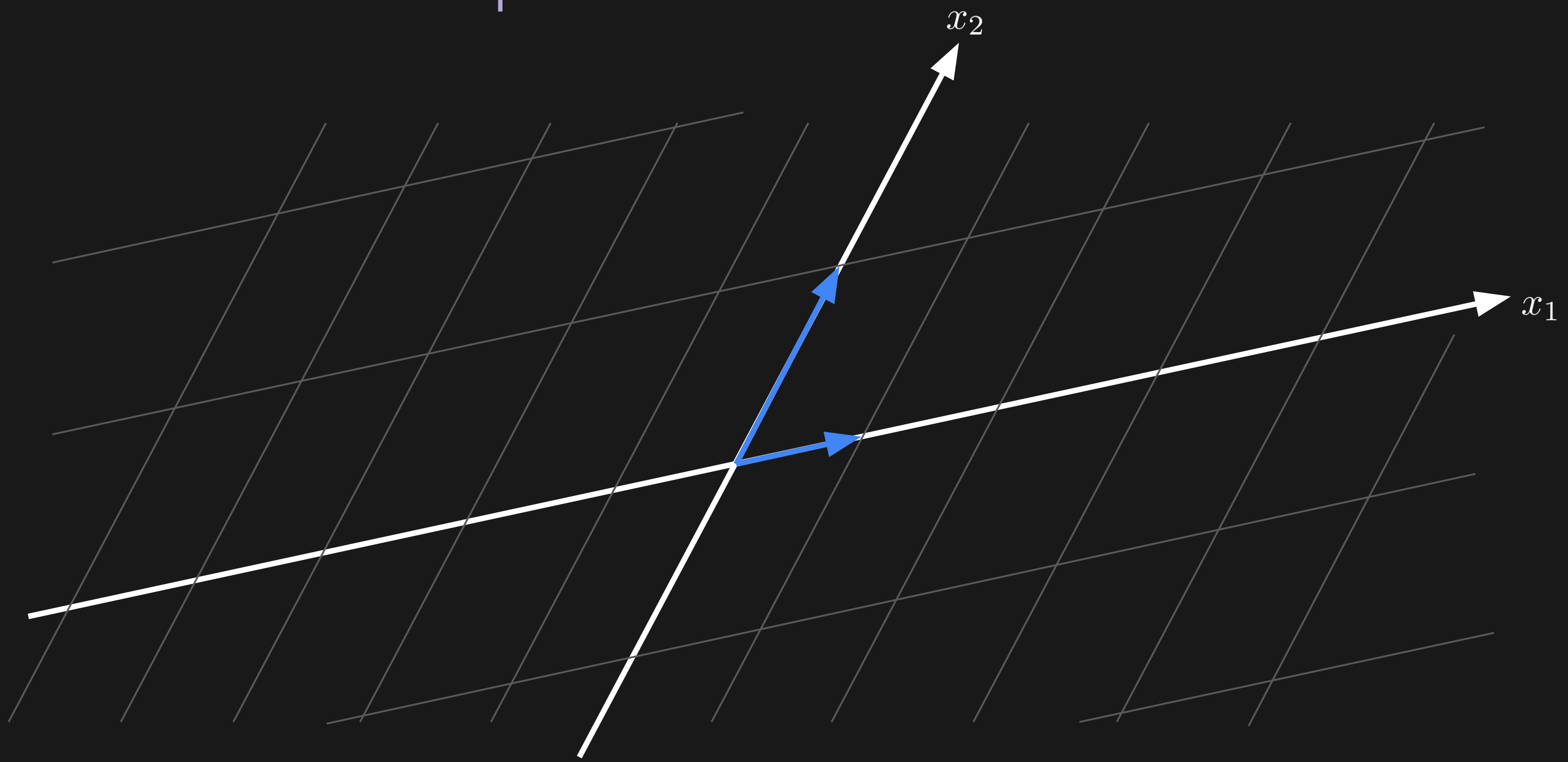
Geometric Interpretation



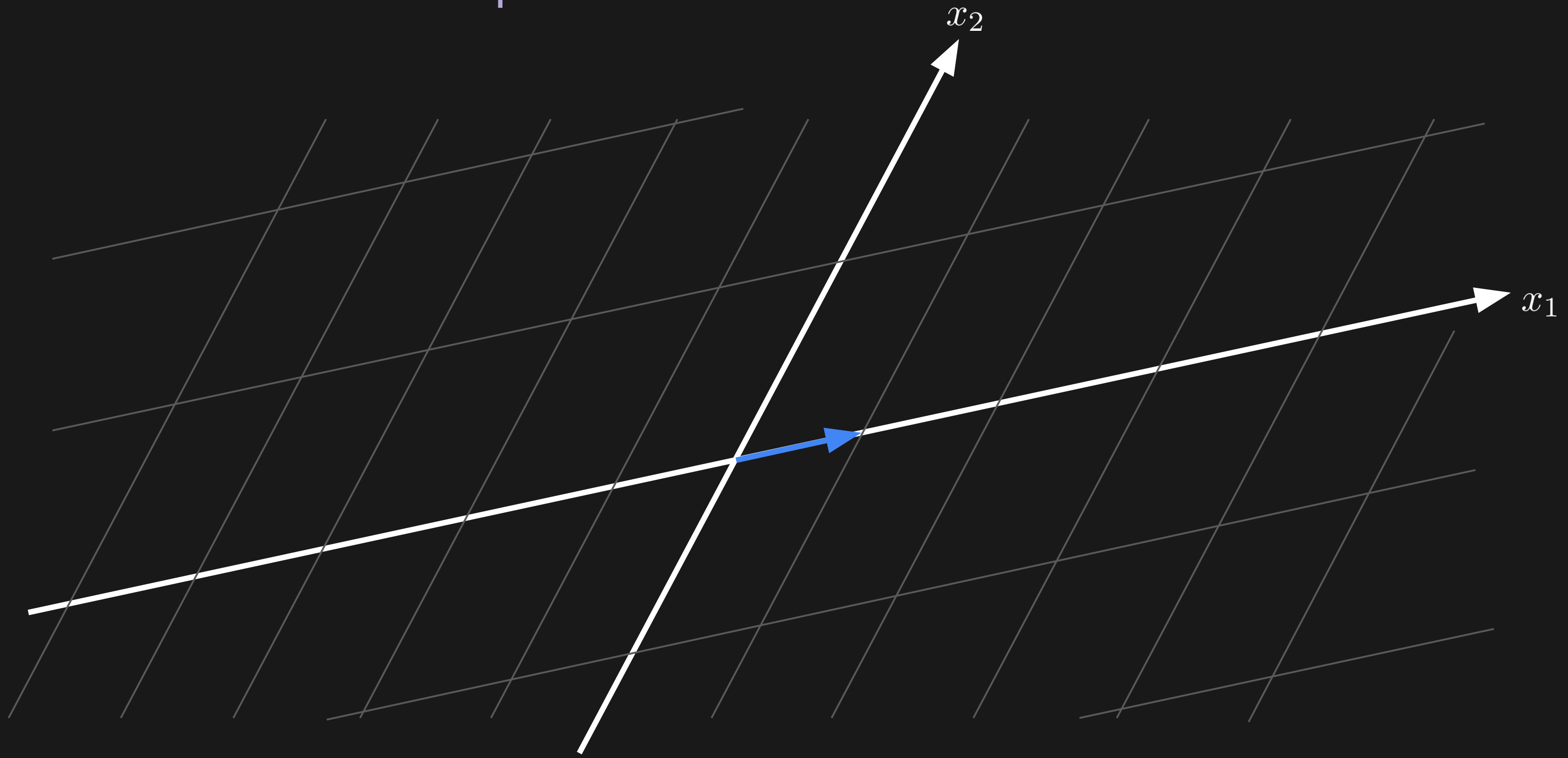
Geometric Interpretation



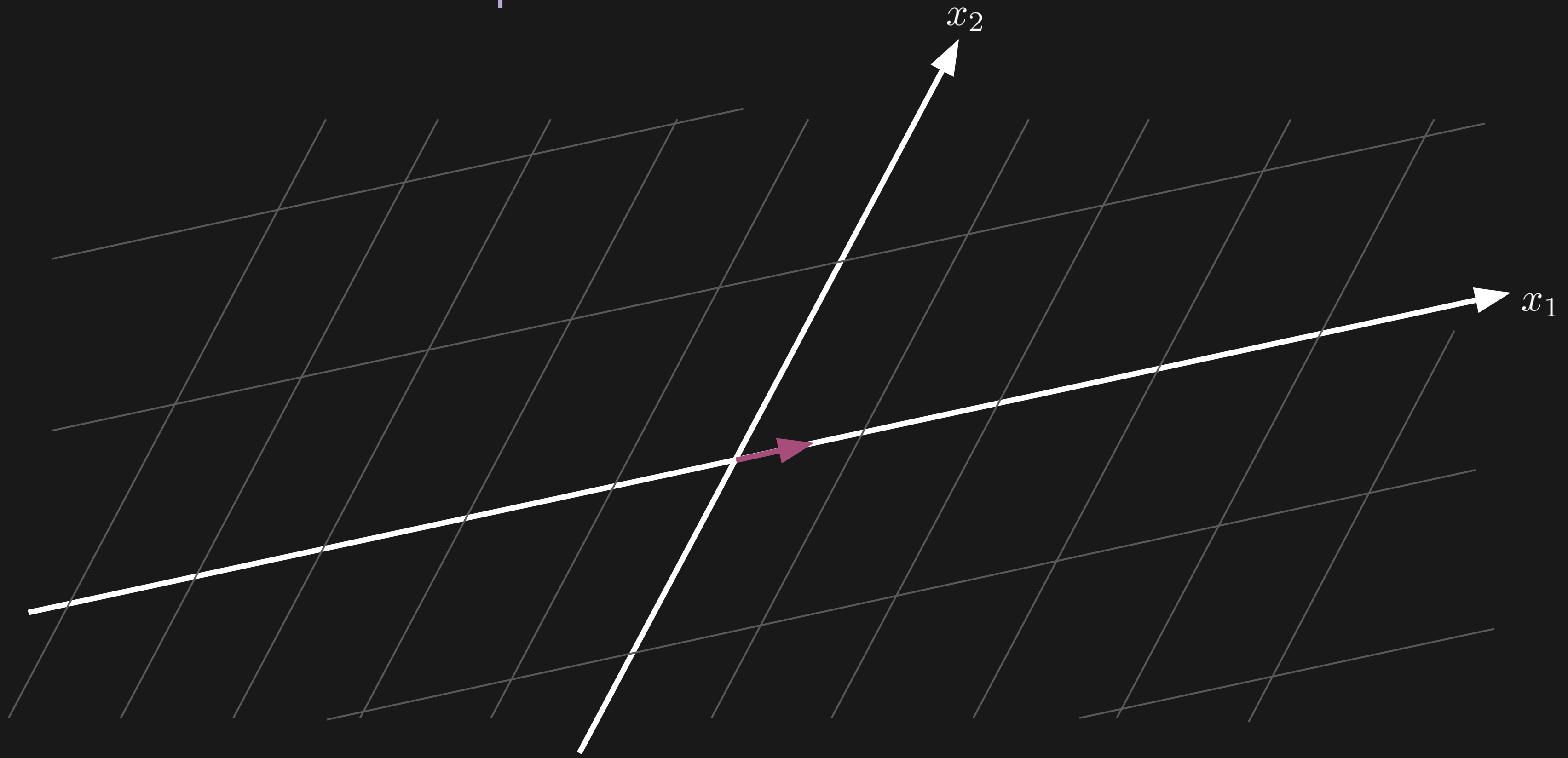
