動画情報に基づく環境音楽実時間生成ソフトウェアアーキテクチャの考察

―データ取得部分を対象として―

2015SE002 青山純　2015SE046 松本尚大

指導教員：野呂昌満

1　はじめに

近年，動画共有サービスの普及により，個人による動画配信を容易に行える環境が整い，動画に適切な音楽を付加する必要性も高まっている [2]．このような背景の下，動画制作の労力軽減と，映像の魅力を高めるために，動画情報から独自の音楽を生成する研究が行なわれてきた．

既存の動画から独自の音楽を生成することを目的として，動画ファイルを読み取り独自の音楽を付加する研究 [5] や，既存の音楽を同期させ雰囲気にあった動画を生成する研究 [3] が行なわれてきた．これらの研究はすべて一括処理を前提としており，実時間での音楽生成に対応できない．

本研究の目的は，実時間での音楽生成について考察することである．動画制作者の労力軽減と，映像の魅力を高めることを目指してソフトウェアアーキテクチャを設計する．設計したソフトウェアアーキテクチャに基づき，プロトタイプを試作しアーキテクチャの有用性を確認する．

我々は，映像解析部分を担当する．パイプとフィルタのソフトウェアアーキテクチャを用いることによって，フィルタ処理の順序変更や付け替えを可能にする．加えて，映像解析と音楽生成処理を並行に行なうことによって，実時間性を実現する．

2　関連技術

2.1　映像解析

映像解析とは，映像から有意な事象を取捨選択する処理のことである．本研究では，音楽を自動生成するために必要となる色調や明度のデータを，映像解析を行なうことで取得する．

2.2　パイプとフィルタ（Pipes and Filters）

パイプとフィルタ（Pipes and Filters）は，データストリームを処理するシステムのアーキテクチャスタイルである．このアーキテクチャにおいて，処理ステップはフィルタコンポーネントにカプセル化され，データは隣接する複数のフィルタ間をパイプコンポーネントを介して引き渡される．以下に各コンポーネントの機能を示す．

- フィルタ
  パイプラインの処理単位であり，以下の種類のものがある．
  -必要なデータ項目を追加することにより情報量を増やす．
  -不要なデータを除去することにより適正なデータとする．
  -表現形式の変換．
- パイプ
  データのソースと 1 番目のフィルタの間，最後のフィルタとデータシンクの間，個々のフィルタ間の接続部分．
- データシンク
  パイプラインの終端から結果を集める．

フィルタコンポーネントの交換，組み合わせの変更が可能で，要求の変更に柔軟に対応できる．図 1 にパイプとフィルタのアーキテクチャ例を示す．

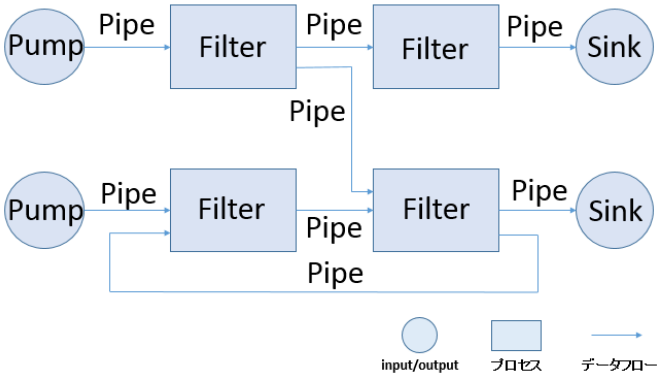

図 1　パイプとフィルタ（Pipes and Filters）のアーキテクチャ例

2.3　音楽心理学

音楽心理学は，心理学および音楽学の派生学問とみなされる．音楽的行動や音楽体験の解明と理解を目的として，音楽を聞くことによる反応を組織的に観察し，データとして収集する．その得られたデータに基づいて仮説を実証するという実験心理学である．研究対象としては以下等が挙げられる．

