Chapter 7

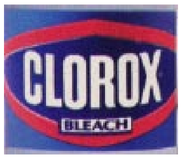
The SEDL Image Retrieval System

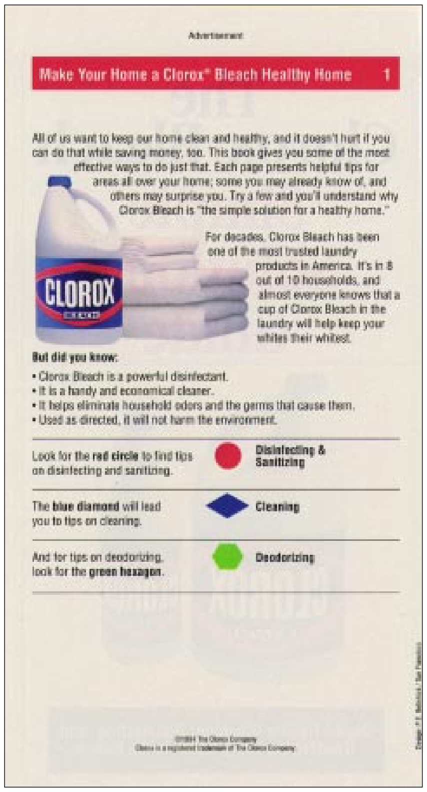
The goal of our content-based retrieval system is to find quickly all database images that contain a region similar to a given query pattern. In other words, processing a query consists of solving many pattern problems with the same pattern input. We will illustrate the general strategy described in this chapter with two specific databases: (1) a color database of scanned product advertisements, and (2) a shape database of Chinese character bitmaps. For the color database, the query patterns are product logos. The goal is to find all the ads for the given product and for products with a similar logo. For the shape database, the query patterns are characters that occur repeatedly within other Chinese characters. The goal is to find all the characters that contain strokes similar to those in the query. Examples of a query pattern and a database image that should be returned for the query are shown in Figure 7.1 for both the color and the shape case.

Our image retrieval system is called SEDL, which stands for Scale Estimation for Directed Location. As its name implies, SEDL performs a directed (as opposed to exhaustive) pattern search once it has computed an estimate for the size at which the pattern might occur within an image. If we imagine a search rectangle moving and changing size and orientation over the image, then this rectangle will eventually converge or settle (SEDL) on the image area where the system believes the pattern exists. A few promising initial placements of search rectangles are efficiently computed at query time without having to examine image areas that obviously do not contain the pattern. The initial size of the search rectangles is determined by the scale estimate.

In the shape pattern problem solved by SEDL, images (and patterns) are sets of curves such as might be produced by edgel detection and linking, or by any standard drawing program. For our Chinese character database, the first preprocessing step is to reduce each bitmap character to a set of curves by computing its medial axis.

(a)





Advertisement

Make Your Home a Clorox® Bleach Healthy Home 1

All of us want to keep our home clean and healthy, and it doesn't hurt if you can do that while saving money, too. This book gives you some of the most effective ways to do just that. Each page presents helpful tips for areas all over your home; some you may already know of, and others may surprise you. Try a few and you'll understand why Clorox Bleach is "the simple solution for a healthy home."

For decades, Clorox Bleach has been one of the most trusted laundry products in America. It's in 8 out of 10 households, and almost everyone knows that a cup of Clorox Bleach in the laundry will help keep your whites their whitest.

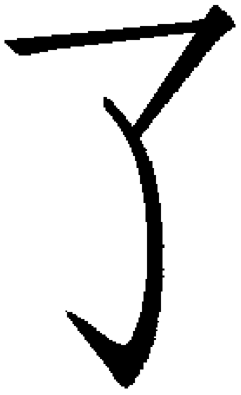
But did you know:

- Clorox Bleach is a powerful disinfectant.
- It is a handy and economical cleaner.
- It helps eliminate household odors and the germs that cause them.
- Used as directed, it will not harm the environment.

Look for the **red circle** to find tips on disinfecting and sanitizing.	●	**Disinfecting & Sanitizing**
The **blue diamond** will lead you to tips on cleaning.	◆	**Cleaning**
And for tips on deodorizing, look for the **green hexagon**.	⬢	**Deodorizing**

©1994 The Clorox Company
Clorox is a registered trademark of The Clorox Company

(b)



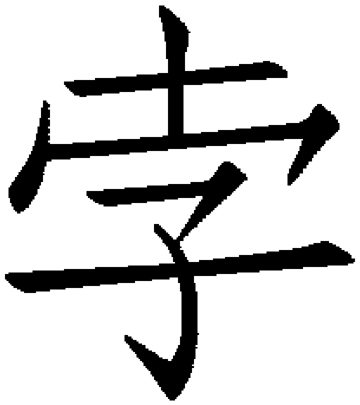


Figure 7.1: Query Patterns and Related Database Images. Here we show examples of a query pattern and a database image that should be retrieved in (a) the color advertisement database, and (b) the shape Chinese character database.

The image signature used in SEDL is a distribution that records what attributes occur where in an image, as well as the amount of the attribute present at the recorded location. For the advertisement database, the attribute is color; for the Chinese character database, the attribute is orientation (of ink along image curves). In general, a pattern "occurs" in an image to the extent that there is a transformation of the pattern attribute locations that aligns similar attributes in the query and image. In the color case, we try to align uniform color regions in the pattern and query with similar colors. SEDL's signature distance function will be small whenever each pattern region is close to an image region of similar color. In the shape case, we try to align pieces of pattern curves with pieces of image curves of similar orientations. SEDL's signature distance function will be small whenever ink along pattern curves is close to ink along image curves of similar orientation. The image signatures and signature distance function used in SEDL are the subject of section 7.1.

There are three phases in SEDL: (1) scale estimation, (2) initial placement selection, and (3) match refinement and verification. See Figure 7.2. In the first phase, only the image and query attributes (and not their locations in the image plane) are used to estimate the scale at which the pattern might occur in the image. The scale estimate is computed using the EMD in attribute space between the image and query signatures marginalized over position. The scale estimation algorithm is discussed in section 7.2. The goal of the initial placement phase is to identify efficiently a handful of promising image regions where the pattern is likely to occur. An image region is "promising" if its color signature is similar to that of the pattern. The size of the regions examined is determined by the scale estimate returned by the previous phase. Only the coarse position information of whether or not an attribute is located in a particular region is used during this phase. The initial placement phase is described in section 7.3. Finally, for each initial placement of the query at the estimated scale, we check for positional consistency of the attributes, modifying the attribute locations by some transformation if this will help improve the match. This last refinement and verification stage is the subject of section 7.4. We will describe the three phases in SEDL mainly as applied to the color pattern problem. The chapter concludes with results of the entire matching process in the advertisement database (section 7.5.1) and the Chinese character database (section 7.5.2).

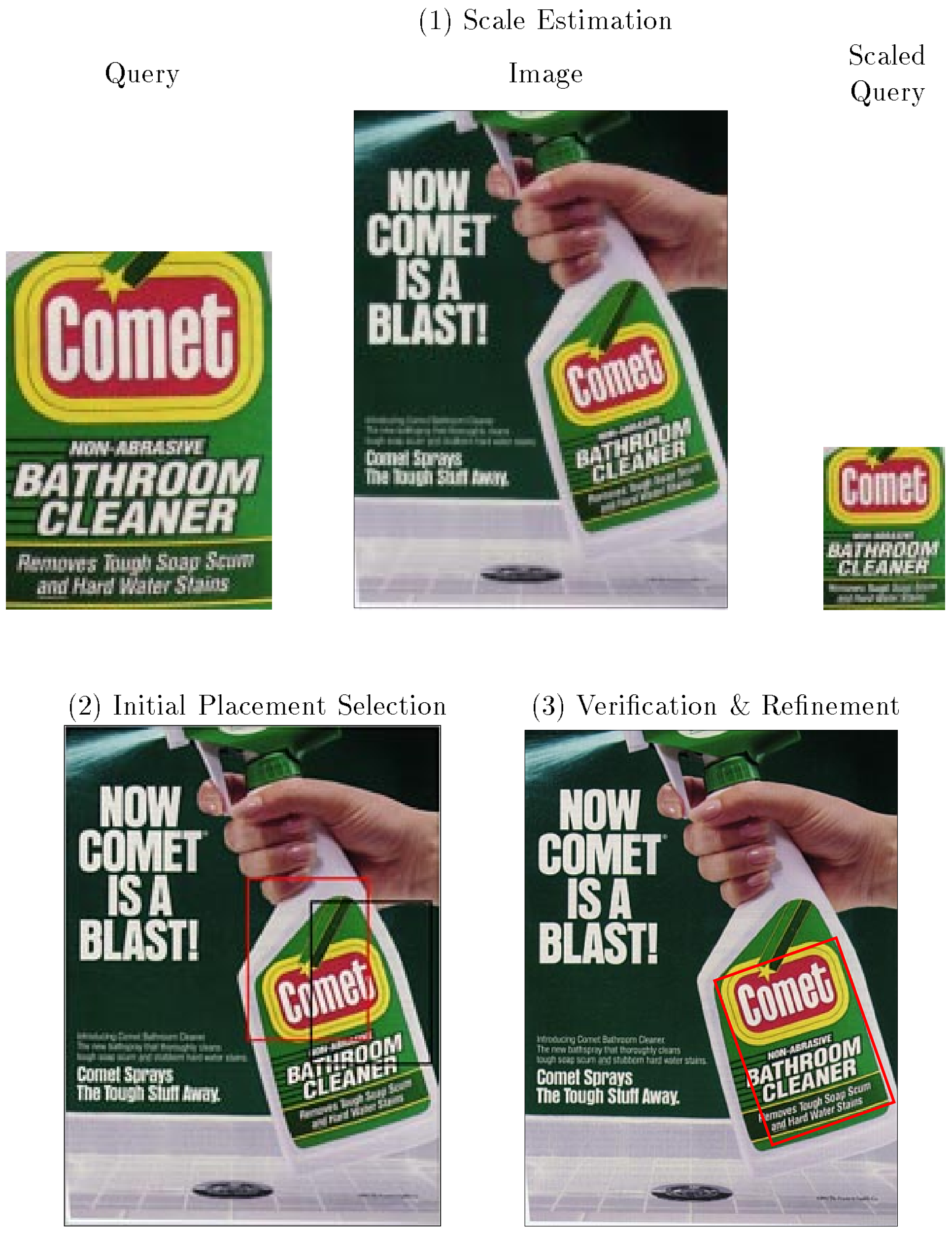

(1) Scale Estimation
Query
Image
Scaled Query
(2) Initial Placement Selection
(3) Verification & Refinement
Comet
NON-ABRASIVE
BATHROOM CLEANER
Removes Tough Soap Scum and Hard Water Stains
NOW COMET IS A BLAST!
Comet Sprays The Tough Stuff Away.

Figure 7.2: The Three Phases in SEDL.

7.1 SEDL's Image Signatures and Distance Function

The image signature $\mathbf{X}$ is a discrete distribution of mass in a combined attribute-position space:

$$\begin{aligned}\mathbf{X} &= \{\ (X_1, W_1), \ldots, (X_M, W_M)\ \} \\ &= \{\ ((A_1, P_1), W_1), \ldots, ((A_M, P_M), W_M)\ \},\end{aligned}$$

where distribution point $X_I = (A_I, P_I)$ consists of a point A_I in attribute space and a point P_I in image position space. The distribution pair $(X_I, W_I) = ((A_I, P_I), W_I)$ indicates that there is a region of size W_I located at P_I with attribute A_I. For the advertisement database, P_I is the centroid of a region of roughly constant color, its attribute A_I is the average color in the region, and its weight W_I is the area of the region. This idea is conveyed in the example in Figure 7.3. For the Chinese character database, P_I is the centroid of a small curve or segment, its attribute A_I is the average orientation along the curve, and its weight W_I is the length of the curve over which the average is taken. In the shape case, a *region* refers to a segment of a curve.

Throughout this chapter, we denote the signature of a database image by $\mathbf{X}$ as given above, and the signature of a query pattern by $\mathbf{Y}$, where

$$\begin{aligned}\mathbf{Y} &= \{\ (Y_1, U_1), \ldots, (Y_N, U_N)\ \} \\ &= \{\ ((B_1, Q_1), U_1), \ldots, ((B_N, Q_N), U_N)\ \}.\end{aligned}$$

Once again, we assume the normalization $W_\Sigma = U_\Sigma = 1$. A query pattern is similar to part of a database image to the extent that there is a transformation of the pattern region positions $\{\ Q_J\ \}$ that aligns regions in $\mathbf{X}$ and $\mathbf{Y}$ that have similar attributes.

In the color case, a small set of dominant image colors $\{a_i\}_{i=1}^m$ is computed by clustering in color space; in the shape case, a small set of dominant orientations $\{a_i\}_{i=1}^m$ is computed by clustering image curve orientations. In both cases, all A_I are members of the set $\{a_i\}_{i=1}^m$. A similar statement holds for the attributes B_J in the pattern attribute $\times$ position signature. The signature creation processes for the color and shape pattern problems are discussed in sections 7.5.1.1 and 7.5.2.1, respectively.

In order to define a signature distance function, we first define a distance function d_{ap} in the combined attribute-position space. Given a distance function d_{attr} between attributes, we use a linear combination distance

$$d_{\mathrm{ap}}((A, P), (B, Q)) = d_{\mathrm{attr}}^{\tau}(A, B) + \lambda d_{\mathrm{pos}}(P, Q), \tag{7.1}$$

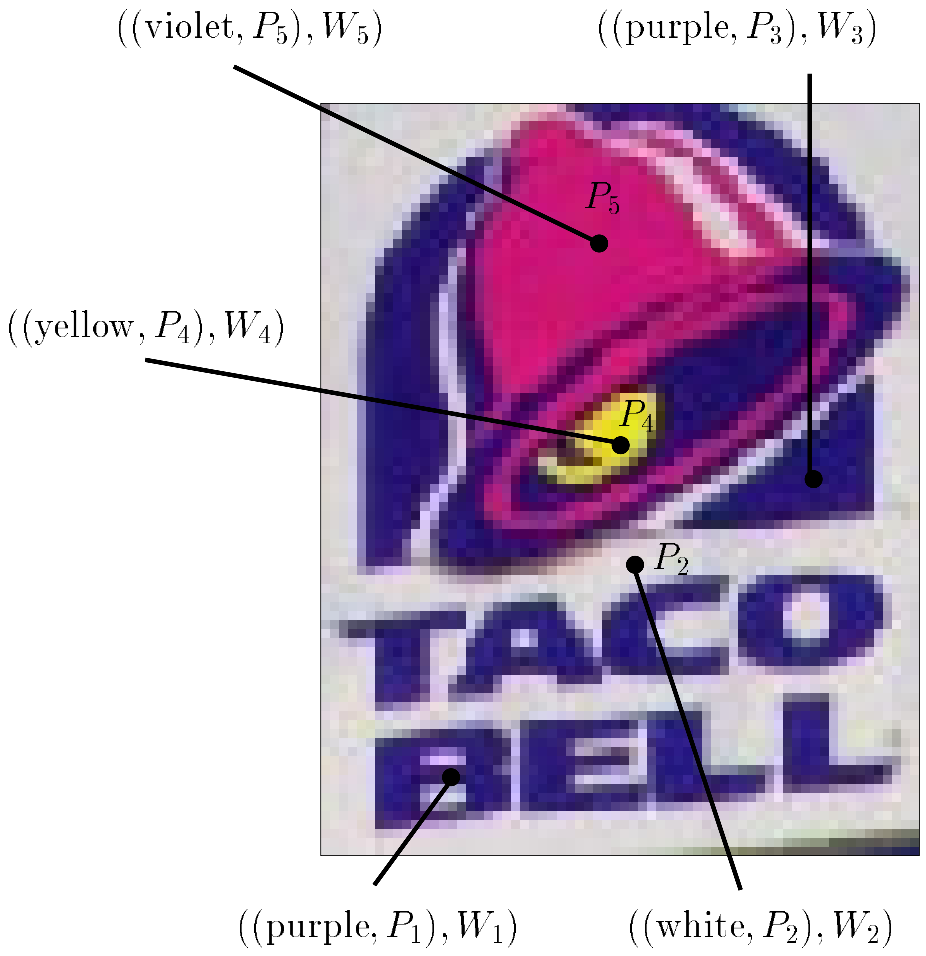


Figure 7.3: Signature in Color × Position Space. Not all signature elements are labelled.

where

$$d^{\tau}_{\text{attr}}(A,B) = \begin{cases} d_{\text{attr}}(A,B) & \text{if } d_{\text{attr}}(A,B) \leq \tau \\ \infty & \text{otherwise} \end{cases} \quad \text{and} \quad d_{\text{pos}}(P,Q) = ||P-Q||_2^2.$$

The attribute distance is set to infinity if it is above some user defined threshold τ. This is to prevent matching of attributes which are very different but happen to be at the same position in the image plane. For colors in CIE-Lab space, we use $d_{\text{attr}}(A,B) = ||A-B||_2$, the Euclidean distance. For orientations in radians, we use $d_{\text{attr}}(A,B) = \min_{k \in \mathbf{Z}} |(A+k\pi) - B|$, the cyclic L_1 distance with period $T = \pi$ radians described in section 6.4.1.4. The pixel locations of region centroids are normalized by dividing by the minimum image dimension. If, for example, a database image has width W and height H pixels with $W \leq H$, then the column coordinate $P_x \in [0,1]$ and the row coordinate $P_y \in [0, H/W]$, where $P = (P_x, P_y)$.

The factor λ in front of the position distance d_{pos} is chosen as follows. Given two "equal" distances D_{attr} and D_{pos} in attribute and position space, and the relative importance $\alpha \in (0,1]$ of attribute differences versus position differences, we choose λ so that $(1-\alpha)/\alpha = (\lambda D_{\text{pos}})/D_{\text{attr}}$. We use $\alpha = 25\%$ attribute and $(1-\alpha) = 75\%$ position. We emphasize the position information because it was not used in the scale estimation or initial placement phases, and, if these phases worked correctly, the verification and refinement phase starts in image areas with similar attributes to those in the pattern. On the other hand, we want the attribute values to guide our matcher, so we cannot make α too small. We set $D_{\text{pos}} = (0.10)^2$, which corresponds before normalization to the square of 10% of the minimum image dimension. In the color case, we set $D_{\text{attr}} = 5$ CIE-Lab units; in the shape case, we set $D_{\text{attr}} = 10^\circ \doteq 0.175$ radians.

Armed with the ground distance d_{ap} in the combined attribute-position (what-where) space, we define the signature distance function SD as

$$\text{SD}(\mathbf{X},\mathbf{Y}) = \min_{\psi \in \Psi} D(\psi, \mathbf{X}, \mathbf{Y}),$$

where

$$D(\psi, \mathbf{X}, \mathbf{Y}) = \sum_{J=1}^{N} U_J d_{\text{ap}}((A_{\psi(J)}, P_{\psi(J)}), (B_J, Q_J)),$$

and

$$\Psi = \{\ \psi : [1..N] \rightarrow [1..M]\ \}$$

is the set of functions from $[1..N]$ to $[1..M]$. If the image attribute $A_{\psi(J)}$ at position $P_{\psi(J)}$ is matched to the query attribute B_J at position Q_J, then we pay a cost of the distance in the combined attribute-position space times the weight of the query attribute. The allowable matchings Ψ are unconstrained; each query region ((attribute,position) pair) can match any image region. Note that this is very much a one way distance – it is small when every query region is near some image region with a similar attribute value, but not necessarily vice-versa. Also note that the image weights do not appear in our distance function. These weights, however, are used explicitly in the scale estimation and initial placement phases.

As mentioned previously, similarity of a query pattern to a database image is judged modulo a transformation of region locations. Thus we want to compute the distance

$$\begin{aligned} \mathrm{SD}_{\mathcal{G}}(\mathbf{X},\mathbf{Y}) &= \min_{g\in\mathcal{G}} \quad \mathrm{SD}(\mathbf{X}, g(\mathbf{Y})) \\ &= \min_{g\in\mathcal{G},\psi\in\Psi} \quad D(\psi,\mathbf{X},g(\mathbf{Y})), \end{aligned}$$

where

$$g(\mathbf{Y}) = \{\ ((B_1, g(Q_1)), U_1), \ldots, ((B_N, g(Q_N)), U_N)\ \}$$

only modifies the region locations, not the region attributes. In full, we have

$$\begin{aligned} \mathrm{SD}_{\mathcal{G}}(\mathbf{X},\mathbf{Y}) &= \min_{g\in\mathcal{G},\psi\in\Psi} \sum_{J=1}^{N} U_J d_{\mathrm{ap}}((A_{\psi(J)}, P_{\psi(J)}), (B_J, g(Q_J))) \\ &= \min_{g\in\mathcal{G},\psi\in\Psi} \sum_{J=1}^{N} U_J (d_{\mathrm{attr}}^{\tau}(A_{\psi(J)}, B_J) + \lambda d_{\mathrm{pos}}(P_{\psi(J)}, g(Q_J))). \end{aligned}$$

In section 7.4, we show how to find a local minimum of $\mathrm{SD}_{\mathcal{G}}(\mathbf{X},\mathbf{Y})$ using the same alternation idea behind our EMD iteration. We also discuss how the distance function $\mathrm{SD}_{\mathcal{G}}$ compares with the distance function $D_{\mathcal{G}}^{\mathrm{ICP}}$ used in the ICP registration algorithm (see section 2.5.2) and with the EMD, including our reasons for choosing SD over these other distance measures. As with any iterative optimization scheme in a space with local minima, the key to finding the global minimum is to have a good starting point for the iteration. For us, this means selecting a good initial transformation $g^{(0)} \in \mathcal{G}$. In turn, this means selecting a good initial scale and placement of the pattern within the image. The scale estimation and initial placement steps are the subjects of the next two sections.

7.2 The Scale Estimation Phase

Our scale estimation algorithm considers only the attributes in an image, not their location. It operates on the previously described signature distributions once position has been marginalized out. If we marginalize the attribute-position distributions $\mathbf{X}$ and $\mathbf{Y}$ over position, we get the attribute distributions

$$\begin{aligned}\mathbf{a} &= \{\, (a_1, w_1), \ldots, (a_m, w_m) \,\} \quad \text{and} \\ \mathbf{b} &= \{\, (b_1, u_1), \ldots, (b_n, u_n) \,\}\,,\end{aligned}$$

respectively, where

$$w_i = \sum_{I:A_I=a_i} W_I \qquad \text{and} \qquad u_j = \sum_{J:B_J=b_j} U_J.$$

This marginalization throws away the positions of attributes and combines the weights for the same attribute at different image locations into a single weight for that attribute. If, for example, the attribute-position image signature $\mathbf{X}$ notes 10% red at the top of an image and 20% red at the bottom, then the attribute signature $\mathbf{a}$ simply notes 30% red in the image. Note that the attribute signatures also have total weight one ($w_\Sigma = u_\Sigma = 1$) since the attribute-position signatures were normalized to have weight one ($W_\Sigma = U_\Sigma = 1$).

Our initial scale estimate is obtained by running the algorithm in section 4.5 on the attribute signatures $\mathbf{a}$ and $\mathbf{b}$. Recall that this estimation algorithm scales down the weights of the query signature $\mathbf{b}$ until there is no further improvement in the EMD between the image signature $\mathbf{a}$ and the modified query signature. The scale c^0 at which this occurs is taken as the scale estimate if $\text{EMD}(\mathbf{a}, c^0\mathbf{b}) \leq \tau$, where τ is a threshold on how well the attributes must match to obtain a visually acceptable match. If not, SEDL assumes that the pattern does not occur within the image, and the initial placement and refinement phases are not performed.

The main property of the scale estimation algorithm is that the error that it makes is roughly equal to the *minimum* amount of background clutter over all query attributes, where the amount of background clutter for an attribute is the amount of that attribute present in the image but not part of the query occurrence. If there is a query attribute that occurs in the image only within the query occurrence, then the scale estimate will be very accurate, regardless of the amount of background clutter for other query attributes. Please refer back to section 4.5 for details and examples from the advertisement database.

7.3 The Initial Placement Phase

All the initial transformations of the query for our iteration will have scale c^0 as returned by the scale estimation step. The selection of initial placements of the query at scale c^0 uses the optimal flow $F^0 = (f^0_{ij})$ returned by $\text{EMD}(\mathbf{a}, c^0\mathbf{b})$. This flow provides important information about what attributes in the image match what attributes in the query. In the color case, for example, the optimal flow tells us which image colors match which query colors. Since the EMD achieved by this flow is small ($\leq \tau$, otherwise this phase would not be considered), the attributes matched by F^0 are similar.