Geometric Interpretation



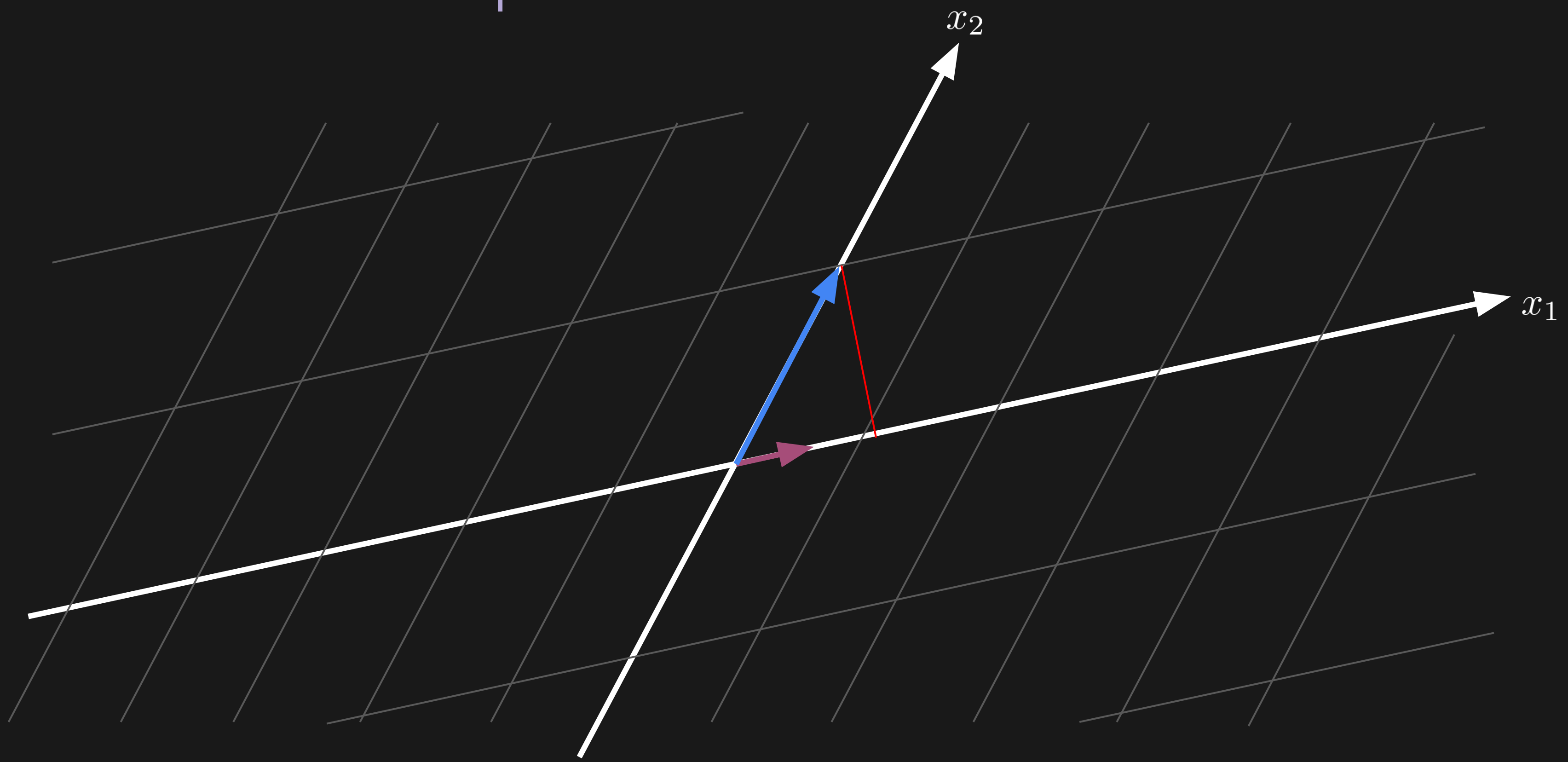
Geometric Interpretation



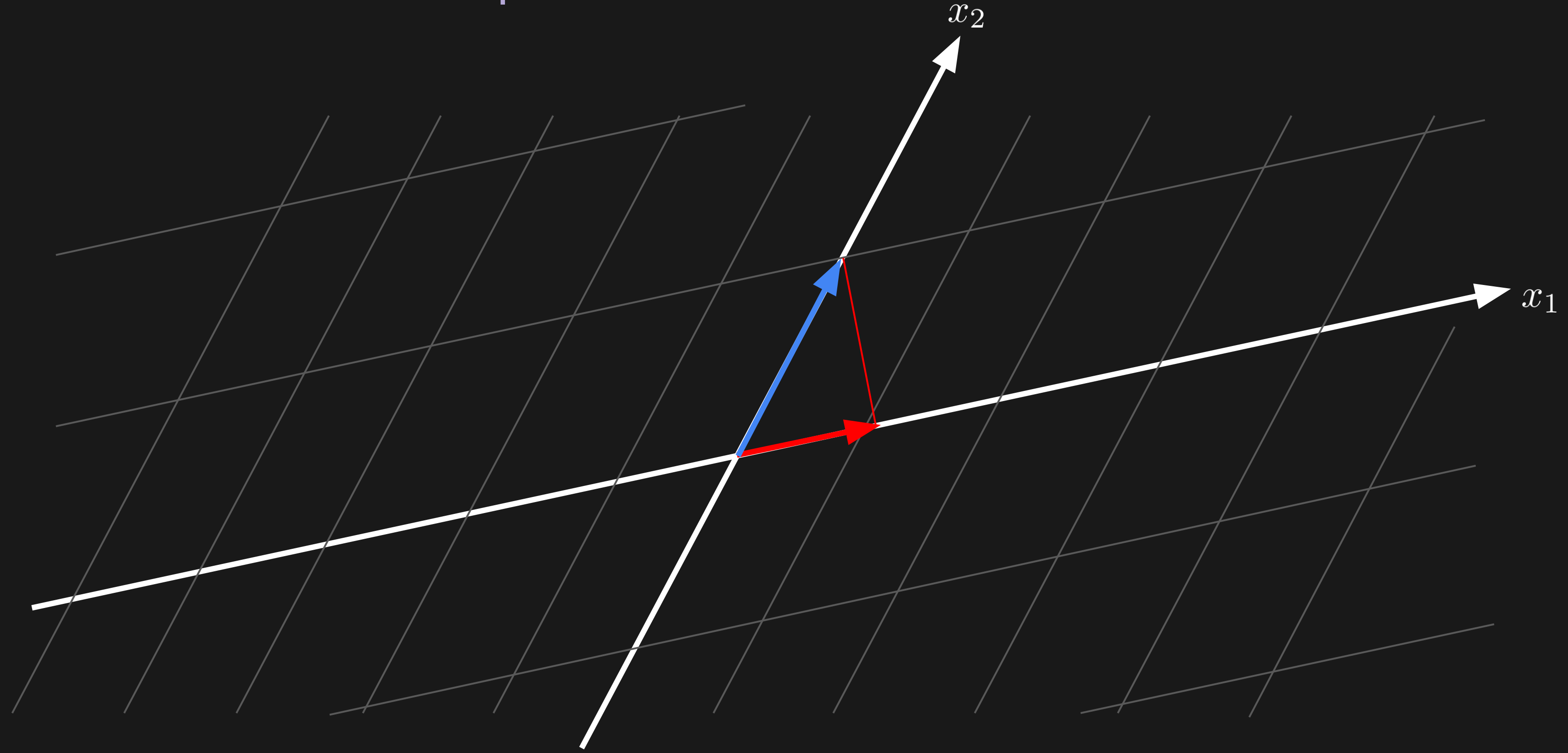
Geometric Interpretation



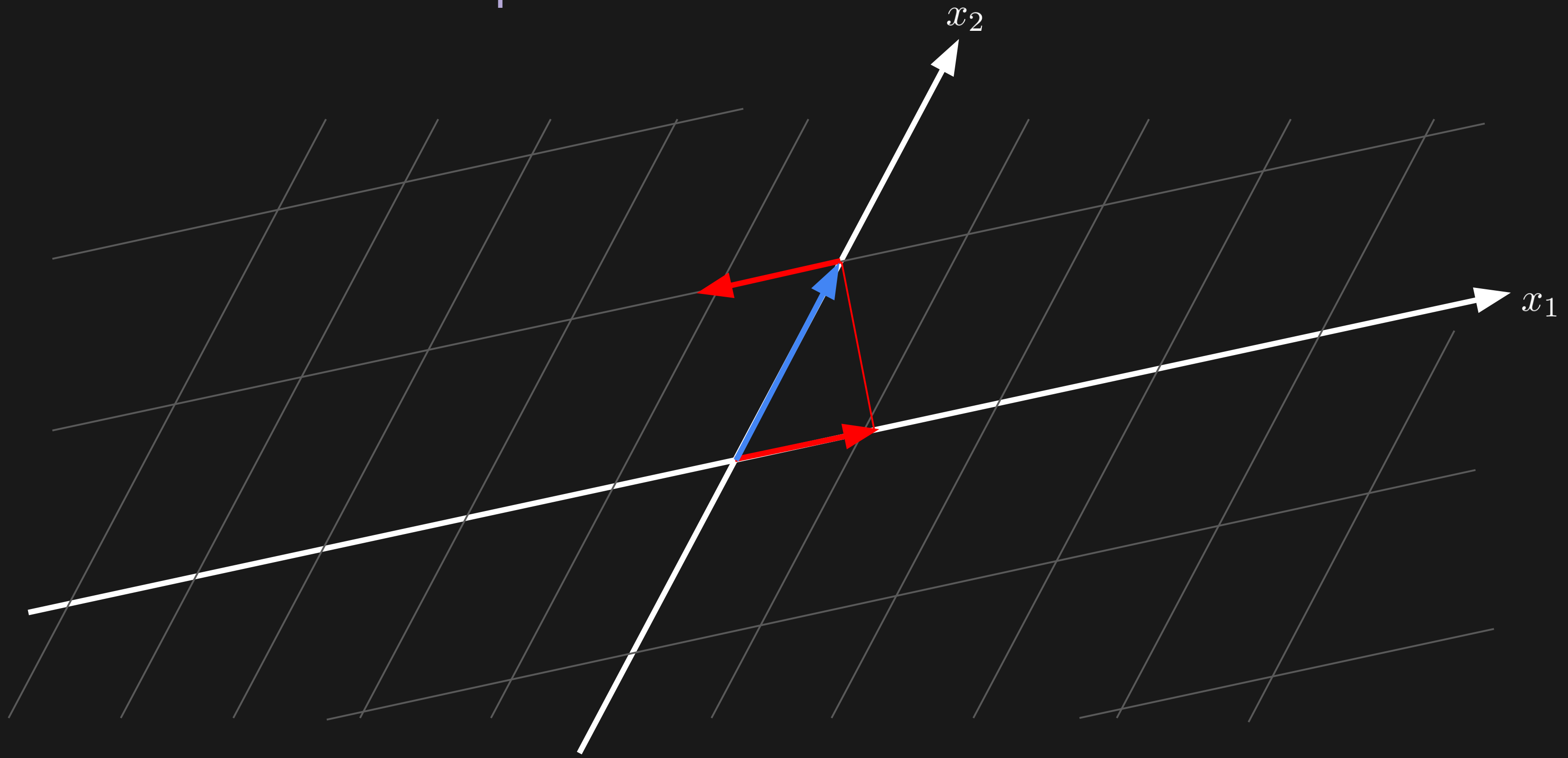