- 食事中，運動中など，場面に応じたリスニングの状態
- 音楽に対する感情的反応

音楽心理学について，谷口は次のように述べている [4]．

- 調と色彩の対応関係については，一般に，長調の曲を聴くと明るく楽しい気分になり，短調の曲を聴くと暗く悲しい気分になる，といった気分の変化との対応から色彩的な表象を描く場合が多い．
- 音の高さと色彩の対応関係については，高音になるに

つれて明るい色になり，低音になるにつれて暗い色になるという関係が成立する．

これらの性質を基に，色彩と音楽の対応づけを行ない，音楽の生成を行なう．

3　実時間性を考慮したアーキテクチャの設計

3.1　本研究で試作するソフトウェアの概要

環境音楽実時間生成ソフトウェアにおける，音楽生成までの工程は，つぎの 2 つに分割可能である．

- 映像解析
- 解析したデータの変換

本研究で試作するプロトタイプでは，カメラから得た映像を解析することによって，色調のデータである RGB の値，明度のデータである HSV の値を取得する．RGB はコード進行の生成に，HSV はメロディの生成に用いる．

映像解析を行なった後，取得したデータに対して音楽データを生成する．本研究の音楽生成部分では，音楽向けの統合開発環境である Max を用いる．Max の API 仕様で音楽データを標準入力として受け取るモジュールが存在することから，音楽データに変換をすることで，データ取得部分に音楽生成部分との互換性をもたせることが可能となる．

音楽データへの変換を行なった後，音楽生成部分で音楽データをもとに生成された音楽を映像と同期させることで，実時間での環境音楽の映像への合成を実現する．

我々は，プロトタイプを試作するさいの映像解析部分において，フィルタ処理に掛かる手間を大きく減らすことが可能である映像解析ライブラリである OpenCV を用いる．

そのさい，コンピュータ上で最も普及しており一般的である MIDI データを扱う．

3.2　設計指針

本研究では，実時間性を考慮するために，以下を前提にソフトウェアアーキテクチャの設計を行なう．

- つねに生成され続けるデータであるストリームデータを対象とする．
- 映像解析と，取得したデータを基にした音楽生成処理を並行に行なう．

実時間での音楽生成をするために，並行処理を前提とし，映像のデータはストリームデータとする．処理の順序変更や追加を容易にし，後からでもソフトウェアの拡張を可能にするために，ストリームデータを処理するのに適したアーキテクチャスタイルであるパイプとフィルタ（Pipes and Filters）を用いる．

実時間での音楽生成に関連して，以下の解決すべき課題がある．

- 映像解析と音楽処理に時間を要するので，映像と音楽に僅かなずれが生じる．
- 映像を音楽と同時に出力する必要がある．

本研究では，映像を短時間一時保存し，その間に音楽生成を行なうことでこの問題を解決する．

動画から音楽を生成する既存の研究では，動画全体を先に読み取ってから音楽を生成する方法が主流である．この方法を用いる利点としては，動画の長さや，動画全体の構成を考慮してから音楽生成を開始できるという利点がある．しかし，この方法ではライブ配信などのさいの音楽生成には対応できない．並行に映像解析と音楽生成処理を行なうことで，ソフトウェアに実時間性を付加でき，この問題を解決することが可能である．