The total weight of query attributes that match attribute i in the image is

$$v_i = \sum_{j=1}^{n} f_{ij}^0,$$

where the total weight of the scaled query signature is c^0. Basically, v defines the query signature in terms of the image attributes. In the color advertisement database, for example, we can think of v_i as the amount of image color i that occurs within the query (assuming that the scale estimate is fairly accurate). The first step in our initial placement selection is to compute the probability or confidence that an image region with attribute i is part of the query pattern (if it occurs within the image). The probability is given by the quotient

$$0 \leq q_i = \frac{v_i}{w_i} \leq 1.$$

The total amount of attribute i in the image is w_i, while the amount of attribute i that occurs within query regions is estimated to be v_i. Thus, if we randomly choose an image pixel from all the pixels with attribute i, the probability that this pixel belongs to a query region is $q_i = v_i/w_i$. The flow feasibility conditions imply $v_i \leq w_i$.

In Figure 7.4, we show an example that illustrates confidences in attributes for the color pattern problem. Yellow is a 98% confidence color since virtually all the yellow in the image is needed to match yellow in the scaled query. The confidence for yellow is not 100% because the pattern is slightly underestimated in scale. About 89% of the purple in the image is matched to purple in the scaled query. Purple is also a high confidence color. The confidence for purple is not 100% because the pattern does not include all the purple on the Ziploc box, and the pattern is slightly underestimated in scale.

For the example in Figure 7.4, white, black, and brown are low confidence colors. Although the white on the Ziploc box within the image is needed to match white in the scaled query, there is a lot of other white in the image (the writings "I will keep my sandwich fresh" and "Ziploc has the lock on freshness", the chalk next to the Ziploc box, and the white on the Ziploc box that is not part of the pattern). The black and brown in the image are not needed to match any color in the pattern. The main idea in the initial placement phase is only to examine image regions that contain (relatively) high confidence colors. In this example, SEDL never needs to examine the upper half of the Ziploc advertisement at

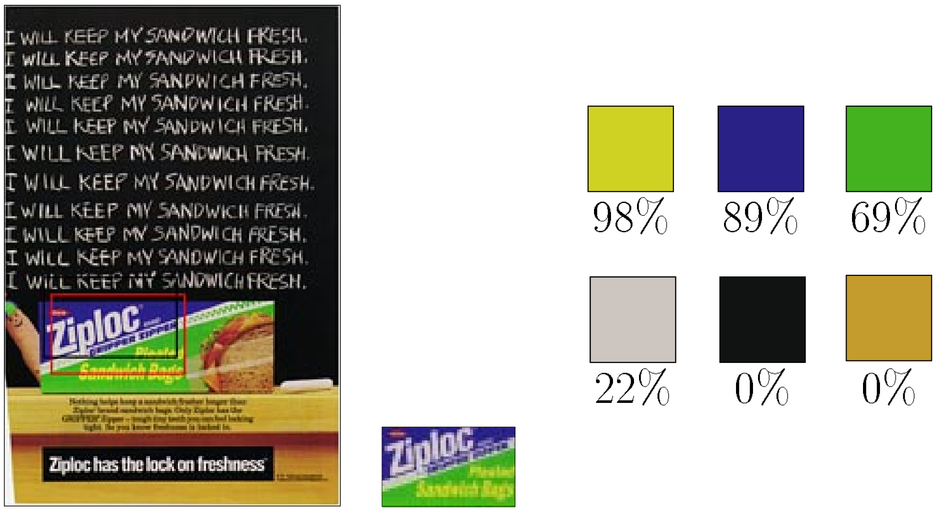

I WILL KEEP MY SANDWICH FRESH.
Ziploc
Sandwich Bags
Ziploc has the lock on freshness
98%
89%
69%
22%
0%
0%

Figure 7.4: An Example of Color Confidences. The left half of the figure shows a Ziploc advertisement, along with the Ziploc query pattern scaled according to SEDL's scale estimate. The right half of the figure shows the confidences computed for various query colors. See the text for further explanation.

query time since it only contains low confidence colors (black and white).

The result of the first step in the initial placement phase is a set I^* of image attributes that have significantly high confidence ($q_i \geq \beta$, with, for example, $\beta = 0.50 = 50\%$). The regions in the image plane with high confidence attributes are good places to check further for initial placements. Pseudocode for this step is shown below.

```
function $G$ = InitialTransformations($\mathbf{X}$,$\mathbf{a}$,$c^0$,$F^0$,$l_{\max}$,$\beta$)
    $v_i = \sum_{j=1}^{n} f_{ij}^0, \quad i = 1, \ldots, m$
    $q = v/w$
    $I^* = \{ i : q_i \geq \beta \} \cup \{ \arg\max_i q_i \}$
    ⋮
end function
```

Note that we always include the image attribute of highest confidence in the set I^*, even when this maximum confidence is less than the threshold β.

The second step in the initial placement phase is to compare the image signature in rectangular regions of the image plane to the query signature expressed in image attributes (v). The aspect ratio of the rectangles is equal to the aspect ratio of the bounding box around the query pattern, while the area of the rectangles is c^0 (the scale estimate computed in the previous phase). Let R_0 denote the origin-centered rectangle with these dimensions. This canonical rectangle will be centered at various image attribute positions for signature comparison. If we denote the image signature over window R by w^R, then the similarity measure used is the Histogram Intersection $H(w^R, v)$ between w^R and v:

$$H(w^R, v) = \sum_{i=1}^{m} \min(w_k^R, v_k).$$

The higher this value, the more similar the signature in local area R is to the query signature. Please refer back to section 2.2 for more details concerning Histogram Intersection.

The locations at which to center R_0 are exactly the locations of high confidence image attributes in I^*; we do *not* try every image location. Thus, the search for promising image regions is directed (the D in SEDL) by the relatively high confidence (i.e. low background clutter) colors. We keep track of the best placements in a variable called *optPlacements*. Associated with each placement is a Histogram Intersection value. The user supplies the maximum number $l_{\max}$ of initial similarity transformations to compute. Pseudocode for the algorithm described thus far is shown below.

```
function $G$ = InitialTransformations($\mathbf{X}$,$\mathbf{a}$,$c^0$,$F^0$,$l_{\max}$,$\beta$)
    $v_i = \sum_{j=1}^{n} f_{ij}^0, \quad i = 1, \ldots, m$
    $q = v/w$
    $I^* = \{ i : q_i \geq \beta \} \cup \{ \arg\max_i q_i \}$
    optPlacements = {}
    foreach $i^* \in I^*$
        foreach $((A_I, P_I), W_I) \in \mathbf{X}$ such that $A_I = a_{i^*}$
            if (dist($P_I$,optPlacements) too small) continue
            $R = R_0 \oplus P_I$ /* center $R_0$ at $P_I$ */
            $T = \{ ((A_K, P_K), W_K) : P_K \in R \}$ /* rectangular range search */
            /* compute image attribute histogram over $R$ */
            $w^R = 0 \in \mathbf{R}^m$
            foreach $((A_K, P_K), W_K) \in T$
                $\hat{\imath}$ = attribute#($A_K$)
                $w_{\hat{\imath}}^R$ += $W_K$
            end foreach
            /* compute Histogram Intersection */
            HI = $H(w^R, v)$
            ⋮
end function
```

Computing the local histogram w^R requires a 2D rectangular range search for image regions inside R. This can be done in output sensitive time $O(\log M + k)$ using $O(M \log M)$ space or in time $O(\sqrt{M} + k)$ using only $O(M)$ space ([15]), where M is the number of regions in the image attribute-position distribution, and k is the number of such regions with location inside R. Both range searching results referred to above require $O(M \log M)$ preprocessing time. Note that we are really computing an approximation to the local image signature inside R since we count an entire region as inside R if its centroid is in R. Also note that if a placement is too close to a previously chosen good placement, then we do not even try the new placement. This is an efficiency consideration that help keeps the running time low.

It remains to explain exactly why we choose to keep a placement and which previously chosen placement is discarded. Here are the rules. (1) If optPlacements is full, we do not include a placement with Histogram Intersection (HI) value less than the minimum HI value currently in optPlacements. (2) If a placement P is close to a placement $\widehat{P}$ already in optPlacements, and P has a higher HI value than $\widehat{P}$, then we replace $\widehat{P}$ by P. (3) If the HI value of P does not exceed the HI value of all similar placements currently in optPlacements, then we do *not* include P. (4) If we get this far without deciding the fate of P, then its HI value is higher than some currently selected placement and there are no nearby placements in optPlacements. If optPlacements is not full, then we simply add P to optPlacements. Otherwise, we replace the placement with the lowest HI value with P.

For the refinement stage that comes next, we actually need the similarity transformation $g = (s, \theta, t_x, t_y)$ that transforms the query bounding box into the R_{P_*} for each P_* in optPlacements. Just before computing this transformation for a particular P_*, we do a small local search around P_* to see if the placement P_* can be improved. Pseudocode for the whole initial placement phase is shown below.

```
**function** $G$ = InitialTransformations($\mathbf{X}$,$\mathbf{a}$,$c^0$,$F^0$,$l_{\max}$,$\beta$)
    $v_i = \sum_{j=1}^{n} f_{ij}^0, \quad i = 1, \ldots, m$
    $q = v/w$
    $I^* = \{ \ i \ : \ q_i \geq \beta \ \} \cup \{ \ \arg\max_i q_i \ \}$
    optPlacements = {}
    **foreach** $i^* \in I^*$
        **foreach** $((A_I, P_I), W_I) \in \mathbf{X}$ such that $A_I = a_{i^*}$
            **if** (dist($P_I$,optPlacements) too small) **continue**
            $R = R_0 \oplus P_I$ /* *center* $R_0$ *at* $P_I$ */
            $T = \{ \ ((A_K, P_K), W_K) \ : \ P_K \in R \ \}$ /* *rectangular range search* */
            /* *compute image attribute histogram over* $R$ */
            $w^R = 0 \in \mathbf{R}^m$
            **foreach** $((A_K, P_K), W_K) \in T$
                $\hat{\imath}$ = attribute#($A_K$)
            $w^R_{\hat{i}}$ += $W_K$
        end foreach
        /* compute Histogram Intersection */
        HI = $H(w^R, v)$
        /* accept placement $P_K$? */
        if ((size(optPlacements) == $l_{\max}$) and
            (HI $\leq$ min HI in optPlacements)) continue
        if (HI > HI of similar placement $\widehat{P}$ in optPlacements)
            replace $\widehat{P}$ with $P_K$ /* replace */
            continue
        elseif (HI $\leq$ HI of all similar placements in optPlacements)
            continue /* do not add */
        end if
        if (size(optPlacements) < $l_{\max}$)
            add ($P_K$,HI) to optPlacements /* add */
        else
            remove min HI placement in optPlacements /* replace */
            add ($P_K$,HI) to optPlacements
        end if
      end foreach
    end foreach
    /* compute similarity transformations to return */
    $G = \{\}$
    foreach $(P, HI)$ in optPlacements
        /* check small perturbations of placement P */
        $P_{ij} = P + i \times (\text{height}(R_0)/4) + j \times (\text{width}(R_0)/4)$     $i = -1, 0, 1, j = -1, 0, 1$
        $P_{\max} = \arg\max_{i=-1,0,1,j=-1,0,1}$ HI(placement $P_{ij}$)
        add ($g$ : query bounding box $\mapsto R_0 \oplus P_{\max}$) to $G$
    end foreach
    return($G$)
end function
```

The only expensive step here is the rectangular range search. The number of outer loop iterations and range searches is kept small by only considering locations of high confidence attributes which are significantly different from placements already chosen.

7.3.1 Experiments with the Color Pattern Problem

In this section, we illustrate the performance of the initial placement algorithm for the color pattern problem. In all of the examples in this section, we use $l_{\max} = 2$ placements and $\beta = 0.50 = 50\%$. The placement with the higher histogram intersection score is shown in red, while the other placement is shown in black. For each (image,pattern) pair, we give the attribute-position distribution sizes (M for the image and N for the pattern), the marginal attribute distribution sizes (m colors for the image and n colors for the query), and the running time T_2 for the second phase. The time T_2 does not include the running time of the first phase to produce the scale estimate. The parameters given to the scale estimation algorithm are $c_{\min} = 0.001 = 0.1\%$, $\varepsilon_c = 0.001$, and $\varepsilon_d = 0.0001$ CIE-Lab space units.

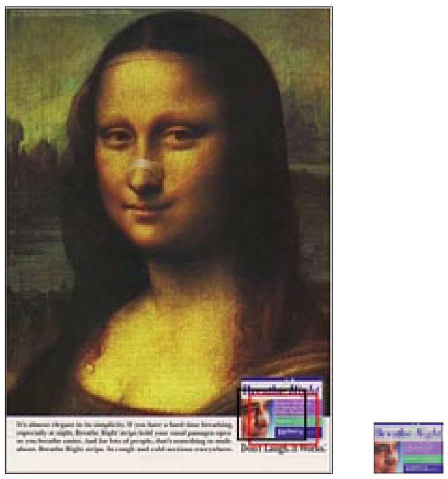
The results in Figures 7.5–7.9 are excellent, despite sometimes imperfect scale estimates. Recall that the scale estimation algorithm uses only the amounts of colors present in the image and the pattern, and not their positions. When the pattern appears in more than place in the image, the scale estimate is therefore likely to be greater than the scale of any single pattern occurrence. This is the case for examples shown in Figure 7.7(c) and Figures 7.8(a),(d). In each of these overestimated scale, multiple pattern occurrence examples, the initial placements cover all occurrences of the pattern.

The scale estimate would be guaranteed to be at least the sum of the pattern occurrence sizes if the occurrences in the image had exactly the same colors as the pattern itself and if we did not perform the color clustering efficiency step that represents the image and query as faithfully as possible with as few colors as possible. Figure 7.7(d) shows an example in which the scale is underestimated. In this example, both initial placements are contained within the pattern occurrence in the image.

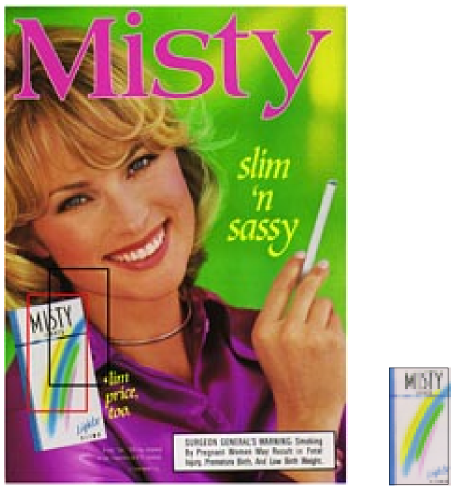
At least one initial placement in all the examples in Figures 7.5 through 7.8 is very accurate, and in some cases near perfect as in Figure 7.5(c) and Figures 7.6(a),(d). Sometimes the heuristics used for choosing the initial placements without considerable overlap perform poorly, as in Figure 7.7(a). In Figure 7.9, we show some examples in which we search for a logo in advertisements for products with a similar logo to that of the query. In all these examples, the initial placements occur over image regions with color content which is similar to the color content of the query logo.

7.4 The Verification and Refinement Phase

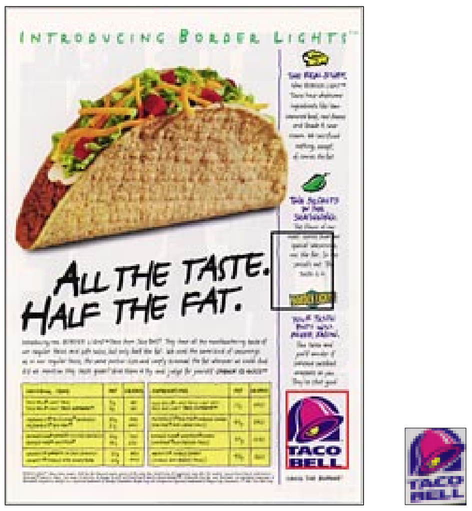
Recall SEDL's notion of visual similarity that each pattern region is close to an image region of similar color or curve orientation (in the shape case). In the verification and refinement phase, SEDL iteratively improves its estimate for the pattern scale, orientation, and location, starting from the scale and locations determined by the scale estimation and initial placement phases. See Figure 7.10 for this main idea illustrated in the color case. The leftmost column contains a pattern that occurs in the image in the rightmost column. The pattern and image signatures in color $\times$ position space are shown in the middle columns. There is a similarity transformation g^* of the pattern regions that aligns pattern regions with image regions of similar colors. Our goal is to find g^* starting from $g^{(0)}$ given by the



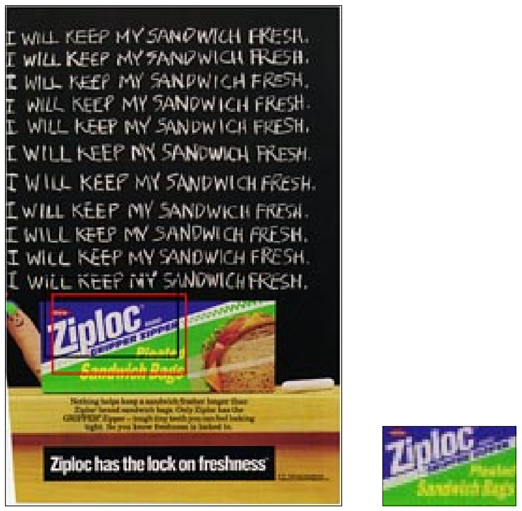
(a)


Misty
slim 'n sassy
MISTY
slim price, too.
SURGEON GENERAL'S WARNING: Smoking By Pregnant Women May Result in Fetal Injury, Premature Birth, And Low Birth Weight.

(b)

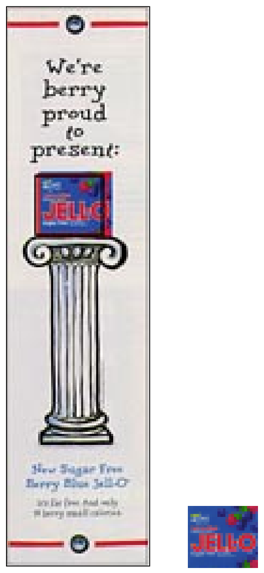


(c)

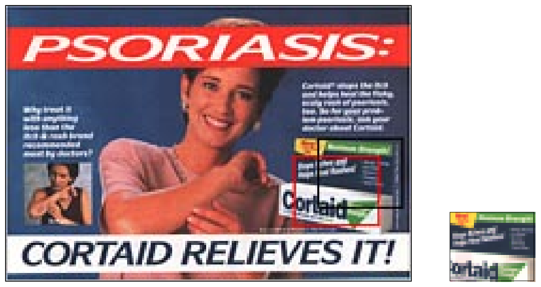

I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
I WILL KEEP MY SANDWICH FRESH.
Ziploc
Sandwich Bags
Ziploc has the lock on freshness

(d)

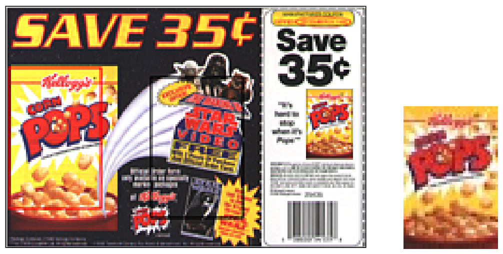
Figure 7.5: Initial Placement Results – Example Set 1. (a) $M = 491$, $m = 20$, $N = 203$, $n = 14$, $T_2 = 0.012$s. (b) $M = 1002$, $m = 41$, $N = 180$, $n = 18$, $T_2 = 0.010$s. (c) $M = 763$, $m = 35$, $N = 265$, $n = 25$, $T_2 = 0.007$s. (d) $M = 871$, $m = 31$, $N = 84$, $n = 8$, $T_2 = 0.008$s.


We're berry proud to present:
JELL-O
New Sugar Free Berry Blue Jell-O

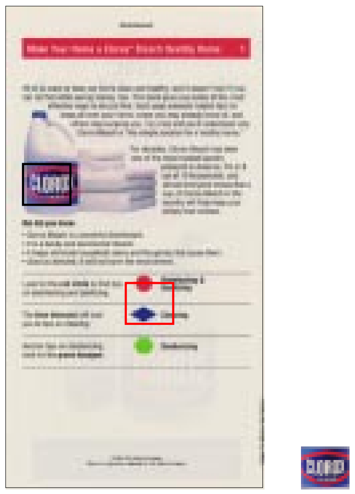
(a)



(b)

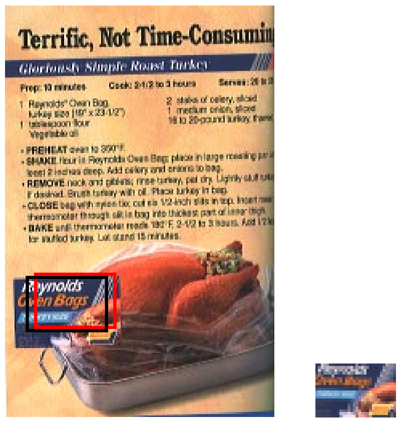


(c)

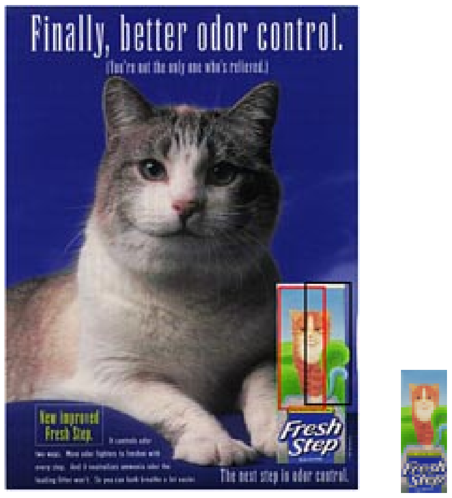


(d)

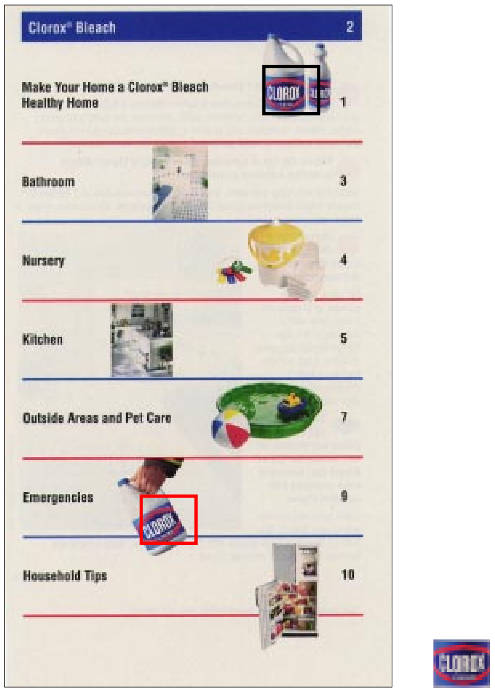
Figure 7.6: Initial Placement Results – Example Set 2. (a) $M = 894$, $m = 29$, $N = 160$, $n = 14$, $T_2 = 0.021$s. The scale estimation and initial placement are near perfect, although the heuristic for selecting initial placements without sizeable overlap performed poorly. (b) $M = 596$, $m = 16$, $N = 127$, $n = 9$, $T_2 = 0.004$s. (c) $M = 1348$, $m = 35$, $N = 325$, $n = 20$, $T_2 = 0.100$s. (d) $M = 382$, $m = 13$, $N = 228$, $n = 19$, $T_2 = 0.007$s. The lower scoring initial placement is only slightly offset from the correct placement. The higher scoring placement is also a good guess at the pattern occurrence given that the exact locations of the colors inside the rectangle are not used.



