LIRe 源代码分析 4：建立索引（DocumentBuilder）[以颜色布局为例]

2013年11月02日 16:45:49　阅读数：5235

===

LIRe源代码分析系列文章列表：

LIRe 源代码分析 1：整体结构

LIRe 源代码分析 2：基本接口（DocumentBuilder）

LIRe 源代码分析 3：基本接口（ImageSearcher）

LIRe 源代码分析 4：建立索引（DocumentBuilder）[以颜色布局为例]

LIRe 源代码分析 5：提取特征向量[以颜色布局为例]

LIRe 源代码分析 6：检索（ImageSearcher）[以颜色布局为例]

LIRe 源代码分析 7：算法类[以颜色布局为例]

===

前几篇文章介绍了LIRe 的基本接口。现在来看一看它的实现部分，本文先来看一看建立索引（（DocumentBuilder））部分。不同的特征向量提取方法的建立索引的类各不相同，它们都位于“net.semanticmetadata.lire.impl”中，如下图所示：

由图可见，每一种方法对应一个DocumentBuilder和一个ImageSearcher，类的数量非常的多，无法一一分析。在这里仅分析一个比较有代表性的：颜色布局。

颜色直方图建立索引的类的名称是ColorLayoutDocumentBuilder，该类继承了AbstractDocumentBuilder，它的源代码如下所示：

[java]

```java
/*
 * This file is part of the LIRe project: http://www.semanticmetadata.net/lire
 * LIRe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * LIRe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with LIRe; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 *
 * We kindly ask you to refer the following paper in any publication mentioning Lire:
 *
 * Lux Mathias, Savvas A. Chatzichristofis. Lire: Lucene Image Retrieval 鈥�
 * An Extensible Java CBIR Library. In proceedings of the 16th ACM International
 * Conference on Multimedia, pp. 1085-1088, Vancouver, Canada, 2008
 *
 * http://doi.acm.org/10.1145/1459359.1459577
 *
 * Copyright statement:
 * --------------------
 * (c) 2002-2011 by Mathias Lux (mathias@juggle.at)
 *     http://www.semanticmetadata.net/lire
 */
package net.semanticmetadata.lire.impl;

import net.semanticmetadata.lire.AbstractDocumentBuilder;
import net.semanticmetadata.lire.DocumentBuilder;
import net.semanticmetadata.lire.imageanalysis.ColorLayout;
import net.semanticmetadata.lire.utils.ImageUtils;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;

import java.awt.image.BufferedImage;
import java.util.logging.Logger;

/**
 * Provides a faster way of searching based on byte arrays instead of Strings. The method
 * {@link net.semanticmetadata.lire.imageanalysis.ColorLayout#getByteArrayRepresentation()} is used
 * to generate the signature of the descriptor much faster.
 * User: Mathias Lux, mathias@juggle.at
 * Date: 30.06.2011
 */
public class ColorLayoutDocumentBuilder extends AbstractDocumentBuilder {
    private Logger logger = Logger.getLogger(getClass().getName());
    public static final int MAX_IMAGE_DIMENSION = 1024;

    public Document createDocument(BufferedImage image, String identifier) {
        assert (image != null);
        BufferedImage bimg = image;
        // Scaling image is especially with the correlogram features very important!
        // All images are scaled to guarantee a certain upper limit for indexing.
        if (Math.max(image.getHeight(), image.getWidth()) > MAX_IMAGE_DIMENSION) {
            bimg = ImageUtils.scaleImage(image, MAX_IMAGE_DIMENSION);
        }
        Document doc = null;
        logger.finer("Starting extraction from image [ColorLayout - fast].");
        ColorLayout vd = new ColorLayout();
        vd.extract(bimg);
        logger.fine("Extraction finished [ColorLayout - fast].");

        doc = new Document();
        doc.add(new Field(DocumentBuilder.FIELD_NAME_COLORLAYOUT_FAST, vd.getByteArrayRepresentation()));
        if (identifier != null)
            doc.add(new Field(DocumentBuilder.FIELD_NAME_IDENTIFIER, identifier, Field.Store.YES, Field.Index.NOT_ANALYZED));

        return doc;
    }
}
```