3.3　実時間性を考慮した具象アーキテクチャ

フィルタの入れ替えや順序の変更を行なうことで，システムへの要求変更に柔軟に対応できる．図 2 に，我々が提案するアーキテクチャの静的構造を示す．

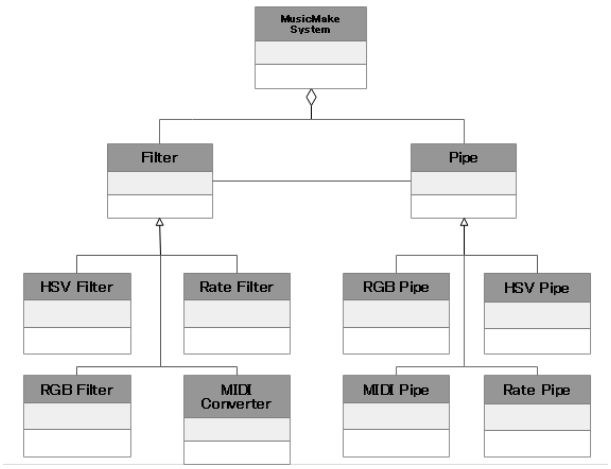

図 2　アーキテクチャの静的構造（クラス図）

映像から音楽生成に必要なデータを抽出するために，映像に関する処理を Filter の多相型で定義した．Pipe は拡張によって，Filter の変更や並び替えが必要なとき，Pipe を書き換えるだけでソフトウェアの振舞いの変更が可能になるので，多相型で定義した．各フィルタの処理内容を以下に示す．

- RGB Filter
  映像からピクセル単位で RGB の値を抽出する．
- HSV Filter
  映像からピクセル単位で HSV の値を抽出する．
- Rate Filter
  抽出した値の画面全体に対する割合を求める．
- MIDI Converter
  抽出した値に対応した MIDI データを生成する．

図 3 に，我々が提案するアーキテクチャの動的振舞いを示す．web カメラの映像データは buffer に一時保存すると同時に，RGB Filter と HSV Filter に同時に送られ，抽出した色調と明度の値が RGB Pipe と HSV Pipe を通して，

Rate Filter へ送られる．Rate Filter で求められた割合のデータは，MIDI Converter へ送られ，MIDI Converter で MIDI データに変換する．生成した MIDI データは，音楽向けの統合開発環境である Max のモジュールが標準入力として受け取り，Max は受け取った MIDI データを基に音楽の生成を行なう．音楽生成処理が完了後，生成した音楽とストレージ上に一時保存した映像を同時に再生して，音楽と映像の同期処理を可能にする．

パイプとフィルタを用いることで基本的な実時間性は担保されるが，更なる実時間の追求のために，独立する処理は並行に行なう．プロトタイプでは，RGB Filter と HSV Filter の入力がカメラからの映像であり，他のフィルタの処理結果に依存しないので，このふたつを並行に行なうことで処理時間の短縮を計る．

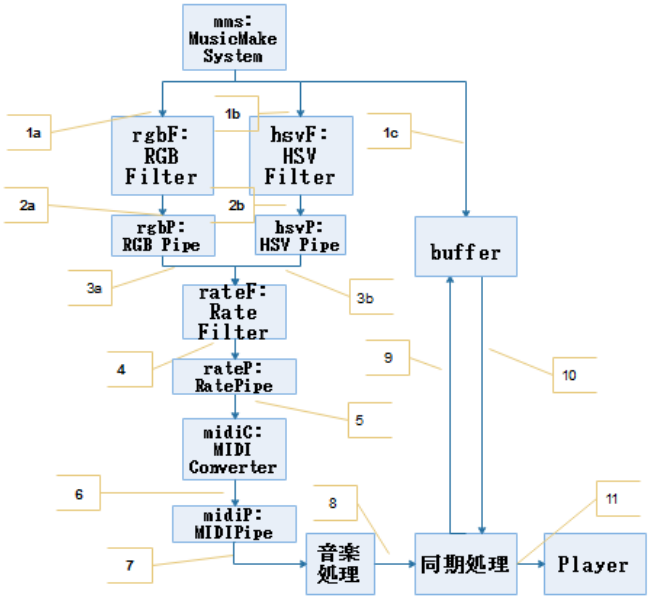

図 3　アーキテクチャの動的振舞い（コミュニケーション図）

4　MIDI データの生成

4.1　MIDI データ

MIDI (Musical Instrument Digital Interface) とは，電子楽器の演奏データを機器間でディジタル転送するための世界共通規格である．