Geometric Interpretation



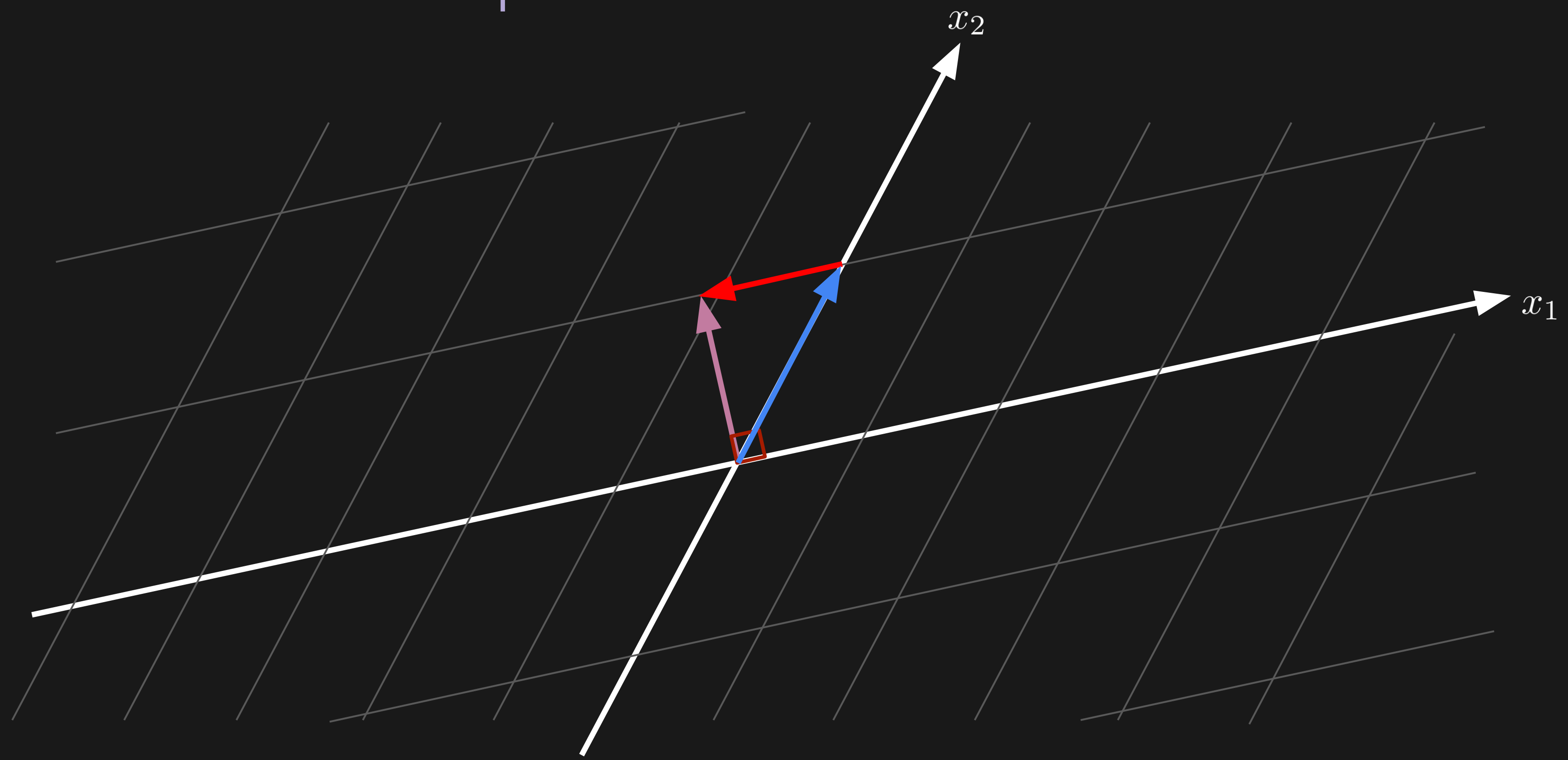
Geometric Interpretation



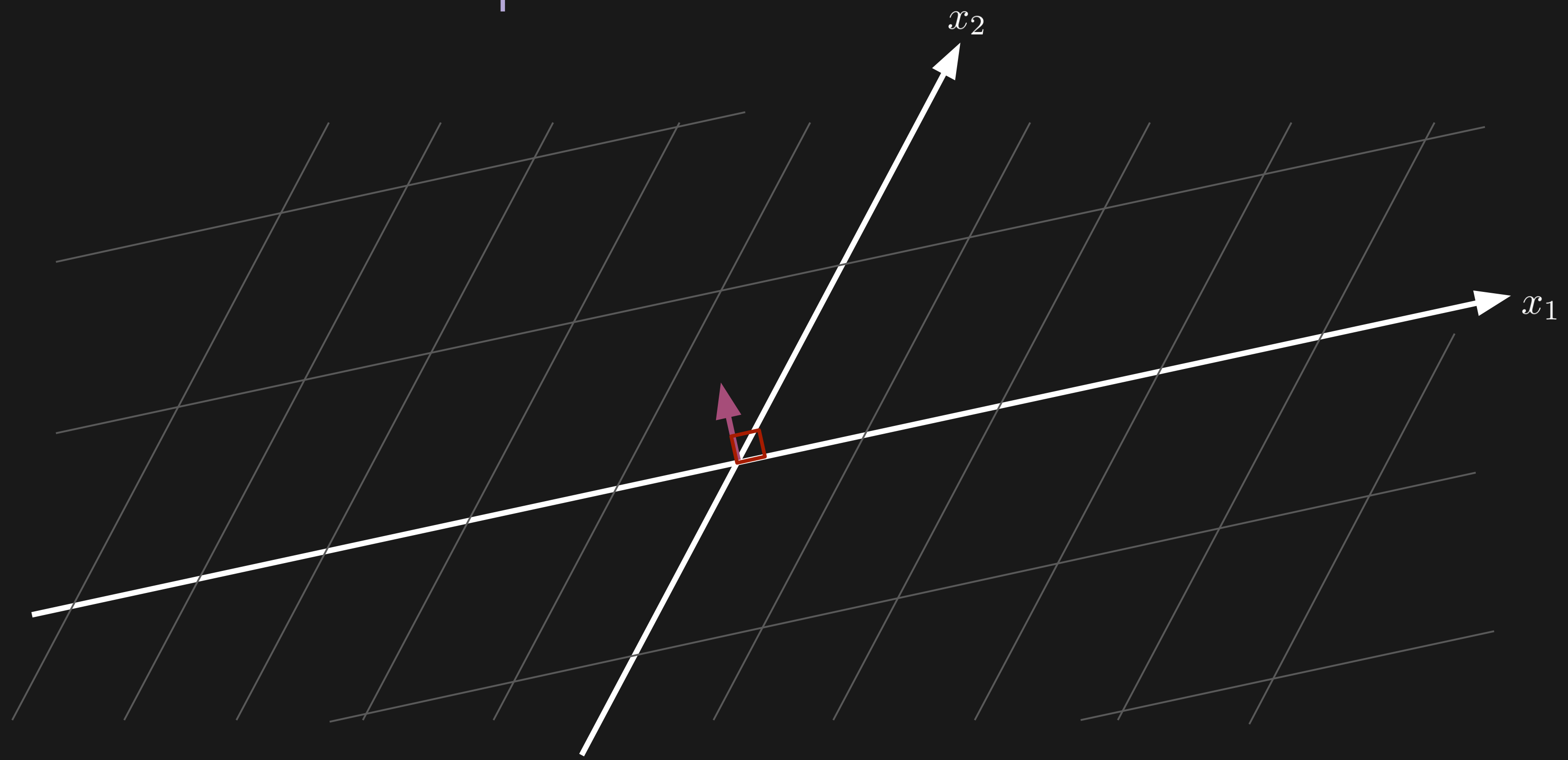
Geometric Interpretation



