
undelete_jpg

undelete_jpg is a program that recovers JPG images from SD cards and hard drives.

Introduction

When deleting an image from an SD card, only the directory entry to the file is removed but the content still remains on the device. undelete_jpg recognizes that undeleted content and recovers JPG files. The recovered files may be corrupted because cameras write new files into deallocated device space, but undelete_jpg will try its best to output only valid JPG files.

Implementation

A JPG file is comprised of a list of JPG markers such as SOI (start of image), SOS (start of scan), EOI (end of image), etc. More about JPG markers [here](#). When possible, undelete_jpg uses mmap() to map the device content into memory, otherwise it uses read(). When reading through the content of the device, undelete_jpg uses memchr(), which is typically optimized with SIMD instructions, to detect SOI markers. If it can it will use mmap, otherwise it will use read()

undelete_jpg is fast: on a Macbook Pro with a 2.8 GHz Quad-Core Intel Core i7, undelete_jpg goes through a cached file at **10 GB/s**. **This is actually the memory bandwidth on this machine**. The throughput is well over the read speed of typical SD cards (10MB/s up to 500 MB/s).

Usage

Run the following in your terminal:

```
1 git clone https://github.com/saintmarina/undelete_jpg.git
2 cd undelete_jpg
3 make
4 sudo ./undelete_jpg /dev/block_device
```

Notes: * `sudo` is required for block device access. * Recovered JPG files are written in the current directory. * Ways to get device path is platform dependent. Follow section below for more detailed instructions.

How to get device path on macOS

To list all mounted drives and it's partitions on a macOS, run the following:

```
1 diskutil list
```

Find your device's path in the list. As an example, see what the list looks like on my machine.

```
anna — anna@Annas-MBP — ~ — -zsh — 80x24
~ » diskutil list
/dev/disk0 (internal, physical):
#  TYPE NAME              SIZE IDENTIFIER
0  GUID_partition_scheme  *251.0 GB disk0
1  EFI EFI                  314.6 MB disk0s1
2  Apple_APFS Container disk1 190.0 GB disk0s2
3  Microsoft Basic Data BOOTCAMP 60.7 GB disk0s3

/dev/disk1 (synthesized):
#  TYPE NAME              SIZE IDENTIFIER
0  APFS Container Scheme - +190.0 GB disk1
   Physical Store disk0s2
1  APFS Volume Macintosh HD - Data 168.3 GB disk1s1
2  APFS Volume Preboot              82.2 MB disk1s2
3  APFS Volume Recovery             528.5 MB disk1s3
4  APFS Volume VM                   1.1 GB disk1s4
5  APFS Volume Macintosh HD         11.1 GB disk1s5

/dev/disk2 (external, physical):
#  TYPE NAME              SIZE IDENTIFIER
0  FDisk_partition_scheme *128.0 GB disk2
1  Windows_FAT_32 EOS_DIGITAL 128.0 GB disk2s1
```

From the example above the path looks like this:

```
1 sudo ./undelete_jpg /dev/rdisk2s1
```

Add **r** in the beginning of your device path. **r** stands for raw.

How to get device path on Linux

To list devices and their partitions on linux, run:

```
1 sudo fdisk -l
```

Find your device in the partition table.

As an example, see what the partition table looks like on my machine.

```
linux — vagrant@vagrant: ~/Exercism/cs50/recovery — ..cs50/recovery — ssh ◀ vagrant ssh — 80x35
[master] recovery » sudo fdisk -l
Disk /dev/sda: 64 GiB, 68719476736 bytes, 134217728 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4ec46e60

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1   *      2048 134215679 134213632   64G 8e Linux LVM

Disk /dev/mapper/vagrant--vg-root: 63 GiB, 67687677952 bytes, 132202496 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/vagrant--vg-swap_1: 980 MiB, 1027604480 bytes, 2007040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdb: 119.3 GiB, 128043712512 bytes, 250085376 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdb1    63 250085375 250085313 119.3G  c W95 FAT32 (LBA)
[master] recovery »
```

Example of running the program on linux:

```
1 sudo ./undelete_jpg /dev/sdb1
```

Other projects

PhotoRec is another opensource battle tested tool to recover all kinds of files, including JPGs.

License

This project is licensed under the MIT License.