本ソフトウェアで考慮すべき点は，以下の 2 点である．

- フィルタ間でデータの受け渡しが多いこと．
- 実時間性．

MIDI を用いることで，サンプリングした音源を用いるよりもデータ転送に掛かる時間を低減でき，加えて，対応する機器も多く，より自然な音に近づけることができる．

- ノートナンバー
  MIDI においてピアノ鍵盤に数値を当てはめて，数字で表現したものである．

4.2　コード進行の決定

本研究では，コード進行の発展の容易さを考慮し，音楽生成で最も基本となる 4 小節で，ひとつの循環コードを生成し作曲を行なう．MIDI Converter のコード生成の条件を変更することで，他の小節数の循環コードを生成することが可能である．

音楽によって人が感じる印象は非常に曖昧なものであり，計算式として定義する事は難しい．下記の場合分けの式は山本らの研究 [2] において，質問紙法を用いて最も映像と調和して聞こえるコード進行を調査し，式として表現したものである．本研究では，これに従ってコードの生成を行なう．

画面全体における RGB の R(Red) 成分と B(Blue) 成分を用いて，計算式と移行するコードを以下のように定義する．

$$
\begin{aligned}
7 < \frac{R}{B} &\quad \Rightarrow \quad 100\%\ \text{メジャーコードに移行} \\
\frac{5}{3} < \frac{R}{B} \leq 7 &\quad \Rightarrow \quad 75\% : 25\% = \text{メジャー} : \text{マイナー} \\
\frac{3}{5} < \frac{R}{B} \leq \frac{5}{3} &\quad \Rightarrow \quad 50\% : 50\% = \text{メジャー} : \text{マイナー} \\
\frac{1}{7} \leq \frac{R}{B} \leq \frac{3}{5} &\quad \Rightarrow \quad 25\% : 75\% = \text{メジャー} : \text{マイナー} \\
\frac{R}{B} < \frac{1}{7} &\quad \Rightarrow \quad 100\%\ \text{マイナーコードに移行}
\end{aligned}
$$

短期で試作して実験するために，ハ長調を対象とし，ダイアトニックコードと，ペンタトニックスケールの音を用いて構成されるコードを使用する．現在のコードがトニックコードであればトニックコードまたはサブドミナントコードに，サブドミナントコードであればトニックコードまたはドミナントコードに，ドミナントコードであればトニックコードまたはサブドミナントコードに移行する．

4.3　メロディラインの決定

音楽心理学 [4] において，明度が低ければ低い音，明度が高ければ高い音を連想することが確認されている．本研究では音楽心理学を基にするので，HSV，特に Value(明度) の値を用いてメロディラインを生成する．

画面全体における明度の値の平均を求め，その値に応じたノートナンバーを求める計算式を以下のように定める．明度平均を B，ノートナンバーの値を N とすると，

$$N = \frac{B}{10} + 60$$

コードの生成と同様に，メロディの生成式も山本らの研究に従って行なうものとする．

メロディが単調になるのを防ぐために，ノートナンバーに-2 から +2 の値をランダムで加算する．この時のノート

ナンバーの値が C メジャースケール上の音でない場合，メロディが不協和音になるのを防ぐために，ノートナンバーの値に更に +1 を加算する．

5　考察

5.1　関連研究との比較

茂出木の研究 [3] では既存の映像ファイルを全て読み込み，映像に適した音楽パターンをあらかじめ用意したデータ群から選んで合成しているのに対し，本研究では生成され続ける映像データの読み込みと音楽データの生成を随時行なう．以下に，関連研究と本研究のそれぞれの利点と欠点を述べる．

＜茂出木の研究＞

- 利点
  - あらかじめ用意した音をデータ群から選び出すので，質の高い音を鳴らし易い．
  - 映像解析に時間を掛けることで，動画の雰囲気に沿った音楽が生成し易い．
- 欠点
  - 実時間性がない．
  - データ群に無い音を鳴らすことができない．