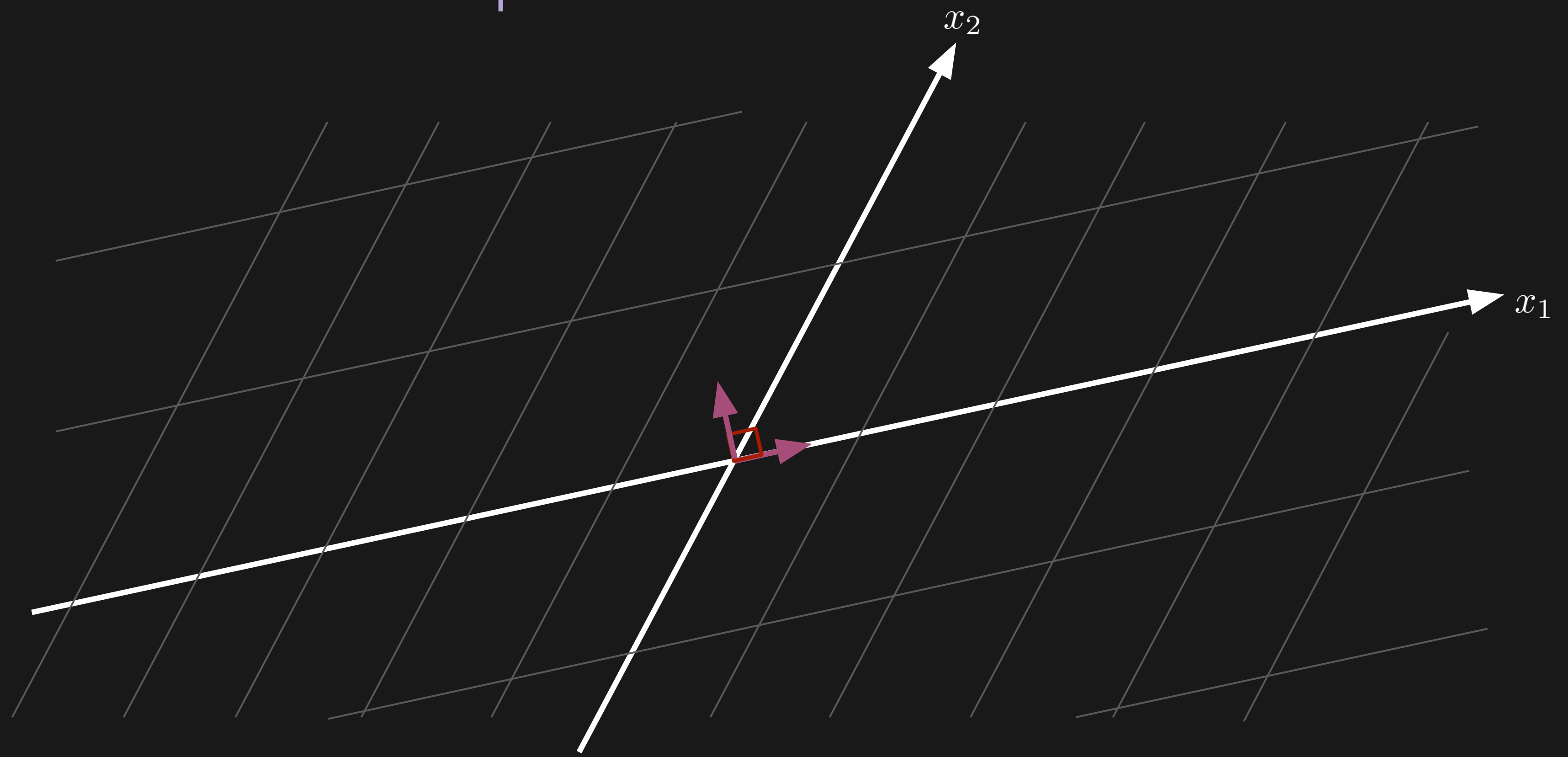
Geometric Interpretation



Geometric Interpretation



Geometric Interpretation



Gram-Schmidt Procedure

In words:

- Take first vector in set (call it a_1) and normalize it. Call this new vector q_1
- Project second vector in set (call it a_2) onto q_1 and subtract that from a_2.
  Call this new vector $\tilde{q}_2$
- Normalize $\tilde{q}_2$ and call it q_2
- Project a_3 onto q_1 and q_2 and subtract both projections from a_3.
  Call this new vector $\tilde{q}_3$
- Normalize $\tilde{q}_3$ and call it q_3
- Keep going...

