Validation and Decomposition of Partially Occluded Images with Holes

Julien Allali[1], Pavlos Antoniou[2], Costas S. Iliopoulos[2], Pascal Ferraro[1], Manal Mohamed[2]

[1] LaBRI, University of Bordeaux I, UMR5800, 33405 Talence, France
[2] Dept. of Computer Science, King's College London, London WC2R 2LS, England, UK

Partially occluded images

- A partially occluded image consists of a set of objects where some may be partially occluded by others.
- The algorithm presented here validates a one-dimensional image x of length n, over a given set of objects all of equal length and each composed of two parts separated by a transparent hole
- We want to "*cover*" a string using a set of "*objects*". These objects may "*occlude*" each other and may be separated by a hole

Well studied problem

- Validating partially occluded images is a classical problem in computer vision and its computational complexity is exponential.
- Iliopoulos and Simpson focused on the theoretical aspect of the problem and produced a sequential on-line algorithm for validating occluded one-dimensional images
- Iliopoulos and Reid provided a linear time solution to the problem in the presence of errors
- They also presented an optimal $O(\log \log n)$-time algorithm using parallel computation and solved the problem for discrete two-dimensional partially occluded images in linear time

Contribution of this work

- Based on the above analyses, we extend the previous work by considering the validity of a family of images, that we call *valid images with holes*.
- Given a set of objects $s_1, \dots s_k$, each composed of two parts separated by a small transparent hole, an image x of length n is a valid image with hole, if x is iteratively obtained from a string $z = \#^n$ by substituting substrings of z by some objects s_i , for some $i \in \{1..k\}$ and a special "background" symbol $\#$.
- We focus on designing an on-line algorithm for testing images in one dimension for validity, with restricted set of objects, e.g., objects of the same length, that are consisting of two parts separated by a hole of small size.

Basic Definitions

Valid Image over set of Objects:

Definition

Let x be a string of length n over an alphabet Σ and let the dictionary $\mathcal{O} = \{s_1, \ldots, s_m\}$ be a set of strings called the objects *also over Σ. Then x is called a* valid image *if and only if $x = z_i$ for some $i \geq 0$, where*

$$\begin{aligned} z_0 &= \#^n \\ z_{i+1} &= prefix_p(z_i)\ s_l\ suffix_q(z_i)\,. \end{aligned} \tag{1}$$

for some $s_l \in \mathcal{O}$ and $p, q \in \{0, \ldots, n-1\}$ such that $p + |s_m| + q = n$. □

Basic Definitions

- Previous equation is called the *substitution rule* and the sequence $z_0, z_1, \ldots, z_i$ is called the *generating sequence* of x
- The number of distinct generating sequences was proved to be exponential

Example of generating sequence

An example of such generating sequences for a specific string is as follows. Let $\mathcal{O} = \{s_1 = abc, s_2 = acde, s_3 = ade, s_4 = dc, s_5 = abd\}$. Then $x = abababacdedcdcade$ is a valid image over $\mathcal{O}$ with generating sequence:

$$z_0 = \#^{17},$$
$$z_1 = \underline{abc}\#^{14},$$
$$z_2 = abc\#^{11}\underline{ade},$$
$$z_3 = ab\underline{abc}\#^{9}ade,$$
$$z_4 = abab\underline{abc}\#^{7}ade,$$
$$z_5 = ababab\underline{acde}\#^{4}ade,$$
$$z_6 = abababacde\underline{dc}\#^{2}ade,$$
$$z_7 = abababacdedc\underline{dc}ade.$$

Not Uique

Note that the generating sequence of x is not unique. The following sequence:

$$z_0 = \#^{17},$$
$$z_1 = \underline{abd}\#^{14},$$
$$z_2 = ab\underline{abc}\#^{12},$$
$$z_3 = abab\underline{abc}\#^{10},$$
$$z_4 = abababc\#^{7}\underline{ade},$$
$$z_5 = abababc\#^{3}\underline{dc}\#^{2}ade,$$
$$z_6 = abababc\#^{3}dc\underline{dc}ade,$$
$$z_7 = ababab\underline{acde}dcdcade.$$

also generates x as a valid image over $\mathcal{O}$.

Valid Image with Hole

Valid Image over Set of Objects with Hole:
Let x *be a string of length* n *over an alphabet* Σ *and let the dictionary* $\mathcal{O} = \{s_1, \ldots, s_k\}$ *be a set of strings called the* objects, *where each object* s_i *is composed of two strings* s_i^l *and* s_i^r *separated by a hole of length* h. *Then* x *is called a* valid image *if and only if* $x = z_i$ *for some* $0 \leq i$, *where*

$$\begin{aligned} z_0 &= \#^n \\ z_{i+1} &= \mathrm{prefix}_p(z_i)\; s_m \;\mathrm{suffix}_q(z_i)\,. \end{aligned} \tag{2}$$

for some $s_m \in \mathcal{O}$ *and* $p, q \in \{0, \ldots, n-1\}$ *such that* $p + |s_m| + q = n$.

