Detection of cats in images

ANNA MARIA WALACH (s121540)
Technical University of Denmark
02238 Biometric Systems

The LaTeX `acmtrans` document style formats articles in the style of the ACM transactions. Users who have prepared their document with LaTeX can, with very little effort, produce camera-ready copy for these journals.

Categories and Subject Descriptors: D.2.7 [**Software Engineering**]: Distribution and Maintenance—*documentation*; H.4.0 [**Information Systems Applications**]: General

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: cat, detection, images

---

1. INTRODUCTION

There are more and more projects that uses live streaming and social media power to help and safe feral cats. Some of them are meant to control the population in the area [Society 2015], other focus on raising awareness about feral cats, spaying and neutering importance [Kittens 2015] or just to increase changes of finding a new home for homeless cats and kittens [Room 2015].

People are also more interested in health and safety of their own pets. Lower prices of video technologies and smarthone popularity helped to create various apps for controlling your pet lifestyle. [Zillians 2015] created an automatic feeder with cat identification system and connected it to the smartphone app. It allows owner to supervise the amount of dry food and water his cats are consuming everyday, to receive alarms in case of abnormalities and even spy on them while eating. [Company 2015] app is designated to help people find their lost cats and dogs. According to American Human Society, almost 3.5 million pets are lost each year and about 80% of dogs and 98% of cats are never re-united with their families. The authors of the smarthone app allows the owners to made profiles for their pets (including pictures) and in case the pet is missing, you launch the "alarm" on that pet, together with last seen location. If anyone finds a homeless animal, he may send the photo of it to the PiP app, which automatically try to match one of the lost pets with found pet.

1.1 Problem description

2. THE STATE OF THE ART

3. METHODOLOGY

3.1 Overfull hbox - Stretching/filling one horizontal line

To solve a line break due to "Overfull `\hbox`", here is a plain TeX solution; here `\hsize` is the default setting of acmtrans.sty:

```
\hbox to \hsize{line sentence to be stretched}
```

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Fig. 1. An example of a program centered in a figure

This can be used in a list environment as well but `\hsize` declared to a reduce dimension:

```
\hbox{\vbox{\hsize = less than the default setting
\hbox to \hsize{line sentence to be stretched}}}
```

3.2 Programs

Good formatting of programs requires a knowledge of their semantics, and is beyond the scope of a document production system. While "pretty printers" are useful for handling the many pages of a real program, the short examples that are published in articles should be formatted by hand to improve their clarity. The LaTeX `tabbing` environment makes the formatting of programs relatively easy, especially if the user defines commands for his particular language constructs. One may also use the `verbatim` environment.

The ACM transactions style requires that programs be formatted with different size fonts, depending upon whether they appear in the text or in a figure, but that is handled by the figure macro which automatically sets the correct font size. Moreover, programs in running text should be indented two ems on each side (as provided by the `quote` environment), and programs in regular figures should be centered. (Programs in "narrow figures" (q.v.) are left or right justified automatically).

Here is an example of a program:

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Figure 1 shows how the same program looks in a figure.

In addition to formatting programs, the `tabbing` environment may be used for similar displayed material such as BNF syntax specifications and rewrite rules.

4. FIGURES AND TABLES

4.1 Figures

The ordinary LaTeX `figure` environment works as usual. Figure 2, which is Figure 6 of Nielson [1985], a bogus reference, was produced in this way. Note that figures

	$\perp$	F	T
$\perp$	$\perp$	$\perp$	T
F	$\perp$	F	T
T	$\perp$	T	T

Fig. 2. The truth table for the parallel-or.

Table I. The truth table for the parallel-or.

	$\perp$	F	T
$\perp$	$\perp$	$\perp$	T
F	$\perp$	F	T
T	$\perp$	T	T

should never appear in the text or at the bottom of a page. (If you use the figure placement optional argument, use only `t` or `p` or both; do not use `h` or `b`).

Some figures (and tables) have no caption except for the figure number. For such figures (and tables), one uses a `\nocaption` command, which has no argument, instead of the `\caption` command.

In addition to this method of formatting figures, the ACM transactions also uses figures with side captions, as in Figure **??**. Such a figure is produced with the `narrowfig` environment. This environment has a single mandatory argument, which is the width of the figure. Note that if the figure is generated by `tabbing` or `tabular`, one can safely overestimate the size. It works just like the ordinary `figure` environment, except it must contain only one `\caption` or `\nocaption` command, which must come after the figure itself.

The `narrowfig` environment should obviously not be used unless the figure is narrow enough to leave a reasonable amount of space beside it for the caption. The ACM seems to have no consistent policy for choosing which style of figure to employ.

4.2 Tables

The ordinary LaTeX `table` environment can be used, but it requires the user to add formatting commands to match the ACM transactions style. This formatting is performed automatically if the `acmtable` environment is used instead, producing the result shown in Table I, which shows the same table displayed in Figure 2.

4.3 Bibliography

(1) [Cytron et al. 1991] → `\cite{cytron-et-al-toplas91}`

(2) Briggs et al. [1994] → `\citeN{briggs-cooper-torczon-toplas94}` or

The list will be updated as we find unique cases.

REFERENCES

Archer, Jr., J. E., Conway, R., and Schneider, F. B. 1984. User recovery and reversal in interactive systems. *ACM Trans. Program. Lang. Syst. 6,* 1 (Jan.), 1–19.

Company, T. P. R. 2013 (accessed June 22, 2015). *PiP - reuniting lost pets with their families.* `http://www.petrecognition.com/`.

Dunlop, D. D. and Basili, V. R. 1985. Generalizing specifications for uniformly implemented loops. *ACM Trans. Program. Lang. Syst. 7,* 1 (Jan.), 137–158.

Gerndt, M. 1989. Automatic parallelization for distributed-memory multiprocessing systems. Ph.D. thesis, University of Bonn, Bonn, Germany.

Heering, J. and Klint, P. 1985. Towards monolingual programming environments. *ACM Trans. Program. Lang. Syst. 7,* 2 (Apr.), 183–213.

Kittens, T. 2015 (accessed June 22, 2015). *Rescue Kitten Reality TV.* `http://www.tinykittens.com/`.

Knuth, D. E. 1984. *The TEXbook.* Addison-Wesley, Reading, Mass.

Korach, E., Rotem, D., and Santoro, N. 1984. Distributed algorithms for finding centers and medians in networks. *ACM Trans. Program. Lang. Syst. 6,* 3 (July), 380–401.

Kornerup, J. 1994. Mapping powerlists onto hypercubes. M.S. thesis, The University of Texas at Austin. (In preparation).

Lamport, L. 1986. *LATEX: A Document Preparation System.* Addison-Wesley, Reading, Mass.

Nielson, F. 1985. Program transformations in a denotational setting. *ACM Trans. Program. Lang. Syst. 7,* 3 (July), 359–379.

Nielson, F. 1986. Program transformations in a denotational setting. *ACM Trans. Program. Lang. Syst. 7,* 3 (July), 359–379.

Reid, B. K. 1980. A high-level approach to computer document formatting. In *Proceedings of the Seventh Annual Symposium on Principles of Programming Languages.* ACM, New York, 24–31.

Room, T. C. 2015 (accessed June 22, 2015). *Foster Kitten Cam.* `https://livestream.com/FosterKittenCam`.

Society, L. A. P. 2015 (accessed June 22, 2015). *Trap-Neuter-Return (TNR) Pilot.* `http://www.tinykittens.com/cats`.

Zillians. 2015 (accessed June 22, 2015). *CatFi - Smart Feeder.* `http://catfi.com/`.

Received February 1986; November 1993; accepted January 1996