Gram-Schmidt Procedure

Step 1

$$\tilde{q}_1 = a_1$$

$$q_1 = \tilde{q}_1 / \|\tilde{q}_1\|$$

Step 2

$$\tilde{q}_2 = a_2 - (q_1^T a_2) q_1$$

$$q_2 = \tilde{q}_2 / \|\tilde{q}_2\|$$

Step 3

$$\tilde{q}_3 = a_3 - (q_1^T a_3) q_1 - (q_2^T a_3) q_2$$

$$q_3 = \tilde{q}_3 / \|\tilde{q}_3\|$$

Step i

$$\tilde{q}_i = a_i - (q_1^T a_i) q_1 - (q_2^T a_i) q_2 - \ldots - (q_{i-1}^T a_i) q_{i-1}$$

$$q_i = \tilde{q}_i / \|\tilde{q}_i\|$$

Gram-Schmidt Procedure

Rearrange by solving for a_i

$$\tilde{q}_i = a_i - (q_1^T a_i)q_1 - (q_2^T a_i)q_2 - \ldots - (q_{i-1}^T a_i)q_{i-1}$$

$$a_i = (q_1^T a_i)q_1 + (q_2^T a_i)q_2 + \ldots + (q_{i-1}^T a_i)q_{i-1} + \tilde{q}_i$$

Remember that $\tilde{q}_i = \|\tilde{q}_i\| q_i$, so

$$a_i = (q_1^T a_i)q_1 + (q_2^T a_i)q_2 + \ldots + (q_{i-1}^T a_i)q_{i-1} + \|\tilde{q}_i\| q_i$$

$$a_i = r_{1i}q_1 + r_{2i}q_2 + \ldots + r_{i-1,i}q_{i-1} + r_{ii}q_i$$

QR Factorization

We can write this in matrix form:

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

$$A = QR$$

This is the QR factorization (or QR decomposition) of $A \in \mathbb{R}^{m \times n}$

QR Factorization

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

$$A = QR$$

Note that Q has orthonormal columns and R is upper triangular

Linearly Dependent Columns

- So far we've assumed our vectors are linearly independent
- What if they're linearly dependent?
  - We can just use the same G-S procedure
  - Sometimes r_{ii} will be zero – this means a_i is linearly dependent!
  - In general R won't be upper triangular anymore – but it will be in upper staircase form

Full QR Factorization

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

- The QR we did earlier is sometimes called the “thin” or “reduced” QR

Full QR Factorization

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

$$A = Q_1 R_1$$

Full QR Factorization

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

- Q_2 describes the subspace of $\mathbb{R}^m$ that is “missing” from the range of A

Full QR Factorization

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

- How do we compute Q_2?
  - Pick any full-rank matrix $\tilde{A}$ with the same number of rows as A
  - Then just compute the thin QR factorization of $\begin{bmatrix} A & \tilde{A} \end{bmatrix}$
    - For example, use $\begin{bmatrix} A & I \end{bmatrix}$
  - The matrix Q we get is just $\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ for the original matrix A

Numpy Example

```
import numpy as np

m = 5  # number of rows of A
n = 3  # number of columns of A

# Thin QR
A = np.random.rand(m, n)
Q, R = np.linalg.qr(A)
```

```
# Full QR
A_extended = np.hstack((A, np.eye(m)))
Q, R = np.linalg.qr(A_extended)
```

Numpy Example

```
Thin QR:

A:
[[0.98 0.11 0.22]
 [0.7  0.16 0.54]
 [0.93 0.54 0.98]
 [0.89 0.19 0.35]
 [0.52 0.27 0.61]]

Q:
[[-0.53 -0.53  0.27]
 [-0.38 -0.15 -0.82]
 [-0.51  0.73  0.27]
 [-0.48 -0.24  0.26]
 [-0.28  0.32 -0.33]]

R:
[[-1.84 -0.56 -1.15]
 [ 0.    0.35  0.63]
 [ 0.    0.   -0.23]]
```

Numpy Example

```
Full QR:

A_extended:
[[0.98 0.11 0.22 1.   0.   0.   0.   0.  ]
 [0.7  0.16 0.54 0.   1.   0.   0.   0.  ]
 [0.93 0.54 0.98 0.   0.   1.   0.   0.  ]
 [0.89 0.19 0.35 0.   0.   0.   1.   0.  ]
 [0.52 0.27 0.61 0.   0.   0.   0.   1.  ]]

Q:
[[-0.53 -0.53  0.27  0.6   0.  ]
 [-0.38 -0.15 -0.82 -0.1  -0.38]
 [-0.51  0.73  0.27  0.08 -0.37]
 [-0.48 -0.24  0.26 -0.77  0.21]
 [-0.28  0.32 -0.33  0.19  0.82]]

R:
[[-1.84 -0.56 -1.15 -0.53 -0.38 -0.51 -0.48 -0.28]
 [ 0.    0.35  0.63 -0.53 -0.15  0.73 -0.24  0.32]
 [ 0.    0.   -0.23  0.27 -0.82  0.27  0.26 -0.33]
 [ 0.    0.    0.    0.6  -0.1   0.08 -0.77  0.19]
 [ 0.    0.    0.    0.   -0.38 -0.37  0.21  0.82]]
```

Numpy Example

```
Full QR:

A_extended:
[[0.98 0.11 0.22 [redacted] ]
 [0.7  0.16 0.54 [redacted] ]
 [0.93 0.54 0.98 [redacted] ]
 [0.89 0.19 0.35 [redacted] ]
 [0.52 0.27 0.61 [redacted] ]]

Q:
[[-0.53 -0.53  0.27  0.6   0.  ]
 [-0.38 -0.15 -0.82 -0.1  -0.38]
 [-0.51  0.73  0.27  0.08 -0.37]
 [-0.48 -0.24  0.26 -0.77  0.21]
 [-0.28  0.32 -0.33  0.19  0.82]]

R:
[[-1.84 -0.56 -1.15 [redacted] ]
 [ 0.    0.35  0.63 [redacted] ]
 [ 0.    0.   -0.23 [redacted] ]
 [ 0.    0.    0.   [redacted] ]
 [ 0.    0.    0.   [redacted] ]]
```

Proof of the Rank-Nullity Theorem

$$\mathbf{rank}(A) + \mathbf{nullity}(A) = n$$

$$A \in \mathbb{R}^{m \times n}$$

Proof of the Rank-Nullity Theorem

- Let U_1 be a matrix whose p columns form an orthonormal basis for the nullspace of A (i.e., $\mathbf{nullity}(A) = p$)
- Perform full QR on U_1 to get Q_1, Q_2 and R_1
  - Turns out $Q_1 = U_1$. Why? Because U_1 already has orthonormal columns!
  - For consistency, let's rename Q_2 to U_2
- Note that U_2 has $n - p$ columns... hm...