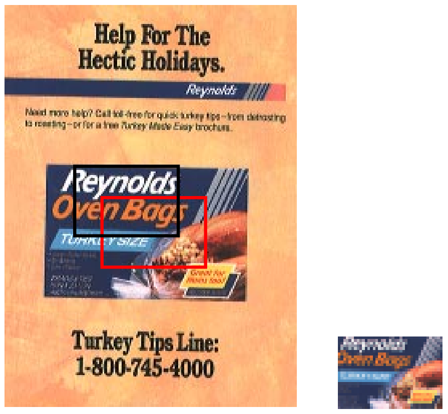
(a)



(b)

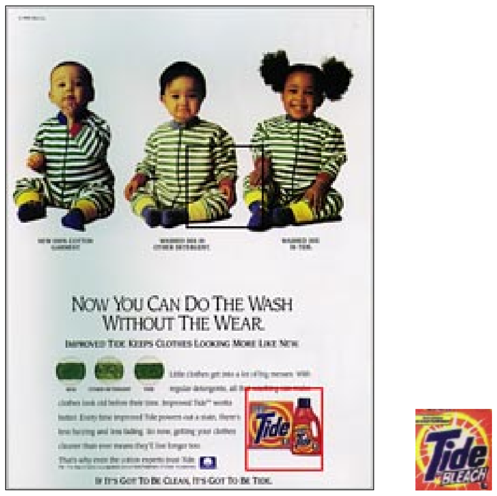

Clorox® Bleach
2
Make Your Home a Clorox® Bleach Healthy Home
1
Bathroom
3
Nursery
4
Kitchen
5
Outside Areas and Pet Care
7
Emergencies
9
Household Tips
10

(c)

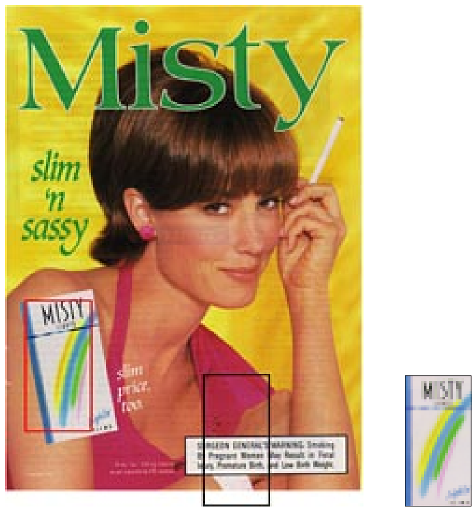

Help For The Hectic Holidays.
Reynolds
Turkey Tips Line: 1-800-745-4000
Reynolds Oven Bags
Turkey Size

(d)

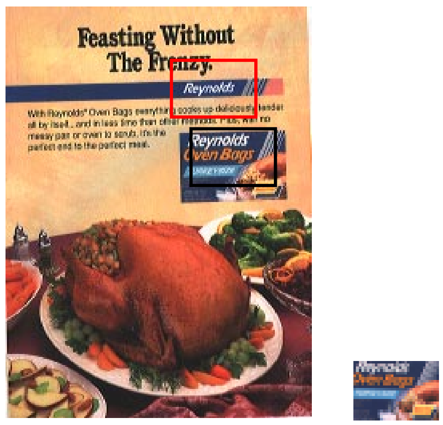
Figure 7.7: Initial Placement Results – Example Set 3. (a) $M = 648$, $m = 20$, $N = 436$, $n = 37$, $T_2 = 0.005$s. (b) $M = 533$, $m = 22$, $N = 191$, $n = 12$, $T_2 = 0.005$s. Although the scale is underestimated, the two initial placements are both contained within the pattern occurrence. (c) $M = 545$, $m = 27$, $N = 228$, $n = 19$, $T_2 = 0.004$s. The pattern scale is overestimated due to the occurrence of the pattern at three different image locations. The two initial placements contain all three pattern occurrences. (d) $M = 703$, $m = 21$, $N = 436$, $n = 37$, $T_2 = 0.008$s. As in (b), the scale is underestimated but the two initial placements are both contained within the pattern occurrence.


Now You Can Do The Wash Without The Wear.
Improved Tide Keeps Clothes Looking More Like New
If It's Got To Be Clean, It's Got To Be Tide.
Tide
BLEACH

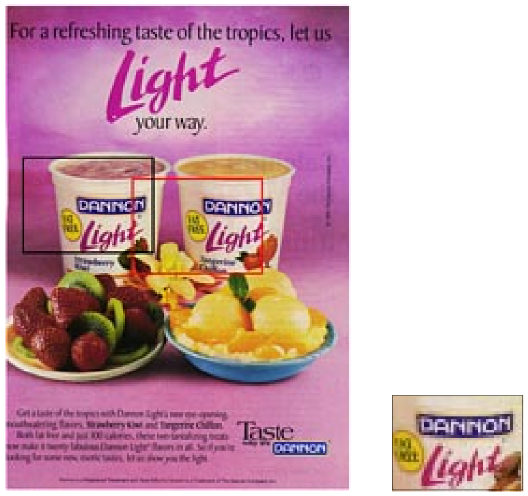
(a)



(b)



(c)



(d)

Figure 7.8: Initial Placement Results – Example Set 4. (a) $M = 898$, $m = 30$, $N = 354$, $n = 36$, $T_2 = 0.023$s. The pattern occurs twice and the scale is overestimated. The higher scoring placement contains both instances of the pattern. (b) $M = 669$, $m = 32$, $N = 180$, $n = 18$, $T_2 = 0.003$s. (c) $M = 873$, $m = 23$, $N = 436$, $n = 37$, $T_2 = 0.003$s. (d) $M = 888$, $m = 43$, $N = 444$, $n = 43$, $T_2 = 0.009$s. Both initial placements overlap one pattern occurrence.

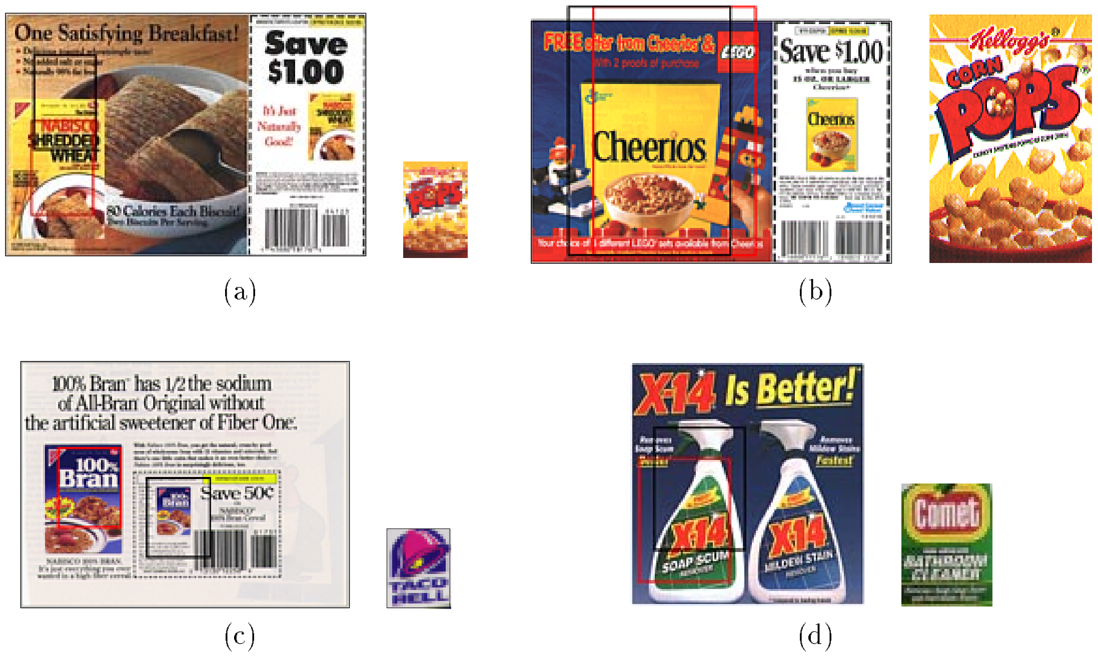


Figure 7.9: Initial Placement Results – Example Set 5. In each of these examples, we search for a pattern that does not occur within an image. (a) $M = 410$, $m = 9$, $N = 325$, $n = 20$, $T_2 = 0.008$s. The yellow and red of the Cornpops logo matches the yellow and red of the shredded wheat box. (b) $M = 381$, $m = 9$, $N = 325$, $n = 20$, $T_2 = 0.006$s. The yellow and red of the Cornpops logo matches the yellow and red of the cheerios box and leggos beneath the box. (c) $M = 409$, $m = 9$, $N = 265$, $n = 25$, $T_2 = 0.007$s. The 100% Bran boxes contain white, purple, and a little bit of yellow as in the Taco Bell logo. (d) $M = 420$, $m = 9$, $N = 175$, $n = 10$, $T_2 = 0.009$s. The X14 label is similar to the Comet logo in its color composition.

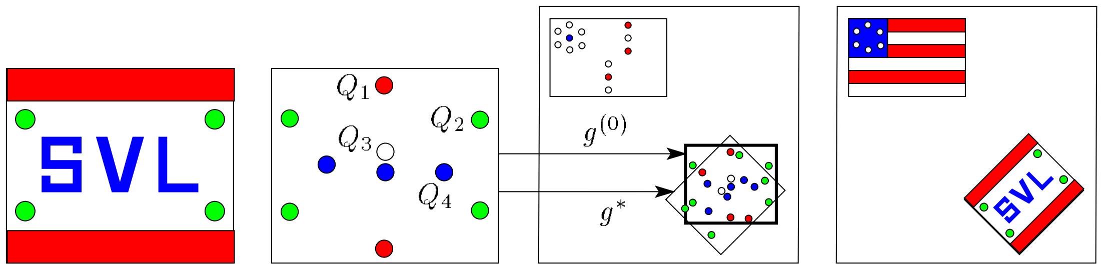


Figure 7.10: The Verification and Refinement Phase. See the text for discussion.

scale estimation and initial placement phases. Notice the importance of starting close to an optimal transformation. The pseudo American flag in the upper left-hand corner of the image contains red, white, and blue just like the query pattern. The search rectangle might be pulled toward this incorrect pattern if the iteration starts too close to the American flag.

We can use the same alternation idea behind the FT iteration to find at least a local minimum of our signature distance function under a transformation set:

$$\mathrm{SD}_{\mathcal{G}}(\mathbf{X},\mathbf{Y}) = \min_{g\in\mathcal{G},\psi\in\Psi} \sum_{J=1}^{N} U_J(d^{\tau}_{\mathrm{attr}}(A_{\psi(J)}, B_J) + \lambda d_{\mathrm{pos}}(P_{\psi(J)}, g(Q_J))).$$

For a fixed transformation $g^{(k)}$, we solve

$$\psi^{(k)} = \min_{\psi\in\Psi} \sum_{J=1}^{N} U_J(d^{\tau}_{\mathrm{attr}}(A_{\psi(J)}, B_J) + \lambda d_{\mathrm{pos}}(P_{\psi(J)}, g^{(k)}(Q_J))). \tag{7.2}$$

This is trivial since Ψ is the set of unconstrained matchings from $[1..N]$ to $[1..M]$. The solution is

$$\psi^{(k)}(J) = \arg\min_{I\in[1..M]} (d^{\tau}_{\mathrm{attr}}(A_I, B_J) + \lambda d_{\mathrm{pos}}(P_I, g^{(k)}(Q_J))) \qquad J = 1, \ldots, N. \tag{7.3}$$

That is, the correspondence step (7.2) involves N nearest neighbor computations over a set of size M. We shall come back to this computation shortly. The transformation step for a fixed matching $\psi^{(k)}$ is to compute the optimal transformation

$$g^{(k+1)} = \arg\min_{g\in\mathcal{G}} \sum_{J=1}^{N} U_J(d^{\tau}_{\mathrm{attr}}(A_{\psi^{(k)}(J)}, B_J) + \lambda d_{\mathrm{pos}}(P_{\psi^{(k)}(J)}, g(Q_J))). \tag{7.4}$$

In performing the minimization (7.4), we need only compute

$$g^{(k+1)} = \arg\min_{g\in\mathcal{G}} \sum_{J=1}^{N} U_J d_{\mathrm{pos}}(P_{\psi^{(k)}(J)}, g(Q_J)) \tag{7.5}$$

since we only allow transformations g of the attribute-position signature that do not modify the attributes. The minimization (7.5) is easy to compute since we use $d_{\mathrm{pos}} = L_2^2$ in SEDL.

If we let $D^{(k)} = D(\psi^{(k)}, \mathbf{X}, g^{(k)}(\mathbf{Y}))$, then it is easy to show that

$$0 \le D^{(k+1)} \le D^{(k)} \ \ \forall k. \tag{7.6}$$

Indeed,

$$\begin{array}{lrcl} (7.2) \;\Rightarrow & D^{(k)} = D(\psi^{(k)}, \mathbf{X}, g^{(k)}(\mathbf{Y})) & \geq & D(\psi^{(k+1)}, \mathbf{X}, g^{(k)}(\mathbf{Y})) \quad \text{and} \\ (7.4) \;\Rightarrow & D(\psi^{(k+1)}, \mathbf{X}, g^{(k)}(\mathbf{Y})) & \geq & D(\psi^{(k+1)}, \mathbf{X}, g^{(k+1)}(\mathbf{Y})) = D^{(k+1)}, \end{array}$$

from which (7.6) follows. Hence, we have a monotically convergent sequence $< D^{(k)} >$. The result of the previous initial placement phase is a handful (typically ≤ 5) of initial transformations $g^{(0)}$ from which to begin the above iteration.

Now that we have presented the iteration and proved its convergence, let us return to the correspondence step computation (7.3). A brute force computation of $\psi^{(k)}$ requires computing $d_{\text{ap}}((A_I, P_I), (B_J, Q_J))$ for all MN pairs $(I, J) \in [1..M] \times [1..N]$. This computation time can be greatly improved by reorganizing the computation (7.3) as

$$\begin{aligned} \psi^{(k)}(J) &= \arg \min_{i \in [1..m]} \left(\min_{I \,:\, A_I = a_i} (d_{\text{attr}}^{\tau}(A_I, B_J) + \lambda d_{\text{pos}}(P_I, g^{(k)}(Q_J))) \right) \qquad J = 1, \ldots, N \\ &= \arg \min_{i \in [1..m]} \left(d_{\text{attr}}^{\tau}(a_i, B_J) + \lambda \min_{I \,:\, A_I = a_i} d_{\text{pos}}(P_I, g^{(k)}(Q_J))) \right) \qquad J = 1, \ldots, N. \end{aligned}$$

The point sets $P^i = \{ \, P_I \; : \; A_I = a_i \, \}$ can be preprocessed to allow Euclidean nearest neighbor searches in time $O(\log |P^i|)$. This allows us to compute $\psi^{(k)}(J)$ in time $O(\sum_{i=1}^m \log |P^i|)$. Since log is a concave function, Jensen's inequality[1] implies

$$\frac{1}{m} \sum_{i=1}^{m} \log |P^i| \leq \log \left(\frac{1}{m} \sum_{i=1}^{m} |P^i| \right) = \log \left(\frac{M}{m} \right).$$

Therefore, $\sum_{i=1}^m \log |P^i| = O(m \log(M/m))$ and we can compute the entire vector $\psi^{(k)}$ in time $O(Nm \log(M/m))$. If there is any doubt that $m \log(M/m)$ is much smaller than M, just note that $M - m \log(M/m) = m((M/m) - \log(M/m))$.

Combining attribute and position distance in a measure of visual similarity is a very difficult task. The use of a linear combination of the two distances with a constant weighting factor λ is a computationally efficient, reasonably effective solution. We shall discuss this solution and its problems in terms of the color case.

The use of the linear combination d_{ap} (see (7.1)) in the signature distance function $D_{\mathcal{G}}$ captures the fact that two patterns with very similar color regions in very similar locations will appear visually similar. Setting color distances above some threshold to ∞ prevents a small matching cost between nearby regions of very different colors, and heavily penalizes such obvious visual dissimilarities. The threshold τ cannot be too small, however, since we

[1]For a concave function f, Jensen's inequality states that $\sum_{i=1}^m \alpha_i f(x_i) \leq f(\sum_{i=1}^m \alpha_i x_i)$ if $\alpha_i \geq 0 \; \forall i$ and $\sum_{i=1}^m \alpha_i = 1$. With $\alpha_i \equiv \frac{1}{m}$, we get $\frac{1}{m} \sum_{i=1}^m f(x_i) \leq f(\frac{1}{m} \sum_{i=1}^m x_i)$.

need to allow for inexact pattern matches.

The function d_{ap} reasonably trades off small changes in position distance with small changes in color distance, but it does not prevent larger tradeoffs which may lead to a small signature distance without visual similarity; all region pairs $((A, P), (B, Q))$ with the same value $d_{\mathrm{ap}}((A, P), (B, Q)) = d^{\tau}_{\mathrm{attr}}(A, B) + \lambda d_{\mathrm{pos}}(P, Q)$ contribute the same penalty in the signature distance. SEDL's use of a linear combination of color and position distance identifies near exact pattern matches, captures many inexact matches, and identifies obviously dissimilar patterns, but can lead to false positives. Although some of these false positives will be eliminated in the step described below, there is still room for improvement in SEDL's use of color and position information. For example, the perceived color of a region depends on the color of its surrounding regions, but d_{ap} does not take this into account.

Once the alternating iteration (7.2),(7.4) converges, SEDL checks that at least some fraction of the attribute weight inside the final search rectangle R_* can be matched to pattern attribute weight over moderate distances in attribute space. This final check is in recognition of the difficulty of trading off attribute and position distance, and helps eliminate false positives. A visually similar match between areas implies somewhat similar attribute signatures for those areas.

The final check in the verification and refinement phase is performed using the τ-EMD described in section 4.4.2. Recall that the τ-EMD measures the fraction of weight in the lighter distribution which *cannot* be matched to weight in the heavier distribution using only ground distances that do not exceed τ units. Let $\mathbf{x}^{R_*}$ be the approximate image attribute distribution inside R_* (computed by a range search as described in section 7.3), and let $\mathbf{y}_*$ be the pattern attribute distribution scaled by the final scale value c_* (the area of R_*). Then the last step in the verification and refinement phase is to check that $\tau\text{-EMD}(\mathbf{x}^{R_*}, \mathbf{y}_*) < \eta$. If not, then R_* is eliminated from further consideration. We cannot choose τ or η too small since inexact matches require some tradeoffs in attribute and position distance, and we do not want to incorrectly reject such matches. For the product advertisement database, we use $\eta = 0.5$, requiring that at least 50% of the color weight to be matched; for the Chinese character database, we use $\eta = 0.75$, requiring at least 25% of the orientation weight to be matched.

We now discuss our choice of distance function SD in relation to the ICP algorithm used to register shapes. Recall that the ICP algorithm uses the distance function

$$D^{\mathrm{ICP}}(\mathbf{X}, \mathbf{Y}) = \min_{\psi \in \Psi} \sum_{J=1}^{N} ||P_{\psi(J)} - Q_J||_2^2,$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the point sets

$$\mathbf{X} = \{ P_1, \ldots, P_M \} \qquad \text{and} \qquad \mathbf{Y} = \{ Q_1, \ldots, Q_N \} .$$

The ICP iteration seeks to solve the optimization problem

$$D^{\text{ICP}}_{\mathcal{G}}(\mathbf{X}, \mathbf{Y}) = \min_{g \in \mathcal{G}} D^{\text{ICP}}(\mathbf{X}, \mathbf{Y}) = \min_{g \in \mathcal{G}, \psi \in \Psi} \sum_{J=1}^{N} || P_{\psi(J)} - g(Q_J) ||_2^2 .$$

If we eliminate the weights U_J and the attribute part of the ground distance d_{ap}, then our distance function SD reduces to D^{ICP}, and $\text{SD}_{\mathcal{G}}$ reduces to $D^{\text{ICP}}_{\mathcal{G}}$. In fact, the weights U_J are effectively eliminated when our framework is applied to shape-based retrieval since we compute average orientations over equal-sized portions of image curves to produce the image signature, and the weight of an attribute is equal to the (constant) length over which the averages are computed.

The use of orientation attributes which are not modified by any transformation is a very important improvement over the ICP algorithm when registration considers only changes in scale and location (and not orientation). The ICP framework matches only by ink on the page. By using the orientation of the ink, we can avoid matching query ink that is physically close to image ink, but that will not produce a good visual match between the underlying shapes. The point here is applicable in a more general setting than our shape setting: using attributes which are unmodified during iteration gives the matcher something constant upon which to anchor its matching and transformation decisions. In fairness to the ICP algorithm, it is specifically designed to handle differences in orientation.

Another natural question to raise at this time is why we choose to compare the attribute-position distributions $\mathbf{X}$ and $\mathbf{Y}$ with the function SD instead of the EMD. The main answer is speed. We can use the results in section 6.5 on allowing weight-altering transformations to compute the EMD under a transformation set in which the scale changes both the distribution points and the weights (which represent image areas in the color case). This requires several EMD computations for a single comparison under a transformation set. Our distributions are so large, however, that the time for even a single EMD computation per (query,image) pair is too much for a content-based retrieval system which must maintain some interactivity with the user. Even if we could compute the EMD for SEDL's large signatures in an acceptable time, it is still an imperfect measure of image similarity because our representation is imperfect. All masses are concentrated at the centroids of image regions. For large regions, it would be a more accurate representation to spread the mass uniformly over the region. We do not, however, know how to compute the EMD between two such continuous distributions.

SEDL's color $\times$ position signatures are defined (roughly) by the largest possible uniform color regions. If a single red pattern region is represented in the image as three adjacent regions of slightly different reds, there may be a relatively large, unjustified matching penalty using SEDL's signature distance. To help reduce the effects of such problems, region distances are weighted by pattern region area, the theory being that large pattern regions are more stable in appearance and easier to detect as a single region within the image. The excellent results obtained show that this strategy is effective. In the shape case, SEDL avoids such representation problems to a large extent by using a fine sampling of image curves. This strategy is feasible because the shape data is one-dimensional. A fine sampling over the 2D image plane, however, would produce color $\times$ position too large to match in acceptable times for retrieval.

7.5 Results

In this section, we show the results of the entire matching process for the color advertisement database and the shape Chinese character database.

7.5.1 The Product Advertisement Color Database

The advertisement database and product logos used in the experiments in this section were obtained from the web site `http://vis-www.cs.umass.edu/~mdas/color_proj.html` for the FOCUS work by Das et al. ([14]). The twenty five product logos used as queries are shown in Figure 7.11. The database contains 361 advertisements for products with and without corresponding query logos.

7.5.1.1 Creating Signatures

In what follows, a *signel* (signature element) refers to a single element $((A_I, P_I), W_I)$ in the color $\times$ position signature $\mathbf{X} = \{((A_I, P_I), W_I)\}_{I=1}^{M}$ of an image. The signature creation process consists of three steps: (1) clustering in color space to reduce the number of colors in an image, (2) connecting pixels with the same color label to form an initial set of signels, and (3) combining same-colored signels which are close together in the image to form the final set of signels which define the image signature in color $\times$ position space.

