1これからグループ2、プロジェクトMoonlight Vigilの発表を始めます。

2私たちがプロジェクト演習で取り組んだ課題は、フーリエ変換を作成し、それを使った信号処理アプリケーションを開発するというものです。このアプリケーションは主に以下の人に向けて作成されました。まず研究者や学生に向けてです。フーリエ変換は数学の重要な分野であり特に信号処理や偏微分方程式の研究においてフーリエ変換は必須のツールだと考えます。電気工学において信号処理や通信工学においてフーリエ変換を用いられます。例えば信号のフィルタリングや周波数の解析、デジタル信号処理などにおいてフーリエ変換は重要なツールです。次に画像処理エンジニアの方などに使われます。コンピュータビジョンの研究者や画像処理エンジニアは2次元フーリエ変換を使用して画像の特徴抽出、画像の圧縮などを行います。また映像ストリームの解析や映像品質向上のために2次元フーリエ変換は広く利用されます。最後に音響エンジニアや音楽家に用いられます。音響エンジニアは音声信号の解析や音響データのフィルタリング、音声信号の特徴抽出などにおいてフーリエ変換を使用します。音楽家は音楽信号スペクトル解析や楽曲の特徴抽出においてフーリエ変換を使用します

3まずフーリエ変換とは何かを説明します。フーリエ変換は上の公式です。これはf(t)にe^-jωtをかけて-∞から∞の範囲で時間積分をしています。その結果角周波数ωの関数F(x)が得られます。つまり時間の関数を角周波数に変換しています。

フーリエ変換は時系列データや信号を周波数成分に分解するのに用いられます。信号を時間領域から周波数領域に変換し、その成分の強さや位相を解析します。これにより、信号の周波数特性や周波数成分の相対的な重要性を理解し、さまざまな応用に活用されます。今回私たちは1次元と2次元のフーリエ変換について学び、プログラムを作成しました。

41次元フーリエ変換は1次元の信号を周波数成分に分解する数学的手法で与えられた信号を時間領域から周波数領域に変換することで、その信号がどのような周波数成分から構成されているのかを解析することができます。また1次元フーリエ変換を行うことで、音声や振動信号、電気信号などの1次元データを周波数成分ごとに分解し、その情報を利用して音声のスペクトル解析やフィルタリング、信号の圧縮やノイズ除去などに用いられます。1次元フーリエ変換の基本的な考え方について説明します。1次元の信号は時間に対して値が変化するデータ列です。例えば音声信号では音の強さが時間と共に変動します。1次元フーリエ変換ではこの時間領域の信号を周波数領域に変換することで、信号の周波数成分を抽出することができます。具体的な手順としては、与えらえれた信号を一定の時間区間でサンプリングし離散的なデータ点列として扱います。このデータ列を離散フーリエ変換と呼ばれるアルゴリズムを用いて周波数領域に変換します。離散フーリエ変換は離散的な時間データを複素数のスペクトルデータへと変換する操作です。離散フーリエ変換によって得られたスペクトルデータは新服と位相からなる複素数の列として表されます。このスペクトルデータは下の信号がどの周波数成分から成り立っているかを示しており、振幅は周波数成分の強度を表し、位相は波形の位相情報を示します。重要な点として、1次元フーリエ変換によって得られたスペクトルデータを逆フーリエ変換することで周波数領域のデータを時間領域に戻すことも可能です。この逆フーリエ変換によって周波数領域での操作や変換結果を下の信号に戻すことができます。

2次元フーリエ変換は2次元の信号や画像を周波数成分に分析する数学的手法のことです。2次元フーリエ変換は画像処理やパターン認識、通信などの分野で広く利用されています。画像処理では2次元フーリエ変換を用いて画像の周波数スペクトルを解析し、画像の特徴抽出やフィルタリングなどの処理を行います。通信では2次元フーリエ変換を用いて信号の周波数特性を分析し信号の伝送や複合に役立てることができます。2次元フーリエ変換の基本的な考え方は1次元フーリエ変換と同様ですが2次元データに対して適用されます。画像は2次元空間的な情報を持っており、各画素の明るさや色情報が格子状に配置されています。2次元フーリエ変換ではこの画像データを空間領域から周波数領域に変換し画像中の周波数成分を解析することができます。2次元フーリエ変換の手順は与えられた画像をピクセルごとに明るさや色情報を表す値として取得し、2次元配列として表現します。この2次元配列を2次元離散フーリエ変換と呼ばれるアルゴリズムを用いて周波数領域に変換します。2次元離散フーリエ変換によって得られたスペクトルデータは画像中の各周波数成分の新服と位相を表す複素数の2次元配列となります。2次元フーリエ変換によって得られたスペクトルデータ画像中の周波数成分の特徴やパターンを示しています。画像中の周波数成分はエッジやテクスチャなどの細部を表し、低周波成分は画像の全体的な構造や平坦な領域を表します。スペクトルデータを解析することで、画像の周波数成分に関する情報を抽出することが可能です。重要なポイントとしては2次元フーリエ変換によって得られたスペクトルデータを逆2次元フーリエ変換することで、周波数領域のデータを下の画像に戻すことが可能です。逆2次元フーリエ変換によって下の画像を復元する際にはスペクトルデータの振幅や位相情報を適切に組み合わせることが重要です。

