

CORBA を利用した画像データベースにおける 処理履歴の再利用

田幡 勝 深津 博樹 有次 正義 金森 吉成

群馬大学工学部情報工学科

1 はじめに

我々は、画像データベース内の画像に対し一連の画像処理を実行することによって生じる画像処理結果および画像処理履歴を版として管理するモデルを提案している^[1]。本論文では、ある画像で得た処理履歴を別の画像に対して再利用する方法およびその実装に焦点をあてて議論する。この版管理機能は、分散環境での画像データベースの利用を考慮し、分散オブジェクト管理技術 CORBA^[3] を用いて実装される。

2 分散環境下での版管理の概要

画像データベースに格納されている画像データを、画像データオブジェクト (Image Data Object, IDO) とする。画像データオブジェクトに対し画像処理を実行するオブジェクトが、画像処理オブジェクト (Image Processing Object, IPO) である。これらのオブジェクトは、ネットワーク上に分散して実装される^[1, 2]。これらの分散したオブジェクトの操作は汎画像オブジェクト (Generic Image Object, GIO) を通じて行われる。アプリケーションプログラムが画像処理の依頼を GIO へ行ったとき、GIO がさらに IPO へ画像処理の依頼を行う。このとき、アプリケーション側からは、GIO が画像処理を実行しているように見える。そして、一連の画像処理を実行した結果生じた画像処理結果および画像処理履歴を GIO が管理する。

これらの分散したオブジェクトに本研究では CORBA を用いて対処する。IDO, IPO および GIO に対し、OMG IDL^[3] を用いてインタフェースをそれぞれ定義する^[2]。このインタフェース定義に従って、図 1 の各モジュール間で ORB を通じたメッセージ送信が行われる。

3 画像処理履歴の管理

対話的に画像処理を実行するとき、GIO は次の目的に対して画像処理結果および画像処理履歴を管理する。

- 一連の画像処理過程における途中結果を版として管理し、特定の時点の画素値データを再利用する。
- 画像処理履歴の導出手順を再利用する。

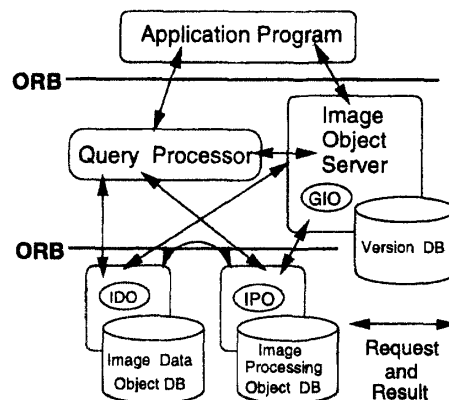


図 1: アーキテクチャの概要

この二つの目的に対し、版の導出に関する情報を表現するために version_of オブジェクトを導入する。このオブジェクトの主な属性を以下に挙げる。

- 版の導出元となる version_of オブジェクトへの参照のリスト。
- 版の導出先となる version_of オブジェクトへの参照のリスト。
- 版を導出したとき用いた画像処理オブジェクト名。
- 版を導出したとき用いた画像処理のパラメータ。
- 版を識別するために用いる名前。
- IDO への参照または数値データ。この属性の示す値が、版として用いられる。

一連の画像処理を実行した結果、図 2 の左側のような構造が得られる。このように GIO が処理過程における途中結果を管理することで、特定の時点での画像の状態を再利用することを可能にしている。

IPO DB に格納されている IPO は、ある画像処理演算をメソッドとして保持しており、画像処理の依頼があったときその画像処理演算を実行し結果を返す。一方、一つの OMG IDL インタフェースに対して複数の実装が可能であることから、IPO のインタフェースに対して全く別の実装を与えることも可能である。そこで、処理履歴の再利用を行うために、version_of オブジェクトによって管理された画像処理の導出手順の情報を属性として保持した IPO も実装する。この IPO は、図 2 の右側に示すように、アプリケーションプログラムが GIO に対して IPO の生成を依頼したとき、その GIO が管理する版の導出履歴の属性に従って生成され、version DB に保存される。

Reusing Version Histories on Image Databases under a CORBA Environment

Masaru Tabata, Hiroki Fukatsu, Masayoshi Aritsugi, Yoshinari Kanamori

Department of Computer Science, Gunma University

このとき、GIOによって途中結果として管理されていたIDOは処理履歴を再利用する際に必要がないため破棄される。このIPOに画像処理の依頼があったとき、画像処理履歴の属性に基づいてIPO DB内のIPOを検索して、それに画像処理を依頼し画像処理履歴を再現する。この処理履歴に基づくIPOは、IPO DBに格納されている個別の画像処理に相当するIPOとは内部の実装が異なっている。しかし、それぞれのIPOのインターフェースはIDLによって共通のものが定義されている。このため、GIOはそれぞれのIPOの実装の違いを意識することなく、これらを同一の種類オブジェクトとして扱うことができる。