Proof of the Rank-Nullity Theorem

- Idea: since the columns of $\begin{bmatrix} U_1 & U_2 \end{bmatrix}$ form an orthonormal basis for $\mathbb{R}^n$, we can express any vector $x \in \mathbb{R}^n$ as

$$x = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = U_1 x_1 + U_2 x_2$$

  for some $x_1 \in \mathbb{R}^p$, $x_2 \in \mathbb{R}^{n-p}$
- Any vector in the range of A can be expressed as Ax for some $x \in \mathbb{R}^n$
- Combining, we get $Ax = AU_1 x_1 + AU_2 x_2 = AU_2 x_2$ (since $AU_1 = 0$)

Proof of the Rank-Nullity Theorem

- For simplicity, let $V = AU_2 \in \mathbb{R}^{m \times (n-p)}$, so $Ax = Vx_2$
- This means the columns of V span the range of A... but is V a basis?
- Yes – turns out the columns of V are independent!
  - Let's find some $z \in \mathbb{R}^{n-p}$ such that $Vz = AU_2z = 0$
  - This means $U_2z \in \mathbf{null}(A)$, so $U_2z = U_1w$ for some $w \in \mathbb{R}^p$
  - But $z = U_2^TU_1w = 0$, so $Vz = 0 \Rightarrow z = 0$ and the columns of V are independent
- Columns of V are a basis for A, so

$$\mathbf{rank}(A) = n - p \Rightarrow \mathbf{rank}(A) + \mathbf{nullity}(A) = n$$

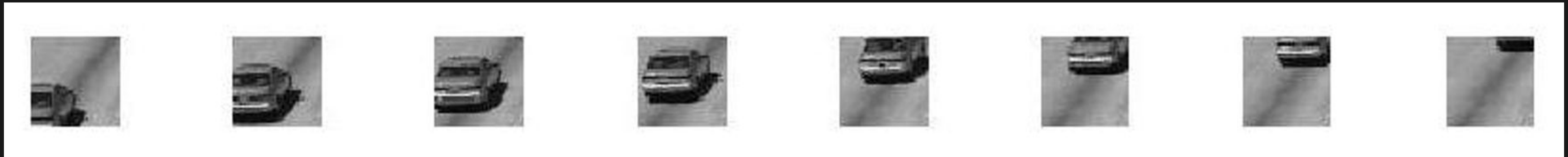
Application: Background Subtraction

QR DECOMPOSITION-BASED ALGORITHM FOR BACKGROUND SUBTRACTION

Mahmood Amintoosi[1,2], Farzam Farbiz[1,3], Mahmood Fathy[1], Morteza Analoui[1], Naser Mozayani[1]

Figure 1. Sample frames of a traffic movie[1].

Application: Background Subtraction



Application: Background Subtraction

Application: Background Subtraction

$$A^b = \begin{bmatrix} X^b_{1,1} & X^b_{1,2} & \cdots & X^b_{1,t} \\ X^b_{2,1} & X^b_{2,2} & \cdots & X^b_{2,t} \\ \vdots & \vdots & \vdots & \vdots \\ X^b_{N,1} & X^b_{N,2} & \cdots & X^b_{N,t} \end{bmatrix}$$

Application: Background Subtraction

$$A = QR = Q \begin{bmatrix} 3 & 3 & 3 & 3 & 1 & 2 & \cdots \\ 0 & 0 & 0 & 0 & 2 & 1 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 4 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Application: Background Subtraction

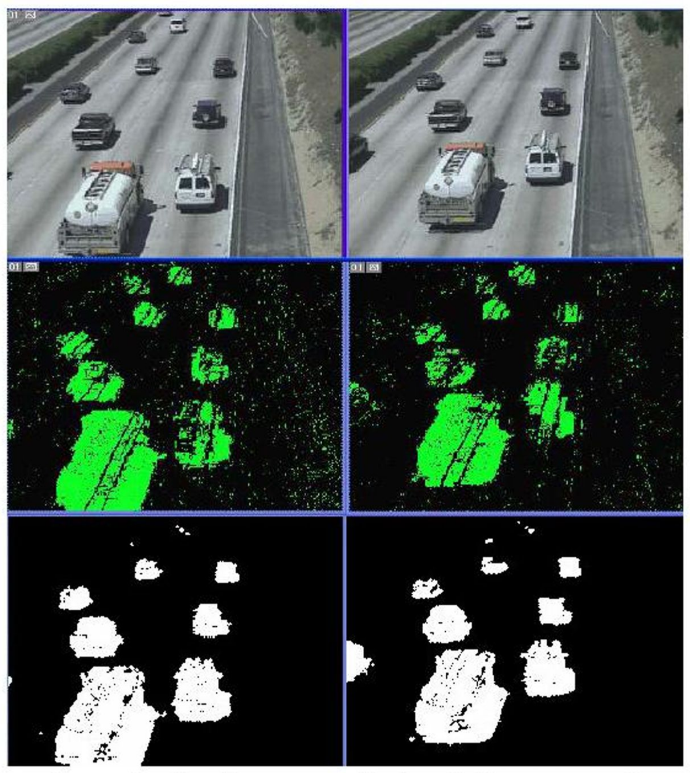


Figure 5. From top to bottom: original frames, foreground object detection using GMM (taken from [11]) and the proposed approach.

Next Time

- Eigenvectors