5今回私たちは基本的な信号（sample wave）、典型的な信号である矩形波（square wave）・三角波（triangle wave）・ノコギリ波（sawtooth wave）・低周波から高周波までの全域が含まれる波（chirp signal：チャープ信号）などを離散データとして作り、それらに対して離散フーリエ変換を施してフーリエ変換の結果であるパワースペクトルを編集することで、逆フーリエ変換を施した復元データ（復元画像）を調整できる機能を組み込むように作りました。ここにおいてパワースペクトルとは、フーリエ変換によって得られる信号や画像の周波数成分のエネルギー分布を表すスペクトルです。

6時間領域または空間領域に存在する信号や画像を周波数領域に変換することで、各周波数成分の強さを可視化します。パワースペクトルは周波数成分の振幅の絶対値の2乗を取ることで得られ、非負の実数となります。一般的に対数スケールで表示され、より広い周波数範囲を視覚的に把握することが可能です。

次にパワースペクトルの利用方法についてお話しします。パワースペクトルは、周波数特性を理解するために使用されます。特定の周波数帯域の強さやエネルギーの分布を把握することで、信号の周期性や特徴を分析することができます。例えば、音声処理では音声信号のパワースペクトルを解析して、音声の音高やスペクトログラムを表示します。またパワースペクトルを利用して特定の周波数帯域の成分を除去するフィルタリング処理が行われます。特定の周波数帯域のノイズを除去したり、特定の周波数成分を強調したりすることが可能です。

フーリエ変換を行うと複素数の形式で表された周波数成分が得られます。フーリエ変換によって得られる複素数の実部と虚部は、それぞれ余弦波と正弦波の周波数成分に対応しています。実部は余弦波の成分を表し、振幅と位相の情報を持ちます。虚部は正弦波の成分を表し、振幅と位相の情報を持ちます。これらの情報を組み合わせることで、元の信号の周波数成分や特性を理解することができます。用意した波について説明します。

7矩形波をフーリエ変換すると、特徴的な周波数成分のスペクトルが現れます。矩形波は周期的な波形であり、フーリエ変換によってその周波数成分が解析されるため、特定の周波数でピークが現れます。矩形波の周期をTとし、振幅を1とすると、矩形波の数学的表現は右上のようになります。これをフーリエ変換すると*このようになります。(ちゃんと式言って)*ここで、F(ω)はフーリエ変換後の周波数スペクトルを表し、ωは角周波数（ラジアン/秒）を表します。実部はこのように、虚部は0となります。これらの式から分かるように、矩形波のフーリエ変換の実部は正の実数の成分を持ち、虚部はゼロです。つまり矩形波のフーリエ変換では、ω・T/2=k・π（ここでkは整数）のときにピークが現れます。つまり周波数が整数の倍数となるところでピークが発生します。この特性により、矩形波の周波数スペクトルは、基本周波数（最も低い周波数）とその全体数倍の周波数成分を持つことになります。基本周波数成分の振幅が最も大きく、その倍数の周波数成分に対して振幅は次第に小さくなっていきます。矩形波のフーリエ変換結果は、このような周波数成分の特性を持つため、特定の周波数を持つ波形を分析するのに役立ちます。

8私たちが開発したプログラムではバタフライ演算を用いて高速フーリエ変換を行いました。バタフライ演算は、フーリエ変換の計算過程で複素数の乗算や加算を行う部分を効率的に実行するためのアルゴリズムです。通常のフーリエ変換では、乗算や加算の回数が信号のデータサイズに比例して増加しますが、バタフライ演算を使用することで計算量を削減することができます。バタフライ演算では、データを複素数のペアとして扱い、それらのペアに対して同時に乗算や加算を行います。具体的には、データをビット反転した順序で並べ替え、複素数のペアごとに計算を行います。バタフライ演算は再帰的に行われ、階層的な構造を持ちます。1次元フーリエ変換では、バタフライ演算はデータを2つのグループに分割し、それぞれのグループ内でバタフライ演算を行います。次に、グループをさらに分割し、再びバタフライ演算を行います。これを繰り返すことで、最終的にフーリエ変換が計算されます

。2次元フーリエ変換では、2次元のデータを行方向と列方向に分割し、それぞれの方向でバタフライ演算を行います。最初に行ごとにバタフライ演算を行い、その結果を保持します。次に、列ごとにバタフライ演算を行い、最終的な2次元フーリエ変換の結果を得ます。