＜本研究＞

- 利点
  - 実時間性がある．
  - 独創的な音楽が生成できる．
- 欠点
  - 実時間で音を生成するので，音質を犠牲にした．
  - 実時間性を考慮しない方法に比べて音楽を動画の雰囲気にあわせ難い．

まとめると，実時間性の観点で，本研究が勝る．

5.2　ソフトウェアアーキテクチャの有用性に関して

実時間での処理を前提としてシステムの構築を行なうさい，以下の理由からシステムの処理ステップの交換や変更が可能な柔軟性を持たせる必要がある．

- 複数の開発者によって開発が可能
- 複数の処理ステップに分割が可能
- システムへの要求を変更することが容易

これらを全て満たすアーキテクチャスタイルが Pipes and Filters アーキテクチャスタイルであり，上記の理由から本研究では Pipes and Filters アーキテクチャスタイルを用いてアーキテクチャ設計を行なった．

試作したソフトウェアが音楽生成のために用いるデータは色調と明度のデータである．データの取得範囲を変更したフィルタに付け替えることで，異なる音が生成されることを確認した．このことからソフトウェアアーキテクチャの有用性を確認できる．

フィルタ処理による処理の変更や付替えを活用することで，各用途に応じた実用化が可能である．以下にソフトウェアの拡張の例を示す．

- ジェスチャ認識による音楽生成
  本ソフトウェアを拡張することで，人物の顔の表情を読み取り，表情によって瞬間的な映像の暖かみを判断して，音楽を生成することが可能となると考える．
- 映像の場面の転換を考慮した音楽生成
  各ピクセルの RGB の値を合計した数を X とすると，この X に一定値以上の変動が起きたとき，画面が大きく転換した場面であると考え，このタイミングで転調や新しい曲を開始する．

ジェスチャ認識や場面転換を考慮した音楽生成が可能なソフトウェアに拡張することで，演劇など，動きや場面転換を多く用いるものも音楽生成の対象とすることができる．

本研究のプロトタイプでは，それぞれのフィルタに渡すデータが同じであり，処理速度が十分である．しかし，以下のような場合等に，より実時間に処理を行なうために優先度に着目してスケジューリングを行なう必要がある．

- 並行に実行されるフィルタの結果をまとめて特定のパートの音楽を生成する場合
- CPU 性能や言語仕様によって，同時に実行可能な処理の数に限度がある場合
- フィルタの処理速度に大きな差がある場合

6　おわりに

本研究では，動画情報に基づく環境音楽実時間生成ソフトウェアアーキテクチャの設計を行ない，設計したアーキテクチャに基づきプロトタイプを試作した．試作したプロトタイプのフィルタを，データ取得範囲を変更したものに付け替えることで，異なる音が生成されることから，ソフトウェアアーキテクチャの有用性を確認した．今後の課題としては，音楽生成及び各データの取得に，ユーザの嗜好性を考慮することが挙げられる．これにより，幅広いユーザのニーズに答えられるソフトウェアの実現が可能となる．加えて，パイプラインのスケジューリングについて考慮していく必要がある．

参考文献

[1] Buschmann, F., MeunierHans, R., Rohnert, H., Sommerlad, P., and Stal, M.:*Architecture, A System of Patterns: Vol. 1*, WILEY, 1996.

[2] 桑田和也, 宝珍輝尚: 視聴覚素材における音と動画の調和について, 日本感性工学会論文誌, 2009.

[3] 茂出木敏雄, 映像コンテンツ解析による BGM サウンドトラック自動生成, *IEEJ Trans.* Vol. 125, No.7 (p1004-1010), 2005.

[4] 谷口高士, 音は心の中で音楽になる音楽心理学への招待, 北大路書房, 2000.

[5] 山本敏夫, 宝珍輝尚, 野宮浩揮: 動画をもとにした自動作曲, 情報処理学会関西支部大会, 2009.