从源代码来看，其实主要就一个函数：createDocument(BufferedImage image, String identifier)，该函数的流程如下所示：

1.如果输入的图像分辨率过大（在这里是大于1024），则将图像缩小。

2.新建一个ColorLayout类型的对象vd。

3.调用vd.extract()提取特征向量。

4.调用vd.getByteArrayRepresentation()获得特征向量。

5.将获得的特征向量加入Document，返回Document。

其实其他方法的DocumentBuilder的实现和颜色直方图的DocumentBuilder差不多。例如CEDDDocumentBuilder的源代码如下所示：

[java]

```java
/*
 * This file is part of the LIRe project: http://www.semanticmetadata.net/lire
 * LIRe is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.
 *
 * LIRe is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with LIRe; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 *
 * We kindly ask you to refer the following paper in any publication mentioning Lire:
 *
 * Lux Mathias, Savvas A. Chatzichristofis. Lire: Lucene Image Retrieval 鈥�
 * An Extensible Java CBIR Library. In proceedings of the 16th ACM International
 * Conference on Multimedia, pp. 1085-1088, Vancouver, Canada, 2008
 *
 * http://doi.acm.org/10.1145/1459359.1459577
 *
 * Copyright statement:
 * ~~~~~~~~~~~~~~~~~~~~
 * (c) 2002-2011 by Mathias Lux (mathias@juggle.at)
 *     http://www.semanticmetadata.net/lire
 */
package net.semanticmetadata.lire.impl;

import net.semanticmetadata.lire.AbstractDocumentBuilder;
import net.semanticmetadata.lire.DocumentBuilder;
import net.semanticmetadata.lire.imageanalysis.CEDD;
import net.semanticmetadata.lire.utils.ImageUtils;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;

import java.awt.image.BufferedImage;
import java.util.logging.Logger;

/**
 * Provides a faster way of searching based on byte arrays instead of Strings. The method
 * {@link net.semanticmetadata.lire.imageanalysis.CEDD#getByteArrayRepresentation()} is used
 * to generate the signature of the descriptor much faster.
 * User: Mathias Lux, mathias@juggle.at
 * Date: 12.03.2010
 * Time: 13:21:35
 *
 * @see GenericFastDocumentBuilder
 * @deprecated use GenericFastDocumentBuilder instead.
 */
public class CEDDDocumentBuilder extends AbstractDocumentBuilder {
    private Logger logger = Logger.getLogger(getClass().getName());
    public static final int MAX_IMAGE_DIMENSION = 1024;

    public Document createDocument(BufferedImage image, String identifier) {
        assert (image != null);
        BufferedImage bimg = image;
        // Scaling image is especially with the correlogram features very important!
        // All images are scaled to guarantee a certain upper limit for indexing.
        if (Math.max(image.getHeight(), image.getWidth()) > MAX_IMAGE_DIMENSION) {
            bimg = ImageUtils.scaleImage(image, MAX_IMAGE_DIMENSION);
        }
        Document doc = null;
        logger.finer("Starting extraction from image [CEDD - fast].");
        CEDD vd = new CEDD();
        vd.extract(bimg);
        logger.fine("Extraction finished [CEDD - fast].");

        doc = new Document();
        doc.add(new Field(DocumentBuilder.FIELD_NAME_CEDD, vd.getByteArrayRepresentation()));
        if (identifier != null)
            doc.add(new Field(DocumentBuilder.FIELD_NAME_IDENTIFIER, identifier, Field.Store.YES, Field.Index.NOT_ANALYZED));

        return doc;
    }
}
```

文章标签： lire 源代码 索引 检索 lucene

个人分类： LIRe MPEG7/图像检索

所属专栏： 开源多媒体项目源代码分析