The color clustering step uses an algorithm due to Rubner et al. ([65]) which averages pixel colors together when the colors are contained in a small enough rectangular region of the color space. As in [65], we use the perceptual CIE-Lab color space ([88]). The

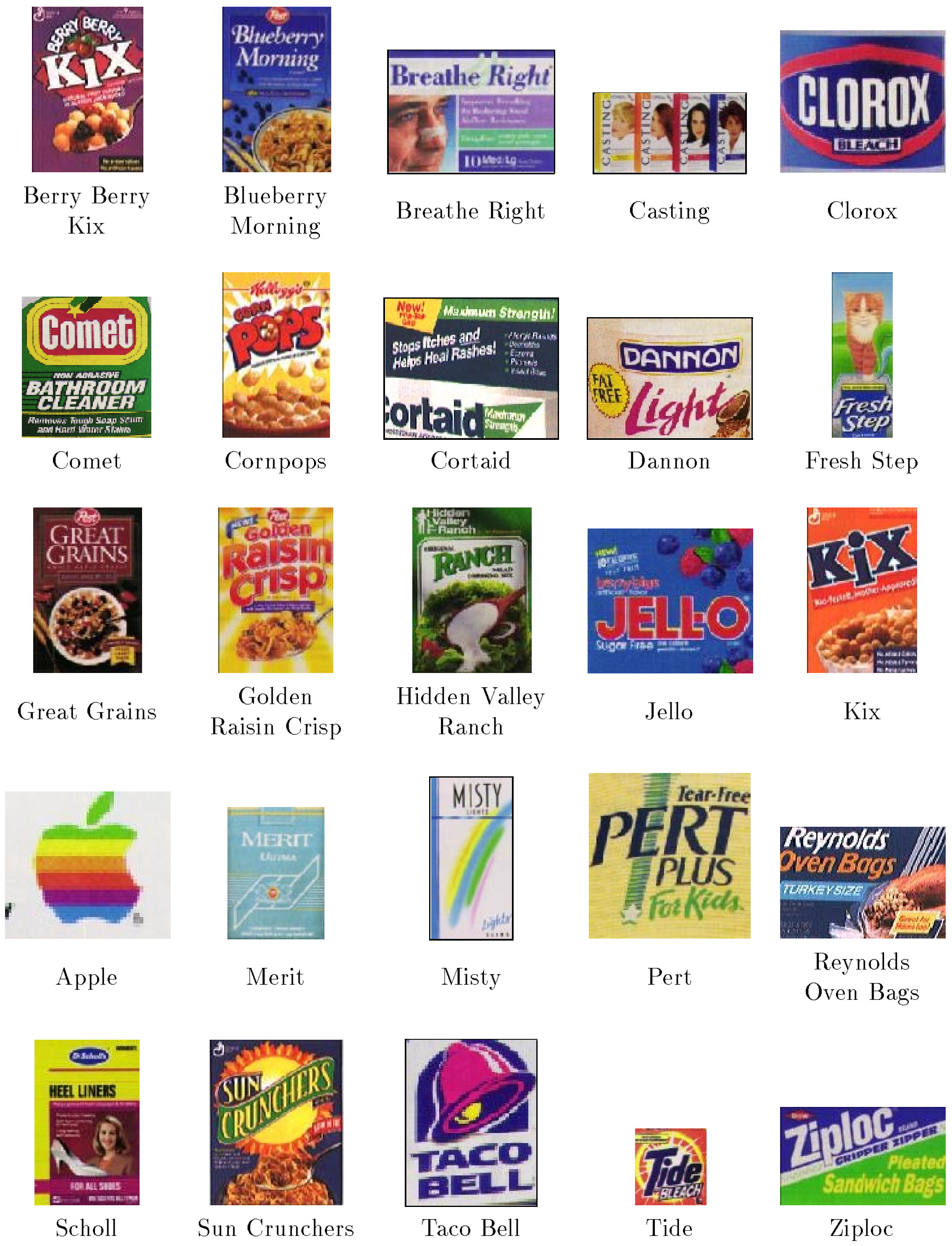


Figure 7.11: Queries for the Color Product Advertisement Database.

L component is a luminance channel, the a component is a red-green channel, and the b component is a blue-green channel. The CIE-Lab space is *perceptual* in the sense that Euclidean distance in this space matches perceptual distance between two colors that are not very different. The initial box which contains all colors ($0 \leq L \leq 100$, $-120 \leq a \leq 120$, $-120 \leq b \leq 120$) is subdivided alternately along each dimension of the color space until the box region size is small enough so that the average color inside the box approximates well all colors inside the box. There is a second stage of the clustering algorithm that merges similar clusters that were broken by fixed boundaries used in the first clustering stage. See [65] for details. By choosing the "small enough" threshold properly, we can usually reduce the image to a small number (5–40) of colors with very little loss of color accuracy. By *reduce*, we mean replace the color at each image pixel with the nearest color cluster representative.

After an image is reduced, we compute all sets of connected pixels with the same color label using Heckbert's seed fill algorithm ([29]). Each of these sets is an initial region in a region-merging process whose ultimate goal is to define the image signature in color $\times$ position space. Associated with each region is a color signature for the pixels contained in the region, along with the position centroid for the region pixels of each color cluster. The color signature for an initial region contains only one color.

When two regions R^i, $i = 1, 2$, are merged into $R = R^1 \bigcup R^2$, we must combine their color signatures and compute the position centroids in R of each color cluster. This is straightforward. Suppose that the image is reduced to n colors $c_1, c_2, \ldots, c_n$, and that region R^i, $i = 1, 2$, has color signature $\mathbf{x}(R^i) = \{ k^i_1, \ldots, k^i_n \}$, where k^i_j is the number of pixels in region R^i with color c_j. Also, let p^i_j denote the centroid of the pixels in R^i with color c_j (if $k^i_j > 0$). Then $\mathbf{x}(R) = \{ k_1, \ldots, k_n \}$, where $k_j = k^1_j + k^2_j$, and $p_j = (k^1_j p^1_j + k^2_j p^2_j)/(k^1_j + k^2_j)$ if $k^1_j + k^2_j > 0$. Here p_j is the centroid of all pixels in $R = R^1 \bigcup R^2$ with color c_j.

Each final region contributes a number of color $\times$ position signels ((color, position), weight) equal to the number of colors contained within the region. For example, if R as given above is a final region, then we get a color $\times$ position signel $((c_j, p_j), k_j)$ for each $k_j > 0$. The role of a region is to define which image pixels of the same color are okay to combine into one signel. Note that merging two regions with no color clusters in common does not change the color $\times$ position signature derived from the set of all regions.

The region merging process maintains a priority queue of all region adjacencies. The next pair of regions to be merged is the adjacent pair with the smallest priority. If we denote the area of a region R by area(R), then the priority for region adjacency (R^1, R^2) is

$$\text{priority}(R^1, R^2) = \text{EMD}(\mathbf{x}(R^1), \mathbf{x}(R^2)) \times \min(\text{area}(R^1), \text{area}(R^2)). \tag{7.7}$$

Here we assume that all region color signatures have been normalized to have total weight equal to one before the EMD is computed. Given our color clustering and seed fill steps to produce the initial set of regions, what ordering on region merges does the priority (7.7) imply?

Consider the situation before any region merging has been performed. The connection of pixels with the same color label typically results in many tiny initial regions of a few pixels or less (more about this later). It is even common to have single pixel initial regions. On the other hand, the EMDs between adjacent initial regions cannot be very small. The EMD between two adjacent initial regions cannot, for example, be zero because this implies that the adjacent initial regions have the same color and, therefore, would have been connected by the seed fill algorithm. Furthermore, the colors of the initial regions are the result of clustering in color space in which colors within a certain distance from each other are merged into a single cluster. Therefore, the EMD between two adjacent initial regions cannot be arbitrarily small.

If the EMD between adjacent regions is, for example, at least e units in CIE-Lab space, then the size of the smaller region in any region adjacency with priority at most p is at most p/e. The minimum priority during the region merging process tends to increase as the process proceeds since the EMDs between adjacent regions and the region sizes both tend to increase. While adjacencies with priority less than p are being processed, regions of size more than p/e are very unlikely to be merged with an adjacent region (the merging process would need to create an adjacency with EMD less than e units for this to happen). The result is that, despite the symmetry in the priority (7.7), region adjacencies with a large EMD and a small minimum size are generally eliminated before adjacencies with a small EMD and large minimum size. Of course, among adjacencies with small size the priority (7.7) gives preference to eliminating those with small EMD over those with large EMD. The causes of these small size adjacencies in the initial set of regions are discussed next.

The first region adjacencies that will be eliminated are those with similar colored regions in which at least one of the regions is very small (the EMD factor and the area factor are both small). Such adjacencies typically arise in areas of the image that have a gradual shading change. The color clustering algorithm may choose two color clusters to represent the lighter and darker colors in such an area, but what happens to the colors which are roughly equidistant from these clusters in color space is much less clear. The color clustering algorithm does not use the positions of the colors in the image plane, so it cannot use a connectedness criterion to help make this decision. Furthermore, the Euclidean distance is only an approximation to perceptual distance in the CIE-Lab color space. If a color is close perceptually to two color clusters, the distances in CIE-Lab space to the clusters will be small but might not properly distinguish which cluster is perceptually closer. A conceptual group of same-colored pixels may be disconnected into several very small groups of connected pixels; there may even be some pixels that are not connected to any pixels of the same color label. Such a fragmented group is likely to be contained in a larger region with a similar color. By merging the very small fragment regions into the larger surrounding region, the fragments are combined into one signel as desired.

The next set of region adjacencies that will be eliminated are those with fairly different color descriptors in which at least one of the regions is very small (the EMD factor is relatively large, but the area factor is small). Such adjacencies typically arise in the advertisements when there is very small lettering in the advertisement. Such lettering is usually placed on a relatively high contrast background to aid legibility. For the purposes of our system, summarizing a paragraph of tiny black lettering with one signel of black at the position centroid of the paragraph is a wise compression since we know a priori that we are not interested in such tiny scale features. This compression is exactly what happens as each individual letter (connected black pixels) region is merged with its background region.

A mistake is made if a merge is performed between two regions with large, spatially separated signels of the same color. In this case, the weight of the combined signel in the final signature will be large and its position will be inaccurate. If there are large amounts of the same color in the regions, then the regions themselves must be relatively large and the EMD between the regions will be relatively small. Adjacencies with large regions and small EMD are, as mentioned above, generally processed after those with at least one small region.

The big question, of course, is when to stop the merging process. Currently we use a very simple criterion which halts the region merging once a pre-specified number of regions has been reached. In a general image database setting, we may or may not have the time and resources to pick a different number of regions for each image entered into the database. For all images in the advertisement database, we stop when there are 128 regions. This number was found acceptable for the complexity of a general advertisement. Examples of the final regions for three database images are shown in Figure 7.12. The problem of finding a "natural" stopping point is a very difficult one, even with a specified range of feature scales in which we are interested. For us, "natural" means that the correct conceptual grouping has been done. For all the query logos, we stop the region merging process when there are 32 regions.

It is interesting to see the results of allowing the region merging process to proceed down to two image regions. See Figure 7.13. The priority (7.7) produces a natural hierarchy of

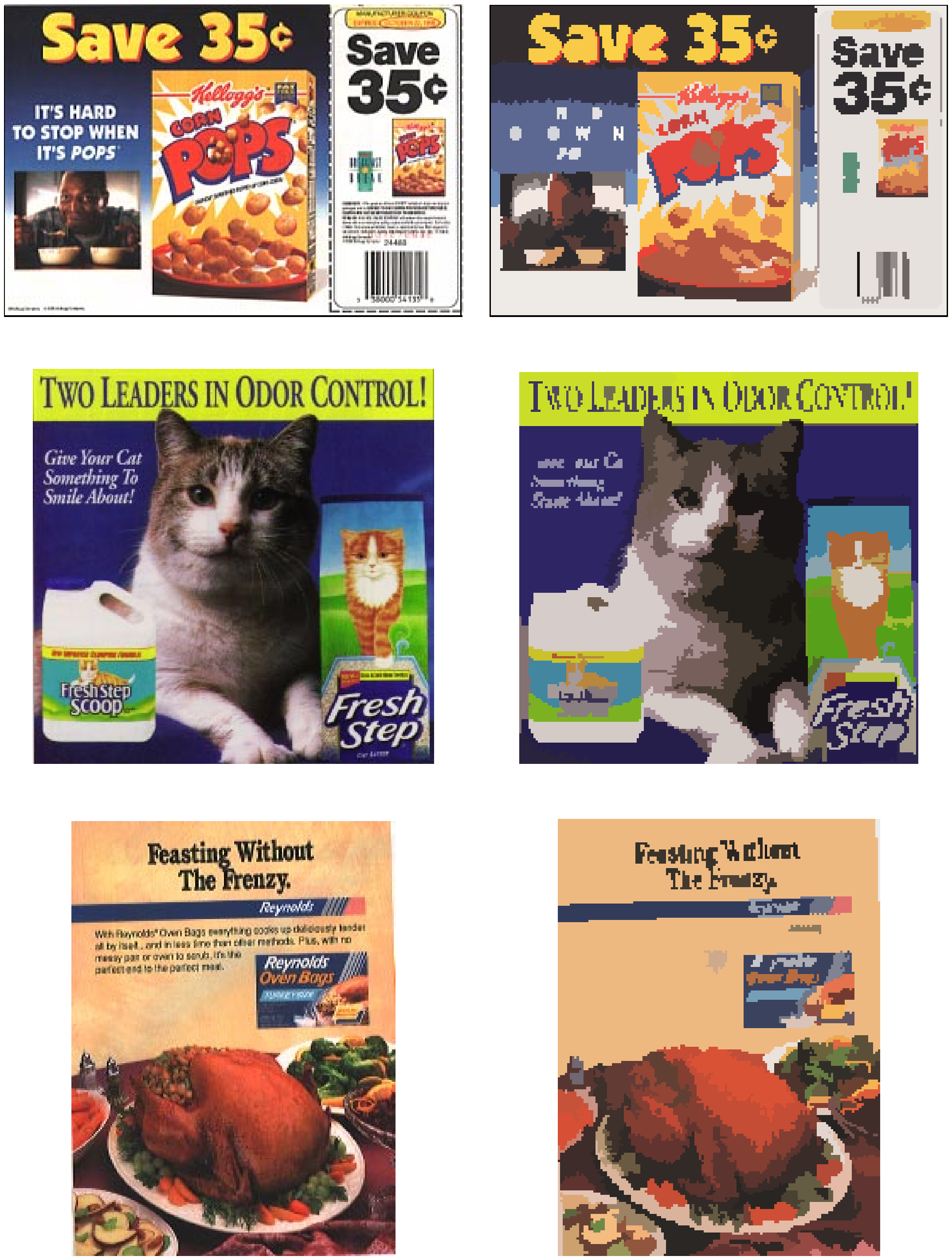
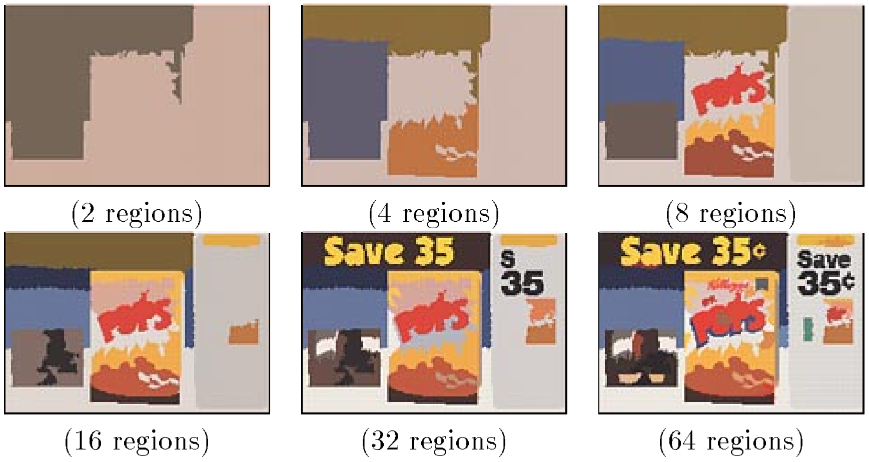
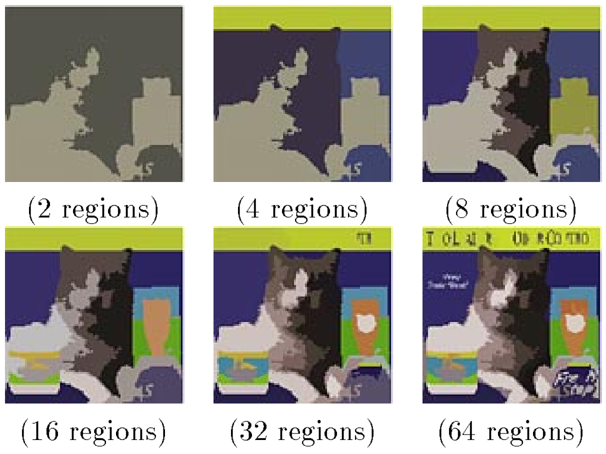

Save 35¢
IT'S HARD TO STOP WHEN IT'S POPS
Kellogg's CORN POPS
Save 35¢
TWO LEADERS IN ODOR CONTROL!
Give Your Cat Something To Smile About!
Fresh Step Scoop
Fresh Step
Feasting Without The Frenzy.
Reynolds
Reynolds Oven Bags

Figure 7.12: Region Merging Results. (left) Original image. (right) The original image reduced to 128 regions. Each region is colored with the average color of its pixels.

(a)



(b)



(c)

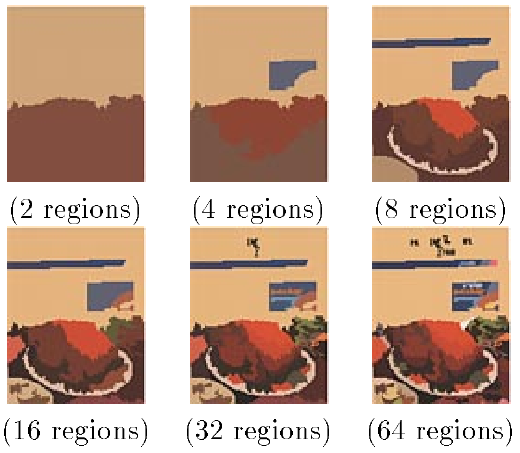


Figure 7.13: More Region Merging Results. Here we show results of allowing the region merging process to proceed down to two image regions. (a) Cornpops advertisement on the top left of Figure 7.12. (b) Fresh Step advertisement on the middle left of Figure 7.12. (c) Reynolds Oven Bag advertisement on the bottom left of Figure 7.12.

segmentations over varying levels of detail (the number of regions).

7.5.1.2 Query Results

We now present some results of SEDL's pattern problem strategy applied to the advertisement database. The "correct" answers for a given product logo are the advertisements for that product. Some exceptions have been made for advertisements that contain a version of the product logo which is substantially different in color from the query logo. See Figure 7.14(a),(b) for example ads that we will not penalize SEDL or FOCUS for missing. At the same time, we retain more challenging cases when the advertisement does not contain the exact logo but is close enough that a good system should be able to handle the small differences. See Figure 7.14(c),(d) for two such examples. The first column in Figure 7.15 shows in parentheses the number of correct answers for each query logo. The FOCUS results were obtained directly from the FOCUS web demo at `http://vis-www.cs.umass.edu/~mdas/color_proj.html` on 8/17/98. See section 2.6 for a review of the FOCUS retrieval system.

The *recall* of a query for a given number R of returned images is the number of correct retrievals in the top R retrievals divided by the number of correct answers. From the first and second columns in Figure 7.15, we see, for example, that SEDL's recall for the Dannon query is $3/4 = 75\%$ and for the Clorox query is $5/5 = 100\%$.

The *precision*, on the other hand, is a measure of the spread of the correct images returned. If there were four correct retrievals in the top twenty, then a query result is more precise if the correct retrieval ranks are 1, 2, 3, and 4 than if the ranks are 3, 7, 10, and 12. Some researchers measure the precision as the number of correct retrievals divided by the rank of the last correct retrieval. This measure, however, does not distinguish between the rank sets 1,2,3,12 and 9,10,11,12; both have precision $4/12 = 33\%$ under the previous definition. We use a definition that accounts for the distribution of ranks by weighing ranks close to one more heavily than ranks closer to R. If there are n correct retrievals in the top R with ranks $1 \leq r_1 < r_2 \cdots < r_n$, then the precision is

$$\rho = \frac{\sum_{i=1}^{n}(R+1-r_i)}{\sum_{i=1}^{n}(R+1-i)}. \tag{7.8}$$

Here we give rank 1 a weight of R, rank 2 a weight of $R-1$, and, in general, rank j a weight of $R+1-j$. The denominator in (7.8) normalizes the precision so that $0 < \rho \leq 1$, with $\rho = 1$ indicating perfect precision ($r_i \equiv i$). If there are no correct retrievals (i.e. $n = 0$), then we set $\rho = 1$ since this bad result is already punished in the recall statistics. The goal, of course, is to obtain simultaneously high recall and precision.

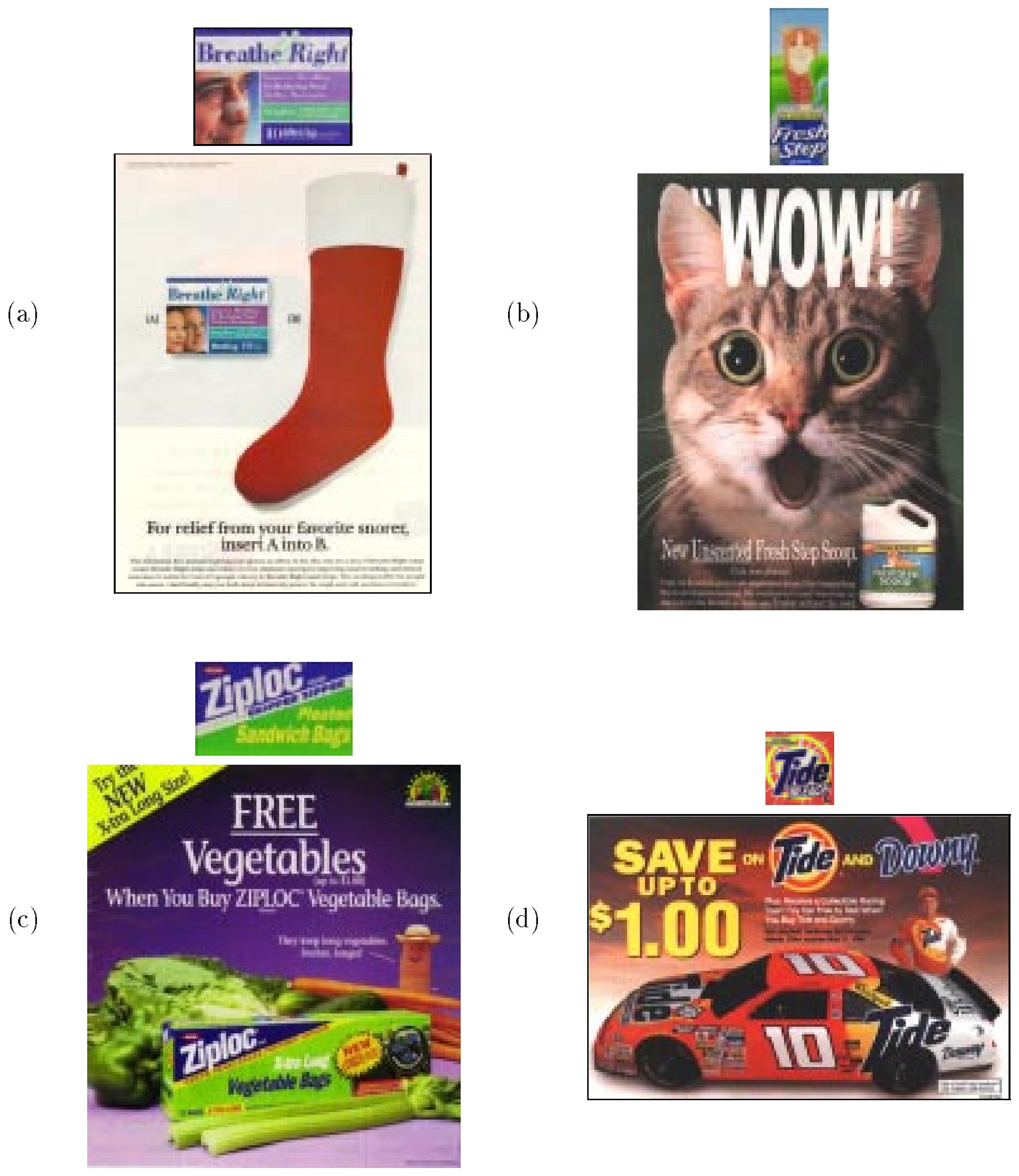

(a)
Breathe Right
For relief from your favorite snorer, insert A into B.
(b)
Fresh Step
"WOW!"
New Unscented Fresh Step Scoop.
(c)
Ziploc Pleated Sandwich Bags
Try the NEW X-tra Long Size!
FREE Vegetables
When You Buy ZIPLOC Vegetable Bags.
Ziploc X-tra Long Vegetable Bags
(d)
Tide
SAVE ON Tide AND Downy
UP TO $1.00

Figure 7.14: Tough Advertisements to Retrieve. (a) The layout of the Breathe Right logo occurrence is the same as that of the query logo, but the colors are different. (b) The advertisement is for "Fresh Step Scoop", while the logo is for "Fresh Step". The layout for the labels of these two products is quite different. (c) The labels for "Ziploc Sandwich Bags" (the query) and "Ziploc Vegetable Bags" (in the advertisement) are very similar but not exactly the same. (d) The Tide advertisement does not contain the Tide box (the query), but it does contain the core Tide logo at the top center.