次の特徴としてグレースケール画像やカラー画像にもフーリエ変換を行えるようにしました。カラー画像に対してフーリエ変換を行う場合、各RGBチャンネルごとに変換を行います。各チャンネルは、画像中の異なる色成分を表しており、それぞれのチャンネルに対してフーリエ変換を行うことで、各色成分の周波数スペクトルを取得することができます。これにより、画像の色情報や彩度の変化、色相の特徴などを解析することができます。さらに、RGBチャンネルの組み合わせによっては、特定の色の強調や抽出などの画像処理を行うことも可能です。一方、グレースケール画像に対してフーリエ変換を行う場合、画像の輝度情報に着目します。グレースケール画像は、明るさの情報のみを持つ画像であり、輝度値を表す単一のチャンネルで構成されています。このチャンネルに対してフーリエ変換を適用することで、画像内の異なる周波数成分やパターンを特定することができます。これにより、画像内のエッジやテクスチャ、周期的なパターンなどを検出することができます。また、グレースケール画像においても、フーリエ変換後の周波数スペクトルを逆変換することで、元の画像に復元することも可能です。カラー画像やグレースケール画像に対するフーリエ変換は、画像処理や信号解析において幅広い応用があります。例えば、エッジ検出、フィルタリング、ノイズ除去、テクスチャ解析、圧縮、特徴抽出などの画像処理タスクにおいて、フーリエ変換を利用することで高度な解析や処理が可能になります。また、画像認識やパターンマッチング、画像の比較や類似性の評価などの応用にも活用されています。

9次に工夫店についてお話しします。一次元においても、二次元においても、フーリエ変換の結果であるパワースペクトルを対話的に編集するようにする仕様にしました。この機能により、ユーザーは信号や画像の周波数成分に直接的にアクセスし、自由に調整することが可能です。さらに手を加えればノイズの除去や特定の周波数成分の強調、フィルタリングのカスタマイズなど、幅広い応用が期待されます。

また可能な限りオブジェクト指向なプログラムにしました。まずオブジェクト指向とは、アラン・ケイさんによって提唱されたプログラムの作り方のことです。

「ある役割を持ったモノ」ごとにクラスを分割し、モノとモノとの関係性を定義していくことでシステムを作り上げようとする**システム構成の考え方**であり、オブジェクト間のメッセージ送受信によってプログラミングする手法のことです。プログラムをオブジェクト指向にすることには多くの利点があります。まずオブジェクト指向のプログラミングでは、機能やデータをオブジェクトとしてカプセル化します。これによりコードを小さな独立したモジュールに分割することができます。各オブジェクトは特定の役割や責任を持ち他のオブジェクトとのインタラクションを通じて機能を実現します。このモジュール性によりコードの再利用性が向上し、保守性や拡張性も向上します。次にオブジェクト指向のアプローチでは、新しい機能を既存のオブジェクトに追加することが比較的容易です。既存のオブジェクトのコードを変更することなく、新しいクラスやオブジェクトを作成して機能を拡張することができます。また継承やポリモーフィズムなどの特性を活用することで、機能の追加やカスタマイズが容易になります。最後にオブジェクト指向のアプローチはチームでのプロジェクト開発においても効果的です。各オブジェクトは独立しており、開発者は自身の担当範囲に集中できます。また、オブジェクトの再利用性やインターフェースの明確化により、チーム内のコミュニケーションや協力が円滑化します。

今思ったらオブジェクト指向の説明いらんかったかもしれん。どうするかは君たちが決めて

10デモンストレーションするで〜

11 次に反省点についてお話しします。先ほども話したようにプログラムはオブジェクト指向であることが望ましいですが時間不足で全てのプログラムをオブジェクト指向にすることができませんでした。アプリケーション全体がオブジェクト指向で設計されていなかったため、一貫性が欠如してしまいました。一部の機能やコンポーネントがオブジェクト指向になり他の部分は手続き型のままとなると、コードの理解や保守が難しくなります。全体的なアーキテクチャをオブジェクト指向に統一することで、コードの一貫性を高め、開発者間のやり取りや修正が円滑に行えるようになると認識しています。またコードの再利用性が低下してしまいました。オブジェクト指向の特性を利用して、より多くのコードを再利用可能にすることで、開発効率を向上させることができると考えています。

次に拡張期のが皆無になってしまいました。要求仕様社にはノイズ除去、シグナル強調、特徴検出、領域分割、類似検索、再帰的な周波数分離、肌理（きめ）の解析、欠損情報の復元などを行おうとするユーザを対象として作るように示唆されている。私たちのプログラムではそのような機能が皆無なため実装ができるような状態ではない。

最後にテストが不十分という点です。テストの不足は、品質に影響を及ぼす可能性があります。