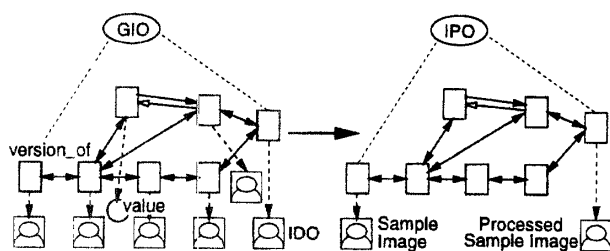


図 2: 処理履歴を保持する画像処理オブジェクトの生成

4 画像処理履歴の再利用

version DBに保存された処理履歴を画像処理オブジェクトとして再利用する手順を図3に基づき以下に示す。

1. アプリケーションプログラムが、画像処理 (IPO-A) のリクエストをGIOへ送る。
2. GIOは、currentのversion_ofオブジェクトが保持するIDO-Aの参照を受け取る。
3. GIOは、IPO-AにIDO-Aへの画像処理の実行を依頼する。
4. 画像オブジェクトサーバ内で画像処理手順を保持するIPO-Aは、画像処理の依頼を受信したとき、点線部分に示す自分自身のコピーIPO-Bを画像処理オブジェクトサーバに生成し、IPO-Bに画像処理の依頼を転送する。
5. IPO-Bは、自分自身が保持するversion_ofオブジェクトの管理する画像処理手順の情報に基づいて、IPO DBに格納されたIPO(この場合、dither)にIDO-Aへの画像処理の実行を依頼する。
6. IPO DB側のIPOは、画像処理を実行するためにIDO-Aの保持する画像データを受け取り、画像処理を実行する。その結果、画像処理手順の途中結果IDO-Bが画像処理オブジェクトサーバ内に生成される。以下、IPO-Bが保持する画像処理履歴の情報に従って5~6と同様の操作(5b~6bおよび5c~6c)を繰り返し実行する。
7. 全ての画像処理手順を完了したとき、IPO-Bは途中結果として生成したIDO-BとIDO-Cを削除し、画像処理の最終結果をIPO-Aに返す。

8. IPO-Aは、画像処理の結果をGIOへ返す。
9. GIOは、画像処理の結果を基に新たな版としてIDO-Dを画像オブジェクトサーバ上に生成する。また、そのIDOがGIOの版であるという関係を表現するために、version_ofオブジェクトの生成を行う。
10. GIOは、アプリケーション側に新たに生成された版の名前を返す。

処理履歴を保持するIPOのコピーをIPO DB側に作成することで、IOSとIPO DB間での画像処理の途中結果の転送を回避している。これは、長い履歴を再現するときや画像データのサイズが大きな場合に有効となる。

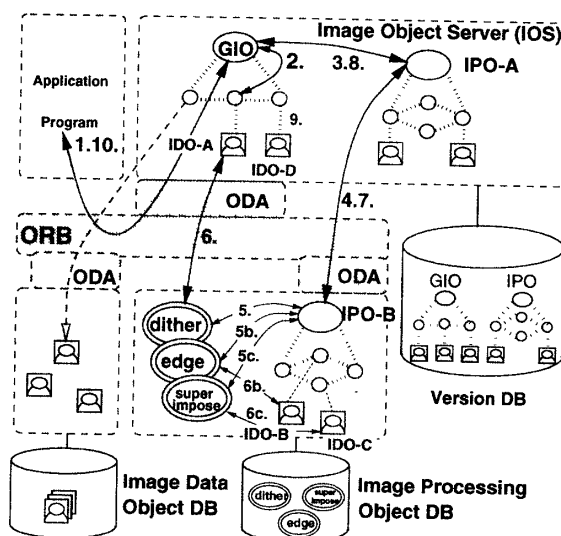


図 3: 画像処理履歴の再利用

5 おわりに

本論文では、画像処理の履歴管理を行うことによって画像処理履歴を再利用できることを示した。CORBAを用いることによって、画像処理履歴の構造を画像処理オブジェクトとして扱えることが可能になっている。

謝辞

本研究の一部は、文部省科学研究費補助金特別研究員奨励費(受付番号3473)による。

参考文献

- [1] 川島亨, 田幡勝, 金森吉成, 増永良文. “画像オブジェクトの版管理モデル”, 電子情報処理学会論文誌 D1, vol.J79-D-I, no.10, pp.843-852, Oct. 1996.
- [2] 田幡勝, 深津博樹, 有次正義, 金森吉成. “分散環境における画像オブジェクトの版管理の設計”, 第8回データ工学ワークショップ(DEWS'97), pp.179-184, Mar. 1997.
- [3] “The Common Object Request Broker: Architecture and Specification”, Revision 2.0. Object Management Group, 1995.