Query	SEDL Ranks	FOCUS Ranks
Berry Berry Kix (2)	8	2, 4
Blueberry Morning (4)	1, 3, 20	1, 3, 4, 9
Breathe Right (3)	1, 2, 3	1, 2, 3
Casting (4)	1, 5, 13	1, 2
Clorox (5)	1, 3, 4, 7, 8	1, 2, 3
Comet (2)	1, 2	—
Cornpops (2)	1, 3	12, 13
Cortaid (2)	1, 2	2, 8
Dannon (4)	1, 3, 5	1
Fresh Step (2)	1, 2	1, 3
Great Grains (2)	2, 4	3
Golden Raisin Crisp (2)	1, 15	1, 2
Hidden Valley Ranch (9)	1, 2, 6, 11	1, 5
Jello (2)	1, 2	1
Kix (3)	1, 5, 11	1
Apple (3)	1, 4	1, 2, 9
Merit (6)	1, 2, 4, 5, 12, 13	1
Misty (6)	1, 2, 3, 14	—
Pert (2)	1, 11	—
Reynolds Oven Bags (5)	1, 2, 3, 9	1, 2, 3, 5
Scholl (4)	1, 2, 9	—
Sun Crunchers (3)	1	1, 9
Taco Bell (2)	1, 2	1, 4
Tide (7)	1, 2, 8, 12	1, 6, 15, 24, 39
Ziploc (4)	1, 2, 14	1, 2, 4

Figure 7.15: Query Results for the Color Advertisement Database. The FOCUS ranks are phase two results. An "—" entry means that there were no correct answers in the returned images.

In the experiments described in this section, the database searched by SEDL is a subset of the database used in the FOCUS web demo. We selected 361 advertisements of the 1200 advertisements and nature images in the FOCUS database.[2] In all the examples in section 7.3.1, using $l_{\max} = 2$ initial transformations was sufficient to find the pattern occurrences. There are, however, some examples that require a couple more initial placements to find the pattern. Here we use $l_{\max} = 4$ initial placements. As in the scale estimation experiments in section 4.5.1, we set the smallest scale in which we are interested to $c_{\min} = 0.001 = 0.1\%$. Finally, for efficiency reasons we do not proceed with the match verification for a particular initial transformation if the first distance iterate $D^{(0)}$ is very large. If this happens, we simply discard the initial placement.

For SEDL's recall and precision statistics, we use $R = 20$ images (approximately 5.5% of the SEDL database). To be fair to FOCUS, we use $R = 40$ images (about 3.3% of the FOCUS database)[3] Figure 7.15 shows the ranks of the correct retrievals for both SEDL and FOCUS after phase two, while Figure 7.16 gives the summary recall and precision statistics. SEDL's total recall of 77.8% is higher than the FOCUS total recall of 53.3%. Both systems achieved high precision with R which is very small compared to the database size. The average FOCUS precision was 96%, while SEDL's average precision was 88%. The final column in Figure 7.16 shows the time SEDL takes for each query. SEDL's average query time is 40.0 seconds. FOCUS has a sizeable advantage over SEDL in speed, requiring less than a second on average for a query in their larger database of 1200 images. There is more to say, however, about the differences between SEDL and FOCUS than just comparing statistics. We return to this comparison in section 7.5.1.3 after we show some SEDL query results.

Figures 7.17–7.24 show the first five or ten images returned by SEDL for several of the query logos. These figures also show the scale, orientation, and location where SEDL believes that the pattern occurs in some of the correct retrievals. Many more examples of where SEDL finds the pattern are given in Figures 7.25–7.27. In general, SEDL is very accurate in localizing the patterns. Most of the pattern occurrences are *not* rotated from their logo form. By allowing an orientation with this database, we are making SEDL's job more difficult. Sometimes the orientation is far from correct (e.g. see the Taco Bell advertisement in the bottom-right of Figure 7.24). No comparison of accuracy with FOCUS

[2]For the query logos, 18 of the 25 are exactly the same as those used in the FOCUS demo. Small modifications were made to the other 7 logos to eliminate areas not part of the actual logo and to make a rectangular pattern. The seven logos which are different are Comet, Cortaid, Dannon, Apple, Merit, Misty, and Tide. The differences are small and the results are still fair to compare.

[3]Only two ranks for correct retrievals by FOCUS were greater than 20, with the maximum being 39 (see the ranks for the Tide query in Figure 7.15).

Query	SEDL Recall	FOCUS Recall	SEDL Precision	FOCUS Precision	SEDL Query Time (secs)
Berry Berry Kix	1/2	2/2	0.65	0.96	38.2
Blueberry Morning	3/4	4/4	0.68	0.95	20.0
Breathe Right	3/3	3/3	1.00	1.00	41.7
Casting	3/4	2/4	0.77	1.00	37.0
Clorox	5/5	3/5	0.91	1.00	33.6
Comet	2/2	0/2	1.00	1.00	33.6
Cornpops	2/2	2/2	0.97	0.72	58.5
Cortaid	2/2	2/2	1.00	0.91	26.6
Dannon	3/4	1/4	0.97	1.00	118.2
Fresh Step	2/2	2/2	1.00	0.98	32.0
Great Grains	2/2	1/2	0.92	0.95	37.3
Golden Raisin Crisp	2/2	2/2	0.67	1.00	35.8
Hidden Valley Ranch	4/9	2/9	0.86	0.96	28.2
Jello	2/2	1/2	1.00	1.00	14.7
Kix	3/3	1/3	0.81	1.00	34.2
Apple	2/3	3/3	0.95	0.95	3.1
Merit	6/6	1/6	0.85	1.00	25.9
Misty	4/6	0/6	0.86	1.00	40.2
Pert	2/2	0/2	0.77	1.00	60.2
Reynolds Oven Bags	4/5	4/5	0.93	0.99	110.8
Scholl	3/4	2/4	0.89	1.00	7.4
Sun Crunchers	1/3	2/3	1.00	0.91	33.7
Taco Bell	2/2	2/2	1.00	0.97	21.6
Tide	4/7	5/7	0.82	0.63	103.0
Ziploc	3/4	3/4	0.81	0.99	4.4
average	70/90 $\doteq$ 77.8%	48/90 $\doteq$ 53.3%	0.88	0.96	40.0

Figure 7.16: Recall, Precision, and Timing Results for the Color Advertisement Database. The FOCUS statistics are for phase two ranks.

is possible since FOCUS does not compute the scale or location of the pattern.

7.5.1.3 SEDL versus FOCUS

In FOCUS, adjacency graph vertices and edges are computed based on a grid imposed on an image, as described in detail in section 2.6. All the blue, for example, in a single grid cell is treated as one region, and there are edges between different color vertices within the same cell and neighboring cells. This reduces the number of graph vertices, but leads to false adjacencies which can cause false matches. The FOCUS authors did an excellent job of choosing the cell size so that the graphs are small enough to match quickly, yet still descriptive enough to yield excellent precision.

SEDL uses the amounts of colors, the sizes of regions, and (roughly) the centroids of uniform color regions, while FOCUS uses only the existence of colors and the possibility that two different color regions are adjacent. We believe that at least the amount of information used in SEDL will be necessary for large databases, and we speculate that FOCUS will need much better position localization, which means fewer false adjacencies in larger graphs requiring more time to match, to maintain high precision. Perhaps the shapes of regions (not used in SEDL or FOCUS) will also be needed as the database size increases. Of course, using too much information may reduce the recall rate since inexact matches need to be allowed. The issue of which and how much information needed for large databases is far from settled.

Finally, recall that the SEDL framework is general enough to be applied to both the color and shape pattern problems, whereas the FOCUS system is specific to the color case.

7.5.2 The Chinese Character Shape Database

In this section, we apply our general pattern retrieval strategy to find shape patterns within a database of 2000 Chinese characters. A small sample of the images in this database is shown in Figure 7.28. This collection of Chinese characters is an ideal database to test our ideas because there are many patterns that occur throughout the database at different scales and locations. Our shape summary of a character is the *medial axis* of the set of black pixels which define the character. The results shown in Figure 7.29 were computed using algorithms and software by Ogniewicz and Kübler ([52]). The skeletons are a very good one-dimensional summary of the characters.

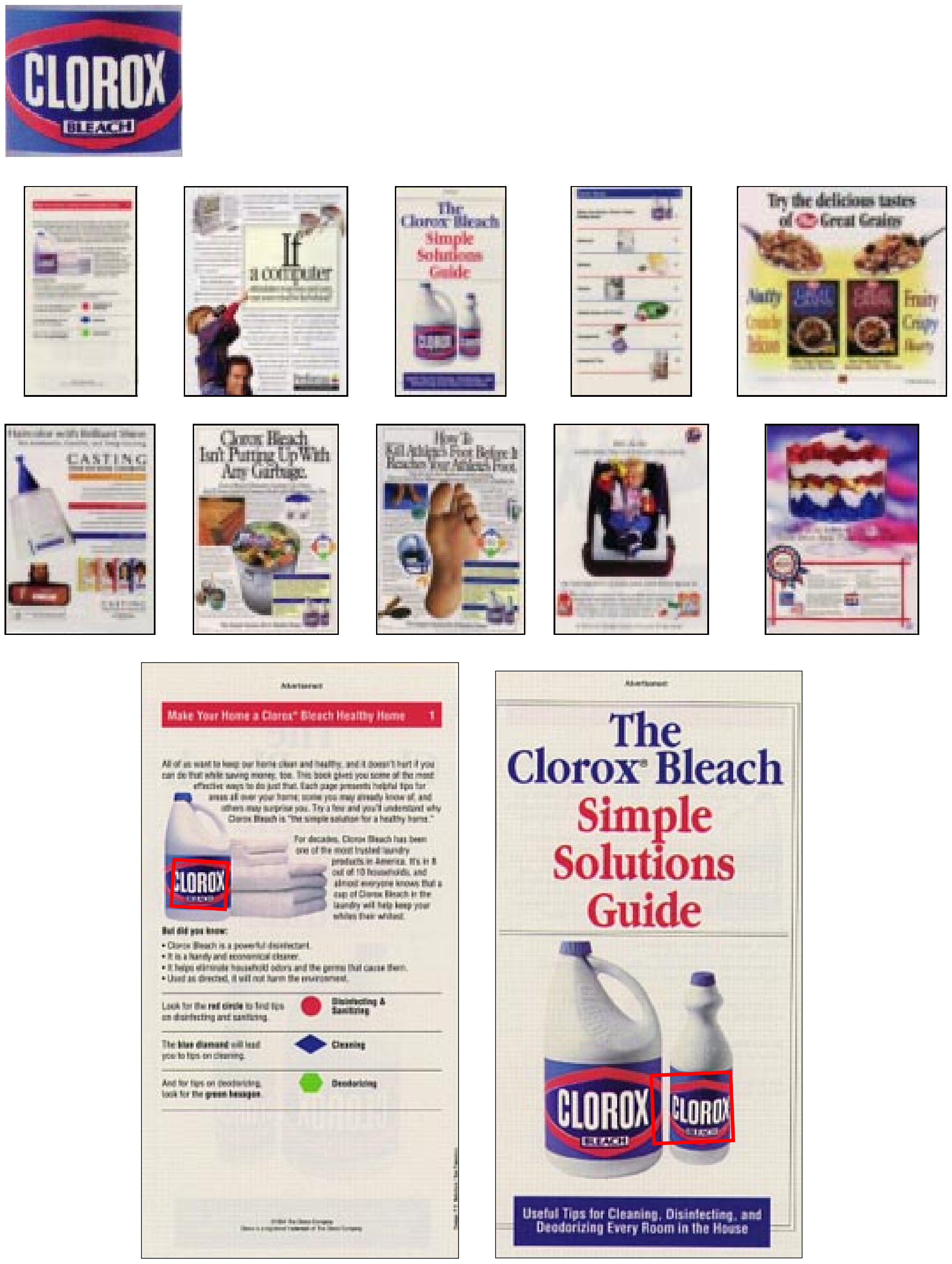


Figure 7.17: Advertisement Query Result – Clorox. (top) Clorox query logo. (middle) The top ten images returned by SEDL. The Clorox advertisements are at positions 1, 3, 4, 7, and 8. (bottom-left) The match is excellent (rank=1st). (bottom-right) One of the Clorox logo occurrences is found, but the final scale is slightly too big (rank=3rd).

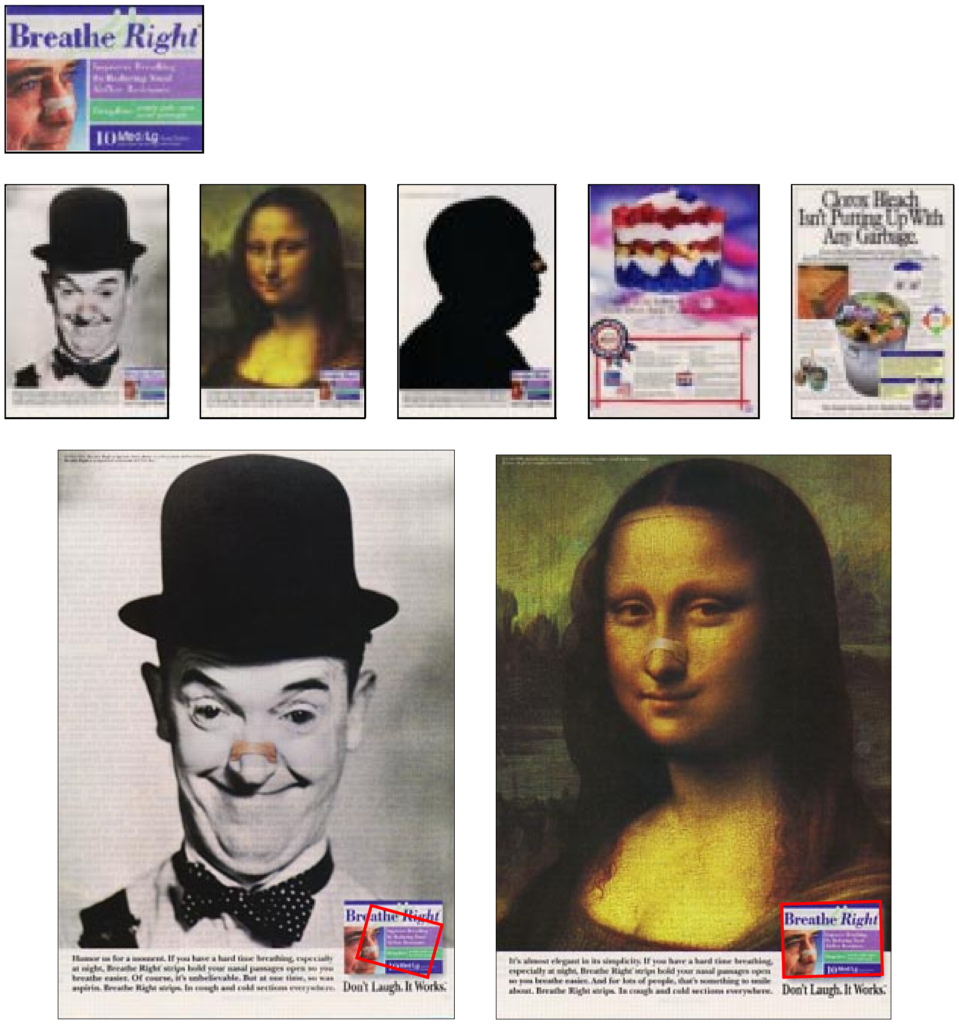


Figure 7.18: Advertisement Query Result – Breathe Right. (top) Breathe Right query logo. (middle) The top five images returned by SEDL. The Breathe Right advertisements are at positions 1, 2, and 3. (bottom-left) The pattern occurence is correctly located, but the orientation is incorrect (rank=1st). (bottom-right) The match is perfect (rank=2nd).

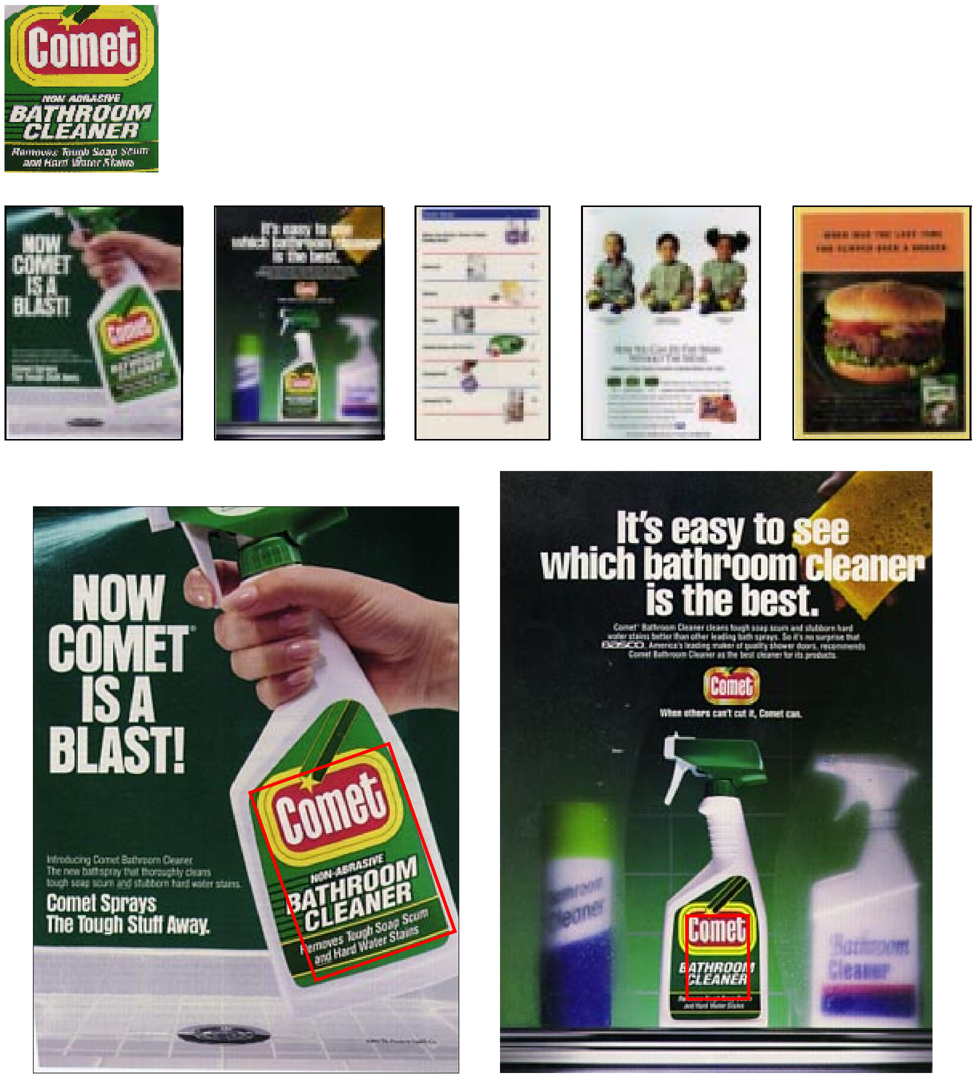

Comet
NON-ABRASIVE
BATHROOM
CLEANER
Removes Tough Soap Scum and Hard Water Stains
NOW COMET IS A BLAST!
Comet Sprays The Tough Stuff Away.
It's easy to see which bathroom cleaner is the best.
When others can't cut it, Comet can.

Figure 7.19: Advertisement Query Result – Comet. (top) Comet query logo. (middle) The top five images returned by SEDL. The Comet advertisements are at positions 1 and 2. (bottom-left) The final pattern placement has the correct orientation and is nearly perfect (rank=1st). (bottom-right) The pattern is correctly located, but the final scale should be a little bit larger (rank=2nd).

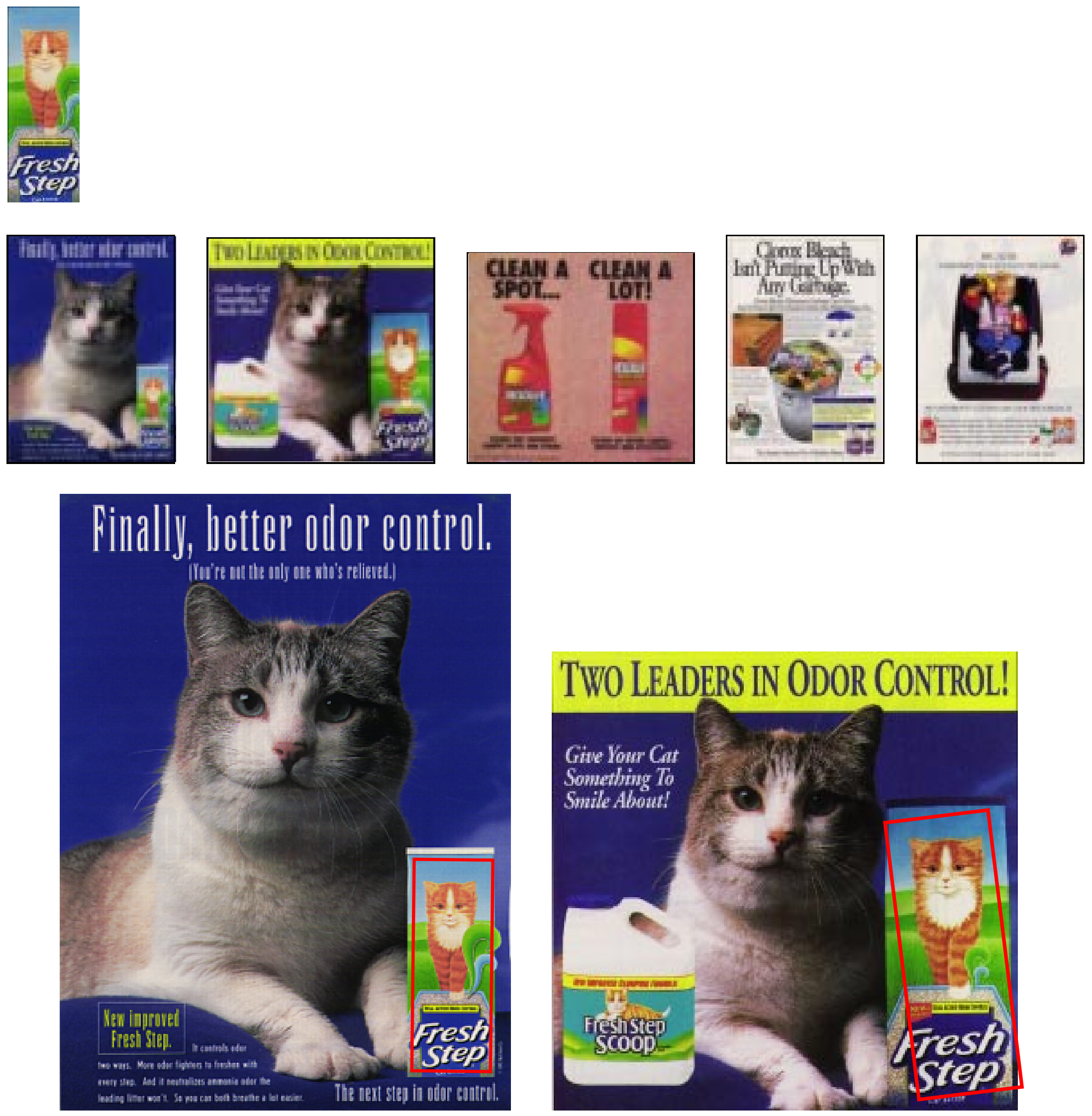

Fresh Step
Finally, better odor control.
(You're not the only one who's relieved.)
New improved Fresh Step.
The next step in odor control.
TWO LEADERS IN ODOR CONTROL!
Give Your Cat Something To Smile About!
Fresh Step Scoop
CLEAN A SPOT...
CLEAN A LOT!
Clorox Bleach Isn't Putting Up With Any Garbage.

Figure 7.20: Advertisement Query Result – Fresh Step. (top) Fresh Step query logo. (middle) The top five images returned by SEDL. The Fresh Step advertisements are at positions 1 and 2. (bottom-left) The match is excellent (rank=1st). (bottom-right) The match is good, but the orientation is incorrect (rank=2nd).