Example of Image with objects with hole

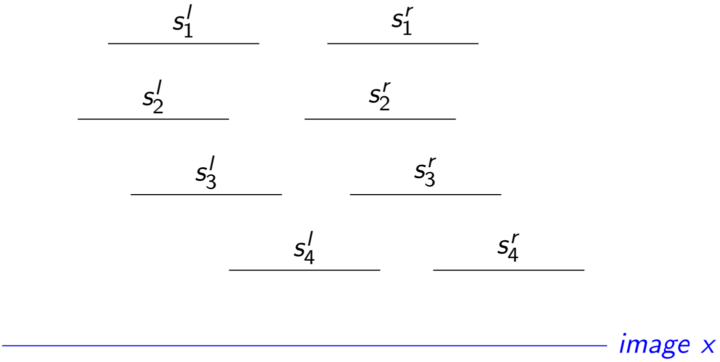

Figure: Image consisting from objects separated by a hole of same length.

Image =
$prefix(s_2^l)\ s_1^l\ suffix(s_3^l)\ substring(s_4^l)\ prefix(s_2^r)\ s_1^r\ suffix(s_3^r)\ suffix(s_4^r)$

Left Part, Hole, Right Part

- Each object $s_i \in \mathcal{O}$ consists of a *left part* (*head*) and a *right part*(*tail*) separated by a transparent hole of length h.
- We denote the left part of s_i as s_i^l and the the right part as s_i^r. For simplicity, we require that $|s_i^l| = |s_i^r|$ and $h \ll |s_i^l|$, for each $s_i \in \mathcal{O}$.

Definition of Valid Image

If x is a valid image over $\mathcal{O} = \{s_1, s_2, \ldots, s_k\}$, then for some $i \in \{1, \ldots, k\}$,

Fact 1: there exists a suffix $\bar{s}_i^r$ of s_i^r that is also a suffix of x.

Fact 2: there exists a prefix $\hat{s}_i^l$ of s_i^l that is also a prefix of x.

Fact 3: there is no suffix of a left part s_i^l that occurs in x ending at position ℓ, where $\ell > n - h - |s_i^r|$.

Fact 4: there is no prefix of a right part s_i^r that occurs in x at position ℓ', where $\ell' < |s_i^l| + h$.

Binding

Given a set of objects $\mathcal{O}$, a string b of length h is a *binding* if it is a concatenation of the following three (possibly empty) parts:

1. **Part 1:** is a sequence of suffixes of left/right parts of objects in $\mathcal{O}$,where the leading (first) suffix is a suffix of a left part of an object.
2. **Part 2:** is a substring of a left/right part of an object.
3. **Part 3:** is a sequence of prefixes of left/right parts of objects in $\mathcal{O}$,where the leading (last) prefix is a prefix of a right part of an object

Theorem

Theorem

The string x is a valid image over $\mathcal{O}$ if and only if

$$x = \hat{s}_i^l y \quad \text{with} \quad i \in \{1..k\}, \tag{3}$$

or

$$x = y\bar{s}_i^r \quad \text{with} \quad i \in \{1..k\}, \tag{4}$$

or

$$x = y\widetilde{s}_i w \quad \text{with} \quad i \in \{1..k\}, \tag{5}$$

or

$$x = y\bar{s}_i^l b\hat{s}_i^r z \quad \text{with} \quad i \in \{1..k\}, \tag{6}$$

where $\hat{s}_i^l$, $\bar{s}_i^r$ and $\widetilde{s}_i$ denote a prefix of the left part s_i^l, suffix of the right part s_i^r and a substring of either parts of s_i respectively, y and w are valid images and b is a satisfied binding.

Theorem

- The above theorem provides the main mechanism for validating images over a set of objects with holes and all of equal length
- Based on definitions and theorem we present the algorithm for validating an image over a set of objects with holes and of equal length item The algorithm is also based on the following principles:

On Line Algorithm

(a) The occurrence of a proper prefix of either a left or a right part of an object in a valid image must be followed by a prefix (not necessarily proper) of a left or a right part of an object.

(b) If the occurrence of a proper prefix of either a left or a right part of an object is followed by an occurrence of a proper suffix of either a left or a right part of an object, then the image is not valid. In a valid image, the occurrence of a proper suffix of an object is always preceded by the suffix of either a left or a right part of an object.

(c) The occurrence of a suffix of either a left or a right part of an object can be followed by either a prefix or a substring or a suffix.

On Line Algorithm

(d) If an occurrence of a suffix of a left part of an object is not followed by either an occurrence of a prefix of its corresponding right part in a distance h or an occurrence of a prefix of a left part of an object in a distance at most h, then the image is not valid. In both cases a satisfied binding should separate the two parts.