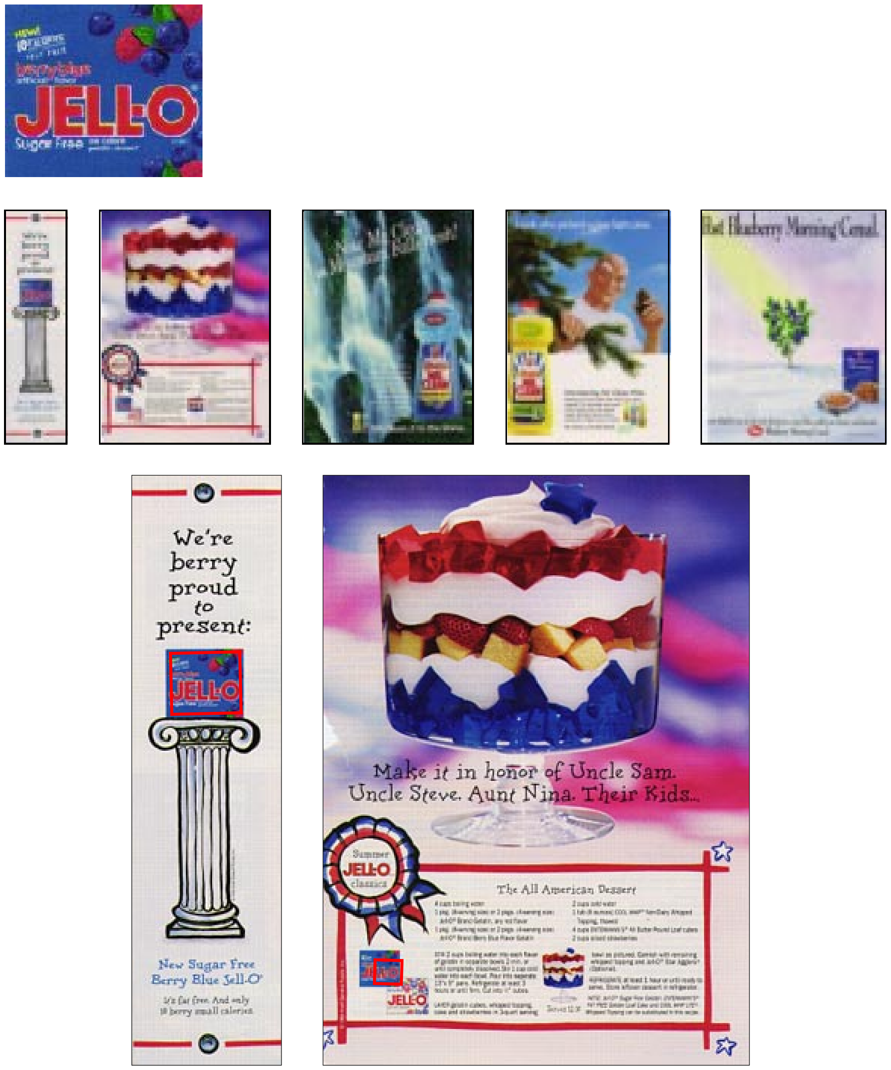

JELL-O
Sugar Free
We're berry proud to present:
New Sugar Free Berry Blue Jell-O
Make it in honor of Uncle Sam. Uncle Steve. Aunt Nina. Their Kids...
Summer JELL-O classics
The All American Dessert

Figure 7.21: Advertisement Query Result – Jello. (top) Jello query logo. (middle) The top five images returned by SEDL. The Jello advertisements are at positions 1 and 2. (bottom-left) The match is perfect (rank=1st). (bottom-right) The final placement overlaps a significant amount of the pattern occurrence, but the scale is too small (rank=2nd).

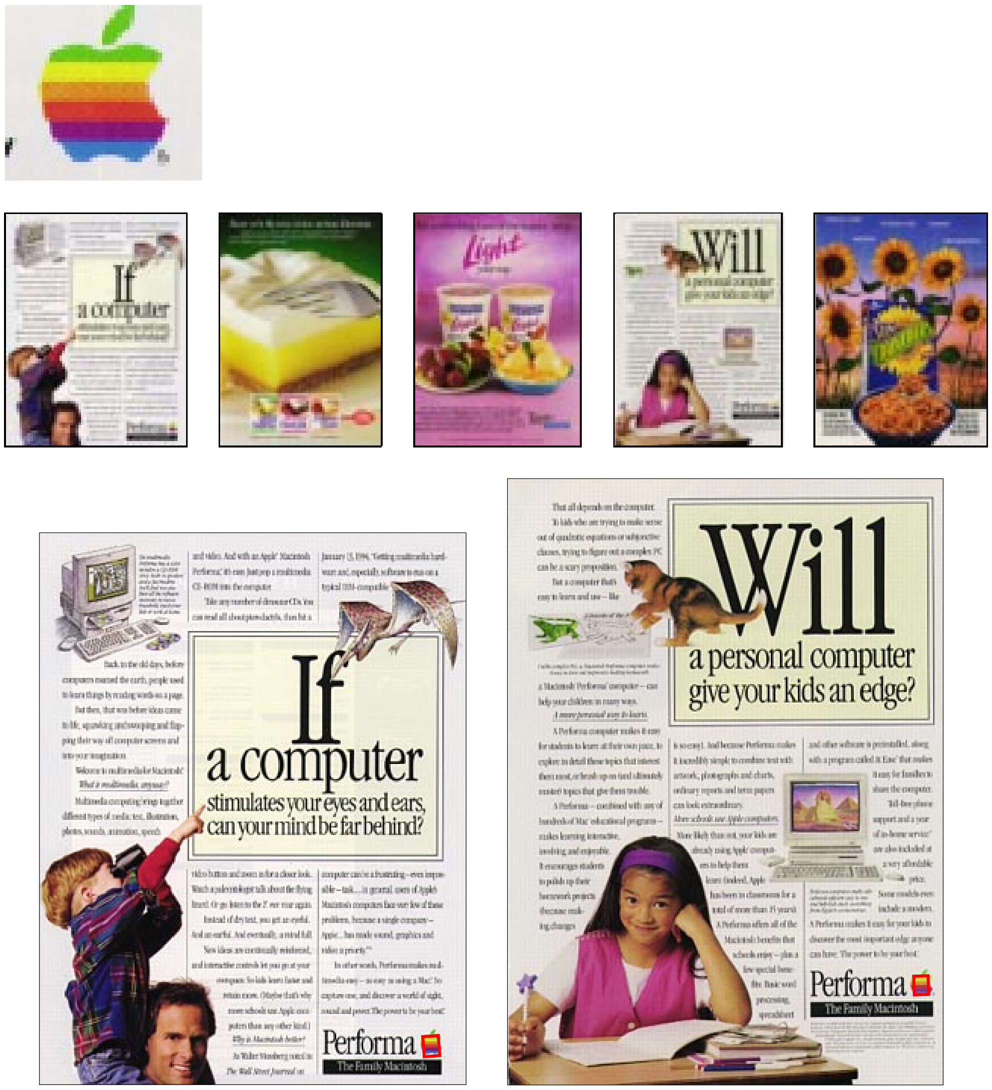


Figure 7.22: Advertisement Query Result – Apple. (top) Apple query logo. (middle) The top five images returned by SEDL. The Apple advertisements are at positions 1 and 4. (bottom-left),(bottom-right) SEDL's algorithm located even these very small scale occurrences of the pattern (ranks=1st,4th).

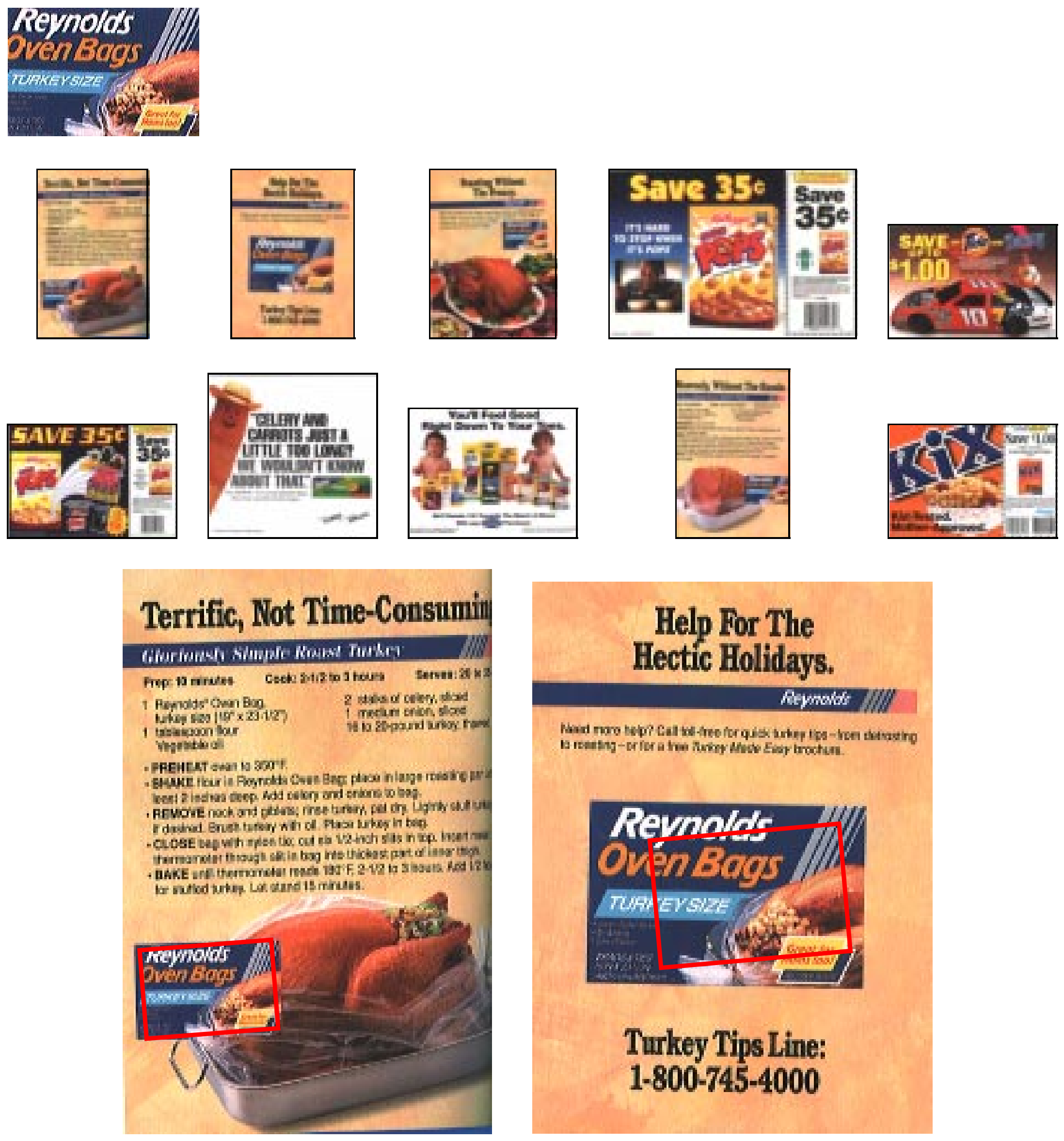

Reynolds
Oven Bags
TURKEY SIZE
Save 35¢
SAVE 35¢
Kix
Terrific, Not Time-Consumin
Gloriously Simple Roast Turkey
Help For The Hectic Holidays.
Reynolds
Turkey Tips Line:
1-800-745-4000

Figure 7.23: Advertisement Query Result – Reynolds Oven Bags. (top) Reynolds Oven Bags query logo. (middle) The top ten images returned by SEDL. The Reynolds Oven Bags advertisements are at positions 1, 2, 3, 9. (bottom-left) The match is excellent (rank=1st). (bottom-right) The final pattern placement is within the pattern occurrence at a scale which too small (rank=2nd).

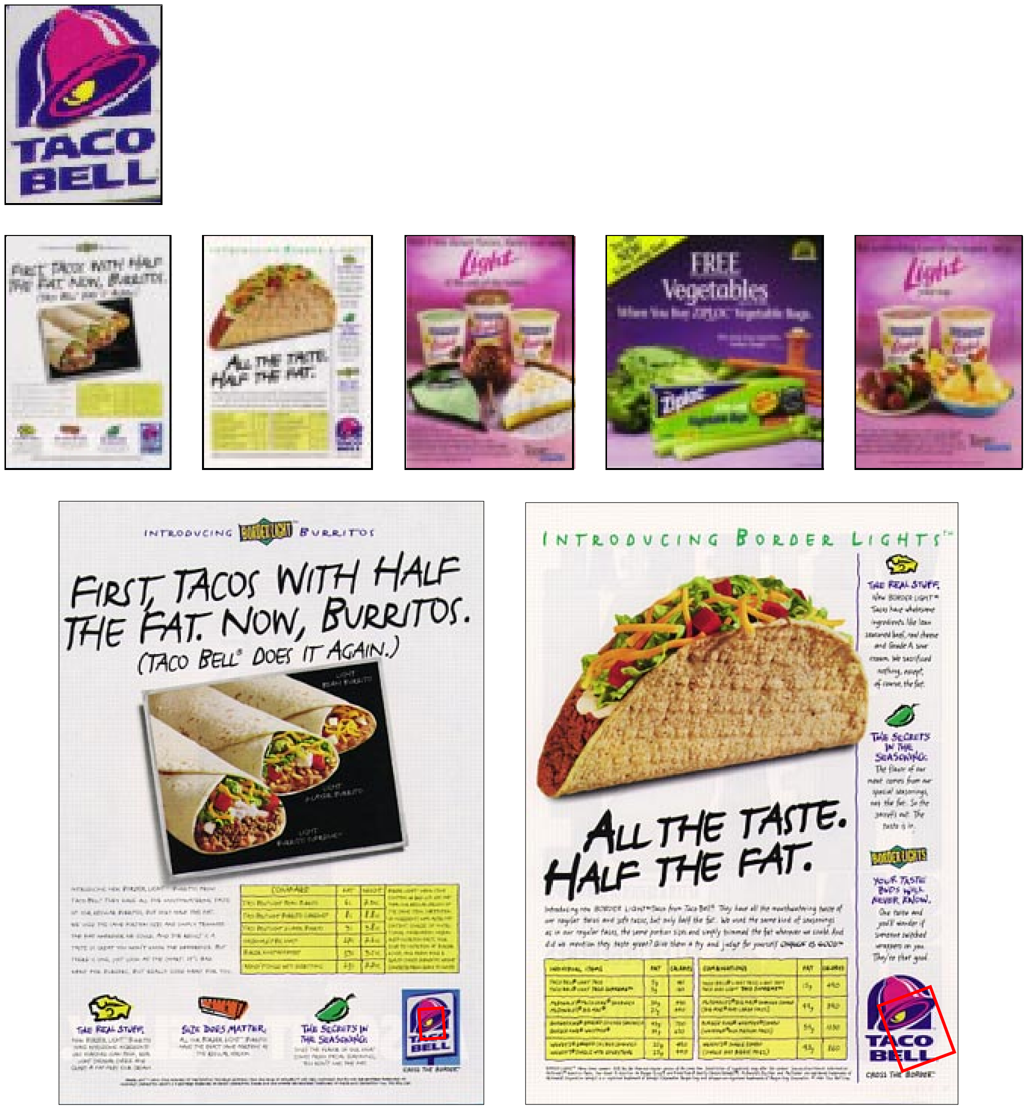

TACO BELL
FREE Vegetables
Light
INTRODUCING BORDER LIGHTS BURRITOS
FIRST, TACOS WITH HALF THE FAT. NOW, BURRITOS.
(TACO BELL® DOES IT AGAIN.)
INTRODUCING BORDER LIGHTS™
ALL THE TASTE. HALF THE FAT.
CROSS THE BORDER™

Figure 7.24: Advertisement Query Result – Taco Bell. (top) Taco Bell query logo. (middle) The top five images returned by SEDL. The Taco Bell advertisements are at positions 1 and 2. (bottom-left) As in the previous Reynolds Oven Bags query, the pattern is correctly located but with an underestimated scale (rank=1st). (bottom-right) The final pattern placement overlaps a large fraction of the pattern occurence, but the orientation is incorrect (rank=2nd).

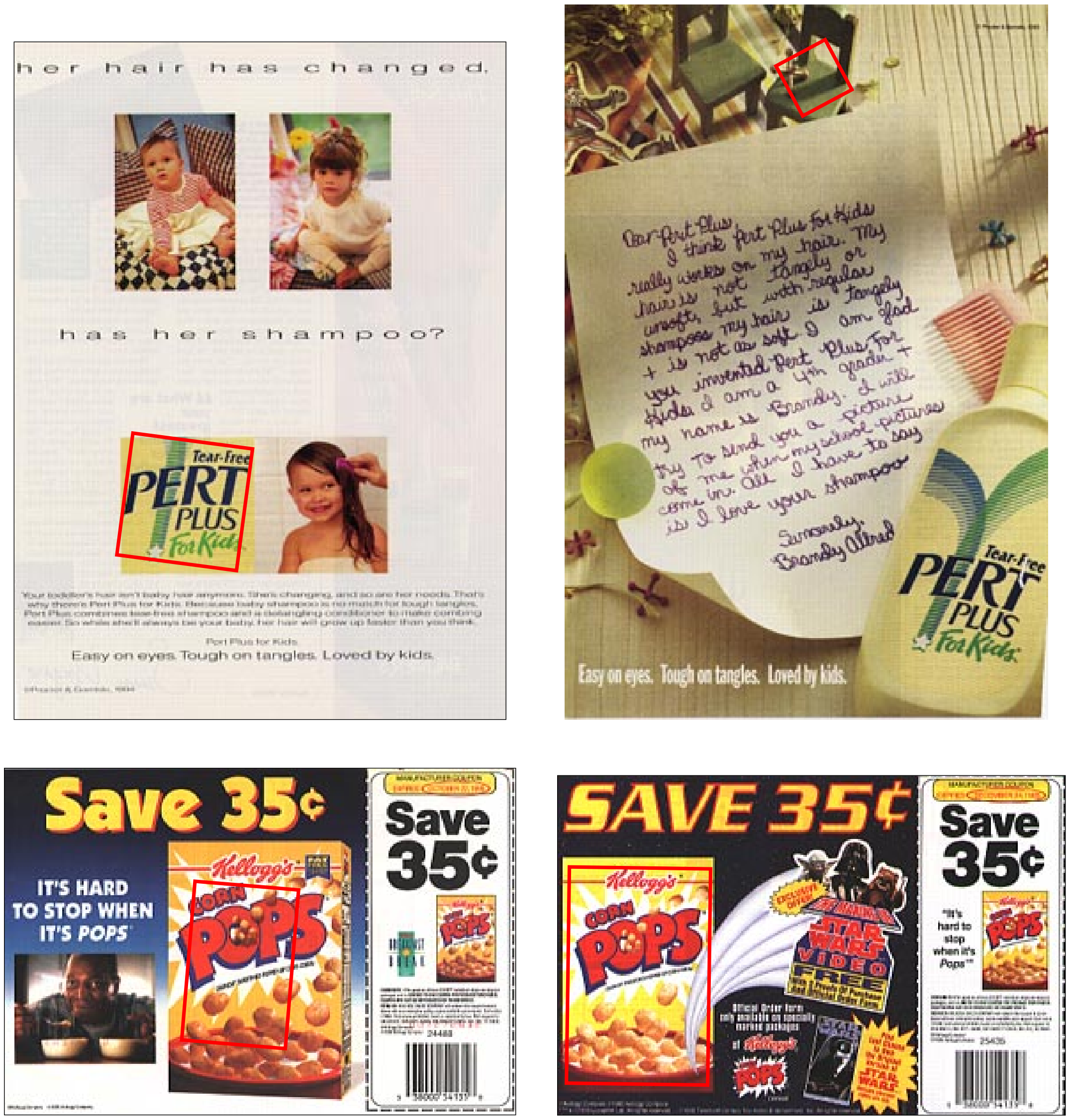

her hair has changed.
has her shampoo?
Tear-Free PERT PLUS For Kids
Easy on eyes. Tough on tangles. Loved by kids.
Save 35¢
IT'S HARD TO STOP WHEN IT'S POPS
Kellogg's CORN POPS
SAVE 35¢
STAR WARS VIDEO FREE

Figure 7.25: Verification and Refinement Results – Example Set 1. (top-left) Pert query. The final transformation has the correct scale and translation, but the orientation is slightly off (rank=1st). (top-right) Pert query. The identifed image area has a similar overall color to the Pert logo, but this final pattern placement is incorrect (rank=11th). (bottom-left) Cornpops query. The position and orientation are correct, but the scale is too small (rank=3rd). (bottom-right) Cornpops query. The final transformation is nearly perfect (rank=1st).

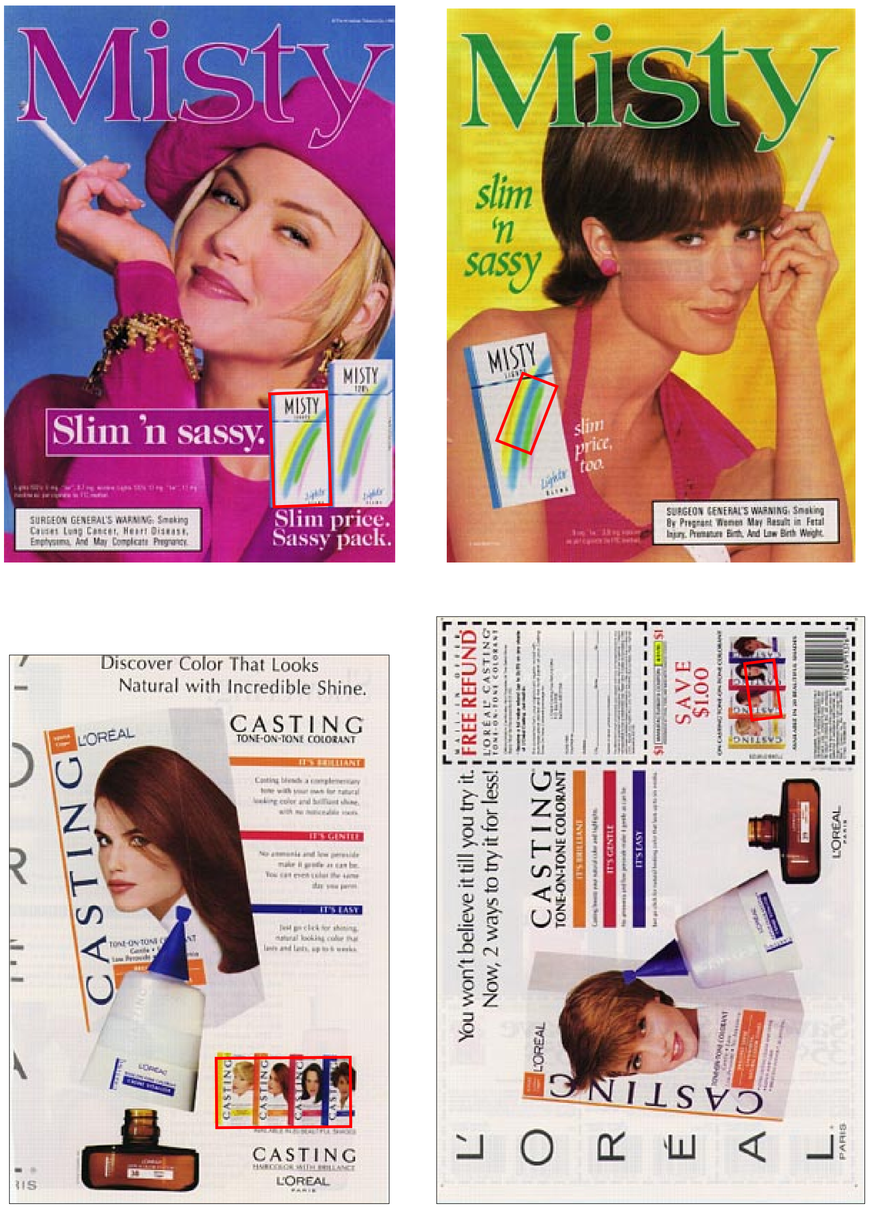


Figure 7.26: Verification and Refinement Results – Example Set 2. (top-left) Misty query. The final pattern placement is perfect (rank=1st). (top-right) Misty query. The pattern occurrence is correctly located, but the scale is too small and the orientation does not follow the slope of the box (rank=2nd). (bottom-left) Casting query. The match is nearly perfect (rank=1st). (bottom-right) Casting query. The final scale is too small, but the final placement has the correct 90° orientation (rank=5th).

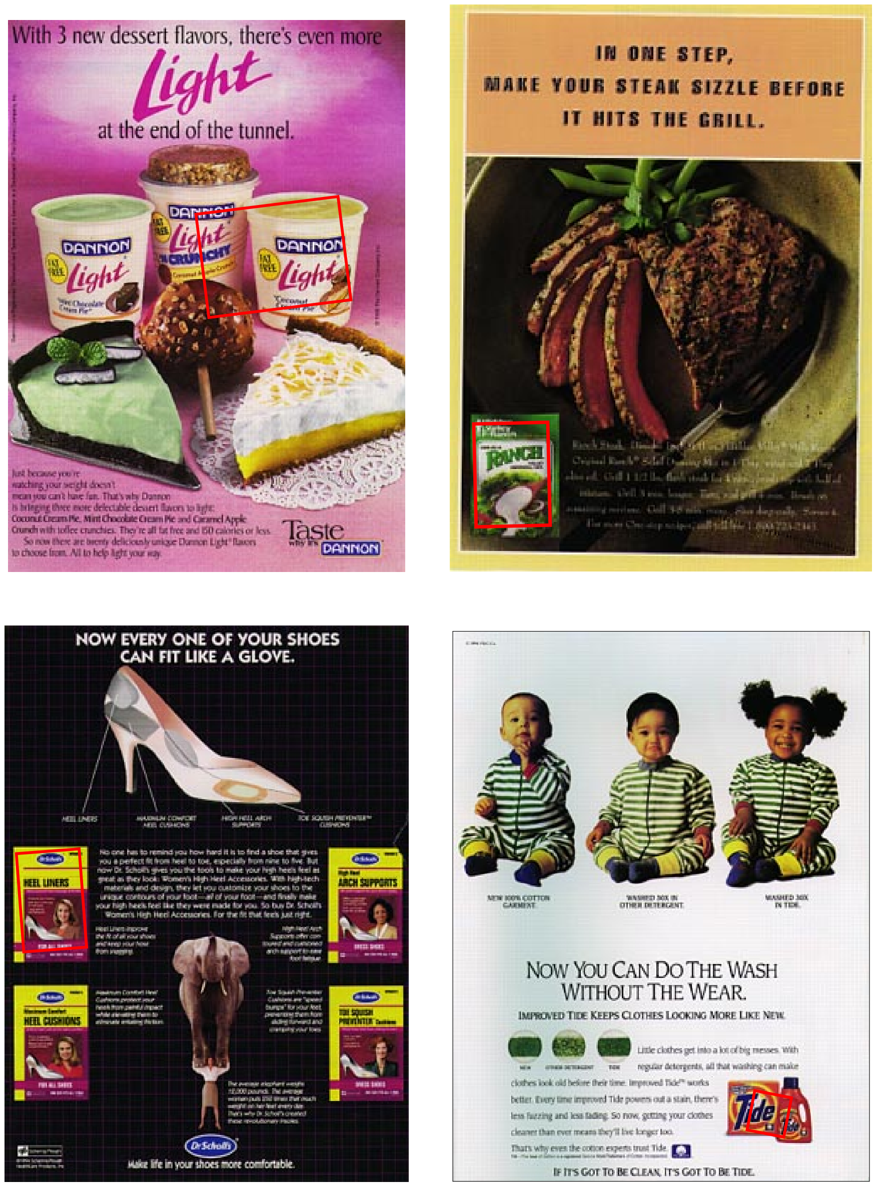

With 3 new dessert flavors, there's even more
Light
at the end of the tunnel.
DANNON
Taste
DANNON
IN ONE STEP,
MAKE YOUR STEAK SIZZLE BEFORE
IT HITS THE GRILL.
RANCH
NOW EVERY ONE OF YOUR SHOES
CAN FIT LIKE A GLOVE.
HEEL LINERS
ARCH SUPPORTS
HEEL CUSHIONS
TOE SQUISH PREVENTER
Dr.Scholl's
Make life in your shoes more comfortable.
NEW 100% COTTON GARMENT.
WASHED 30X IN OTHER DETERGENT.
WASHED 30X IN TIDE.
NOW YOU CAN DO THE WASH
WITHOUT THE WEAR.
IMPROVED TIDE KEEPS CLOTHES LOOKING MORE LIKE NEW.
Tide
IF IT'S GOT TO BE CLEAN, IT'S GOT TO BE TIDE.

Figure 7.27: Verification and Refinement Results – Example Set 3. (top-left) Dannon query. The final position is between two occurrences of the Dannon logo (rank=1st). (top-right) Hidden Valley Ranch query. The match is excellent (rank=1st). (bottom-left) Scholl query. This match is also excellent, although the scale should be a little greater and the orientation is slightly off (rank=1st). (bottom-right) Tide query. The final position is between two occurrences of the Tide logo (rank=1st).

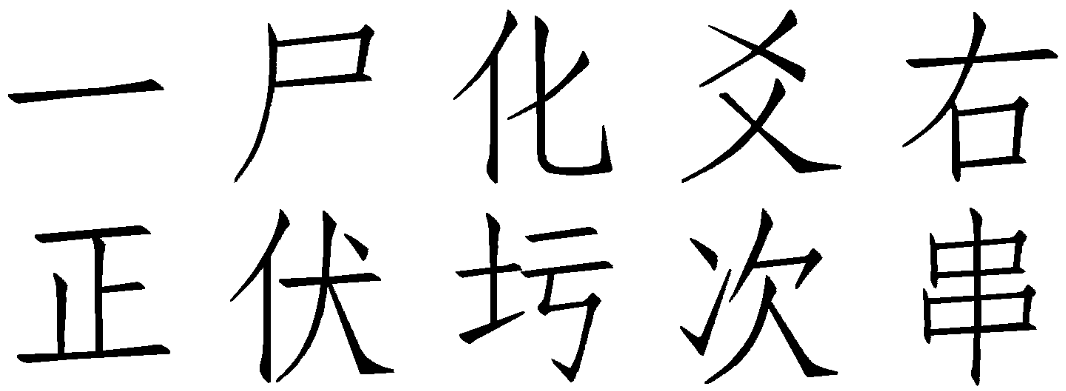


Figure 7.28: Sample Images from the Chinese Character Database. The images are bitmaps.

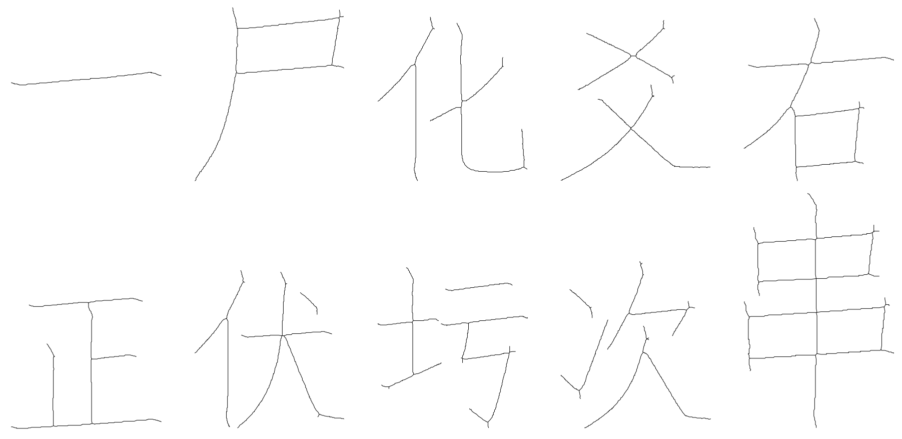


Figure 7.29: Medial Axis Shape Summaries. The shape summary of a Chinese character bitmap is the medial axis of the set of black pixels which define the character. Here we show the summaries for the characters in Figure 7.28.

7.5.2.1 Creating Signatures

The signature creation process operates on the medial axis shape summaries of the characters. The medial axis of a character is represented as a collection of polylines in the image plane. For each polyline, we compute the average position and average orientation of nonoverlapping pieces or samples of a fixed length. The average orientations for an image are clustered to produce a small set of representative orientations $\{ a_1, \ldots, a_n \}$. These orientations are the points in the orientation signature of an image. The weight w_i of each orientation cluster a_i is the total polyline length which is classified as having orientation a_i. The orientation of a polyline sample is the orientation cluster which is closest to the average sample orientation. The weight W_I of $(A_I, P_I) =$ (orientation cluster, average position) in the orientation $\times$ position signature is the sample length, which is the same for every sample except possibly those at the end of a polyline. The orientation signature is the orientation $\times$ position signature marginalized over position.

The same sample length $L = 10$ points is used for every image in the database, where each database image is contained in a square of size 256 points $\times$ 256 points. This results in a relatively dense sampling of the medial axis polylines of a character. Ideally, a database image and a query image are sampled so that the query sample intervals correspond exactly to the sample intervals for the query occurrence within the image. If this were the case, then the average image and query positions would differ by exactly a scaling and translation (assuming that an exact pattern copy with the same orientation appears within the image). Of course this scale and translation is precisely the information that we seek, so it is not possible to obtain the perfect signatures for matching. The average number of orientation clusters is 21.8, while the average number of points in the orientation $\times$ position signature is 119.8.

7.5.2.2 Query Results

The query results shown in this section are the result of matching image and query signatures under the transformation set $\mathcal{G}$ which allows changes in scale and location but *not* in orientation, and ground distance $d = L_{1,\pi}$, the cyclic L_1 distance with period $T = \pi$. We use a cyclic distance so that there is a small distance between an orientation close to 0 and an orientation close to π. The minimum scale parameter is set to 0.25=25% of the total length of the character medial axis. We choose two initial transformations from which to begin the phase 3 iteration that verifies positional consistency and adjusts the scale and location of the match. Finally, we do not allow matches between signels whose orientations differ by more than $\tau = 20°$.

Each of Figures 7.30–7.44 shows the results of a single query into the Chinese character database. The query pattern and its top thirty matches are shown in top row of these figures. In the bottom row, we show where SEDL believes that the query occurs in a selected subset of the matches. Overall, the results of our pattern retrieval algorithm in the Chinese character database are excellent. Almost all the top matches for a query contain the query pattern or a pattern which is very similar to the query pattern. Occurrences of a query are usually well localized, although there are a few examples in which the pattern scale is too large (see e.g. Figure 7.30(iv) and Figure 7.36(viii)) and/or the pattern location is slightly in error (see e.g. Figure 7.32(i) and Figure 7.39(ii)).

There are two points to keep in mind when examining the query results. First, there is no penalty for extra ink on the page that obscures the pattern occurrence; one may have to look closely to see that the pattern is indeed present. This is the case, for example, in Figure 7.30(viii), Figure 7.33(iii), and Figure 7.33(viii). Second, SEDL just matches ink on the page and does not take note of whether or not strokes are connected, or how strokes are connected. Consider the example shown in Figure 7.31(vii) in which the scale of the pattern occurrence is overestimated. This overestimation is understandable because the boxed region contains ink in a very similar layout to that of the query pattern. SEDL does not know that the right part of the query pattern should be one continuous stroke. It only sees that there is ink of the correct orientation in roughly the correct position, and it pays a small price for the incorrect location. A similar example is given in Figure 7.31(vi). The fact that a pattern which is similar to the query pattern appears is merely a coincidence of the juxtaposition of two separate parts of the retrieved character. Neither part alone contains the query. Of course, there are examples in which no matter how hard one looks, nothing similar to the query pattern is present. Examples of such errors are shown in Figure 7.30(vi) and Figure 7.36(vii). Our combination of orientation and position distances produces a small overall distance even though the higher level structure of the pattern is clearly not present in these cases.

The running time for each of the queries in Figures 7.30–7.44 is shown in Figure 7.45. The average time for a single query into our 2000 character database is 95.7 seconds. Thus the average time to compare one (query,image) pair is approximately 0.05 seconds.

7.5.2.3 Possible Modifications

SEDL's signature distance function is asymmetric. The distance is small whenever each piece of all pattern curves is near a similarly-oriented piece of an image curve. There is no penalty for image ink which does not match pattern ink. As we saw in the previous section,

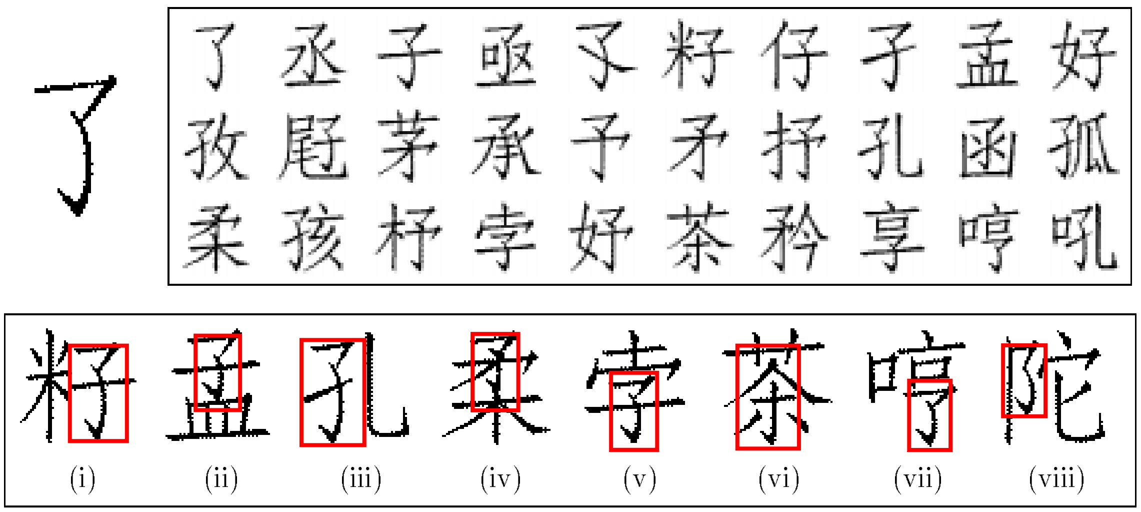

了
了 丞 子 亟 孓 籽 仔 孑 孟 好
孜 [illegible] 茅 承 予 矛 抒 孔 函 孤
柔 孩 杼 孛 好 茶 矜 享 哼 吼
籽 孟 孔 柔 孛 茶 哼 陀
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.30: Chinese Characters Query Result – Example 1. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 5th, (ii) 9th, (iii) 18th, (iv) 21st, (v) 24th, (vi) 26th, (vii) 29th, (viii) 34th (not shown in (top)).

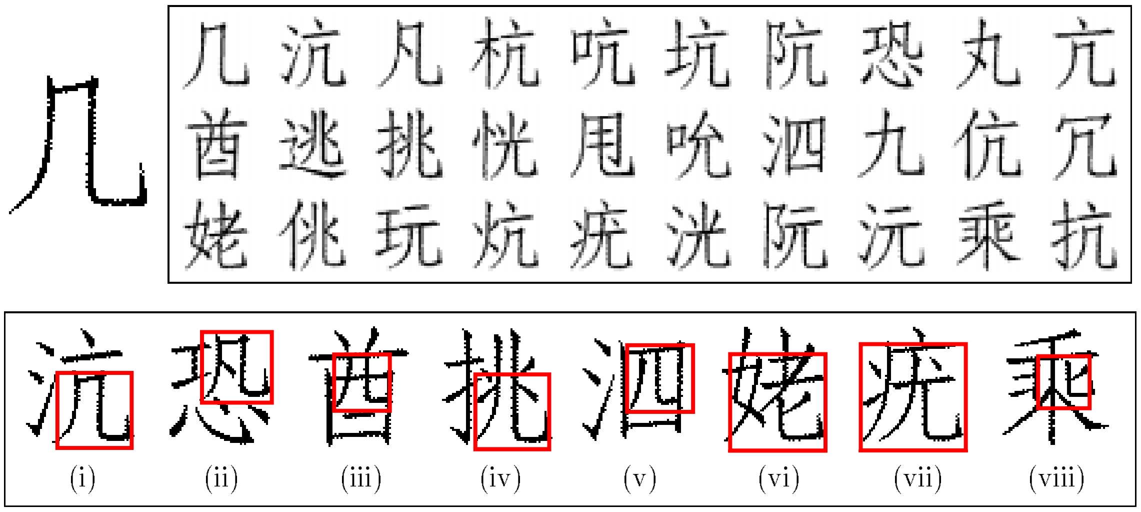

几
几 沆 凡 杭 吭 坑 阬 恐 丸 亢
酉 逃 挑 恍 甩 吮 泗 九 伉 冗
姥 佻 玩 炕 疣 洸 阮 沅 乘 抗
沆 恐 酉 挑 泗 姥 疣 乘
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.31: Chinese Characters Query Result – Example 2. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 2nd, (ii) 8th, (iii) 11th, (iv) 13th, (v) 17th, (vi) 21st, (vii) 25th, (viii) 29th.

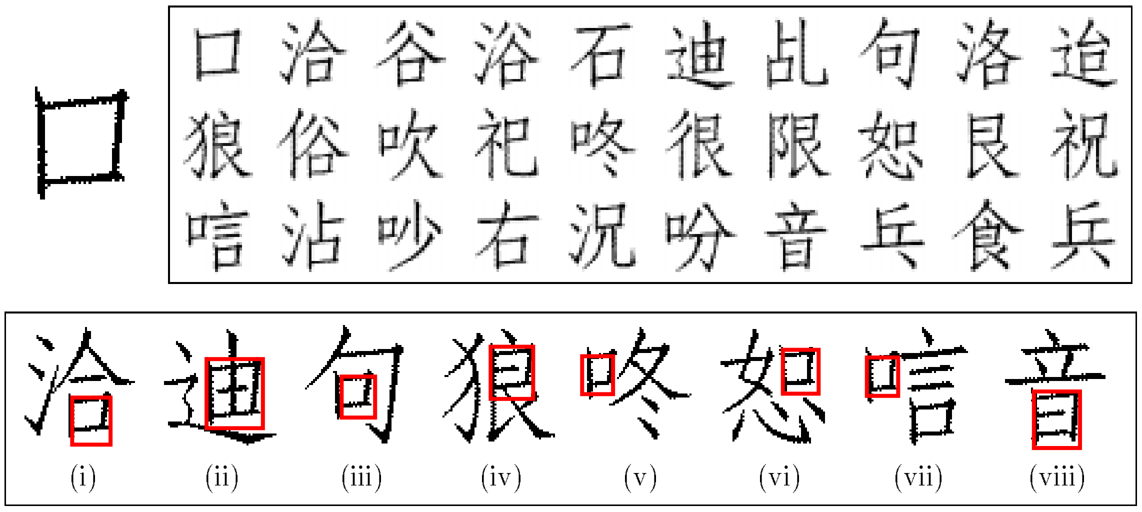

口
口 洽 谷 浴 石 迪 乩 句 洛 逭
狼 俗 吹 祀 咚 很 限 恕 艮 祝
唁 沾 吵 右 況 吩 音 乓 食 兵
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.32: Chinese Characters Query Result – Example 3. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 2nd, (ii) 6th, (iii) 8th, (iv) 11th, (v) 15th, (vi) 18th, (vii) 21st, (viii) 27th. The scale is slightly overestimated in (ii). In this case, the query pattern appears five times (the large square and the four smaller squares contained inside the big one) and SEDL finds the largest occurrence.

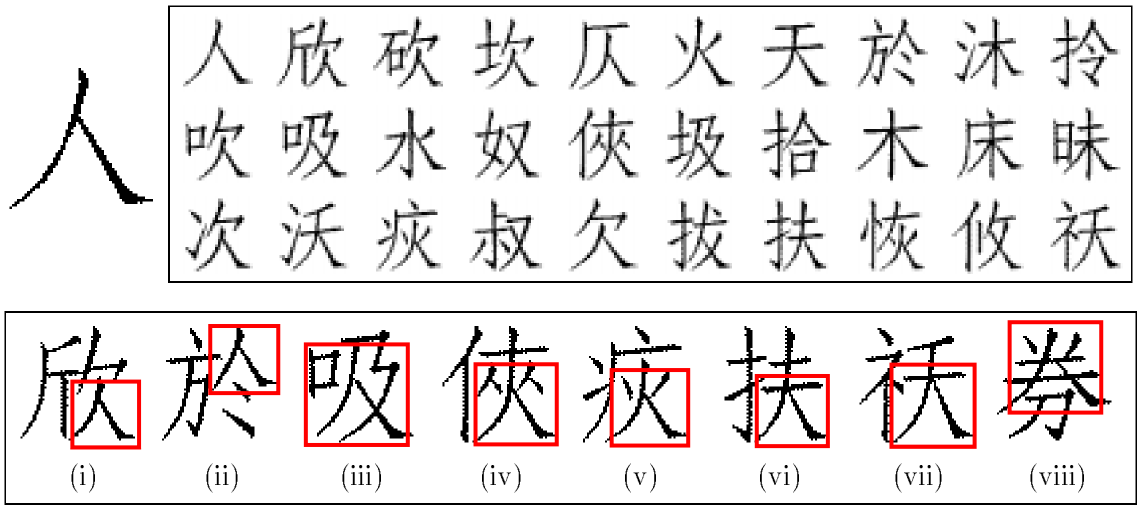

人
人 欣 砍 坎 仄 火 天 於 沐 拎
吹 吸 水 奴 俠 圾 拾 木 床 昧
次 沃 痰 叔 欠 拔 扶 恢 攸 祆
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.33: Chinese Characters Query Result – Example 4. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 2nd, (ii) 8th, (iii) 12th, (iv) 15th, (v) 23rd, (vi) 27th, (vii) 30th, (viii) 37th (not shown in (top)).

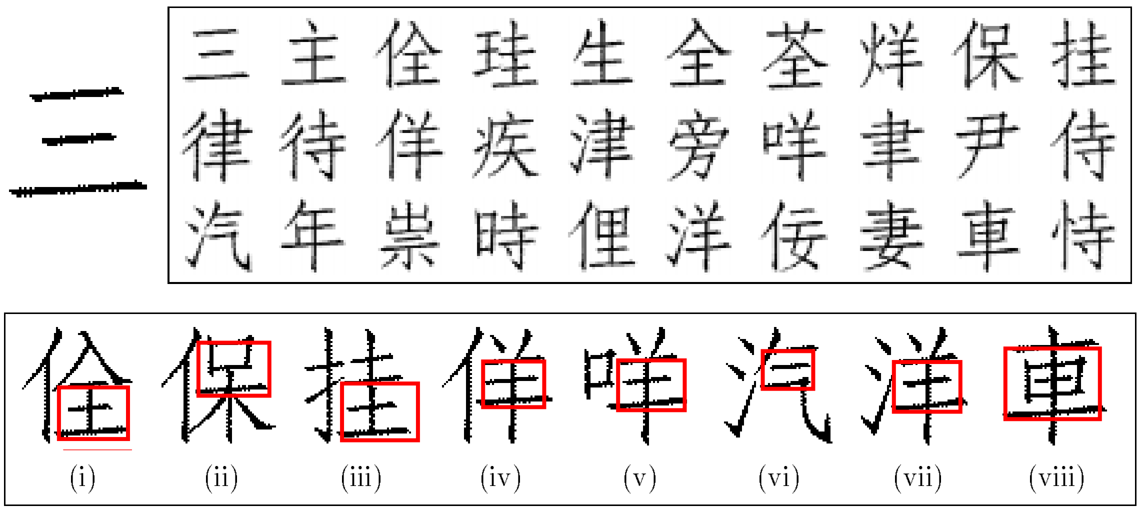

三
三 主 佺 珪 生 全 荃 烊 保 挂
律 待 佯 疾 津 旁 咩 聿 尹 侍
汽 年 祟 時 俚 洋 侫 妻 車 恃
佺 保 挂 佯 咩 汽 洋 車
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

Figure 7.34: Chinese Characters Query Result – Example 5. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 3rd, (ii) 9th, (iii) 10th, (iv) 13th, (v) 17th, (vi) 21st, (vii) 26th, (viii) 29th.

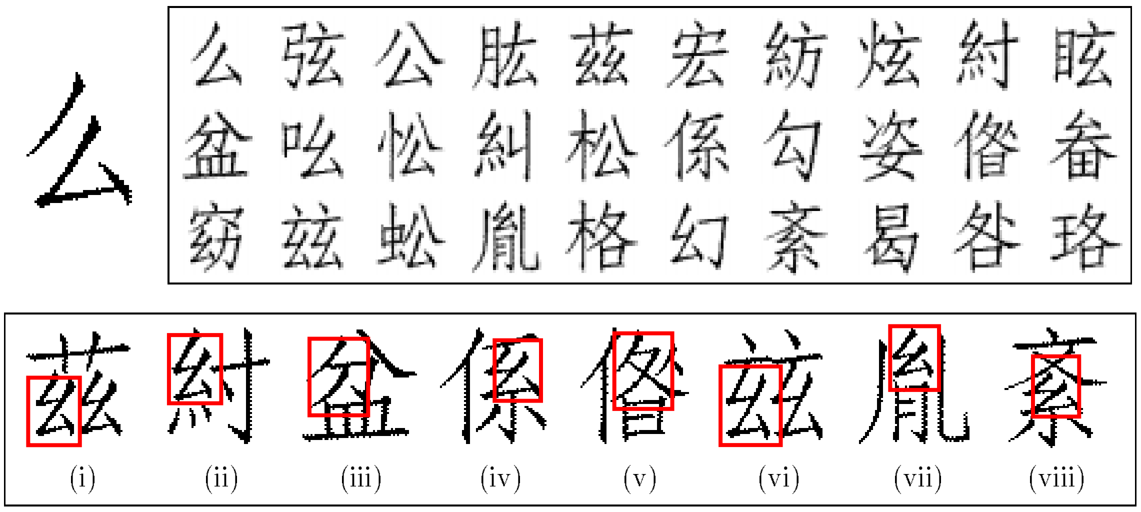

厶
厶 弦 公 肱 茲 宏 紡 炫 紂 眩
盆 吆 忪 糾 松 係 勾 姿 倄 畚
窈 玆 蚣 胤 格 幻 紊 曷 咎 珞
茲 紂 盆 係 倄 玆 胤 紊
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

Figure 7.35: Chinese Characters Query Result – Example 6. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 5th, (ii) 9th, (iii) 11th, (iv) 16th, (v) 19th, (vi) 22nd, (vii) 24th, (viii) 27th. A pattern similar to the query occurs in (iii) and (v) because of the juxtaposition of two separate parts of the characters. In both cases, the surrounding ink obscures the pattern appearance.