(e) The occurrence of a substring in a valid image may be preceded by and followed by valid images.

Preprocessing stage

1. We preprocess the set of objects.
2. We compute the suffix tree of the set of the left and right parts of all objects in $\mathcal{O}$. This data structure will allow us to perform a constant time on-line checks whether a suffix, or a substring of s_j^l/s_j^r occurs in any position of x.
3. We will also build the Aho-Corasick automaton for the set of the left and right parts of all objects in $\mathcal{O}$ that will allow us to compute the largest prefixes of s_j^l/s_j^r occurring in x.

Main Algorithm

Let $\hat{s}_j^l = x[\ell..i]$ be the longest prefix of a left part of an object in $\mathcal{O}$ that is also a suffix of $x[1..i]$. A prefix of a left part of an object is preceded by either a valid image, or a proper prefix of left/right part an object or a substring of an object.

Main Algorithm

- If *valid*$[\ell - 1]$ is marked *TRUE*, then $x[1..\ell - 1]$ is a valid image and position ℓ could be the beginning of a valid sub-image, thus we mark *prefix*$[i]$ = *TRUE*, *first-prefix* $= \ell$ and *last-prefix* $= i$.
- If *prefix*$[\ell - 1]$ is marked *TRUE*, then we have a chain of prefixes, thus we mark *prefix*$[i]$ = *TRUE* and *last-prefix* $= i$.
- If there is no prefix of a left/right part of an object or a valid image preceding $\hat{s}^l_j$, then $x[1..i]$ is valid if and only if $x[\textit{previous-valid}[\ell - 1] + 1..\ell - 1]$ is a substring of left/right part of an object or $x[\textit{previous-valid}[\ell - 1] + 1..i]$ is a prefix of a satisfied binding. If $x[\textit{previous-valid}[\ell - 1] + 1..\ell - 1]$ is a substring then ℓ is the start of a valid image.

Main Algorithm

Let $\hat{s}_j^r = x[\ell..i]$ be the longest prefix of a right part of an object in $\mathcal{O}$ that is also a suffix of $x[1..i]$. Similarly, a prefix of an object is preceded by either a proper prefix of left/right part an object or a substring of an object.

Main Algorithm

- If *prefix*$[\ell-1]$ is marked *TRUE* and *first-prefix* $\leq \ell - h + |s_j^r|$, then we have a chain of prefixes thus we mark *prefix*$[i]$ = *TRUE* and *last-prefix* $= i$. If $\hat{s}_j^r = s_j^r$ (a complete left part), then $x[1..i]$ is a valid image and we mark the relevant array as *TRUE*.
- If *l-suffix*$[j][\ell-h-1]$ is marked *TRUE* and $x[\ell-h..\ell-1]$ is a satisfied binding then we have a prefix of a valid image (Eq. (6)), thus we mark *prefix*$[i]$ = *TRUE* and *last-prefix* $= i$. If $\hat{s}_j^r = s_j^r$ (a complete left part), then $x[1..i]$ is a valid image and we mark the relevant array as *TRUE*.

Let $\bar{s}_j^l = x[\ell..i]$ be the longest suffix of a left part of an object in $\mathcal{O}$ that is also a suffix of $x[1..i]$. If *valid*$[\ell-1]$ then *l-suffix*$[j][i]$ is marked *TRUE*.

Finally, let $\bar{s}_j^r = x[\ell..i]$ be the longest suffix of a right part of an object in $\mathcal{O}$ that is also a suffix of $x[1..i]$. Note that, in a valid image, a suffix $\bar{s}_j^r$ is always preceded by a valid image.

- If *previous-valid*$[\ell - 1] \geq \ell - 1$, then $x[1..i]$ is valid.
- If there is no valid image preceding $\bar{s}_j^r$, then $x[1..i]$ is valid if and only if the length of $i-$*previous-valid*$[\ell - 1] < |s_j|$.

Theorem

Theorem

Algorithm 1 validates an image x over a set $\mathcal{O}$ of objects of equal length and all and each composed of two parts separated by a hole in linear $O(|x| + |\mathcal{O}|)$ time.

Proof

Proof.

The construction of the Aho-Corasick automaton and the suffix tree of the dictionary $\mathcal{O}$ both require $O(|\mathcal{O})|$ time.
At Stage i, finding the largest suffix that is a prefix of some part of an object requires constant time. At Stage $i-1$, we have traced on the Aho-Corasick automaton the largest prefix of a part of an object that is a suffix of $x[1..i-1]$; on Stage i, we can either extend this prefix with one symbol, $x[i]$, or we can follow the failure link that lead to the largest such prefix. $\square$

Conclusion

As future work, the algorithm may be modified in order to deal with a set of objects of different lengths. Another interesting problem is the computation of the depth of an object in an image, *i.e.* the number of rules applied after the placement of an object in an image.

Thank you

Thank you for your attention.