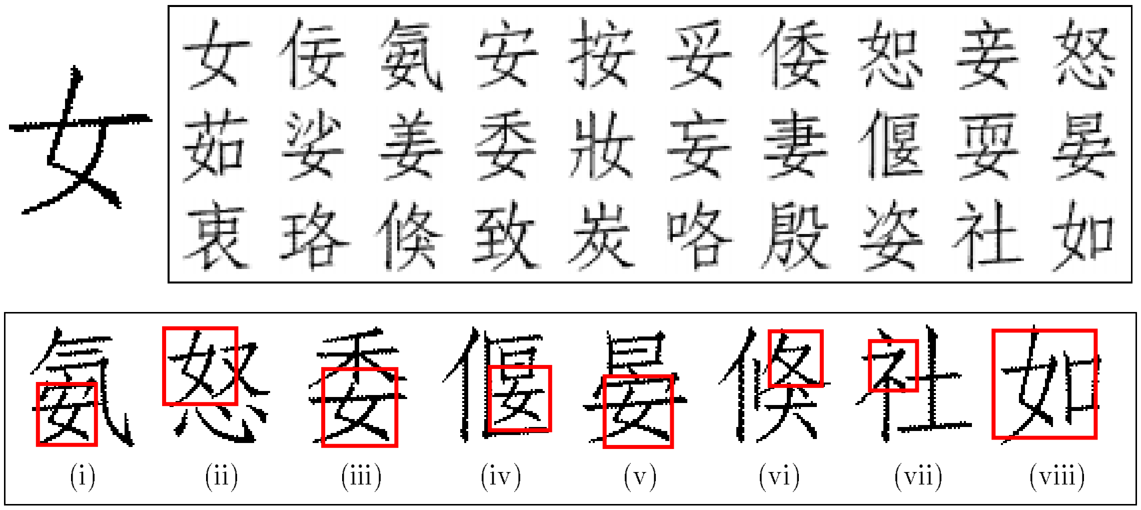

女
女 佞 氨 安 按 妥 倭 恕 妾 怒
茹 娑 姜 委 妝 妄 妻 偃 要 晏
東 珞 倏 致 炭 咯 殷 姿 社 如
氨 怒 委 偃 晏 倏 社 如
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

Figure 7.36: Chinese Characters Query Result – Example 7. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 3rd, (ii) 10th, (iii) 14th, (iv) 18th, (v) 20th, (vi) 23rd, (vii) 29th, (viii) 30th.

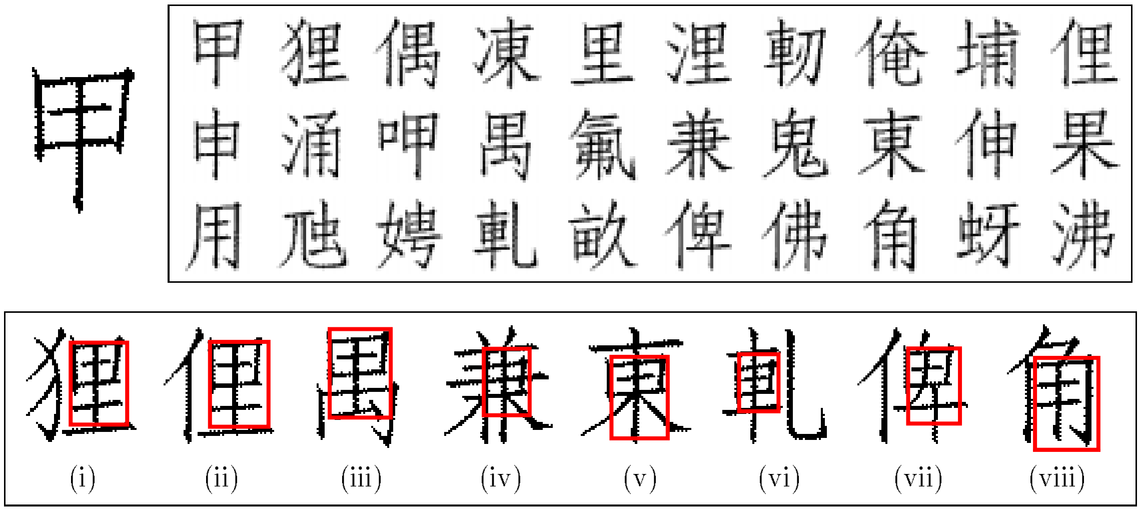

甲
甲 狸 偶 凍 里 浬 軔 俺 埔 俚
申 涌 呷 禺 氟 兼 鬼 東 伸 果
用 虺 娉 軋 畝 俾 佛 角 蚜 沸
狸 俚 禺 兼 東 軋 俾 角
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

Figure 7.37: Chinese Characters Query Result – Example 8. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 2nd, (ii) 10th, (iii) 14th, (iv) 16th, (v) 18th, (vi) 24th, (vii) 26th, (viii) 28th. It is difficult to see the pattern occurrence in (iv) and (viii) because of its surroundings.

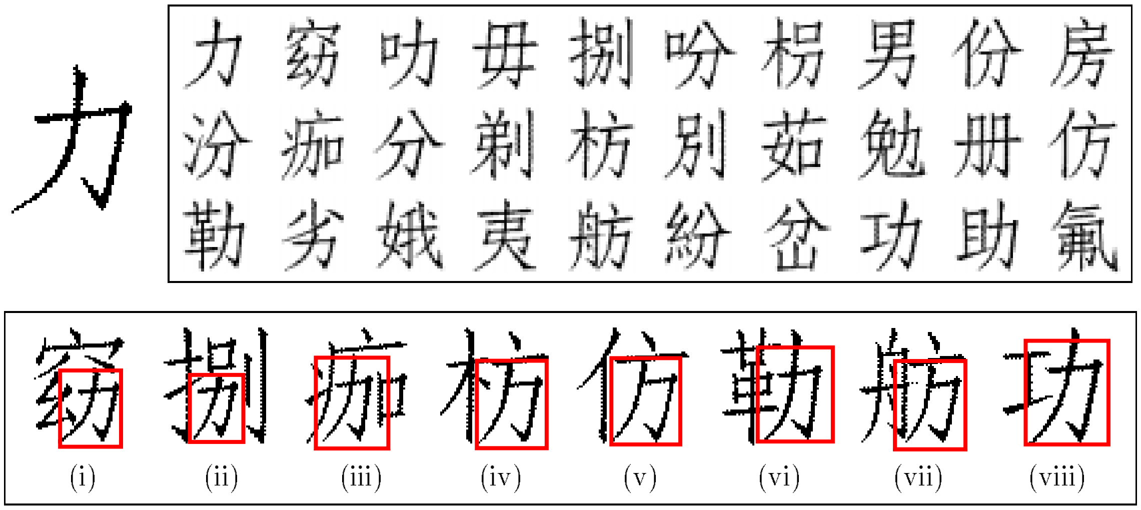

力
力 窈 叻 毋 捌 吩 枴 男 份 房
汾 疬 分 剃 枋 别 茹 勉 册 仿
勒 劣 娥 夷 舫 紛 岔 功 助 氣
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.38: Chinese Characters Query Result – Example 9. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 2nd, (ii) 5th, (iii) 12th, (iv) 15th, (v) 20th, (vi) 21st, (vii) 25th, (viii) 28th.

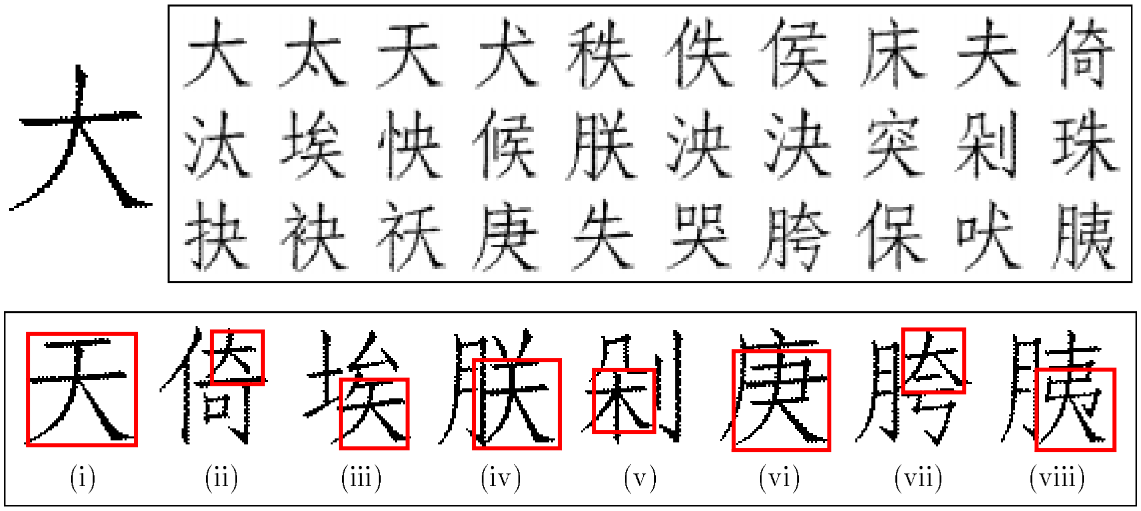

大
大 太 天 犬 秩 佚 侯 床 夫 倚
汰 埃 快 候 朕 泱 決 突 剁 珠
抉 袂 祆 庚 失 哭 胯 保 吠 腆
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.39: Chinese Characters Query Result – Example 10. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 3rd, (ii) 10th, (iii) 12th, (iv) 15th, (v) 19th, (vi) 24th, (vii) 27th, (viii) 30th.

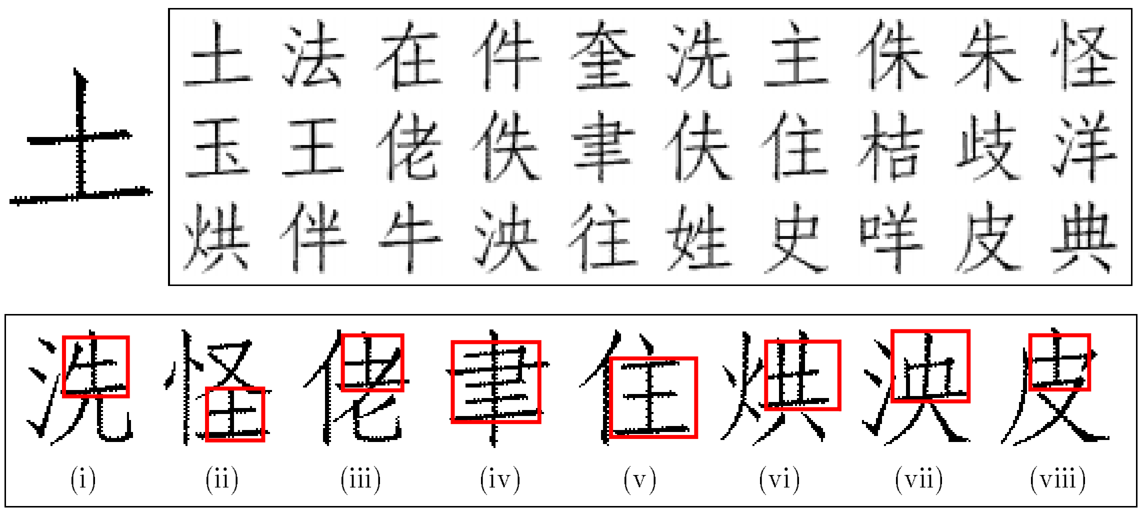

土
土 法 在 件 奎 洗 主 侏 朱 怪
玉 王 佬 佚 聿 伕 住 桔 歧 洋
烘 伴 牛 泱 往 姓 史 咩 皮 典
洗 怪 佬 聿 住 烘 泱 皮
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

Figure 7.40: Chinese Characters Query Result – Example 11. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 6th, (ii) 10th, (iii) 13th, (iv) 15th, (v) 17th, (vi) 21st, (vii) 24th, (viii) 29th. The extra horizontal lines in (iv) make it difficult to see the pattern occurrence.

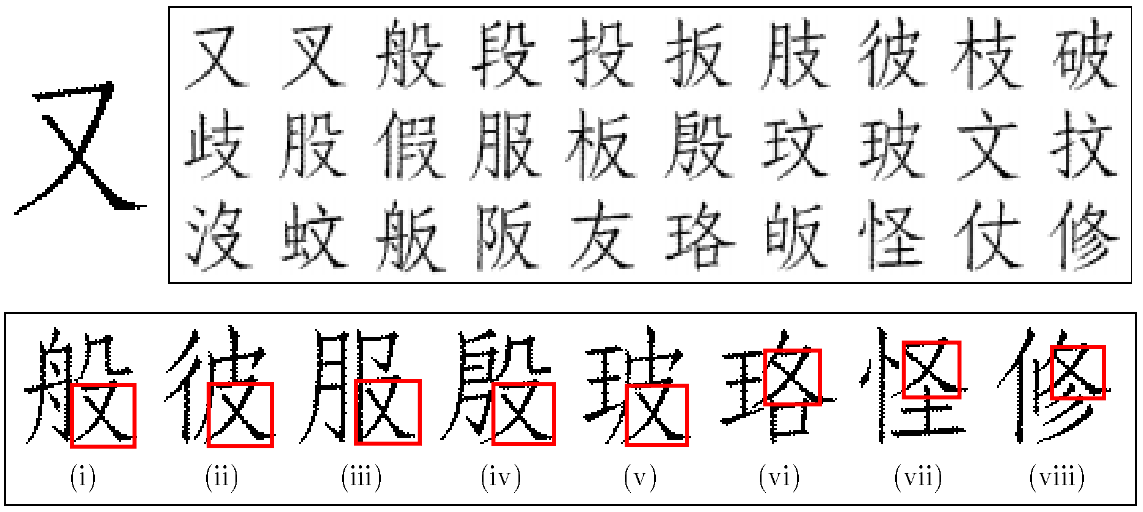

又
又 叉 般 段 投 扳 肢 彼 枝 破
歧 股 假 服 板 殷 玟 玻 文 技
没 蚊 舨 阪 友 珞 皈 怪 仗 修
般 彼 服 殷 玻 珞 怪 修
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

Figure 7.41: Chinese Characters Query Result – Example 12. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 3rd, (ii) 8th, (iii) 14th, (iv) 16th, (v) 18th, (vi) 26th, (vii) 28th, (viii) 30th.

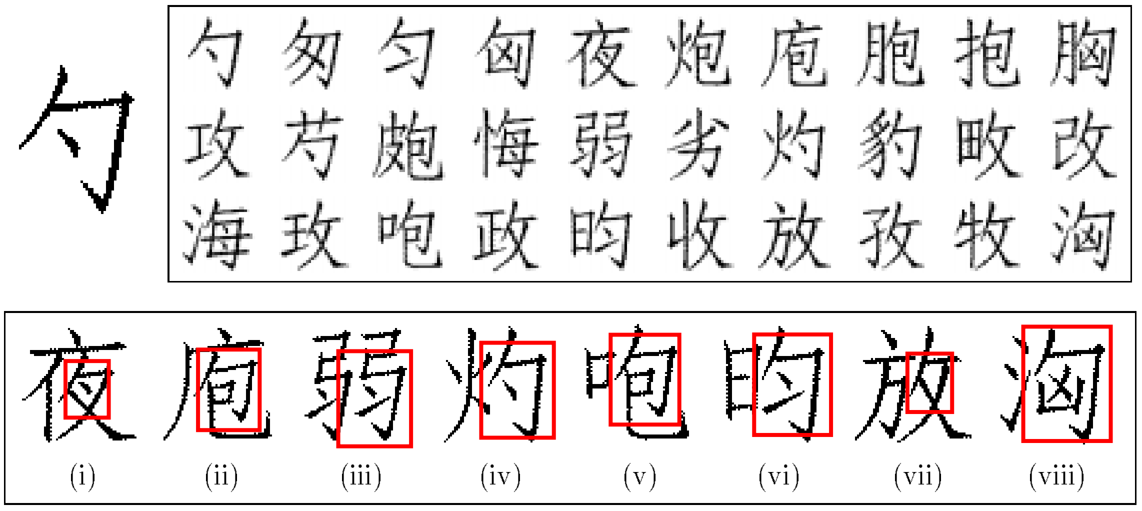

勺 匆 匀 匈 夜 炮 庖 胞 抱 胸
攻 芍 匏 悔 弱 劣 灼 豹 畋 改
海 玫 咆 政 昀 收 放 孜 牧 汹
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.42: Chinese Characters Query Result – Example 13. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 5th, (ii) 7th, (iii) 15th, (iv) 17th, (v) 23rd, (vi) 25th, (vii) 27th, (viii) 30th. Most of these examples show inexact matches of the query and a region of a database character. The differences in appearance are relatively small except for example (vii).

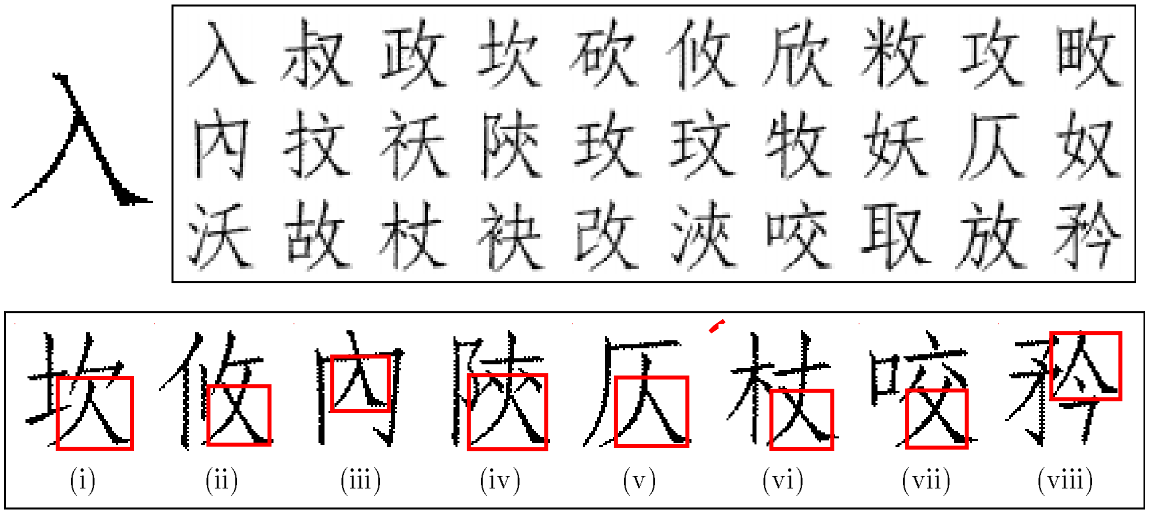

入 叔 政 坎 砍 攸 欣 敉 攻 畋
内 技 祆 陕 玫 玟 牧 妖 仄 奴
沃 故 杖 袂 改 浹 咬 取 放 矜
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.43: Chinese Characters Query Result – Example 14. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 4th, (ii) 6th, (iii) 11th, (iv) 14th, (v) 19th, (vi) 23rd, (vii) 27th, (viii) 30th.

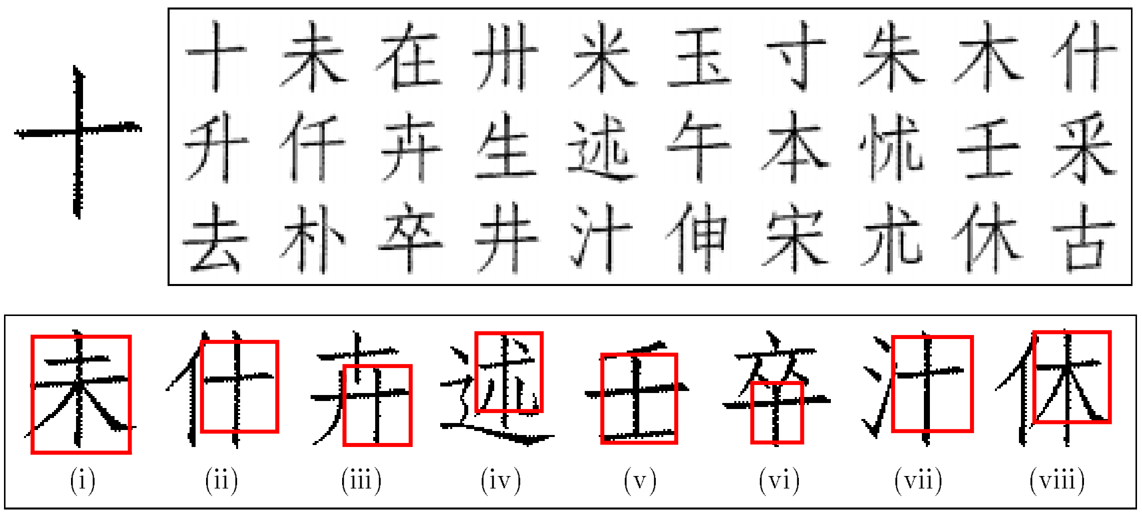

十 未 在 卅 米 玉 寸 朱 木 什
升 仟 卉 生 述 午 本 怵 壬 采
去 朴 卒 井 汁 伸 宋 朮 休 古
(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

Figure 7.44: Chinese Characters Query Result – Example 15. (top) the query pattern and the top thirty matches. (bottom) some selected query localization results. The ranks of these selected retrievals in (top) are: (i) 2nd, (ii) 10th, (iii) 13th, (iv) 15th, (v) 19th, (vi) 23rd, (vii) 25th, (viii) 29th.

such "extra" ink in the vicinity of the pattern occurrence can obscure the visual similarity between that area of the image and the pattern. The local image signature computed during the final τ-EMD check of the verification and refinement phase can be used to solve this extra ink problem if it is indeed considered a problem for a particular application. A penalty can be added to the signature distance which is proportional to the difference between the total amount of image ink inside the final search rectangle and the amount of ink in the scaled pattern. An implementation of this idea should err on the side of smaller rather than larger penalties since the computed pattern scale and location will not be perfect.

A big difference between the shape and color cases is that, as currently implemented, SEDL does not find rotated versions of a given shape pattern within an image. Instead, it uses curve orientations to direct or guide its search for a possibly scaled version of the given pattern. A possible solution to find rotated shape patterns within the SEDL framework is to use relative orientations instead of absolute orientations ([33]). For example, we might compute the attribute portion of a signel by taking the difference in tangent orientations at the endpoints of a short curve piece instead of the average orientation over the curve piece. SEDL's search for a possibly rotated, scaled version of a given shape pattern can be directed by the relative orientation attribute since relative orientations will not change when a similarity transformation is applied to the pattern curves.

	Time (secs)			
Query	Phase 1	Phase 2	Phase 3	Total
了	39.7	7.6	37.2	84.5
几	40.4	9.8	77.5	127.7
口	16.2	13.9	86.2	116.3
人	40.0	8.4	30.3	78.7
三	14.1	12.6	33.2	59.9
么	36.4	9.4	37.8	83.6
女	46.6	8.9	72.4	127.9
甲	26.1	14.4	103.4	143.9
力	40.4	9.6	75.0	125.0
大	46.5	8.4	62.7	117.6
土	24.3	11.6	50.7	86.6
又	45.3	8.9	54.1	108.3
勺	41.3	9.0	60.5	110.8
入	32.8	8.7	30.5	72.0
十	22.7	7.9	31.7	62.4
Average				95.7

Figure 7.45: Query Times for the Chinese Character Database.